



Даниела Иванова Борисова

**ЕДНО- И МНОГОКРИТЕРИАЛНИ МОДЕЛИ И АЛГОРИТМИ
ЗА ОПТИМАЛНО ПРОЕКТИРАНЕ, ПЛАНИРАНЕ И
УПРАВЛЕНИЕ НА ИНЖЕНЕРНИ СИСТЕМИ**

ДИСЕРТАЦИЯ

за присъждане на научна степен

“Доктор на науките”

Научна специалност: 01.01.12 ”Информатика”

Професионално направление 4.6 ”Информатика и компютърни науки”

Съдържание

Използвани основни термини.....	v
Увод.....	vii
Цел и задачи	x
Резюме.....	xi
Глава 1: Едно- и многокритериална оптимизация и вземане на решения при инженерни приложения	1
1.1. Оптимизационен модел	1
1.2. Методи за решаване на еднокритериални задачи	5
1.2.1. Точни методи.....	5
1.2.2. Приблизени методи	6
1.3. Методи на многокритериалната оптимизация	8
1.3.1. Обща формулировка на задачата на многокритериалната оптимизация	8
1.3.2. Парето-оптималност	9
1.3.3. Класификация на методите за решаване многокритериални задачи, в зависимост от начина на задаване на предпочитанията	10
1.3.4. Скаларизационен подход за решаване на задачи на многокритериалната оптимизация	11
1.4. Комбинаторна оптимизация.....	15
1.4.1. Някои типични приложения на комбинаторната оптимизация	18
1.5. Общи подходи при вземането на решения	21
1.5.1. Процес на вземане на решение	22
1.5.2. Видове системи за подпомагане вземането на решения.....	23
Глава 2: Оптимално проектиране на модулни инженерни системи, вземащи предвид съществуващите взаимовръзки между модулите	27
2.1. Персоналните компютри като пример за модулна инженерна система	29
2.2. Обобщен модел за оптимално проектиране на модулни инженерни системи, вземащи предвид съществуващите взаимовръзки между модулите.....	30
2.3. Еднокритериална оптимизационна задача за проектиране на персонални компютри	36
2.3.1. Числено тестване	37
2.4. Многокритериална оптимизационна задача за проектиране на персонални компютри	40
2.4.1. Числено тестване	42
2.5. Обобщен алгоритъм за итеративно и рационално проектиране на конфигурираната на персонални компютри.....	44

Глава 3: Оптимално проектиране на модулни инженерни системи, вземащи предвид съществуващите взаимовръзки между модулите и външните експлоатационни условия 51

- 3.1. Влияние на външните експлоатационни условия, на примера на проектиране на уреди за нощно виждане 52
- 3.2. Детерминиран оптимизационен модел на уреди за нощно виждане..... 54
- 3.3. Стохастичен оптимизационен модел на уреди за нощно виждане..... 57
- 3.4. Обобщен модел за оптимално проектиране на модулни инженерни системи, вземащи предвид съществуващите взаимовръзки между модулите и външните експлоатационни условия 61
- 3.5. Многокритериални модели за определяне на външните условия на наблюдение, съвместими с параметрите на уреди за нощно виждане..... 62
 - 3.5.1. Граници на изменение на външните условия на наблюдение, съвместими с определено разстояние на действие на уреда..... 63
 - 3.5.2. Минимално допустими външни условия, при които се постига една и съща стойност за параметъра разстояние на действие..... 69
- 3.6. Методи за проектиране на модулни инженерни системи..... 71
 - 3.6.1. Метод на итеративно проектиране 71
 - 3.6.2. Метод на рационално проектиране 73
 - 3.6.3. Метод на оптимално проектиране 75
 - 3.6.4. Обобщен алгоритъм за оценка на параметрите на проектираните уреди за нощно виждане 76

Глава 4: Оптимално проектиране на инженерни системи, вземащи предвид съществуващите взаимовръзки, външните експлоатационни условия и икономическата ефективност 79

- 4.1. Специфика на проектирането на инженерни системи, отчитащи вътрешни и външни зависимости, и икономическата ефективност – ветроенергийни паркове 80
- 4.2. Разположение на ветровите турбини във ветроенергийни паркове 85
- 4.3. Еднокритериален оптимизационен модел за оптимално проектиране на ветроенергийни паркове..... 88
 - 4.3.1. Числено тестване 93
- 4.4. Многокритериален оптимизационен модел за оптимално проектиране на ветроенергийни паркове..... 100
 - 4.4.1. Числено тестване 101
 - 4.4.2. Алгоритъм за приложение на многокритериалния подход при проектиране на ветроенергийни паркове 107

Глава 5: Модели и алгоритми за оптимално планиране.....109

- 5.1. Въведение 109
- 5.2. Оптимално разписание за зависима обработка на детайли върху множество машини..... 112
 - 5.2.1. Обобщен модел за определяне на оптимални разписания за зависима обработка на детайли върху множество машини..... 113

5.2.2. Числено тестване	113
5.3. Определяне на оптимални разписания за смесена обработка на детайли върху множество машини	117
5.3.1. Паралелен алгоритъм за определяне на оптимално разписание с минимална продължителност	117
5.3.2. Числено тестване	118
5.4. Модели за линейно разкрояване	124
5.4.1. Обобщен модел за линейно разкрояване.....	125
5.4.2. Числено тестване	126
5.5. Оптимален брой на персонал и график за работа	131
5.5.1. Обобщен модел за определяне броя на персонала	132
5.5.2. Алгоритъм за определяне на оптимален брой оператори.....	134
5.5.3. Числено тестване	136
Глава 6: Модели и алгоритми за оптимално управление при предсказващо поддържане.....	139
6.1. Въведение.....	139
6.2. Определяне на оптимална стратегия, базирана на оценките на приходите и разходите.....	140
6.2.1. Еднокритериални модели за определяне на оптимална стратегия, базирана на оценката за приходи и разходи за предсказващо поддържане.....	141
6.2.2. Числено тестване	146
6.3. Определяне на оптимална стратегия, отчитаща условията на неопределеност .	148
6.3.1. Числено тестване	154
6.4. Определяне на оптималното разположение на сензори за мониторинг на състоянието.....	156
6.4.1. Обобщен еднокритериален оптимизационен модел	158
6.4.2. Числено тестване	159
6.4.3. Обобщен многокритериален оптимизационен модел.....	162
6.4.4. Числено тестване	163
Глава 7: Модели и алгоритми за обоснован избор на алтернативи.....	167
7.1. Въведение.....	167
7.2. Обобщен алгоритъм за избор чрез класиране	168
7.2.1. Числено тестване	170
7.3. Многокритериален оптимизационен модел за избор на k най-добри алтернативи	176
7.3.1. Числено тестване	178
7.4. Избор на алтернатива с отчитане влиянието на околната среда	182
7.4.1. Многокритериален модел за избор на уреди за нощно виждане с отчитане на външните условия на наблюдение	182

7.4.2. Числено тестване	183
7.5. Избор на алтернатива чрез групово вземане на решения	187
7.5.1. Обобщен модел на групово вземане на решения	189
7.5.2. Числено тестване	189
Глава 8: Софтуерни инструменти за приложения в инженерни системи....	193
8.1. Уеб-базирани системи за подпомагане проектирането на модулни инженерни системи	193
8.1.1. Система за подпомагане проектирането на персонални компютри.....	195
8.1.2. Система за подпомагане проектирането на уреди за нощно виждане (очила, прицели и бинокли).....	196
8.1.3. Система за подпомагане проектирането на ветроенергийни паркове	201
8.1.4. Система за подпомагане на групово вземане на решения	205
8.2. Специализирани софтуерни системи.....	207
8.2.1. Специализирани софтуерни системи, използващи LINDO API	209
8.2.2. Специализирани софтуерни системи, използващи оптимизационен модул, работещ в средата на електронни таблици	213
Заклучение.....	217
Авторска справка	219
Научни и научно-приложни приноси	219
Апробация на резултатите	221
Декларация	221
Списък на научноизследователските и приложни проекти по темата на дисертацията с участието на автора	222
Списък на публикациите, в които са представени резултати от дисертационния труд	223
Списък на цитиранията на публикации от дисертационния труд	225
Библиография	231

Използвани основни термини

инженерна система – система от технически структури (машини, апарати и устройства) и свързаните с тях производствени процеси и експлоатационни условия.

модулна система – система, съставена от стандартизирани модули, които могат да бъдат самостоятелно създадени и след това използвани при проектирането на системата.

оптимизация – използване на едно- и многокритериални задачи за определяне на оптимумите на зададени целеви функции.

оптимално проектиране – проектиране на модулни системи, вземащи предвид съществуващите взаимовръзки между модулите, външните експлоатационни условия и икономическата ефективност.

оптимално планиране – оптимизация на производствени разписания и графици, линейно разкрояване и избор на алтернатива/и.

оптимално управление – определяне на стратегия за оптимално предсказващо поддържане.

предсказващо поддържане – подход, подпомагащ определянето състоянието на инженерна система в експлоатация, за да се предскаже кога и как да се извършва техническо обслужване.

зависима обработка на детайли – наличие на зададена последователност на обработката на детайлите.

смесена обработка на детайли – наличие на зададена последователност на обработката на част от детайлите, а останалите детайли се обработват в произволен ред.

итеративен метод за проектиране – проектиране на модулна инженерна система чрез итеративен избор на съвместими модули и определяне параметрите на проектираната система.

рационален метод за проектиране – проектиране на модулна инженерна система чрез избор на съвместими модули, удовлетворяващи зададени ограничения за параметрите на проектираната система.

оптимален метод за проектиране – проектиране на модулна инженерна система чрез оптимизация на избора на съвместими модули.

Увод

Обект на изследванията в настоящия дисертационен труд са *инженерните системи*, разглеждани като системи от технически структури (машини, апарати и устройства) и свързаните с тях производствени процеси и експлоатационни условия. В проектирането, изграждането и поддръжката на всяка инженерна система, инженерите трябва да вземат различни технологични и управленски решения на няколко етапа. Крайната цел на всички тези решения е или да се сведат до минимум разходите или да се увеличат желаните ползи. Като се има предвид изключителния напредък в областта на компютърните технологии през последните години, оптимизационните методи могат да бъдат все по-пълноценно използвани за намиране на оптимални решения на големи и сложни проблеми, отколкото когато и да е било досега. В резултат, оптимизацията днес се разглежда като необходим инструмент за всеки инженер, работещ в различни области на индустрията. За всеки практически проблем желаните ползи могат да бъдат изразени като функция на определени управляващи променливи, за която функция да се определят условията, които водят до намирането на максималната (или минималната) ѝ стойност. Оптимизацията, прилагана за решаване на различни инженерни проблеми, в най-широк смисъл, може да бъде обобщена в няколко направления като – оптимално проектиране, оптимално планиране и оптимално управление. Терминът *оптимално проектиране* се отнася до проектиране на модулни системи, вземащи предвид съществуващите взаимовръзки между модулите, външните експлоатационни условия и икономическата ефективност. Под *оптимално планиране* се има предвид оптимизация на производствени разписания и графици, линейно разкрояване и избор на алтернатива/и. В настоящия дисертационен труд се третира една подобласт на областта на *оптималното управление*, свързана с поддръжката на инженерни системи – определянето на стратегия за оптимална поддръжка, определяне разположението на сензори и избор на алтернатива/и за ремонт. Терминът *оптимално* се използва в смисъла на формулиране и решаване на оптимизационни задачи спрямо зададени критерии и при спазване на определени технически, технологични, конструктивни, икономически и др. ограничения.

Актуалността на проблема се обуславя от съвременните тенденции в създаването и използването на високоефективни инженерни системи, които от една страна да бъдат конкурентни на пазара и от друга страна да удовлетворяват различни

потребителски изисквания. Тези тенденции се изразяват в стремежа да се използват новите достижения в областта на приложната математика.

Използването на оптимизация в областта на инженерното проектиране става все по-достъпна, благодарение бързо развиващата се ИТ индустрия. Голяма част от процеса на проектиране е и вероятно винаги ще бъде интуитивна, но както аналитичните техники, така и оптимизацията допринасят за значителното подобряване процеса на проектиране. Един от най-трудните аспекти на този процес е да се намали възможността за скъпо струващи грешки, които ще се проявят в процеса на масовото производство и ще компрометират функционалността на произвежданата система/устройство. Формулирането и използването на достатъчно адекватен модел за проектираната инженерна система е несъмнено по-добрата алтернатива от интуитивното вземане на решения. Въпреки това, често е необходимо да се разглеждат много функции едновременно, което води до необходимост от компромисни решения. Това определя процеса на инженерното проектиране на реални системи като сложен процес. В този контекст оптималното проектиране в настоящия дисертационен труд се разглежда като процес на проектиране, включващ формулирането и решаването на едно- и многокритериални оптимизационни задачи.

Ролята на моделирането, като метод на научното познание и метод за решаване на инженерни проблеми винаги се оценява достатъчно високо. Постиженията на математиката позволяват използването на математическото моделиране за различни обекти и процеси. Особено значение придобива моделирането в съвременните условия на ускорен научно-технически прогрес, при нуждата от постигането на висока ефективност с ограничени финансови, материални, трудови, енергетични и времеви ресурси. Сега е трудно да се открие област от човешката дейност, в която да не е използвано такова моделиране. В перспектива, за всяка една система могат да се създадат модели и те да се използват преди фактическата реализация на системата, т.е. всеки информационен, технически или организационен проект може да бъде предшестван от моделиране и симулиране, с цел предварителна оценка на заложената функционалност.

Друг аспект на използване на оптимизацията е прилагането ѝ в организацията на производствените процеси в съвременните предприятия. Целта е да се осигури не само постигането на производствените цели, но и да се осигури оптимално разпределение на материалните и човешки ресурси. При това, големината на разглежданите предприятия не е показател за сложността на съществуващите проблеми. Възможно е малките и

средните предприятия да имат по-сложни организационни проблеми от някои големи фирми и компании. Причина за това е изострената конкуренция, в резултат на която предприятията са принудени да разширяват спектъра на дейностите си. Важен е и фактът, че съвременните предприятия не функционират изолирано, а са част от определена икономическа среда, която също оказва влияние – наличието на конкуренти, възможности за финансово подпомагане при изпълнението на определени условия и т.н. Това влияние постоянно расте и в съчетание с останалите фактори определя необходимостта от използване на научни подходи при решаване на организационните проблеми. Използването на едно- и многокритериална оптимизация би подобрило оптимално планиране и използването на ресурсите.

Една съществена особеност на инженерните системи е, че техните компоненти подлежат на износване с течение на времето. Този естествен процес на влошаване на технологичните параметри в процеса на експлоатация може да бъде ускорен от външни фактори или да бъде забавен чрез мониторинг, диагностика и вземане на съответни превантивни мерки. За целта са необходими стратегии за вземане на адекватни решения в условията на неопределеност или непълна информация за състоянието на системата или осъществяване на т.нар. *предсказващо поддържане*. Един от основните проблеми, който трябва да бъде решен, е да се вземе решение кога да се ремонтира оборудването и кога да се закупи нова машина. Превантивният ремонт спестява разходи за закупуване и монтиране на ново оборудване и осигурява печалби от продължаване на производството, но това решение трябва да бъде добре обосновано, чрез използване на подходящ научен подход. Едно погрешно решение би довело до скъпо струващи последици, свързани с повреди на машини, необходимост от подмяна и съответно спиране на производството. Съвременните ИКТ позволяват използване на разнообразна информация за подобни решения, чрез бази от данни, и експертни системи. Едно перспективно направление е използването на математически модели и оптимизационни задачи, подпомагащи мениджмънта при вземането на добре обосновани решения за оптимално предсказващо поддържане.

Цел и задачи

Цел на настоящият дисертационен труд е да се предложат едно- и многокритериални модели, методи и алгоритми, подпомагащи оптимално проектиране, планиране и управление на инженерните системи. За реализиране на тази цел е необходимо да се изпълнят следните задачи:

- да се предложат модели и методи за оптимално проектиране на модулни инженерни системи, вземащи предвид съществуващите взаимовръзки между модулите, външните експлоатационни условия и икономическата ефективност,
- да се предложат модели и алгоритми за оптимално планиране на използването на ресурси,
- да се предложат модели, методи и алгоритми за оптимално управление на предсказващото поддържане,
- да се предложат методи и алгоритми за обоснован избор на алтернативи.

Методологията на изследванията включва освен разработването на адекватни математически модели, методи и алгоритми, така и оценка на тяхната ефективност и практическа приложимост, чрез числено тестване на базата на данни за реални проблеми.

Резюме

Дисертационният труд е структуриран в осем глави. В *Глава 1* е представен обзорец анализ на методите за решаване на едно- и многокритериални оптимизационни задачи. За решаването на различни класове оптимизационни задачи съществуват множества от разработени и широко използвани алгоритми и методи, най-често използваните от които, са анализирани в тази глава. Много от тези методи, които са доказали във времето своята ефективност, са реализирани в утвърдени в практиката софтуерни системи, които могат да бъдат използвани за получаване на числени решения на конкретни практически проблеми. В главата е направен и анализ на някои от най-често срещаните проблеми, свързани с използването на комбинаторната оптимизация. Горните съображения определят основното направление на изследванията в областта на инженерната оптимизация, като разработване на подходящи оптимизационни модели и алгоритми за използването им при решаване на различни класове инженерни проблеми.

В *Глава 2* са описани резултатите от използването на предложения обобщен оптимизационен модели за оптимално проектиране на модулни инженерни системи, вземащи предвид съществуващите взаимовръзки между модулите. Основен проблем при проектирането на този клас системи е определянето на оптимална комбинация от модули, удовлетворяваща както зависимостите между модулите, така и различни потребителски изисквания към параметрите на проектираната система. Като пример, достатъчно добре илюстриращ проблемите от този клас, е използван примера за проектиране на персонални компютри (ПК). Отчитайки специфичните особености на проектирането на ПК, са предложени алгоритми за рационално проектиране, подходящи за реализиране в съответна програмна система. На база на предложените математически модели са формулирани едно- и многокритериални оптимизационни задачи, позволяващи удовлетворяване на различни потребителски изисквания. Предложен е обобщен алгоритъм за итеративно и рационално конфигуриране на ПК. Този алгоритъм включва различни сценарии за конфигуриране на ПК в зависимост от потребителските изисквания, като за всеки сценарий са представени съответни алгоритмични реализации.

Глава 3 разглежда оптималното проектиране на модулни инженерни системи, вземащи предвид както съществуващите взаимовръзки между модулите, така и условията на експлоатация. Този клас проблеми е илюстриран на примера на уреди за нощно виждане (УНВ), чиито параметри зависят освен от параметрите на използваните

модули и от външните условия на наблюдение. Отчитайки спецификата на подобни класове проблеми, предложени са детерминиран, стохастичен и обобщен модел за проектиране, позволяващи да се симулират различни устройства и да направи предварителна оценка на параметрите на проектираната система/устройство. Предложен е обобщен алгоритъм за предварителна теоретична оценка, в който са интегрирани предложените методи за проектиране на УНВ (итеративен и рационален). Описани са два многокритериални модела за определяне на външни условия на наблюдение, при които се постига една и съща стойност на зададен параметър на устройството/системата.

В *Глава 4* са представени получените резултати в областта на оптималното проектиране на инженерни системи, вземащи предвид съществуващите взаимовръзки, външните експлоатационни условия и икономическата ефективност. Този клас проблеми е илюстриран на примера на проектирането на ветроенергиен парк (ВЕП). ВЕП се характеризират със зависимости между модулите (тип, брой и разположение на турбините), наличие на експлоатационни условия (площ на ВЕП и посока на вятъра) и осигуряване на определено ниво на икономическа ефективност. Предложеният математически модел отчита както наличната площ на парка и посоката на вятъра, така и икономическата ефективност. Моделът е използван за формулиране на едно- и многокритериални оптимизационни задачи, в резултат на които се определя едновременно типа, броя и разположението на вятърните турбини за проектирания ВЕП. Предложен е алгоритъм за приложение на многокритериалния подход при проектиране на ВЕП.

В *Глава 5* са описани предложените модели и алгоритми за оптимално планиране на ресурси. Формулиран е обобщен модел за класа проблеми, касаещи определяне на оптимално разписание на зависимата обработка на детайли върху множество машини. Предложен е паралелен алгоритъм за определяне на оптимално разписание с минимална продължителност при смесена (независима и зависима) обработка на детайли върху множество машини. Предложен е и обобщен модел за класа проблеми, касаещи оптимално линейно разкрояване на материали. Моделът цели определянето както на оптималната дължина на заготовките, така и на оптималните схеми за разкрояване за всяка отделна заготовка. Друг клас от проблеми на оптималното планиране е необходимостта от определяне на оптималния брой на необходимия персонал и неговия график за работа, осигуряващ минимален непроизводителен престой. За целта е предложен съответен обобщен модел, приложим

към този клас проблеми. Моделът е използван за формулиране на оптимизационна задача, включена в алгоритъм за определяне на оптимален брой оператори. Използваният критерий за минимален престой на операторите, може да бъде модифициран при необходимост, така че да взема предвид и други изисквания при определяне на оптималния брой на операторите.

В Глава 6 се разглеждат проблемите на оптималното управление на предсказващото поддържане на инженерни системи. Предложени са модели и алгоритми за оптимално предсказващо поддържане на инженерни системи. Формулирани са два обобщени оптимизационни модела, приложими при определяне на стратегия за поддръжка на машина и при определяне на стратегия за ремонт или замяна на отделни детайли на машината. Тези модели са използвани в предложен алгоритъм за определяне на оптимална стратегия за поддръжка. Предложена е система за подпомагане вземането на решения за целите на предсказващото поддържане, която интегрира текущата информация от мониторинг на състоянието на оборудването и използва подходящо моделиране и управление на наличната информация. Решенията за техническо обслужване се базират на актуалните данни от сензорите, като се отчита съществуващата неопределеност и се използва активно участие на лице, вземащо решение. За целите на мониторинга на състоянието при предсказващото поддържане е предложен обобщен модел за определяне оптималното разположение на сензори за мониторинг на състоянието. Моделът е използван за формулиране на еднокритериална оптимизационна задача, определяща разположението на предварително зададен брой сензори. Формулиран е и многокритериален обобщен модел за едновременно определяне на броя и разположението на сензорите.

В Глава 7 са описани модели и алгоритми за обоснован избор на алтернативи. Предложен е алгоритъм за класиране на алтернативи по предварително определени критерии. Идеята на предложения подход се свежда до решаване на многокритериални задачи, определящи подредбата на алтернативите, съгласно задаваните предпочитания на ЛВР. Алгоритъмът е използван за класиране на устройства и за класиране на държави по определени от Световната Банка икономически индикатори. Предложен е и оптимизационен модел, позволяващ определяне на k най-добри алтернативи, чрез решаване на многокритериална оптимизационна задача. Проблемът за обоснован избор, зависещ както от параметрите на системата (устройството), така и от външни експлоатационни условия, е тестван за примера на уреди за нощно виждане. Формулиран е математически модел, в който при зададени експлоатационни условия,

се избира уред, удовлетворяващ зададени потребителски изисквания към параметрите на уреда. Този модел позволява добавяне на ограничения за параметрите. В инженерната практика, често се налага вземане на решения с отчитане мнението на група от експерти. Предложен е обобщен модел за групово вземане на решения. Характерна особеност на модела е, че мненията на експертите от групата за важността на критериите и оценките за всяка от алтернативите, по отношение на тези критерии, се вземат предвид едновременно.

В *Глава 8* са представени разработените изследователски прототипи на софтуерни инструменти за подпомагане вземането на решения. Разработени са уеб базирани приложения за подпомагане проектирането на ПК, подпомагане проектирането на УНВ (очила, прицели и бинокли), за подпомагане проектирането на ВЕП и за подпомагане на груповото вземане на решения. Описани са специализирани софтуерни системи, използващи оптимизационен модул LINDO API за определяне на оптимални разписания и система за оптимално проектиране на ВЕП. Представени са и модели за оптимално разписание и за линейно разкрояване, използващи оптимизационен модул в средата на електронни таблици.

Глава 1:

Едно- и многокритериална оптимизация и вземане на решения при инженерни приложения

1.1. Оптимизационен модел

Основен принцип при построяване на математически модел е да се определят някои от най-важните зависимости и получената схема на взаимодействия да се опише с помощта на адекватен математически апарат. Необходимо е да се определят количествените връзки, параметрите на решението и изходните резултати, които да се използват като индикатор за ефективността на решението. Колкото по-добре математическият модел отразява спецификата на изследваните обекти или процеси, толкова по-полезни ще са получените резултати и препоръки за управление на тези обекти и процеси. Тъй като в реалността изискванията към моделите обикновено са противоречиви, то правилното моделиране на проблемите е почти половината от решаването им.

На Фиг. 1.1. е илюстрирана необходимостта от използването на абстракции, при разработването на модели (Taha, 2010).



Фиг. 1.1. Необходимост от абстракция при построяването на модели

Предположенията за реалния свят се формализират чрез определяне на основни (доминиращи) променливи, описващи поведението на реалната система. Поради това, моделът представлява абстракция на реалния свят и чрез езика на математиката се описва не поведението на реални системи, а предположението за тяхното поведение. В

резултат се достига до концептуален модел на изследваната система, който в съдържателна форма описва състава на системата, нейните компоненти и връзките между тях, основните показатели за качество, променливите величини, както и взаимовръзките с показателите на качеството на системата, както и списък на стратегиите за управление (или решенията), който трябва да се определят в резултат на решаването на поставената задача. Най-общо, концептуалният оптимизационен модел се характеризира с три основни компонента: *целева функция, променливи и ограничения*. Целевата функция или критерият за ефективност се задава като функционална зависимост, изразяваща какво точно трябва да бъде оптимизирано. Обикновено това е функция на печалбите, които трябва да се максимизират, или функция на разходите, които следва да се минимизират. Моделите могат да имат повече от една целева функция, като в тези случаи е необходимо използването на многокритериалната оптимизация. При формулирането на математическия модел е необходимо да се определят кои съществени параметри на модела могат да се разглеждат като променливи, чиито стойности ще влияят при определяне на търсеното решение. Целта на оптимизацията е да се намерят стойностите на тези променливи за дефинирания модел, според който се генерира най-добрата стойност за целевата функция, при спазване на ограничителните условия за въведените променливи. Почти без изключение във всички модели, ще има някакви гранични стойности, които променливите могат да приемат – напр. цена, брой и т.н. Тези ограничения се изразяват чрез математически изрази, разглеждани като функции на променливите в модела. Прието е тези функции да се дефинират като ограничения, тъй като те реално ограничават допустимите стойности на променливите. Този концептуален модел се използва да се извърши по-нататъшна формализация на оптимизационната задача. За целта е необходимо да се детайлизират следните множества на променливите:

- множество променливи, определени като *управляващи променливи* $X = \{x_j\}$, стойностите на които могат да се избират от лицето, вземащо решение (ЛВР);
- множество променливи, които зависят от избора на стратегия за определяне на решението – т. нар. *изходни променливи* на задачата за вземане на решение/я $Y = \{y_i\}$;
- множество променливи, стойностите на които зависят от *параметрите на системата* $A = \{a_k\}$ – вътрешни променливи и външни променливи, изразяващи *външни параметри за системата* $\Omega = \{w_j\}$.

Целевата функция (критерият за ефективност) се определя в зависимост от приетата стратегия за търсене на оптимално решение, от параметрите на моделираната система и от външните за системата параметри, като $f : XA\Omega \rightarrow Y$. В общия случай,

математическият модел на задачите за вземане на оптимално решение има вида (Таћа, 2010):

$$F = f(X, Y, A, \Omega) \rightarrow \max(\min) \quad (1.1)$$

при ограничения

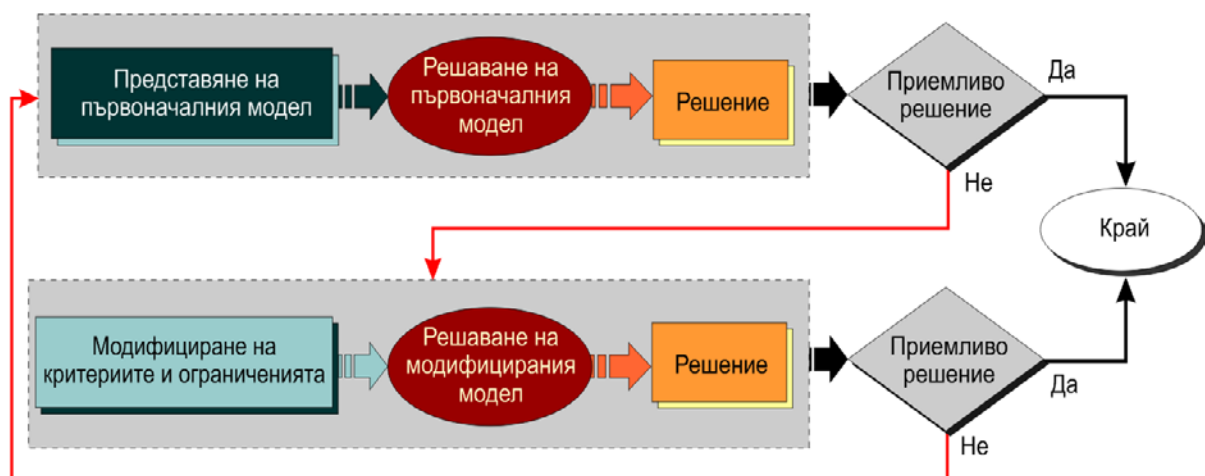
$$g_i(X, Y, A, \Omega) \leq b_i \quad (1.2)$$

$$X_{\min} \leq X \leq X_{\max} \quad (1.3)$$

$$Y_{\min} \leq Y \leq Y_{\max} \quad (1.4)$$

където $f(X, Y, A, \Omega) = [f_k(X, Y, A, \Omega)]$, за $k=1, \dots, K$ – целева функция (критерий за ефективност); $g_i(X, A, \Omega)$ – функция на загубите на i -ия ресурс; b_i – стойност на i -ия ресурс. Много често в реалните инженерни проблеми сред ограниченията (1.3), (1.4) съществуват и ограничения от вида $X \geq 0, Y \geq 0$, изразяващи условия за неотрицателност на променливите.

В голяма част от реалните инженерни проблеми се използва експертизата на лицето, вземащо решение (ЛВР), за интерактивно модифициране или донастройване на първоначалните оптимизационни модели. Обобщеното представяне на този процес е показано на Фиг. 1.2.



Фиг. 1.2. Обобщено представяне на процеса на моделиране

Оптимизационният проблем се решава чрез използване на първоначалния модел. За целта се използва едно- или многокритериална оптимизация, определяща оптимално или Парето-оптимално решение. Чрез използване на инструменти за анализ се определят показателите за ефективност на полученото решение. На базата на този анализ се извършва корекция на пространството на решенията. Модификацията на първоначалния модел се извършва в зависимост от особеностите на използваното математическо моделиране. Възможни са различни подходи на базата на анализа на решенията. Например, използването на математически модели с голяма размерност,

изисква специални подходи за определяне чувствителността на решението спрямо неопределеността на входните данни (Dimov & Georgieva, 2013). Друг възможен подход е оценка на параметрите на първоначалния модел. Първоначалният модел се замества с модифицирания модел и процедурата се повтаря. По време на търсенето на решение междинната информация се съхранява и се използва за подобряване на първоначалния модел. След анализ може да се извърши преглед на коригиращите функции. Както модифицирания модел, така и коригиращите функции са подходящи не само за търсене на оптимално решение, но и за приближаване към оптималното решение. Подобни процедури намаляват изчислителните усилия за решаване на сложни оптимизационни задачи. За намиране на оптимално решение на задачите, в зависимост от вида и структурата на целевата функция и ограниченията, се използват различни методи на математическото програмиране: 1) *Линейно програмиране*, ако функциите $f(X, Y, A, \Omega)$ са линейни относно променливите X ; 2) *Нелинейно програмиране*, ако функциите $f(X, Y, A)$ и/или $g_i(X, Y, A, \Omega)$ са нелинейни относно променливите X ; 3) *Дискретно програмиране*, ако на управляващите променливи X е наложено условие за дискретност, напр. целочисленост x_j – цяло, $j = \overline{1, n}$; 4) *Динамично програмиране* – когато за дадена задача е известно, че оптималното решение може да бъде конструирано на база оптималните решения на поредица подзадачи, при което се избягва необходимостта да се извършват повторно вече направени изчисления; 5) *Геометрично програмиране*, ако целевата функция $f(X)$ и ограниченията $g_k(X)$ могат да се представят като полиноми от вида: $g_k(X) = \sum_{i=1}^n c_i x_1^a x_2^a \dots x_m^a$; 6) *Стохастично програмиране*, ако векторът на външните променливи $\Omega = \{\omega_s\}$ или параметри на системата $A = [a_{ij}]$ са случайни; 7) *Размито математическо програмиране*, когато се налага да се взема решение в условия на неопределеност, т.е. когато някои параметри на системата $A = [a_{ij}]$ и/или външните променливи $\{\omega_s\}$ не са точно известни и ЛВР може или да зададе интервал на техните стойности или да използва свои субективни оценки относно вероятността за появяване на едни или други стойности – т.нар. степени на принадлежност $\mu(a_{ij})$, $\mu_s(\omega_s)$; 8) *Евристично програмиране* – за решаване на задачи, в които е невъзможно да се намери точен оптимум, поради характера на задачите, водещ до огромно количество варианти и значителни изчислителни трудности. В тези случаи се търси удовлетворяващо ЛВР решение, чрез използване на специални техники, които са резултат от натрупан практически опит – т.нар. евристики, позволяващи съществено да се намали броя на разглежданите варианти.

1.2. Методи за решаване на еднокритериални задачи

Методите за решаване на еднокритериалните задачи могат да бъдат класифицирани в два основни класа – *точни* и *приближени* методи.

1.2.1. Точни методи

- **Симплекс методът** е предложен от Dantzig (Dantzig, 1987) и е един от най-широко използваните методи за решаване на линейни еднокритериални задачи. Той е определен като един от върховете 10 алгоритъма на 20 век. Основното свойство на симплекс метода е, че решаването на задачите на линейното програмиране са осъществява итерационно. На всяка итерация алгоритъмът преминава в нова възлова точка в изпъкналия многостен на ограниченията на задачата, която потенциално може да подобри стойността на целевата функция. Този процес на преход от една към друга възлова точка приключва, когато последващо подобрене на стойността на целевата функция е невъзможно. Процесът на реализация на симплекс метода включва голямо количество еднотипни и трудоемки изчисления.

- **Метод на вътрешната точка.** Narendra Karmarkar разработва полиномиален алгоритъм за линейно програмиране, известен още като метод на вътрешната точка (Karmarkar, 1984). Този алгоритъм е крайъгълен камък в линейното програмиране и разбива мита за невъзможност на съществуване на алгоритъм за линейно програмиране, различен от Симплекс метода. Поради съпътстващите го изчислителни трудности не се използва така масово, както Симплекс метода.

От своя страна Симплекс методът е неприложим за решаване на целочислени или смесено-целочислени линейни задачи. Този клас задачи е NP-труден и тяхното решение не може да бъде определено за полиномиално време. Това налага разработването на специални методи за тяхното решаване, най-популярните от които са изброени по-долу:

- **Метод на отсичащите равнини (cutting plane).** Първият общ подход за решаване на целочислената задача на линейното оптимиране е предложен от Гомори (1958 г.) Методът на отсичащите равнини е оптимизационен метод, чрез който итеративно се прецизира допустимото множество или целевата функция с помощта на линейни неравенства, които дефинират равнина, отсичаща текущото решение в пространството на допустимите решения. Тези процедури се използват за намиране решения на смесено-целочислени линейни проблеми, както и за решаване на общи, а не непременно диференцируеми изпъкнали оптимизационни задачи (Balas и др., 1996).

- **Метод на клонове и граници (branch and bound).** Този метод е разработен за решаване на дискретни и комбинаторни оптимизационни задачи (Clausen, 1999). Това е най-разпространеният точен алгоритъм за решаване на задачите на целочислено и смесено-целочисленото програмиране. Разклоняването разделя пространството на решенията на две части, във всяка от които се търси оптималното решение. Известни са различни техники за разклоняване (Wolsey, 1998), като алгоритъмът на клоновете и границите може да се приложи и за нелинейни целочислени задачи.

- **Метод на разклоняване и отсичане (branch and cut).** Това е метод за решаване на целочислени комбинаторни линейни проблеми, когато някои или всички неизвестни величини са ограничени до цяло число (Mitchell, 2002). *Branch and cut* обединява подходите на алгоритмите *branch and bound* и *cutting planes*. В последните години този алгоритъм е особено популярен за решаване на комбинаторни оптимизационни задачи. Главната идея на алгоритъма е да се генерират отсичащи равнини в цялото търсещо дърво на стандартния алгоритъм на клоновете и границите. Целта е във всеки връх на търсещо дърво да се получават по-тесни граници на областта на допустими решения на решаваната задача. Различни техники могат да бъдат използвани за прилагане на отсичащите равнини в алгоритъма (Atamturk & Savelsbergh, 2005).

- **Метод на разклоняване и оценяване (branch and price).** Методът на разклоняване и оценяване е също метод на комбинаторната оптимизация за решаване на линейни целочислени проблеми и смесено-целочислени линейни проблеми с много променливи. Същността му се състои в използване на алгоритъм за генериране на колони за решаване на линейна задача към общата схема на алгоритъма на клоновете и границите. Следваща стъпка в развитието на дървовидно-търсещите алгоритми е комбинирането на алгоритъма на разклоняване и отсичане и алгоритъма на разклоняване и оценяване в т. нар. алгоритъм на разклоняване, оценяване и отсичане (*Branch Price and Cut*), при който динамично се генерират отсичания и променливи (Desrosiers, 2010).

1.2.2. Приближени методи

Целочислените задачи са NP-пълни задачи и намирането на оптимално и дори само на допустимо решение за задачи с по-големи размери е свързано с големи изчислителни трудности. Често пъти е по-важно да се намери бързо приемливо решение, отколкото да се чака дълго за достигане до оптималното решение. В практиката, при описанието на модела на задачата могат да се въвеждат някои гъвкави ограничения, които да се

променят слабо. При точните алгоритми, дори и при малка промяна на ограничение се налага изцяло пререшаване, а това отнема доста време и може да е доста скъпо за реални приложения. Приближените алгоритми не са така чувствителни към малки промени в някои ограничения. Някои от тях решават проблема последователно, като го разделят на части и тогава не се налага изцяло пререшаване на задачата. Приложение намират приближените алгоритми за намиране на подходящо начално решение или за стесняване на допустимата област на решенията и да насочват търсенето на оптимално решение. За решаването на големи практически целочислени задачи, без да се гарантира оптималност на решението, са разработени огромен брой приближени алгоритми (Silver, 2004). Повечето приближени алгоритми са предназначени за задачи на комбинаторната оптимизация, фокусирани върху специфични приложения в областта на разписанията, логистиката, маршрутизацията и т.н. При тези алгоритми се използват предимствата на специфичната структура на областта на допустимите решения на тези класове задачи, при което се постига много по-добра ефективност на работата на алгоритъма в сравнение с тази на алгоритми, които са предназначени за решаване на общата целочислена задача. В зависимост от качеството на получените с тях решения, приближените алгоритми могат да се разделят на следните три групи: 1) приближени алгоритми с произволно зададена точност (абсолютна или относителна); 2) приближени алгоритми с предварително определена точност; при което грешката на приближение не клони към нула; 3) евристични алгоритми – на базата на експерименти или други експертни оценки се предполага, че с голяма вероятност ще намери добро решение на задачата с приемливи изчислителни ресурси, но не може да се гарантира математическа оценка за точността им.

Основните евристични стратегии, които използват приближените алгоритми могат да се определят като конструктивни и локално-подобряващи алгоритми.

- **Конструктивни алгоритми.** При конструктивните алгоритми решението се генерира стъпка по стъпка като се използват данните на задачата. Обикновено не е намерено решение, докато алгоритъмът не е завършил работата си (за разлика от подобряващите алгоритми). Към този клас алгоритми се отнася т. нар. *greedy* алгоритъм, където на всяка стъпка се включва следващ елемент на решението, избран така, че да се получи най-добро локално подобрение. Едно от най-известните приложения на *greedy* алгоритъма е за решаване на задачата за търговския пътник (Golden и др., 1980). Тези алгоритми са от най-бързите приближени алгоритми, но често водят до твърде лоши решения. Поради това, често в конструктивните алгоритми се включват процедури от вида “поглед напред - look ahead” за оценяване бъдещите последици от текущия избор (Atkinson, 1994).

• **Локално-подобряващи алгоритми.** Основната идея на локално-подобряващите алгоритми е много проста – започва се с допустимо решение на задачата, получено най-често с конструктивен алгоритъм. Оценяват се допустимите решения в околността на текущото решение и ако някое от тях е по-добро от текущото решение, то става новото текущо решение на задачата и се изследва неговата околност и т.н., докато не може да се намери ново подобряване и текущото решение на тази стъпка е локален оптимум. Очевидна е важноста на начина на дефиниране на околност на точка (решение), като е важен и въпросът колко голяма да е тази околност, т.е. на какво разстояние от текущата точка да се изследват допустимите решения (Michalewicz & Fogel, 2000). Недостатък на тези алгоритми е, че може да се гарантира само локален оптимум. В последните години се развиха нов тип приближени алгоритми, в които се използват комбинации от различни евристични подходи, известни като метаевристики в смисъл на търсене на по-високо ниво (Glover, 1986). Към този клас приближени алгоритми се отнасят симулираното закаляване (*simulated annealing*) (Dowsland, 1993), табу търсенето (*tabu search*) (Glover & Laguna 2002), популационно-базирани алгоритми (еволюционни и генетични алгоритми, оптимизация на мравките) (Eiben & Smith, 2003; Dorigo & Stutzle, 2004) и др.

1.3. Методи на многокритериалната оптимизация

1.3.1. Обща формулировка на задачата на многокритериалната оптимизация

Реалните инженерни проблеми обикновено се характеризират с наличието на повече от един критерии, които често са противоречиви и несъизмерими един с друг. Това поставя въпроса за това, как да се подобряват различните критерии, за получаване на крайно оптимално решение. Общата формулировка на задачата на многокритериалната оптимизация има вида (Marler & Arora, 2004):

$$\min \mathbf{F}(\mathbf{x}) = [F_1(\mathbf{x}), F_2(\mathbf{x}), \dots, F_k(\mathbf{x})]^T \quad (1.5)$$

при ограничения:

$$g_j(\mathbf{x}) \leq 0, \quad j = 1, 2, \dots, m, \quad (1.6)$$

$$h_l(\mathbf{x}) = 0 \quad l = 1, 2, \dots, e, \quad (1.7)$$

където $\mathbf{F}(\mathbf{x})$ е вектор на критериите (целевите функции), а k е броят на критериите, m е броят на ограниченията, представени като неравенства, e е броят на ограниченията във формата на равенства, $\mathbf{x} = (x_1, x_2, \dots, x_n)^T \in E^n$ е векторът на променливите и n е броят на независимите променливи x_i .

В задачите на многокритериалната оптимизация е необходимо едновременно да се оптимизират няколко критерия. В общия случай не съществува единствена алтернатива, която да оптимизира всички критерии. Всяка алтернатива, подобряваща определен критерий може да бъде търсена като решение на многокритериалната задача, но за крайно решение на задачата е необходимо да бъде избрана една най-предпочитана алтернатива. За избор на най-предпочитана алтернатива е необходима допълнителна информация, която се задава от лицето, вземащо решение (ЛВР). Информацията, която ЛВР задава, отразява неговите предпочитания по отношение на качеството на търсената алтернатива.

1.3.2. Парето-оптималност

Идеята за Парето-оптималността има значително въздействие и в областта на инженерните науки. Парето-оптималността се използва, за да се опише набор от възможности за избор, известни като Парето-множество или Парето-фронт. Определянето на множеството от тези Парето-ефективни избори позволява на инженерите да изразят своите съображения за крайното решение, без да се има предвид целият набор от възможни решения. За разлика от еднокритериалната оптимизация, решението на многокритериалния проблем е по-скоро концепция, отколкото дефиниция. Обикновено не съществува едно глобално решение и често е необходимо да се определи набор от точки, които да удовлетворяват предварително зададеното определение за оптималност. Концепцията за определянето на оптимална точка е Парето-оптималността, която се определя като (Marler & Arora, 2004; Pareto, 1906):

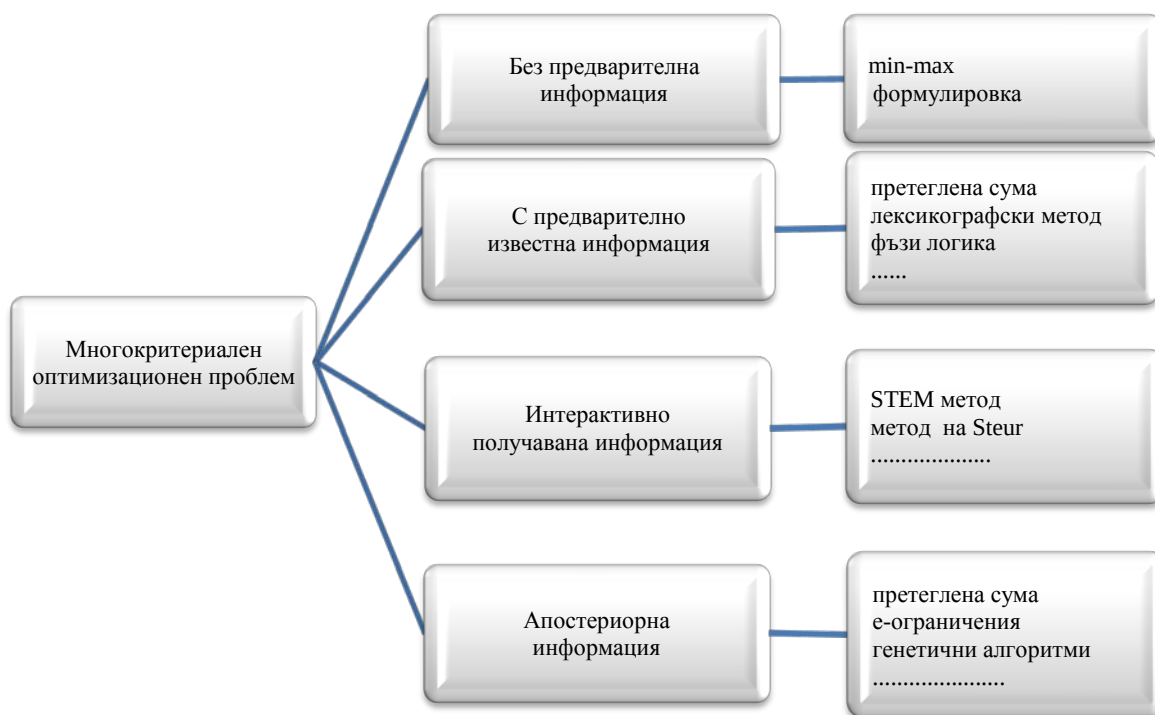
Парето-оптимално решение (Marler & Arora, 2004) – една точка $x^* \in X$, е Парето-оптимална ако не съществува друга точка $x \in X$ такава, че $F(x) \leq F(x^*)$, и $F_i(x) < F_i(x^*)$ за поне една функция. Казано с други думи, една точка е Парето-оптимална, ако няма друга точка, която да подобрява поне една целева функция, без да влошава друга целева функция. Всички Парето-оптимални точки лежат на границата на допустимото критериално пространство (Athan & Papalambros, 1996; Chen, и др., 2000).

Слабо Парето-оптимално решение (Marler & Arora, 2004): Една точка $x^* \in X$, е слабо Парето-оптимална ако няма друга точка, $x \in X$, такава, че $F(x) < F(x^*)$. Или с думи, една точка е слабо Парето-оптимална, ако няма друга точка, която да подобрява всички целеви функции едновременно. Парето-оптималните точки са слабо Парето-оптимално, но слабо Парето-оптималните точки не са Парето-оптимални.

Задачата на многокритериалната оптимизация е задача за многокритериален избор. Тази задача не притежава оптимално решение в смисъла на решението на еднокритериална оптимизационна задача. Поради това е необходимо от слабо Парето-оптималните или от Парето-оптималните решения да се избере едно решение, което най-добре отговаря на глобалните предпочитания на ЛВР. Този избор е субективен и зависи изцяло от ЛВР. Множеството на Парето-оптималните решения на задачата на многокритериалната оптимизация е подмножество на множеството на слабо Парето оптимални решения на съответната задача.

1.3.3. Класификация на методите за решаване многокритериални задачи, в зависимост от начина на задаване на предпочитанията

Методите за решаване на многокритериалната оптимизационна задача могат да се групират в четири класа (Фиг. 1.3), в зависимост от това, кога ЛВР задава своите предпочитания към отделните критерии – никога, преди, по време или след процедурата за оптимизация (Andersson, 2000).



Фиг. 1.3. Методи на многокритериалната оптимизация

Информацията, която ЛВР задава, изразява неговите предпочитания към търсеното решение (т.е. определя важността на отделните критерии). При *апериорните* алгоритми всички предпочитания са известни още в началото на процеса на избор на решение. Използваните алгоритмични техники целят намирането на цялото Парето-

множество или подмножество от него, на базата на тези *априори* зададени предпочитания. При *апостериорните* алгоритми се генерира множеството на всички Парето-оптимални решения за конкретната задача, след което това множество се анализира в съответствие с предпочитанията на ЛВР. Много приближени алгоритми са реализирани в съответствие с този подход. *Интерактивните* алгоритми извеждат Парето-оптимални решения на базата на прогресивно задавани от ЛВР предпочитания по време на процеса на избор на решение.

ЛВР може да изрази своите предпочитания между отделните Парето-оптимални решения чрез стойностите на параметрите на скаларизиращата задача. В крайна сметка, ЛВР трябва да направи избор на най-предпочитано решение и е отговорен за този избор.

1.3.4. Скаларизационен подход за решаване на задачи на многокритериалната оптимизация

Двата основни подхода при решаването на задачи на многокритериалната оптимизация са: 1) апроксимационен подход (Ehrgott & Wiecek, 2005); 2) скаларизационен подход (Miettinen, 1999). Първият подход апроксимира множеството на Парето-оптималните решения (Deb, 2001; Coello и др., 2002; Ehrgott & Wiecek, 2005), като повечето от методите са итеративни и пораждат точки или обекти, апроксимиращи това множество (Vassilev и др., 2005). Някои от тези методи (Ehrgott & Wiecek, 2005) са прецизно обосновани от гледна точка на теоретични доказателства и оптималност, докато други (Coello и др., 2002) са евристични и често нямат теоретична обосновка. За голяма част от задачите на многокритериалната оптимизация не е лесно да бъде направено точно описание на множеството на Парето-оптималните решения, което обикновено включва много голям или безкраен брой решения. Например, решенията, получени с помощта на генетичните алгоритми, в общия случай са приближени Парето-оптимални решения. ЛВР е изолиран по време на процеса на определяне на апроксимиращото множество, а накрая му се представя за оценяване и избор твърде голямо множество от решения, което е сравнително трудна задача на многокритериалния анализ. Голяма част от тези решения не са необходими на ЛВР, а най-предпочитаното решение може дори да не бъде намерено.

Вторият подход към решаването на задачите на многокритериалната оптимизация е т.нар. *скаларизация*. Скаларизацията представлява трансформация на многокритериалната оптимизационна задача в съответна еднокритериална задача или в множество от еднокритериални оптимизационни задачи, наречени скаларизиращи задачи. Основно свойство на всяка скаларизираща задача е, че всяко нейно оптимално

решение е Парето-оптимално (слабо Парето-оптимално) решение на съответстващата задача на многокритериалната оптимизация. Скаларизиращият подход към многокритериалните задачи позволява използването на добре изучените и развити теория и методи на еднокритериалната оптимизация. Известни са различни скаларизиращи техники, водещи до: задача на *претеглената сума*, задача на *е-ограниченията*, задача на *отправната точка*, *класификационно-ориентирани задачи* и т.н. (Miettinen & Makela, 2002).

За да бъде една еднокритериална задача скаларизираща задача на изходната многокритериална задача, тя трябва да отговаря на две изисквания (Sawaragi и др., 1985): 1) чрез промяна на стойностите на параметрите на скаларизиращата задача трябва да може да се намери всяко Парето-оптимално решение на многокритериалната задача; 2) за всяко Парето-оптимално решение на многокритериалната задача трябва да могат да се намерят подходящи стойности на параметрите на скаларизиращата задача, така че нейното оптимално решение да е точно това Парето-оптимално решение на многокритериалната задача.

Многокритериалната оптимизация по своята същност включва вземане на решения и сравняване на различните критерии, които често са под формата на математически функции. Въпреки това, тези функции могат да имат различни размерности и/или значително различни стойности, което прави сравняването им трудно, а понякога дори и невъзможно. Освен това, когато се използва скаларизационния подход, е възможно една и съща целева функция с относително големи стойности да доминира в агрегираната целева функция, като по този начин се обезсмислят усилията за използването на много критерии (Marler & Arora, 2005). Следователно, необходимо е така да се трансформират целевите функции, че те да са съизмерими, т.е. да имат сходни порядъци. Тази процедура се нарича *нормализация* и е удачна за използване както за априорните, така и за апостериорните предпочитания на ЛВР относно важността на критериите.

Един от възможните подходи за нормализация на критериите може да бъде представен по следния начин (Proos и др., 2001):

$$F_i^{transf}(x) = \frac{F_i(x)}{|F_i^{max}|} \quad (1.8)$$

Тази нормализация води до безразмерност на целева функция с горна граница единица (или -1) и неограничена долна граница, като се предполага, че $F_i^{max} \neq 0$.

Разновидност на горния подход за нормализация е следната схема за нормализация (Osyczka, 1978; Salukvadze, 1979):

$$F_i^{transf}(x) = \frac{F_i(x) - F_i^o}{F_i^o}, \quad (1.9)$$

$$F_i^o(x) = \min\{F_i(x) \mid x \in X\} \quad (1.10)$$

При този подход също се осигурява безразмерност на целевата функция, като стойността на долната граница на $F_i^{transf}(x)$ се ограничава до нула, а горната граница е неограничена. Тази трансформация често се нарича и *относително отклонение*.

Друг подход за нормализиране на целевите функции може да бъде реализиран чрез трансформация от вида (Koski & Silvenpoinen, 1987; Chen, и др., 1999):

$$F_i^{transf}(x) = \frac{F_i(x)}{F_i^o}, F_i^o > 0 \quad (1.11)$$

Този подход води получаване на безразмерни стойности на целевата функция с долна граница единица.

Най-често използвания подход за нормализиране на целевите функции, независимо от диапазона им на изменение, се получава чрез следната трансформация (Koski, 1984; Koski & Silvenpoinen, 1987; Rao & Freiheit, 1991):

$$F_i^{transf}(x) = \frac{F_i(x) - F_i^o}{F_i^{\max} - F_i^o} \quad (1.12)$$

При тази трансформация, $F_i^{transf}(x)$ обикновено получава стойност между нула и единица, в зависимост от точността и метода, с които са определени стойностите за $F_i^{\max}(x)$ и $F_i^o(x)$.

В следващата точка са описани най-често използваните скаларизационни методи към многокритериалната оптимизация, намиращи приложение в областта на инженерната практика.

- **Метод на претеглената сума**

Това е най-често използвания метод за решаване на многокритериални проблеми. Методът на претеглената сума е предназначен за решаване на линейни и нелинейни многокритериални задачи. Този метод изисква ЛВР да определи тегловни коефициенти ω_k за важността на всеки от критериите в целевата функция. Общата формулировка има следния вид (Marler & Arora, 2010):

$$\min \sum_{j=1}^k \omega_j f_j(x) \quad (1.13)$$

при ограничения

$$\sum \omega_i = 1, \omega_i > 0 \quad (1.14)$$

$$x \in S \quad (1.15)$$

За да се намерят множество точки от Парето-фронта се решават няколко оптимизационни задачи с различни теглови коефициенти. Този метод е най-интуитивния и директен начин за изразяване на принципа на Парето-оптималността и за получаване на множество точки от Парето-оптималния фронт.

Методът на претеглената сума може да се разглежда както като метод с *априори* (в началото на процеса) зададени предпочитания на ЛВР, така и като метод с апостериорна информация (при други стойности на тегловните коефициенти) относно предпочитанията на ЛВР. На базата на различни тегловни коефициенти използваните алгоритмични техники целят намирането на цялото Парето-множество или на подмножество от него. Този метод е широко използван не само за създаване на множество точки на решенията чрез последователна промяна на коефициентите, но и за осигуряване на една единствена точка на решението, която отразява предпочитанията на ЛВР за конкретно дефинирани стойности на тегловните коефициенти (Marler & Arora, 2010).

- **Метод на компромисното решение**

Методът на компромисното решение цели да минимизира разстоянието до идеална (z^I) или до утопична точка (z^U) (Yu, 1985). Идеалната точка се определя съгласно индивидуалните минимума на всички критерии като:

$$z_j^I := \min z^j(x), \quad x \in X \quad (1.16)$$

- **Целево програмиране**

Специален подход към многокритериалната оптимизация е целевото програмиране, датиращ от началото на 60-те години (Charnes & Cooper 1961; Ignizio, 1976; Lee 1972). Понастоящем съществуват по-нови варианти на този метод (Steuer 1986; Tamiz и др., 1998.) При него за всеки от критериите в целева функция се задава стойност (goal) от ЛВР, като общата идея е да се минимизира отклонението от зададените цели. Обикновено не съществува една точка, която да удовлетворява всички цели. Различните видове методи на целевото програмиране използват различни начини, за да определят коя точка най-добре удовлетворява зададените утопични стойности.

- **Метод на е-ограниченията**

Методът на е-ограниченията е предназначен за решаване на нелинейни многокритериални задачи. При този метод един от критериите се избира за оптимизация, а всички останали се трансформират в ограничения, като се задава съответна граница за всеки от тях. Общата формулировка има следния вид (Mavrotas, 2009):

$$\max\{f_i(x)\} \quad (1.17)$$

при ограничения

$$f_j \geq e_j, \quad j = 1 \dots k, \quad j \neq i, \quad (1.18)$$

$$x \in S \quad (1.19)$$

където x е вектор на променливите, $f_i(x)$ са критериите и S е допустимата област. Различни точки от Парето-фронта могат да бъдат получени чрез промяна в стойностите на ограниченията.

- **Лексикографски метод**

Когато предпочитания на ЛВР към критериите са предварително известни се използва т.нар. *лексикографски* метод. Според този метод, критериите се подреждат по степен на важност, т.е. ЛВР нарежда критериите според неговите предпочитания и ги оптимизира един по един, като последователно се решават съответните еднокритериални оптимизационни проблеми, формулирани като (Marler & Arora, 2004):

$$\min F_j(x), \quad x \in X, \quad j = 1, 2, \dots, k \quad (1.20)$$

при ограничения

$$F_j(x) \leq F_j(x_j^*), \quad j = 1, 2, \dots, i-1, \quad i > 1, \quad i = 1, 2, \dots, k \quad (1.21)$$

В тази формулировка, i представлява позиция на функцията в предпочитаната последователност, а $F_j(x^*)$ представлява оптимума на j -ия критерий, намерен в j -тата итерация. След първата итерация ($j = 1$), не е задължително $F_j(x^*)$ да е същото като независимия минимум на $F_j(x)$, тъй като се въвеждат нови ограничения. При използването на лексикографския метод, решението на всеки отделен еднокритериален проблем дава мярка за ограничаване на този критерий, използвана за определяне на подходящо ограничение в следващия етап, когато следващият по важност критерий се оптимизира и процесът се повтаря. Окончателното решение се определя на последния етап от процедурата.

1.4. Комбинаторна оптимизация

Много от проблемите, свързани с практически приложения имат дискретна природа (Bjorndal и др., 1995). Комбинаторната оптимизация (КО) изучава проблеми, характеризирани се с краен брой възможни решения. Въпреки че, по принцип оптималният резултат към такъв ограничен проблем може да бъде намерен чрез просто преглеждане на всички алтернативи, в действителност този вариант често е невъзможен, особено за практични проблеми с реални размерности, където броят на възможните решения може да бъде достатъчно голям. В действителност се използват

структурните свойства на проблемите за намиране на точни или приблизителни алгоритми за решаването им. Като цяло, проблемите на КО се класифицират според тяхната изчислителна сложност. Този анализ не отразява винаги актуалната изчислителна трудност. Поради тази причина, истинската трудност на проблемите е тази, която води до развитието на различни подходи за решението им. Известно е, че могат да се намерят добри решения за сложни или лошо дефинирани проблеми при непълна информация. Този подход за намиране на решения на базата на субективни експертни оценки или на базата на опита се нарича *евристика* или *евристичен* метод. Евристичните методи не могат да докажат и да гарантират оптималност на получените решения. Във всяка област на решаването на математически проблеми, евристичните методи имат съществено значение. В комбинаторната оптимизация евристичните методи играят важна роля, тъй като позволяват получаване на квази-оптимални решения за приемливо време, като моделират процеса, чрез който човек достига до решение. При използването им трябва да се има предвид, че съществуват мощни моделиращи инструменти, които могат да се използват при комбинаторната оптимизация (смесено целочислено програмиране, графи, и др.) и които дават достатъчни възможности за моделиране на реални проблеми с висока степен на точност. При това, наличието на софтуерни системи за решаване на смесено-целочислени проблеми, към които се свеждат задачите на КО, позволява получаването на доказано оптимални решения. От друга страна, съществуват реални проблеми, за които времето за намиране на това решение е неприемливо голямо и за които съкращаването на това време чрез използване на евристични техники, водещи до достатъчно добри решения, е препоръчително.

В тази връзка, представлява интерес определянето на сложността на най-лошия случай на проблема с използване на евристичните методи. Определя се една горна граница на времето за получаване на решение на проблем с размерност n като функция от n . Понякога обаче съществува несъответствие между теоретичната сложност на най-лошия случай и времето за решаване на средно-сложни проблеми. Една друга мярка за сложността е разглеждането на някакво разпределение на проблемите и очакваната им сложност. Повечето евристики се характеризират с полиномиална сложност. Напредъкът в областта на компютърните изчислителни възможности е по-важен за евристичните методи, отколкото за точните методи, тъй като позволява използване на сложни евристики за решаване на реални проблеми с голяма размерност. Тъй като евристичните методи не гарантират оптимално решение, то се въвежда друга важна

мярка – качество на решението. В идеалния случай, то може да се оцени като се знае една горна граница на решението (оптималното решение) и се определи разликата между нея и полученото решение. Тази граница може да се определи чрез други точни методи или чрез други евристични методи. Много евристични методи имат подобна структура, което води до разработване на обобщени мета-евристики като “*симулирано закаляване*” (Aarts & Korst, 1989), “*табу търсене*” (Glover, 1989) и “*генетични алгоритми*” (Goldberg, 1969). Тези мета-евристики осигуряват алгоритми със съществено участие на потребителя при дефинирането на сложността. Потребителят е решаващ при определянето на това дали времето за решение или качеството на решението е по-важно.

Дървовидните стратегии за търсене играят съществена роля във всеки точен алгоритъм и различните им приложения могат да водят до различна ефективност. При това, ограничаващите процедури играят ключова роля във всеки такъв метод. Ограничаващата процедура се базира на точното решение на един модифициран проблем, който се получава от оригиналния проблем чрез премахване или опростяване на някои от ограниченията му (Bjorndal и др., 1995). Получената чрез този модифициран проблем граница на решението е добра, но за сметка на увеличаване на изчислителните усилия. Класическите „*методи на граници и разклонения*” се базират на модифицирани проблеми, които в много случаи се решават лесно. За някои класове задачи като “*задача за раницата*”, “*несиметричен проблем на търговския пътник*”, “*разписания за една машина*” те осигуряват граница, близка до оптимума. Съществуват обаче проблеми, за които разликата между оптимума и границата е незадоволително голяма. Целта на “*многостенната*” комбинаторика е да осигури близки до оптимума граници чрез методите на линейното програмиране. Ако комбинаторният проблем се дефинира като проблем на целочисленото линейно програмиране, то оптималното решение съответства на “най-добрия” връх от изпъкналия многостен на допустимите точки, описан чрез краен брой линейни ограничения. Използвайки теоретичните резултати за линейното програмиране и наличните ефективни алгоритми могат да се определят достатъчно добри граници за решението. Има два основни проблема при решаване на сложни задачи от комбинаторната оптимизация като задачи на линейното програмиране. Първо, трудно е да се формулират необходимите линейни ограничения. Второ, броят на тези ограничения е експоненциален и не е ясно как да се избере полиномиален брой от тях.

1.4.1. Някои типични приложения на комбинаторната оптимизация

Когато се оптимизират системи, създадени от човека, почти винаги възниква дискретност. Това се дължи на много фактори като: стандартизация и унификация, серийност на производството, създаване на големи структури, в които процесите са прекъснати и взаимодействащи си – транспорт, комуникации и др. В тези случаи възникват т. нар. дискретни и в частност целочислени оптимизационни задачи. Изискването за целочисленост на променливите е отличителна особеност на задачите на дискретна комбинаторна оптимизация. Това позволява представяне на неделимост на обектите или изброимост на множество алтернативи, което значително повишава адекватността на математическото моделиране на реални проблеми. Някои типични дискретни комбинаторни проблеми са описани по-долу.

- **Проблем за раницата**

Много приложения могат да се формулират като *проблем за раницата*. Също така различни проблеми на целочисленото програмиране могат да бъдат модифицирани (чрез премахване или отслабване на ограничения) до проблем за раницата. Семейството на проблемите за раницата изисква избор на едно подмножество от n зададени обекта, така че да се максимизира печалбата (броя, обема, теглото и т.н.) без да се надвиши определен капацитет (обем на раницата):

$$\max \left\{ \sum_{j=1}^n c_j x_j : \sum_{j=1}^n a_j x_j \leq b, 0 \leq x_j - \text{цели}, j = 1, \dots, n \right\} \quad (1.22)$$

където: c_j – цена на обекта, a_j – обем на обекта, b – капацитет (обем на раницата).

Така представената формулировка представлява целочислена задача за раницата, но има различни типове задачи на проблема за раницата. Например, ако $x_j \in \{0,1\}$ – двоична задача (Balev и др., 2008; Kong и др., 2015), а при наличие на повече ограничения (напр. по тегло) – за многомерна задача за раницата. Най-обобщената форма на задачата за раницата с много ограничения е всъщност обобщен проблем на целочисленото програмиране с положителни коефициенти. Въпреки, че от теоретична гледна точка тези проблеми са NP-трудни, някои от тях могат да бъдат решени сравнително бързо поради тяхната специфична структура. Най-важното свойство на проблема за раницата е, че близка до оптимума граница може да бъде получена чрез решаване на съответния модифициран непрекъснат проблем (с изпуснато условие за целочисленост). Наличието на близки до оптимума и бързо намирани горни граници за решението прави възможно разработването на ефективни алгоритми от типа “разклонения и граници”.

За всички проблеми за раницата са разработени ефективни редуциращи алгоритми (като специални техники на “разклонения и граници”), позволяващи да се фиксират някои променливи на техните оптимални стойности и да се намали значително размерността на проблема (Bjorndal и др., 1995).

- **Проблем на търговския пътник**

Може би най-известният комбинаторен оптимизационен проблем е проблемът за търговския пътник. Търговски пътник трябва да обходи N града, като тръгне от един град, приет за начален и се върне в него, без да преминава два пъти през един и същ град, като цената на транспортните разходи да е минимална. Целта е да се определи маршрутът на търговския пътник, ако са известни пътните разходи между градовете, (когато такъв път съществува).

$$\min \left\{ \sum_{i,j=1}^n c_{i,j} x_{i,j} \right\} \quad (1.23)$$

$$\sum_{i=1}^n x_{ij} = 1, \quad j = 1, \dots, n \text{ – във всеки връх се влиза само веднъж} \quad (1.24)$$

$$\sum_{j=1}^n x_{ij} = 1, \quad i = 1, \dots, n \text{ – от всеки връх се излиза само веднъж} \quad (1.25)$$

$$x_{ij} = \begin{cases} 1 & \text{– от връх } i \text{ отива в } j \\ 0 & \text{– от връх } i \text{ неотива в } j \end{cases} \quad (1.26)$$

$$u_i - u_j + nx_{ij} \leq n-1, \quad i, j = 2, \dots, n \quad (1.27)$$

където n са цели порменливи, $u_i \geq 0, i=1, \dots, n$.

Проблемът за търговския пътник е NP-пълна задача и поради това се разработват различни алгоритми за неговото решаване (Laporte, 1992; Battarra и др., 2014; Mahi и др., 2015). Този проблем е част от един по-общ клас, известен като *оптимална маршрутизация*. Проблемите, свързани с определянето на маршрути са важни при реализирането на транспортни услуги, дистрибуция и логистика. Типични примери освен проблемът за “търговския пътник” са проблемите за “маршрути за превозни средства”, “най-къси пътища”, заедно с много техни варианти. Проблемите на търговския пътник са едни от най-добре изучените и са важен модел за тестване на нови алгоритмични идеи (Lawler, и др., 1985), докато алгоритмите за определяне на маршрути са едни от най-динамично развиващите се.

- **Задача за оптимално разкрояване**

Задачата за оптималното разкрояване е NP-пълна оптимизационна задача, която възниква в различни области на индустрията. Проблемът за оптималното разкрояване

може да бъде представен като целочислен линеен модел, формулиран за първи път от Канторович (Kantorovich, 1960).

$$\min \sum_{j=1}^k b_j \quad (1.28)$$

$$\sum_{i=1}^p a_i x_{ij} \leq y_j l_j, \quad j = 1, \dots, k \quad (1.29)$$

$$\sum_{j=1}^k x_{ij} = n_i, \quad i = 1, \dots, p \quad (1.30)$$

където $b_1, \dots, b_k$ – заготовки с еднаква дължина, $a_1, \dots, a_p$ – елементи на дължините, които трябва да се отрежат, а $n_1, \dots, n_p$ – брой на елементите, x_{ij} – променливи за броя елементи от типа $i = 1, \dots, p$, които ще се отрежат от заготовка $j = 1, \dots, k$ и $y_j \in \{0, 1\}, j = 1, \dots, k$.

Проблемът за разкрояване може да бъде двумерен, когато е необходимо да бъдат изрязани правоъгълни (или по-общо) форми от изходната заготовка, която е във формата на лист (Haessler & Sweeney, 1991; Malaguti и др., 2014).

Когато целта е да се запълни голямо пространство с определени по-малки форми по най-икономичен (печеливш) начин възникват т. нар. *проблеми на пакетирането*, които могат да се разглеждат като взаимно допълващи се проблеми за разкрояване (Silva и др., 2014).

• Разписания за обработка на детайли

Съставянето на разписания за обработка на детайли на определени машини е в действителност проблем за оптимално приложение на ограничени ресурси (машини) за дейности, разпределени във времето. Първите изследвания започват в началото на 50-те години и се развиват и през последните години. Множеството от проблемите за разписания за машини са детерминирани проблеми, а стохастичните им разширения са по-рядко разглеждани. Голяма част от проблемите за разписания за машини се определят като NP-трудни. Най-често използвани са алгоритмите на *динамичното програмиране* и на *“разклонения и граници”*. Също така е възможно проблемите за разписания за машини да се формулират като проблеми на целочисленото програмиране, които при някои специални структури на проблемите могат да бъдат лесно решавани. В много случаи трудностите при решаването на проблемите за разписания за машини водят до използване на евристични алгоритми (Morton & Pentico, 1993). Те могат да бъдат разделени на две главни категории – конструктивни и подобряващи евристики. Конструктивните евристики определят допустимо разписание, а подобряващите евристики се опитват да подобрят съществуващо допустимо разписание. Втората категория принадлежи към класа на евристичните алгоритми,

определени като локални (съседни) техники за търсене. Двете най-популярни техники за локално търсене “табу-търсене” и “симулирано закаляване” са използвани най-често. “Табу-търсенето” е добър компромис между качество на решението и изчислителното време (Vaessens и др., 1994).

- **Проектиране на мрежи**

Проблемите за проектиране на мрежи намират приложение в електроразпределението, телекомуникациите, транспорта, логистиката, производството и др. Този клон на комбинаторната оптимизация има съществени заслуги към практиката, осигурявайки намаляване на разходите в сравнение с методите на ръчно проектиране на мрежи. Намаляване на разходите с няколко процента е съществено в приложения с големи финансови инвестиции в инфраструктурата. Проблемът за проектиране на мрежи може да се дефинира като при зададени критерии за проектирането и изисквания за трафика между възлите в мрежата, да се оптимизира мрежата, съединяваща тези възли по отношение на топологията, капацитета на връзките и маршрутите в мрежата. При оптимизацията обикновено се минимизират някакви разходи или закъснения по време, или се максимизират потоците върху мрежите. Основни изисквания при проектиране на мрежи е гарантирането на изпълнението на изискванията за трафика, в т.ч. ограниченията за капацитет на връзките (дъгите), за свързаност на мрежата, за намаляване на средното време за преминаване и т.н. Оптимизационните критерии могат да бъдат всякакви – линейни, изпъкнали, неизпъкнали, дискретни или техни комбинации. Трудностите при проектирането на мрежи касаят най-вече вида на формулираната целева функция (Minoux, 1989). Често пъти евристичните подходи осигуряват приемливи начини за решаване на подобен тип задачи на мрежовото проектиране (Gavish, 1992). Въвеждането на стохастични фактори обикновено води до решения с по-малко печалба (по-големи разходи) за сметка на по-голяма гъвкавост в бъдещето, например при т. нар. дългосрочно планиране. Такива проблеми не могат да бъдат решени от детерминирани модели, тъй като не могат да се базират на достоверна бъдеща информация.

1.5. Общи подходи при вземането на решения

Голяма част от инженерните дейности включва решаване на проблеми и вземане на решения. Този процес често е итеративен. Трябва да бъдат взети много последователни решения докато се постигне желаният резултат. Вземането на решения е процес на избор на една от няколко алтернативи, на базата на анализ на последствията при

приемането на една от тях. Целта е да се намери най-добрата възможност и да се достигне до добро или оптимално решение.

1.5.1. Процес на вземане на решение

Един възможен подход за анализиране на процеса на вземане на решение е разглеждането му като многоетапен процес. Често пъти някои от етапите включват оптимизация на време и ресурси (Stoilova & Stoilov, 2012). Едно обобщение на етапите на този процес може да се илюстрира, както е показано на Фиг. 1.4 (Baker и др., 2002):



Фиг. 1.4. Диаграма на процеса на вземане на решение

- *Дефиниране на проблема* – идентифицират се основните причини, ограничаващите предположения, системата и организационните граници и интерфейси, както и всички заинтересовани страни.
- *Определяне на изискванията* – определят се условията, които всяко приемливо решение на проблема трябва да удовлетворява.
- *Определяне на целите* – общи декларации за намерения и желани стойности.
- *Определяне на алтернативите* – алтернативите предлагат различни подходи за промяна на първоначалното състояние в желаното състояние на процеса. Независимо дали те са съществуващи или са въображаеми, всяка алтернатива трябва да отговарят на определени изисквания.

- *Определяне на критериите* – чрез критериите се прави разлика между алтернативите, като критериите трябва да се основават на целите. Необходимо е да се определят критерии, които са обективни мерки на целите, за да се измери как всяка алтернатива постига целите.
- *Избор на инструмент за вземане на решение* – изборът на подходящ инструмент не е лесна задача и зависи от спецификата на конкретния проблем, както и от целите на ЛВР. Понякога, колкото по-прост е начина, толкова по-добре, но сложните проблеми изискват и сложни методи, използващи например индексни матрици и интуиционистки размити множества (Atanassova и др., 2014).
- *Оценка на алтернативите спрямо критериите* – в зависимост от критерия, оценката може да бъде обективна (фактическа), чрез отношение към някои общоприети и разбираеми измервания (напр. чрез пари) или може да бъде субективна, отразяваща субективната преценка на оценителя.
- *Проверка на решението* – избраните чрез използване на приетите средства за вземане на решения алтернативи, винаги трябва да бъдат проверени спрямо изискванията и целите на проблема. Може да е налице неправилно прилагане на инструмента за вземане на решение. При сложните проблеми, избраните алтернативи също могат да бъдат обект на вниманието на ЛВР и заинтересованите страни, по отношение на необходимост от допълнителни цели или изисквания, които да се добавят към модела на вземане на решение.

1.5.2. Видове системи за подпомагане вземането на решения

Системите за подпомагане вземането на решения (СПВР) представляват клас на компютърно-базирани информационни системи, включително и системи, базирани на знания, които подпомагат процеса на вземане на решения. СПВР са един специфичен клас системи, създадени за да подпомагат вземането на решения по отношение на различни дейности. Правилно проектираните СПВР представляват интерактивна софтуерна система, предназначена да улеснява работата на ЛВР, да събира полезна информация от данни, документи, лични знания, и/или модели за идентифициране и решаване на проблеми и вземането на решения. Конкретната реализация на една СПВР зависи от спецификата на решавания/те проблем/и. Преди етапа на избора на алтернатива се реализира сортиране на алтернативите в зависимост от тяхната целесъобразност спрямо зададените критерии. Могат да се обобщят два главни подхода при този етап – сортиране на алтернативите самостоятелно или обединението им в

групи и сортиране на групите. Това осигурява гъвкавост на избора при сложни проблеми. Съществена особеност при реализацията на една СПВР е наличието на обратна връзка след етап на оценка на избраната алтернатива, осигуряваща при необходимост, възможност за корекции и настройки.

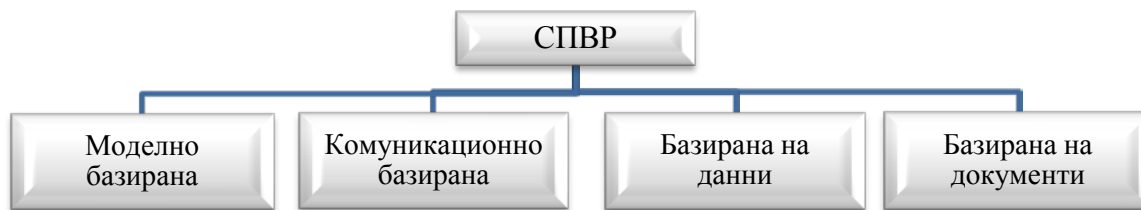
Важно е да се отбележи, че в сферата на СПВР няма универсално приет модел и съществуват много конкуриращи се теории в тази област. Поради това съществуват много начини за тяхното класифициране - например, една част от моделите на СПВР наблягат на връзката с потребителя и с неговите предпочитания. Това може да бъде отразено чрез класифициране по отношение на модела на взаимодействие с потребителя – пасивни модели, активни модели и смесени модели, Фиг. 1.5 (Kwon и др., 2005).



Фиг. 1.5. Модели на СПВР, в зависимост от връзката с потребителя и с неговите предпочитания

Системите за подпомагане вземането на решения, които само събират данни и ефективно ги организират, се наричат обикновено пасивни модели, т.е. те не предполагат конкретни решения, а само показват наличните данни. Активните системи за подпомагане вземането на решения, обработват данните и явно показват решения, базирани на данните. Макар че съществуват много системи, които са способни да бъдат активни, за много организации ще бъде трудно да повярват в компютърен модел и да разчитат на решения взети без човешка намеса. Смесените системи за подпомагане вземането на решения са тези, при които данните се събират, анализират и след това се предоставят на човека като възможно решение, който може да преразгледа или да подобри решението. Това означава, че човешкият елемент и компютърният компонент работят заедно за получаване на най-доброто решение.

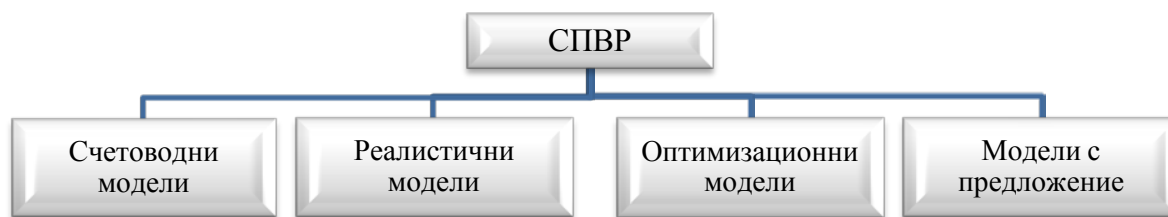
Докато посочените по-горе модели на СПВР отчитат взаимодействието с потребителя, друг популярен модел на СПВР взема под внимание начинът на подпомагане. Това включва моделно-базирана СПВР, комуникационно-базирана СПВР, базирана на данни СПВР, базирана на документи СПВР, и базирана на знания СПВР (Kwon и др., 2005), Фиг. 1.6.



Фиг. 1.6. Модели на СПВР, вземащ под внимание начинът на подпомагане

Моделно-базираната СПВР се реализира, когато лицата, вземащи решения, използват статистически, симулационни и финансови модели, за получаване на решение или стратегия, като тези решения са базирани на модели и не са претоварени с данни. В общия случай, не са необходими големи бази данни за този тип СПВР. Комуникационно-базираните СПВР са тези, при които много сътрудници работят заедно, за да получат поредица от решения, за определяне посоката на движение на решението или стратегията. Този модел се прилага в офис-среда или в интернет-среда. Комуникационно-базираните СПВР използват комуникационните технологии, за да улеснят съвместното вземане на решение. Базираните на данни СПВР поставят акцент върху събраните данни, които след това се манипулират, за да удовлетворят нуждите на ЛВР. Тези данни могат да бъдат вътрешни или външни и в различни формати. Базираните на данни СПВР разчитат на достъпа и манипулацията на времеви последователности от вътрешни данни, а в някои случаи и външни данни и данни, получавани в реално време. Документно-базираните СПВР използват различни документи – текстови документи, електронни таблици и записи от бази данни, за получаването на решения, както и за по-нататъшно манипулиране на информацията, с цел конкретизиране на стратегиите. Базираните на знания СПВР използват специални правила, съхранявани в компютър или се използват от хора, за да се определи дали определено решение трябва да бъде избрано. Така например, определянето на границата на загубите за търговците може да се разглежда като СПВР, базирана на знания.

Голяма част от СПВР са разработени като инструменти за генериране на решения и оценка на алтернативи, чрез анализ от типа “какво-ако” и “търсене на цел”, Фиг. 1.7 (Power, 2005).



Фиг. 1.7. СРВР за генериране на решения и оценка на алтернативи, чрез “какво-ако” и “търсене на цел” анализи

Счетоводните модели улесняват планирането чрез изчисляване на последователността от планирани действия, за да се оценят приходите, балансите и други финансови отчети. Реалистичните модели оценяват бъдещи последствия от действията, въз основа на частично дефинирани модели, включително всички симулационни модели. Оптимизационните модели генерират оптимални решения. Моделите с предложение водят до конкретно предложение за решение на добре структурирани задачи. Тези системи извършват механични изчисления и оставят малка роля за управленско решение.

Съществена роля в СРВР имат оптимизационните модели. Необходимо е да се отбележи, че общи методи за построяването на оптимизационните модели не съществуват. Във всеки конкретен случай се изгражда модел, изхождайки от спецификата на проблема. Оптимизационните модели намират голямо приложение поради факта, че те предлагат като крайно решение най-доброто решение, съгласно дефинирания модел. На базата на построения оптимизационен модел могат да се формулират съответни оптимизационни задачи, чиито решение е търсеното оптимално решение. За решаването на оптимизационните задачи съществуват различни подходи, които могат да бъдат използвани в зависимост от формулировката на задачата (едно- или многокритериални задачи). Правилно формулираните оптимизационни модели ще доведат и до правилно решение на разглеждания проблем.

Глава 2:

Оптимално проектиране на модулни инженерни системи, вземащи предвид съществуващите взаимовръзки между модулите

В настоящата глава е описан метод за оптимално проектиране на модулни инженерни системи, вземащ предвид съществуващите взаимовръзки между модулите. Предложеният обобщен математически модел, позволява оптимален избор на модулите на системата при спазване на всички изисквания за съвместимост между модулите и може да бъде използван за проектиране на подобни класове модулни системи. Моделът е тестван на базата на реален пример, достатъчно добре илюстриращ особеностите на проектирането на инженерни системи от този клас, а именно конфигуриране на персонални компютри (ПК). Моделът е използван за формулиране едно- и многокритериални оптимизационни задачи, решенията на които удовлетворяват както изискванията за съвместимост между модулите, така и разнообразни потребителски изисквания за параметрите на проектираната система. Описан е обобщен алгоритъм за итеративно и рационално конфигуриране на системи от този клас. Този алгоритъм включва различни сценарии в зависимост от потребителските изисквания, като за всеки сценарий е представена съответна алгоритмична реализация. Получените резултати са публикувани в статии D8, D10 и D16.

Процесът на проектиране може да се разглежда като процес на разработване на проект за нов обект, за което са необходими изследвания, моделиране, интерактивна настройка и ре-дизайн. Процесът на инженерния дизайн може да бъде представен чрез следните 10 стъпки (Gomez, 2004; Garrett, 1991):

- 1) Идентифициране на проблема – общо описание
- 2) Определяне на критериите и целите
- 3) Проучване на проблема – в повечето случаи е необходимо получаването на нови знания и информация

- 4) Определяне на спецификациите – подробно описание на проблема и изясняване какво трябва да се постигне при проектирането и какви ограничения ще повлияят на крайното решение
- 5) Определяне на възможни решения – комбиниране на идейните проекти с информацията, получена от т. 3 за определяне на възможни проектни решения
- 6) Анализ на потенциалните решения – определяне кой вариант да се развива – в идеалния случай това е този вариант, който най-добре отговаря на спецификациите и на други ограничения, като например време, разходи или умения
- 7) Подготвяне на работна документация и планиране на всички подробности, които са важни за реализирането на избрания вариант
- 8) Изработване на първоначален модел (прототип)
- 9) Тестване и оценка на дизайна
- 10) Подготвяне на документация – планиране и проектиране за извършване на практическата реализация.

Проблемите на концептуалния дизайн са свързани с неясното естество на първоначалните концепции за дизайн и наличието на различни варианти, които инженерите трябва да тестват. Реалните проблеми на инженерния дизайн обикновено се характеризират с наличието на много често противоречащи си и несъвместими цели. По време на процеса на проектиране, трябва да се направи компромис между различни инженерни характеристики. Дизайнерът може да използва набор от инструменти за моделиране и симулиране, за да се предскажат характеристиките на проектирания обект. Този процес често е итеративен и отнема много време и е ограничен от познанията на конкретния дизайнер. Днес клиентите имат все по-голямо търсене на различни продукти. Предлагането на голямо разнообразие от продукти е важно за всяка фирма, за да привлече различни потребителски сегменти. Последните теории предлагат модулен дизайн на продукта като ключов фактор за успеха на продукта в глобалната конкуренция (Antonio и др., 2007). Те изследват влиянието на модулни продукти на конкурентни производствени като възможности, включително цена, качество на продукта, гъвкавост, и др.

Като се има предвид разрастващата се глобална конкуренция и съкращаване жизнения цикъл на продуктите, има нарастваща нужда от разработването и

производството на достъпни стоки и услуги, които могат да бъдат подходящо персонализирани (Zhang и др., 2014; Avak, 2007)

Модулността в проектирането е един от основните начини, осигуряващи разнообразие на продуктова гама от устройства (Yu и др., 2015). Модулният дизайн като форма на стратегическото производство предлага гъвкавост при интеграцията на модулните компоненти. Модулността има някои съществени предимства – тя води да намаляване на производствените разходи и времето за изпълнение, позволява реализиране на различни изисквания в съответствие с променящите се пазари и технологии, а по отношение на недостатъците – липса на оптимизация на производителността и прекомерно сходство между продуктите (Kidd, 1994; Gershenson & Prasad, 1997; Tseng & Huang, 2008; Dorsey и др., 2006; Kamrani & Nasr, 2008). Модулният дизайн не включва разработването на нови видове компоненти, но изисква познаване на характеристиките на модулите и тяхната съвместимост с други модули. Модулната система се характеризира с функционално разделяне на модулите и използване на добре дефинирани модулни интерфейси, както и използването на индустриални стандарти за интерфейси (AlGeddawy & ElMaraghy, 2013). Това е от особено значение да се прилагат подходи за моделиране в контекста на използване на оптимизационни модели. За формулирането на подходящ оптимизационен модел е необходимо добро познаване на спецификата на моделираната система (вкл. и избор на управляващи променливи, ограничения, критерии), както и познаване на методите да решаване на едно- и многокритериалните оптимизационни задачи. Една от перспективните насоки при проектирането е използването на комбинаторна оптимизация (Levin, 2009) и по-специално на използването на 0-1 целочислени променливи и евристичен методи (Fujita и др., 1999).

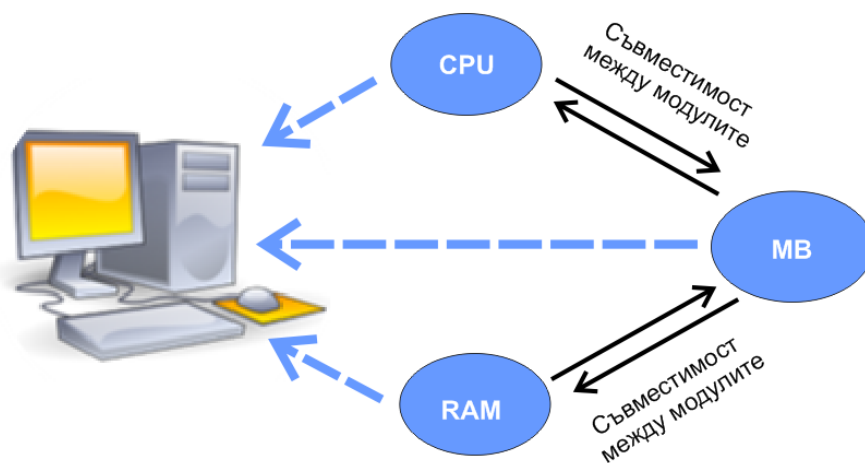
2.1. Персоналните компютри като пример за модулна инженерна система

ИТ индустрията е един от най-бързо развиващите се сектори. В наши дни съществува голямо разнообразие от компютърни системи с различно предназначение. Това е продиктувано както от непрестанното съревнование между фирмите производители, така и от различните потребителски изисквания към функционалността компютърните системи. Тази функционални изисквания се отразяват най-вече върху основните модули на системата – дънна платка (MB), процесор (CPU), памет (RAM) и периферни устройства. Съществен проблем при проектирането на модулни системи е

разнообразието от модули с налични специфични ограничения за съвместимост. За конфигурирането на ПК например, съществуват различни производители на МВ като ASUS, ABIT, MSI, Gigabyte, Intel и др., които предлагат много и различни видове дънни платки със специфични изисквания за съвместимост с процесори, памети и други компоненти. Същото важи както за CPU модулите (напр. произведените от Intel или AMD), така и за RAM модулите. В тази връзка, примерът за конфигуриране на компютър е една добра илюстрация на проблемите, присъщи на проектирането на класа от модулни системи, вземащи предвид зависимостите между отделните модули. Примерът за конфигуриране на ПК от своя страна представлява интересен проблем на комбинаторната оптимизация (McDermott, 1982; Jae-Kyu и др., 1996; Tam & Ma, 2000; Soininen и др., 2000; Tam & Ma, 2002; Kreng & Lee, 2004)..

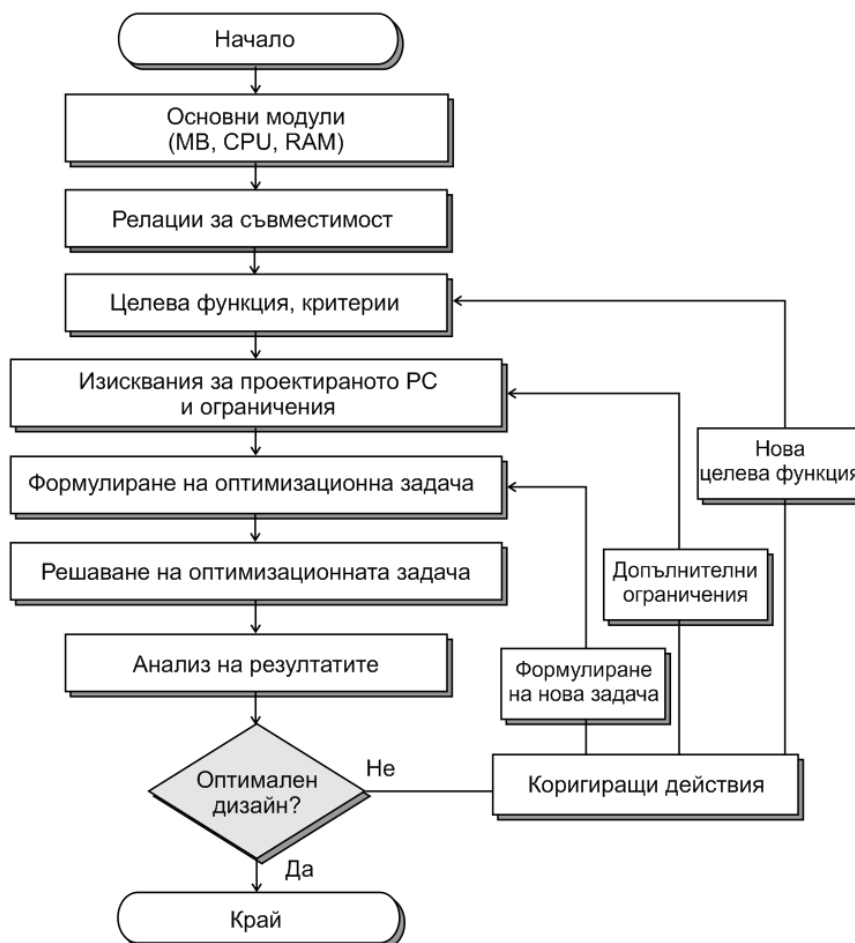
2.2. Обобщен модел за оптимално проектиране на модулни инженерни системи, вземащи предвид съществуващите взаимовръзки между модулите

За разлика от използваните евристични и генетични подходи (McDermott, 1982; Jae-Kyu и др., 1996; Tam & Ma, 2000; Soininen и др., 2000; Tam & Ma, 2002; Kreng & Lee, 2004), цел на обобщен модел е определянето на оптимални или Парето оптимални комбинации на модули за проектираната инженерна система, като следствие от решаването на формулирани едно- или многокритериални оптимизационни задачи. Използваият пример за този клас системи, а именно оптималното проектиране на конфигурирането на ПК се характеризира с необходимост от избор на основни компоненти – дънна платка (МВ), процесор (CPU) и памет (RAM), които пряко се отразяват върху производителността и характеристиките на проектираната система (Фиг. 2.1).



Фиг. 2.1. Основни модули на ПК

Етапите на алгоритма за оптимално конфигуриране на ПК е показан на Фиг. 2.2 (Mustakerov & Borissova, 2013).



Фиг. 2.2. Алгоритъм за оптимално проектиране на конфигурацията на ПК

Алгоритъмът започва с процедура за определяне на множество от модули, които могат да бъдат използвани в процеса на проектирането. След това е необходимо да се определят съвместимите подмножества между отделните модули. На следващия етап се определят критериите, на които трябва да отговаря проектираната конфигурация на ПК. На база на така поставените изисквания към параметрите и определените съвместимости между модулите може да се формулира съответна оптимизационна задача. Тъй като алгоритъмът е насочен към подпомагането на ЛВР при вземане на решение относно параметрите на проектираната конфигурация за ПК, то полученото решение е необходимо да бъде анализирано от ЛВР. Ако ЛВР не е удовлетворен от избраните модули е необходимо да бъдат направени и съответни корекции. В алгоритъма са включени 3 вида коригиращи действия: 1) преформулиране на критериите или добавяне на нови критерии; 2) предефиниране на поставените

потребителски изисквания към дизайна; 3) нова формулировка на задачата за оптимизация. Чрез тези действия ЛВР има възможност да отрази по-точно своите виждания по отношение на проектираната система.

Общата формулировка на обобщения модел за оптимално проектиране на инженерна система, вземаща предвид съществуващите взаимовръзки между модулите има вида:

$$\min/\max f_i(\mathbf{x}), i = 1, 2, \dots, k \quad (2.1)$$

при ограничения:

$$g_j(\mathbf{x}) \leq 0, j = 1, 2, \dots, m \quad (2.2)$$

$$l_q(\mathbf{x}) \geq 0, q = 1, 2, \dots, t \quad (2.2)$$

$$h_l(\mathbf{x}) = 0, l = 1, 2, \dots, e \quad (2.4)$$

където: $f_i(\mathbf{x})$ – критерии (целева функции), k – брой критерии, m , t – брой на ограниченията, представени като неравенства, e – брой ограничения като равенства, $\mathbf{x} = (x_1, x_2, \dots, x_n)^T$ – вектор на променливите, n – брой на независимите променливи.

Съвместимостта между отделните модули на проектираната система се изразява математически чрез ограниченията (2.2) – (2.4). Няма общо правило при математическото моделиране какъв тип ограничения да бъдат използвани. За да се демонстрира приложимостта на метода на оптималното проектиране ще се използва примера на проектирането на конфигурация за ПК. За да се получи работеща компютърна система, при избора на MB, CPU и RAM модулите трябва да се вземе предвид следното:

- дънната платка налага строги изисквания за останалите модули – специфичен тип CPU куплунг, определен чипсет, брой слотове и портове (PCI, AGP, IDE, SATA, E-SATA, USB, FireWire, LAN, RAID и т.н.), както и задава определени спецификации за вида и размера на RAM и периферните устройства.
- компаниите-производители на процесори предлагат процесори за различни куплунзи и с различни функционални възможности. Така например, продуктова линия на Intel¹ предлага Pentium IV и Celeron процесори за куплунг 478; Pentium, Celeron, Core™ 2 Duo, Quad и Extreme за куплунг 775; Core™ i7 и i5 процесори Socket 1366 за Core™ i7 за куплунг 1156. Друг

¹ <http://www.intel.com/content/www/us/en/intelligent-systems/previous-generation/intel-ixp4xx-intel-network-processor-product-line.html>

ведещ производител на процесори – AMD² предлага Athlon™ и Duron™ процесори за куплунг от тип А, Athlon™ 64 и Sempron™ процесори за куплунг AM2, Phenom™ и Athlon™ 64 процесори за куплунг AM2+, Opteron™ и Athlon™ 64 7x процесори за куплунг от тип F. Ето защо, изборът на конкретен процесор трябва да бъде съобразен със съответните спецификации на MB за наличните куплунзи за CPU.

- RAM модулите също трябва да бъдат избрани със спецификации, поддържани от MB за тип памет, брой слотове и максимален размер на паметта, която може да бъде поддържана.
- други компютърни модули, като твърд диск, видео-контролер, RAID-контролер, оптични устройства, хранване, клавиатура и т.н., също могат да се вземат предвид, но изборът на горепосочените основни модули (MB, RAM, CPU) е достатъчен, за да се оцени и илюстрира предложението за подход на проектиране на модулни инженерни системи, използващи оптимизационни модели.

Оптимизирането на дизайна на коя да е система се извършва по отношение на определени критерии за оптимизация, като често се налага едновременно разглеждане на повече от един критерий. Това означава, че е необходимо да се търси многокритериален компромис между съществуващи функционални и потребителски изисквания и ограничения. Всички те трябва да бъдат надлежно определени и включени в математическия модел. Изискванията на потребителите са от съществено значение за приемането на проектираната система. Формализирането на оптимизационния проблем се основава на математически модел на проектираната система. Процесът на проектиране може да се раздели на два проблема: генериране на алтернативни решения и вземане на подходящо решение.

Проектирането на реални модулни системи е сложен комбинаторен проблем, който може да се моделира чрез използване на двоични целочислени променливи, присвоени към различни параметри на модулите. Например, ако $j \in J$ са индексите на различните видове модули, $k \in K^j$ са индексите на параметрите на модул от тип j , $i \in I_j$ са индексите на различните модули от един и същ тип и всеки отделен модул i от тип j има присвоени двоични целочислени променливи x_i^j , то изборът на модули може да бъде изразен чрез избор на неговите параметри като:

² <http://www.amd.com/us/products/desktop/processors/Pages/desktop-processors.aspx>

$$\forall j \in J : (\forall k \in K^j : P^{j,k} = \sum_{i \in I_j} P_i^{j,k} x_i^j), \quad x_i^j \in \{0,1\}, \quad \sum_{i \in I_j} x_i^j = 1 \quad (2.5)$$

Съществените съображения, свързани с проектирането на системата (например, разходи, проблеми със съвместимостта и изисквания на потребителите) трябва да бъдат взети под внимание при процеса на моделиране.

В разглеждания пример за оптимално конфигуриране на компютърна система при избора на MB и CPU, параметърът *цена* може да бъде определен като:

$$P^{MB\text{cost}} = \sum_{i \in I_{MB}} P_i^{MB\text{cost}} x_i^{MB}, \quad (2.6)$$

където $x_i^{MB} \in \{0,1\}$ и $\sum_{i \in I_{MB}} x_i^{MB} = 1$

$$P^{CPU\text{cost}} = \sum_{j \in I_{CPU}} P_j^{CPU\text{cost}} x_j^{CPU} \quad (2.7)$$

където $x_i^{CPU} \in \{0,1\}$ и $\sum_{i \in I_{CPU}} x_i^{CPU} = 1$

По същия начин, и други параметри на модулите могат да бъдат определяни чрез подходящ избор:

$$P^{CPU\text{clock}} = \sum_{i \in I_{CPU}} P_i^{CPU\text{clock}} x_i^{CPU} \quad (2.8)$$

$$P^{CPU\text{core}} = \sum_{i \in I_{CPU}} P_i^{CPU\text{core}} x_i^{CPU} \quad (2.9)$$

При това, проектирането на конкретна модулна техническа система трябва да бъде съобразено с нейните специфични особености. В конкретния пример, при проектиране на персонален например, изборът на RAM трябва да включва не само вид, но и брой на RAM модулите. Това налага въвеждане на неотрицателни целочислени променливи x_i^{ram} за различните видове RAM. Съответните уравнения за параметрите на RAM по отношение на *цена* и *капацитет*, използвайки променливите x_i^{ram} се изразяват като:

$$P^{RAM\text{cost}} = \sum_{i \in I_{RAM}} P_i^{RAM\text{cost}} x_i^{RAM}, \quad x_i^{ram} \in N \quad (2.10)$$

$$P^{RAM\text{size}} = \sum_{i \in I_{RAM}} P_i^{RAM\text{size}} x_i^{RAM}, \quad x_i^{RAM} \in N \quad (2.11)$$

където N е множество на неотрицателните числа.

Задължително условие при проектирането на модулна система е изискването да има съвместимост между отделните модули. Отношенията на съвместимост между модулите могат да се представят чрез съвместими подмножества на множествата на

модулите. Например, при проектирането на ПК, ако подмножеството на процесорите, съвместими с определен вид дънна платка MB_i е означено като I_{CPU}^i , то ограниченията за съвместимост на съответните променливите, присвоени към MB и процесорите може да се изрази като:

$$\forall i \in I_{MB} : x_i^{MB} \leq \sum_{j \in I_{CPU}^i} x_j^{CPU} \quad (2.12)$$

където $I_{CPU}^i \subseteq I_{CPU}$ са индексите на процесорите съвместими с дънни платки от тип i .

Поради спецификата на избора на RAM са необходими два вида ограничения:

- ограничения за съвместимост на RAM и MB

$$\forall i \in I_{MB} : x_i^{MB} \sum_{k \in I_{RAM}^{i+}} x_k^{RAM} \geq 0 \quad (2.13)$$

- ограничения за несъвместимост на RAM и MB

$$\forall i \in I_{MB} : x_i^{MB} \sum_{k \in I_{RAM}^{i-}} x_k^{RAM} \leq 0 \quad (2.14)$$

където $I_{RAM}^{i+} \subseteq I_{RAM}$ и $I_{RAM}^{i-} \subseteq I_{RAM}$ са подмножества на съвместимите и несъвместими RAM с MB_i съответно.

Друг тип ограничения се използват за изразяване на съвместимостта на RAM и MB по отношение размера на паметта $P^{RAMsize}$, която се поддържа от съответния тип MB_i :

$$P^{RAMsize} \leq \sum_{i \in I_{MB}} P_i^{MBram \max} x_i^{MB} \quad (2.15)$$

Броят на наличните слотове $P_i^{MB, RAMslots}$ за конкретния тип MB_i се изразява чрез следната зависимост:

$$\sum_{i \in I_{RAM}} x_k^{RAM} \leq P_i^{MBramslots} \quad (2.16)$$

и изискването, че е необходим поне един модул RAM, за да има функционална компютърна конфигурация се изразява чрез:

$$\sum_{i \in I_{RAM}} x_k^{RAM} \geq 1 \quad (2.17)$$

Така предложеният подход на моделиране позволява лесно да се въвеждат различни потребителски изисквания относно параметрите на проектираната система. Например могат да се въведат изисквания за размера на RAM, за тактовата честота на CPU и за броя ядра на CPU като:

$$P^{RAMsize} \geq P^{RAM \min} \quad (2.18)$$

$$P^{CPUclock} \geq P^{CPUclock \min} \quad (2.19)$$

$$P^{CPUcore} \geq P^{CPUcore \min} \quad (2.20)$$

където $P^{RAM \min}$, $P^{CPUclock \min}$, $P^{CPUcore \min}$ са минималните стойности, определени от потребителя.

Други изисквания за параметрите на проектирана система могат лесно да бъдат въведени по подобен начин.

2.3. Еднокритериална оптимизационна задача за проектиране на персонални компютри

Описаният модел (2.5) – (2.20) може да бъде използван за формулиране на еднокритериални оптимизационни задачи. Като оптимизационен критерий е избрана *цената* на ПК конфигурацията, а различните потребителски изисквания са отразени чрез съответни ограничения: (Mustakeroov & Borissova, 2013):

$$\min (MB^{cost} + CPU^{cost} + RAM^{cost}) \quad (2.21)$$

при ограничения

$$MB^{cost} = \sum_{i \in I} MB_i^{cost} x_i^{mb}, x_i^{mb} \in \{0, 1\}, \sum_{i \in I} x_i^{mb} = 1 \quad (2.22)$$

$$CPU^{cost} = \sum_{j \in J} CPU_j^{cost} x_j^{cpu}, x_j^{cpu} \in \{0, 1\}, \sum_{j \in J} x_j^{cpu} = 1 \quad (2.23)$$

$$CPU^{clock} = \sum_{j \in J} CPU_j^{clock} x_j^{cpu} \quad (2.24)$$

$$CPU^{core} = \sum_{j \in J} CPU_j^{core} x_j^{cpu} \quad (2.25)$$

$$MB^{RAID} = \sum_{i \in I} MB_i^{RAID} x_i^{mb}, MB^{RAIDmin} = 1 \quad (2.26)$$

$$RAM^{cost} = \sum_{k \in K} RAM_k^{cost} x_k^{ram}, x_k^{ram} \in N \quad (2.27)$$

$$RAM^{size} = \sum_{k \in K} RAM_k^{size} x_k^{ram} \quad (2.28)$$

$$\forall i \in I: x_i^{mb} \leq \sum_{j \in J_i} x_j^{cpu}, \quad (2.29)$$

$$\forall i \in I: x_i^{mb} \sum_{k \in K_i'} x_k^{ram} \geq 0, K_i' \subseteq K \quad (2.30)$$

$$\forall i \in I: x_i^{mb} \sum_{k \in K_i''} x_k^{ram} \leq 0, K_i'' \subseteq K \quad (2.31)$$

$$RAM^{size} \leq \sum_{i \in I} MB_i^{RAM \max} x_i^{mb} \quad (2.32)$$

$$\sum_{k \in K} x_k^{ram} \leq MB_i^{RAMslots} \quad (2.33)$$

$$\sum_{k \in K} x_k^{ram} \geq 1 \quad (2.34)$$

$$RAM^{size} \geq RAM^{size min} \quad (2.35)$$

$$CPU^{clock} \geq CPU^{clock min} \quad (2.36)$$

$$CPU^{core} \geq CPU^{core min} \quad (2.37)$$

$$MB^{RAID} = MB^{RAID min} \quad (2.38)$$

2.3.1. Числено тестване

Приложимостта и ефективността на предложения подход за моделиране процеса на избор на основните компютърни модули е тестван чрез формулирани и решени оптимизационни задачи, базирани на реални данни на модули. Параметрите на използваните модули са показани в Таблицы 2.1, 2.2 и 2.3.

Таблица 2.1. Параметри на MB

#	MB	CPU socket	Intel® CPU	RAM type	RAM slots	RAM max, GB	RAM MHz	Price BGL
1	ASROCK P45XE-WIFIN/P45	LGA775	Core™ 2E, Core™ 2 Quad, Core™ 2 Duo, Pentium® Dual Core, Celeron® Dual Core, Celeron®	DDR2	4	16	1200, 1066, 800, 667	178.50
2	ASROCK G31M-S/G31	LGA775	Core™ 2E, Core™ 2 Quad, Core™ 2 Duo, Pentium® EE, Pentium® D, Pentium® 4 E, Pentium® 4, Celeron®	DDR2	2	8	800, 667	69.00
3	Gigabyte G31M-ES2C/G31	LGA775	Core 2 Quad, Core 2 Extreme, Core 2 Duo, Pentium EE, Pentium D, Pentium 4, Celeron	DDR2	2	4	1066, 800, 667	74.50
4	Gigabyte X48-DQ6 /X48	LGA775	Core 2 Quad, Core 2 Extreme, Core 2 Duo, Pentium EE, Pentium D, Pentium 4, Celeron	DDR2	4	8	1200, 1066, 800, 667, 533	227.50
5	ASUS P5S800-VM /SIS661FX	LGA775	Pentium4, Celeron	DDR	2	2	400, 333, 266	36.00
6	ASUS P7P55D/PRO/P55	LGA1156	Core i5, i7	DDR3	4	16	1600, 1333, 1066	315.50
7	INTEL DX58SO/X58/BOX	LGA1366	Core i7	DDR3	4	8	1600, 1333, 1066	421.50

Таблица 2.2. Параметри на CPU

j	CPU	Сокет	Честота	Ядра	Цена
1	CELERON-D 347	LGA775	3.06 GHz	1	59.50
2	Core DUO E5200 /800/2M BOX	LGA775	2.50 GHz	2	109.00
3	Core2 DUO E7600 /1066/3M BOX	LGA775	3.06 GHz	2	251.00
4	Core2 QUAD Q8200S/1333/BOX	LGA775	2.33 GHz	4	381.00
6	C i5-750 /8M/BOX/	LGA1156	2.66 GHZ	4	371.00
6	C i7-860 /8M/BOX/	LGA1156	2.80 GHZ	4	531.00
7	C i7-940 /8M/BOX/	LGA1366	2.93 GHZ	4	979.50

Таблица 2.3 Параметри на RAM

k	RAM	RAM_k^{size} GB	RAM_k^{freq} MHz	Цена
1	DDR A-DATA	1	400	55.00
2	DDR2 KINGSTON	1	667	41.50
3	DDR2 KINGSTON	1	1066	64.50
4	DDR3 KINGSTON	1	1066	42.50
5	DDR2 KINGSTON	2	800	77.50
6	DDR3 A-DATA	2	1333	77.50
7	DDR3 HYPER X KINGSTON	2	1600	83.00

Съвместимостта между отделните модули може да бъде изразена чрез съответни подмножества на модулите както следва:

- съвместимост между MB и CPU:

$$MB_1-CPU = \{CPU_1, CPU_2, CPU_3, CPU_4\}$$

$$MB_2-CPU = \{CPU_1, CPU_2, CPU_3, CPU_4\}$$

$$MB_3-CPU = \{CPU_1, CPU_2, CPU_3, CPU_4\}$$

$$MB_4-CPU = \{CPU_1, CPU_2, CPU_3, CPU_4\}$$

$$MB_5-CPU = \{CPU_1\}$$

$$MB_6-CPU = \{CPU_5, CPU_6\}$$

$$MB_7-CPU = \{CPU_7\}$$

- съвместимост между CPU и MB:

$$CPU_1-MB = \{MB_1, MB_2, MB_3, MB_4, MB_5\}$$

$$CPU_2-MB = \{MB_1, MB_2, MB_3, MB_4\}$$

$$CPU_3-MB = \{MB_1, MB_2, MB_3, MB_4\}$$

$$CPU_4-MB = \{MB_1, MB_2, MB_3, MB_4\}$$

$$CPU_5-MB = \{MB_6\}$$

$$CPU_6-MB = \{MB_6\}$$

$$CPU_7-MB = \{MB_7\}$$

- съвместимост между MB и RAM:

$$MB_1-RAM = \{RAM_2, RAM_3, RAM_5\}$$

$$MB_2-RAM = \{RAM_2, RAM_5\}$$

$$MB_3-RAM = \{RAM_2, RAM_3, RAM_5\}$$

$$MB_4-RAM = \{RAM_2, RAM_3, RAM_5\}$$

$$MB_5-RAM = \{RAM_1\}$$

$$MB_6-RAM = \{RAM_4, RAM_6, RAM_7\}$$

$$MB_7-RAM = \{RAM_4, RAM_6, RAM_7\}$$

- съвместимост между RAM и MB:

$$RAM_1-MB = \{MB_5\}$$

$$RAM_2-MB = \{MB_1, MB_2, MB_3, MB_4\}$$

$$RAM_3-MB = \{MB_1, MB_3, MB_4\}$$

$$RAM_4-MB = \{MB_6, MB_7\}$$

$$RAM_5-MB = \{MB_1, MB_2, MB_3, MB_4\}$$

$$RAM_6-MB = \{MB_6, MB_7\}$$

$$RAM_7-MB = \{MB_6, MB_7\}$$

Формулирани са три еднокритериални оптимизационни задачи, отразяващи различни потребителски изисквания към проектираната компютърна система:

- *Задача 1:* Целева функция (2.21), определяща минилална цена на конфигурацията, ограничения за съвместимост (2.22) – (2.34) и допълнително ограничение за капацитета на паметта $RAM^{size\ min} = 2\ GB$
- *Задача 2:* Целева функция (2.21), определяща минилална цена на конфигурацията, ограничения за съвместимост (2.22) – (2.34) и допълнителни ограничения за паметта и процесора $RAM^{size\ min} = 3\ GB$, $CPU^{clock\ min} = 2.5\ GHz$ и $CPU^{core\ min} = 2$
- *Задача 3:* Целева функция (2.21), определяща минилална цена на конфигурацията, ограничения за съвместимост (2.22) – (2.34), и допълнителни ограничения за паметта, процесора и наличие на RAID:
 $RAM^{size\ min} = 4\ GB$, $CPU^{clock\ min} = 2.5\ GHz$, $CPU^{core\ min} = 2$, $MB^{RAID\ min} = 1$.

Получените оптимални конфигурации, вследствие на решенията на задачите, са показани в Таблица 2.4.

Резултатите от формулираните оптимизационни задачи и техните решения определят оптимални комбинации на MB, CPU и RAM, отговарящи както на условието за съвместимост, така и на зададените потребителски изисквания. Това потвърждава

адекватността и приложимостта на използвания подход за моделиране, за оптимално проектиране на модулни системи. Различните формулировки на задачите показват лекотата на въвеждане на потребителските изисквания към параметрите на проектираната система.

Таблица 2.4. Получени резултати

Задача	Потребителски изисквания				Получени конфигурации			
	RAM	CPU clock	CPU core	RAID	MB	CPU	RAM	Cost (BGL)
1	≥ 2 GB	--	--	--	MB5: ASUS-P5S800-VM /SIS661FX RAID	CPU1: CELEROND 347, 3.06 GHz	RAM1: 2 x 1 GB DDR, A-DATA 400 MHz	205.5
2	≥ 3 GB	≥ 2.5 GHz	≥ 2	--	MB2: ASROCK G31M-S/G31, no RAID	CPU2: Core DUO E5200 /800/ 2M BOX, 2.50 GHz	RAM2: 1 GB DDR2 667 MHz, RAM5: 2 GB DDR2 800 MHz	297.0
3	≥ 4 GB	≥ 3 GHz	≥ 2	1 (yes)	MB1: ASROCK - P45XE-WIFIN/P45 RAID	CPU3: Core2 DUO E7600 /1066/3M BOX, 3.06 GHz	RAM5: 2 x 2 GB DDR2 800 MHz	584.5

Чрез описания подход за проектиране на модулни системи могат да се въвеждат и други целеви функции и/или формулировки на оптимизационните задачи. Следва да се отбележи, че решаваните оптимизационни задачи са базирани на примери за реални проблеми и илюстрират практическото приложение на подхода. Времената на решаваните задачи са от порядъка на секунди при използването на настолен компютър с Intel® Celeron® 2.80 GHz CPU и 1.96 GB оперативна памет и операционна система MS Windows XP ©.

2.4. Многокритериална оптимизационна задача за проектиране на персонални компютри

Описаният по-горе подход за моделиране на проектирането на модулни системи може да се развие до формулиране на многокритериални оптимизационни задачи. Решенията на тези задачи ще определят Парето оптимален избор на модулите, съответстващи на зададените предпочитанията от ЛВР. Един реалистичен пример за използване на многокритериална оптимизация при конфигуриране на ПК може да има като критерии *минимална цена* за модулите, *максимална изчислителна мощност* на процесора

(максимизация на броя ядра и на тактовата честота) и *максимален размер* на RAM. Тези изисквания могат да се изразят чрез следната многокритериална оптимизационна задача (Mustakeroov & Borissova, 2013):

$$\begin{aligned}
 & \min (P^{MB, cost} + P^{CPU, cost} + P^{RAM, cost}) \\
 & \max P^{CPU, core} \\
 & \max P^{CPU, clock} \\
 & \max P^{RAM, size}
 \end{aligned} \tag{2.39}$$

при ограничения

$$P^{MB cost} = \sum_{i \in I_{MB}} P_i^{MB cost} x_i^{MB}, x_i^{MB} \in \{0, 1\}, \sum_{i \in I_{MB}} x_i^{MB} = 1 \tag{2.40}$$

$$P^{CPU cost} = \sum_{j \in I_{CPU}} P_j^{CPU cost} x_j^{CPU}, x_j^{CPU} \in \{0, 1\}, \sum_{j \in I_{CPU}} x_j^{CPU} = 1 \tag{2.41}$$

$$P^{CPU clock} = \sum_{i \in I_{CPU}} P_i^{CPU clock} x_i^{CPU} \tag{2.42}$$

$$P^{CPU core} = \sum_{i \in I_{CPU}} P_i^{CPU core} x_i^{CPU} \tag{2.43}$$

$$P^{RAM cost} = \sum_{i \in I_{RAM}} P_i^{RAM cost} x_i^{RAM}, x_i^{RAM} \in \mathbb{N} \tag{2.44}$$

$$P^{RAM size} = \sum_{i \in I_{RAM}} P_i^{RAM size} x_i^{RAM}, x_i^{RAM} \in \mathbb{N} \tag{2.45}$$

$$\forall i \in I_{MB} : x_i^{MB} \leq \sum_{j \in I_{CPU}^I} x_j^{CPU} \tag{2.46}$$

$$\forall i \in I_{MB} : x_i^{MB} \sum_{k \in I_{RAM}^{I+}} x_k^{RAM} \geq 0 \tag{2.47}$$

$$\forall i \in I_{MB} : x_i^{MB} \sum_{k \in I_{RAM}^{I-}} x_k^{RAM} \leq 0 \tag{2.48}$$

$$P^{RAM size} \leq \sum_{i \in I_{MB}} P_i^{MBram max} x_i^{MB} \tag{2.49}$$

$$\sum_{i \in I_{RAM}} x_k^{RAM} \leq P_i^{MBram slots} \tag{2.50}$$

$$\sum_{i \in I_{RAM}} x_k^{RAM} \geq 1 \tag{2.51}$$

$$P^{RAM size} \geq P^{RAM min} \tag{2.52}$$

$$P^{CPU clock} \geq P^{CPU clock min} \tag{2.53}$$

$$P^{CPU core} \geq P^{CPU core min} \tag{2.54}$$

2.4.1. Числено тестване

Един широко разпространен подход за решаване на многокритериалните оптимизационни проблеми е чрез трансформиране на многокритериалната (векторна) задача в еднокритериална (скаларна) задача (Miettinen & Makela, 2002). Използван е *методът на претеглената сума*, при който информацията за предпочитанията на потребителя към различните критерии се определя посредством съответни тегловни коефициенти w_i , определени за всички критерии (Marler & Arora, 2004). Приложението на този метод трансформира многокритериалната задача (2.39) – (2.54) в еднокритериална оптимизационна задача от вида:

$$\max(w_1 PC^{cost*} + w_2 CPU^{core*} + w_3 CPU^{clock*} + w_4 RAM^{size*}) \quad (2.55)$$

при ограничения (2.36) – (2.50) и допълнително ограничение за тегловните коефициенти

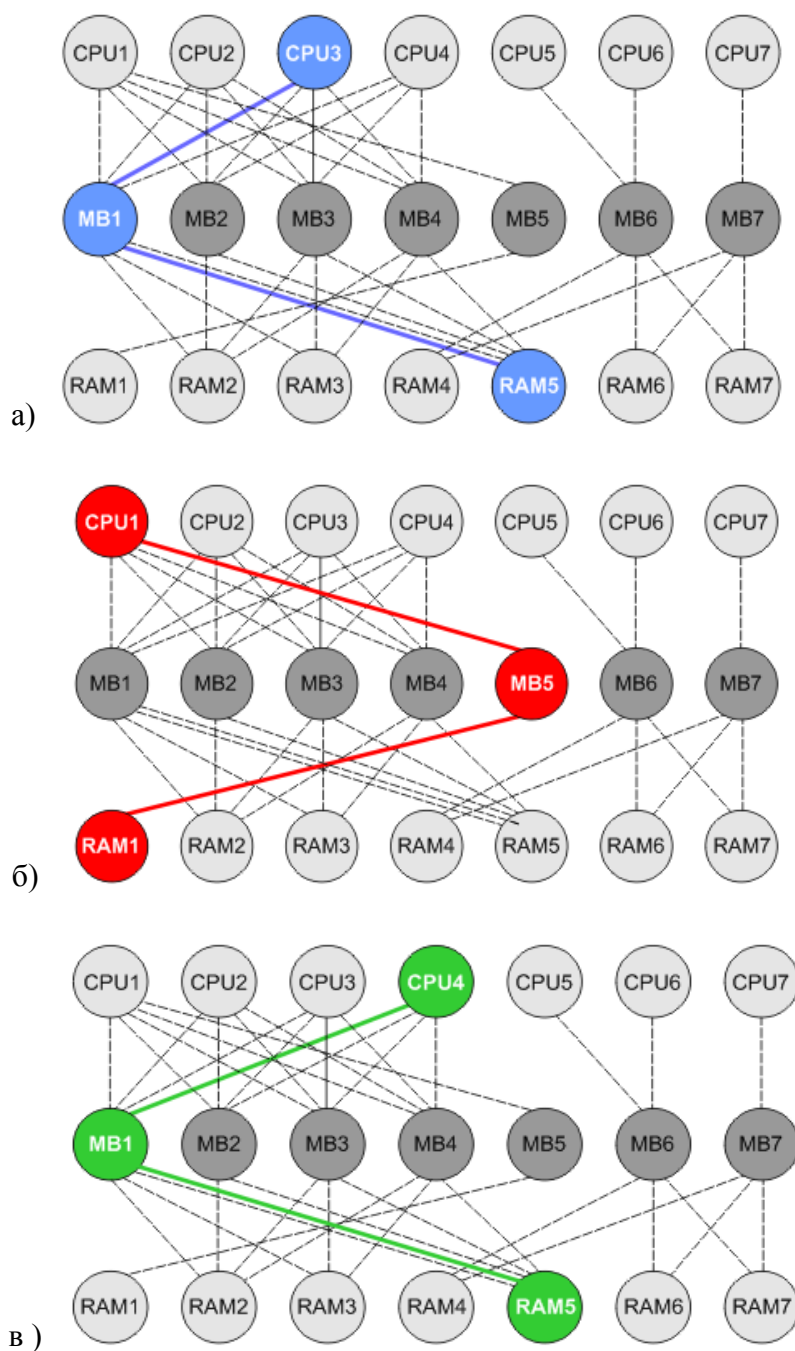
$$\sum_{f=1}^4 w_f = 1 \quad (2.56)$$

Така трансформираната оптимизационна задача (2.55) при ограничения (2.40) – (2.54) и (2.56) е решавана за три различни предпочитания на ЛВР по отношение важността на всеки един от критериите, както е показано в Таблица 2.5.

Таблица 2.5. Получени резултати

ЛВР	Теглови коефициенти				Получени конфигурации			
	w_1 (PC cost)	w_2 (CPU core)	w_3 (CPU clock)	w_4 (RAM size)	MB	CPU	RAM	Цена
ЛВР-1	0.25	0.25	0.25	0.25	MB_1	CPU_3	4 RAM ₅ (4x2=8 GB)	739.50
ЛВР-2	0.70	0.10	0.10	0.10	MB_5	CPU_1	1 RAM ₁ (1x1=1 GB)	150.50
ЛВР-3	0.15	0.35	0.20	0.30	MB_1	CPU_4	4 RAM ₅ (4x2=8 GB)	869.50

Формулираните оптимизационни задачи са решавани чрез LINGO вер. 12 на персонален компютър с процесор Intel® Celeron® 2.93 GHz и 2 GB оперативна памет в средата на операционна система MS Windows XP ©. Времената на решаване на числените примери са от порядъка на секунди, но очевидно зависят от размера на решаваните проблемите. Графичното представяне на получените решения в Таблица 2.8 са илюстрирани на Фиг. 2.3, където с пунктирна линия са показани съществуващите модули и тяхната съвместимост, а с непрекъсната линия са свързани избраните в резултат на решената задача, модули.



Фиг. 2.3. Избрани модули в резултат на решената многокритериална задача:
 а) предпочитания на ЛВР-1; б) предпочитания на ЛВР-2; в) предпочитания на ЛВР-3

Резултатите от решаването на оптимизационните задачи определят Парето-оптимални комбинации на MB, CPU и RAM, удовлетворяващи условията за съвместимост и зададените потребителски изисквания. По този начин се потвърждава приложимостта на предложения многокритериален оптимизационен подход за проектиране конфигурацията на ПК.

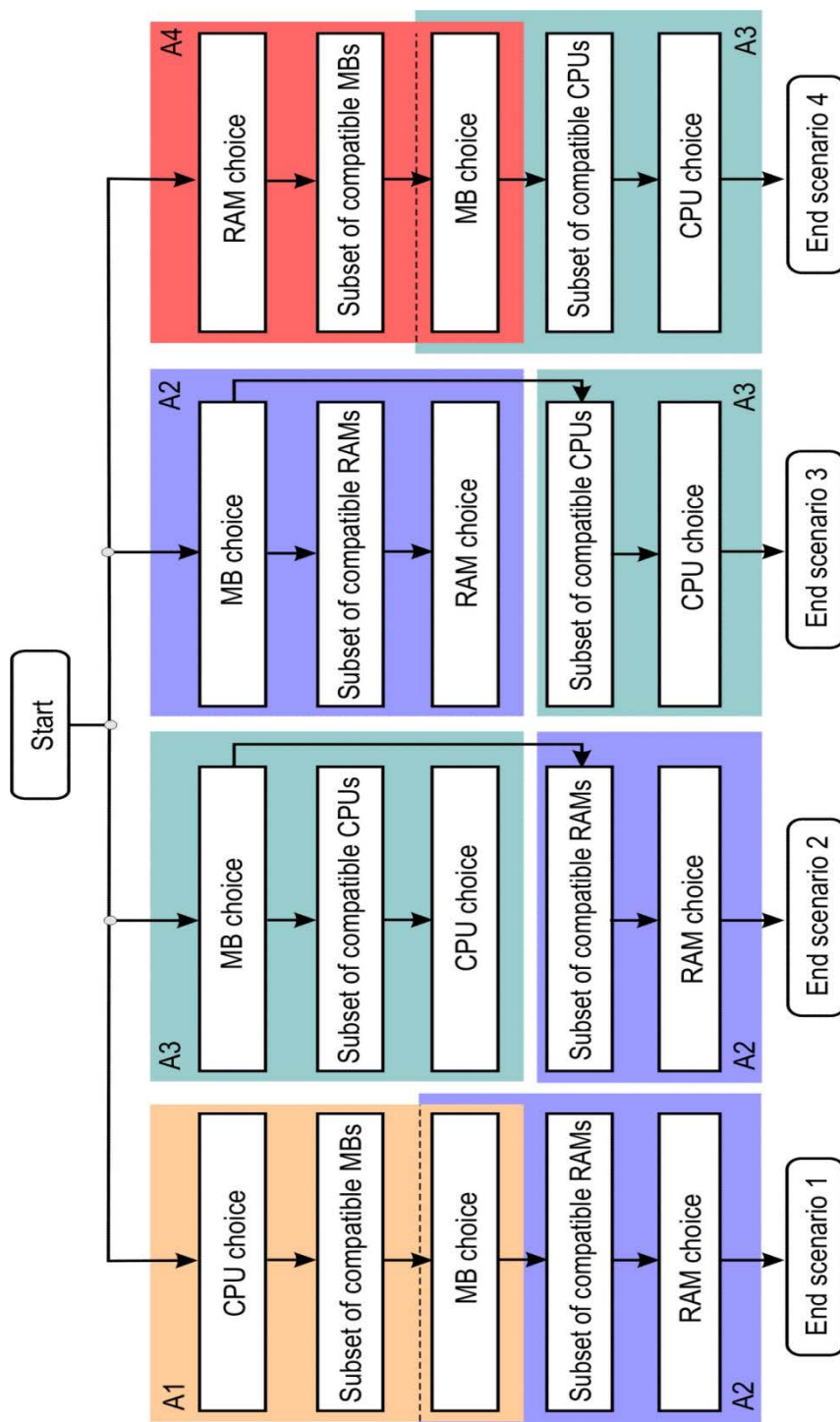
Решенията на многокритериалните задачи показват, че използването на различните набори от тегловни коефициенти (т.е. различни потребителски

предпочитания) водят до определяне на различни комбинации от модули. *Задача 1* (ЛВР-1) е пример за компромис между всички критерии, докато *Задача 2* (ЛВР-2) цели определяне на най-евтината конфигурация, а *Задача 3* (ЛВР-3) търси най-добра изчислителна мощност. По този начин, чрез използване на многокритериална оптимизация, могат да бъдат симулирани различни потребителски предпочитания и да се направи предварителна оценка на проектираната система по отношение удовлетворяване на зададените към нея изисквания.

2.5. Обобщен алгоритъм за итеративно и рационално проектиране на конфигурирането на персонални компютри

При разработването на CAD системи за модулно проектиране не винаги е желано (или възможно) заместването на човешката експертиза. В много случаи, експертизата на проектанта може да бъде подпомогната от автоматизирана система, определяща допустими комбинации от модули, от които той може да избира. Един от основните проблеми при модулното проектиране е реализирането на избор, съответстващ на допустими (съвместими) комбинации от модули. При това, всеки конкретен пример на такава система има специфични изисквания, които трябва да бъдат отразени чрез разработването на подходящи алгоритми и структури на данните. Известно е, че структурите от данни и алгоритмичните реализации са тясно свързани (Вирт, 1986) (Кнут, 1976). Доброто познаване на структурите от данни позволява анализиране на проблемите в дълбочина и води до реализирането на ефективни решения. Разработването на работоспособни алгоритми трябва да отразява възможно най-точно логиката за вземане на решения от експертите. Сложността на алгоритмите може да бъде съществено повлияна от структурите на данните и определянето на най-подходящи за конкретния случай структури на данните ще определи и ефективни алгоритми. Това е илюстрирано чрез предложени структури на данните и съответен обобщен алгоритъм за итеративно и рационално конфигуриране на компютърна система, както е показано на Фиг. 2.4 (Mustakerov & Borissova, 2013).

Предложеният алгоритъм е съставен от 4 клона, съответстващи на 4 сценария за вземане на решение при избора на основните компютърни модули. Отделните клонове на алгоритъма се отличават един от друг по реда на избор на модулите. Първият клон на алгоритъма (отляво-надясно) започва с избор на CPU. Изборът на CPU определя подмножество на съвместими MB, от които да се направи избор на конкретна MB.



Фиг. 2.4. Обобщен алгоритъм за проектиране на конфигурацията на ПК

Изборът на MB на свой ред определя подмножество на съвместими памети, от които да се избира. Вторият клон на алгоритъма започва с избор на MB, минава през избора на CPU и завършва с избор на RAM модула.

Третият клон на алгоритъма също започва с избор на MB, но продължава през избор на RAM и след това избор на CPU. Четвъртият клон на алгоритъма описва сценарий, при който изборът започва от RAM модула.

Изборът на RAM влияе върху избора на MB, който от своя страна определя избора на CPU. Тук няма отношения на съвместимост между RAM и CPU модули, които да бъдат взети предвид. Тези четири алгоритмични клона (A1, A2, A3, A4) са описани чрез псевдо-код, както е показано по-долу (Mustakerov & Borissova, 2013).

A1: Algorithm realization for CPU => MB choice of compatible modules

1. Given the array *CPU-Data[j]* or *CPUm[j]* **set** *j=c*; $\prec$ choice of particular CPU
2. **for** *i = 0 to number of MBs do $\prec$ create list of compatible MBs for "drop-down" menu*
3. **if** *MB-CPU[i,c] > 0 then*
4. *MBc[i] = MB-Data[i,0]*; $\prec$ selectable item of "drop-down" list
5. **else** *MBc[i] = ""*; $\prec$ empty (not selectable) item of "drop-down" list
6. **set** *i = m*; $\prec$ choice of particular MB with index *m*;
7. **end**

A2: Algorithm realization for MB => RAM choice of compatible modules

1. Given the array $MB_Data[i]$ or $MBc[i]$ **set** $i = m$; $\prec$ choice of particular MB
2. **for** $j = 0$ **to** number of RAMs **do** $\prec$ create list of compatible RAMs for "drop-down" menu
3. **if** $MB_RAM-1[m,j] > 0$ **then**
4. $RAMm[j] = RAM_Data[i,0]$; $\prec$ selectable item of "drop-down" list
5. **else** $RAMm[j] = ""$; $\prec$ empty (not selectable) item of "drop-down" list
6. **set** $j = r$; $\prec$ choice of particular RAM with index r
7. **get** N_{ram} $\prec$ get entered RAM modules number
8. **if** $N_{ram} > MB_RAM-2[m,j]$ **then** message **goto** 7
9. **else if** $N_{ram} * RAM_Data[r,1] > MB_RAM-3[m,r]$
 then message **goto** 7
10. **else end**

A3: Algorithm realization for MB => CPU choice of compatible modules

1. Given the array $MB\text{-}Data[i]$ or $MBc[i]$ **set** $i = m$; $\prec$ choice of particular MB
2. **for** $j = 0$ **to** number of CPUs **do**
3. **if** $MB\text{-}CPU[m,j] > 0$ **then** $\prec$ create list of compatible CPUs for

"drop-down" menu
4. $CPUm[j] = CPU\text{-}Data[i,0]$; $\prec$ selectable item of "drop-down" list
5. **else** $CPUm[j] = ""$; $\prec$ empty (not selectable) item of "drop-down" list
6. **set** $j = c$; $\prec$ choice of particular CPU with index c ;
7. **end**

A4: Algorithm realization for RAM => MB choice of compatible modules

1. Given the array *RAM-Data[j]* **set** *j = r*; $\leftarrow$ choice of particular RAM,
2. **for** *i = 0 to number of MBs do*
 if *MB-RAM-1[i,r] > 0 then* $\leftarrow$ create list of compatible MBs for "drop-down" menu
3. *MBr[i] = MB-Data[i,0]*; $\leftarrow$ selectable item of "drop-down" list
4. **else** *MBr[i] = ""*; $\leftarrow$ empty (not selectable) item of "drop-down" list
5. **set** *i = m*; $\leftarrow$ choice of particular MB with index m
6. **get** *N_{ram}* $\leftarrow$ get entered RAM modules number
7. **if** *N_{ram} > MB-RAM-2[m,j]* **then** message **goto** 7
8. **else if** *N_{ram} * RAM-Data[r,1] > MB-RAM-3[m,r]* **then** message **goto** 7
9. **else end**

Структурите от данни, използвани от тези алгоритми, са представени чрез двумерни масиви, изразяващи отношенията за съвместимост между отделните модули, както са показани в Таблицы 2.6, 2.7, 2.8 и 2.9:

Таблица 2.6. Съвместимост между модулите MB и CPU по отношение на сокет

	<i>CPU₀</i>	<i>CPU₁</i>	<i>CPU₂</i>	<i>CPU₃</i>	<i>CPU₄</i>	<i>CPU₅</i>	<i>CPU₆</i>
<i>MB₀</i>	1	1	1	1	0	0	0
<i>MB₁</i>	1	1	1	1	0	0	0
<i>MB₂</i>	1	1	1	1	0	0	0
<i>MB₃</i>	1	1	1	1	0	0	0
<i>MB₄</i>	1	0	0	0	0	0	0
<i>MB₅</i>	0	0	0	0	1	1	0
<i>MB₆</i>	0	0	0	0	0	0	1

Таблица 2.7. Съвместимост между модулите MB и RAM по отношение на честота

	<i>RAM₀</i>	<i>RAM₁</i>	<i>RAM₂</i>	<i>RAM₃</i>	<i>RAM₄</i>	<i>RAM₅</i>	<i>RAM₇</i>
<i>MB₀</i>	0	667	1066	0	800	0	0
<i>MB₁</i>	0	667	0	0	800	0	0
<i>MB₂</i>	0	667	1066	0	800	0	0
<i>MB₃</i>	0	667	1066	0	800	0	0
<i>MB₄</i>	400	0	0	0	0	0	0
<i>MB₅</i>	0	0	0	1066	0	1333	1600
<i>MB₆</i>	0	0	0	1066	0	1333	1600

Таблица 2.8. Съвместимост между модулите MB и RAM по отношение на броя слотове

	RAM ₀	RAM ₁	RAM ₂	RAM ₃	RAM ₄	RAM ₅	RAM ₇
MB ₀	0	4	4	0	4	0	0
MB ₁	0	2	0	0	2	0	0
MB ₂	0	2	2	0	2	0	0
MB ₃	0	4	4	0	4	0	0
MB ₄	2	0	0	0	0	0	0
MB ₅	0	0	0	4	0	4	4
MB ₆	0	0	0	4	0	4	4

Таблица 2.9. Съвместимост между модулите MB и RAM по отношение на капацитета

	RAM ₀	RAM ₁	RAM ₂	RAM ₃	RAM ₄	RAM ₅	RAM ₇
MB ₀	0	16	16	0	16	0	0
MB ₁	0	8	0	0	8	0	0
MB ₂	0	4	4	0	4	0	0
MB ₃	0	8	8	0	8	0	0
MB ₄	2	0	0	0	0	0	0
MB ₅	0	0	0	16	0	16	16
MB ₆	0	0	0	8	0	8	8

Ако и други модули трябва да се вземат предвид, то могат да бъдат въведени по подобен начин и съответни структури на данните и за тези модули.

Целта на описаните алгоритмични реализации е да се използват в подходяща софтуерна система с лесен и интуитивен интерфейс за конфигуриране на ПК. Използването на отделните клонове от обобщения алгоритъм позволява въвеждане на граници при избора на модулите. Например, клонът на алгоритъма A1 започва с избор на конкретен модул CPU, който избор определя колона в двумерния масив $MB-CPU[i,j]$. Списъкът на съвместимите MB с процесорите се създава чрез замяна на всички ненулеви стойности в колоната на избрания процесор с имена на MB, съхранявани в първата колона на двумерния масив $MB-Data[i,j]$. Списъкът на съвместимите MB с CPU се създава чрез замяна на всички ненулеви стойности от колоната на избрания CPU с имена на MB, съхранявани в първата колона на двумерния масив $MB-Data[i,j]$. Клетките, които съответстват на нулеви стойности остават празни като индикация, че тези елементи не са избираеми в “падащото” меню. Масивът $MB-RAM-1[i,j]$ се използва за определяне на списъка на съвместимите RAM памети (клон A2 на алгоритъма), или съвместими MB (клон A4), докато масивите $MB-RAM-2[i,j]$ и $MB-RAM-3[i,j]$ се използват за проверка дали въведения брой RAM модули отговаря на броя слотове MB и поддържания максимален размер RAM. Когато изборът започва с избор на конкретен

модул МВ (клон А3), от определен ред от $MB-CPU[i,j]$ се избира и списъкът от съвместими процесори с избраната МВ чрез използване на данните от първата колона на двумерния масив $CPU-Data[i,j]$. Създаването на списък от съвместими памети и МВ следва същата логика, но се използват три масива за описание на изискваните съвместимости.

Описаните до тук алгоритмични реализации касаят проектирането на конфигурацията на ПК чрез итеративни процедури, които се повтарят докато потребителят бъде удовлетворен от получените резултати. Този процес може да бъде съществено съкратен като се предостави възможност на потребителя да задава различни долни и/или горни граници за някои от параметрите на проектираната система. Описаните структури от данни позволяват лесна и интуитивна алгоритмична реализация на подобен процес, изразяващ се в реализиране на *рационален избор* на модули за проектираната система.

Например, ако потребителят се нуждае от определен *брой на ядра на процесора* N_{core} , то тази стойност може да се използва в следната алгоритмична реализация (алгоритъм А5а):

A5a: Algorithmic realization of intelligent CPU choice	
1.	get N_{core} $\leftarrow$ get entered CPU core number
2.	for $i = 0, k = 0$ to number of CPUs do $\leftarrow$ create list of CPUs fitting to this requirement
3.	if $CPU-Data[j,1] \geq N_{core}$ then
4.	$CPUc[j] = CPU-Data[j,0]$; $\leftarrow$ selectable item of "drop-down" list
5.	else
	$CPUc[j] = ""$; $\leftarrow$ empty (not selectable) item of "drop-down" list
	$k = k + 1$; $\leftarrow$ counter of empty (not selectable) items
6.	if $k = \text{number of CPUs}$ then message goto 1
7.	else set $j = c$; $\leftarrow$ choice of CPU with core number N_{core} ;
8.	continue $\leftarrow$ continuing with next part of modules choice algorithm;

Друг пример, за реализиране на рационалния избор на модули, е изборът на модул, който да удовлетворява зададена горна граница. Например, ако потребителят зададе горна граница за максималната цена $Price_{mb}$ за МВ, то изборът на подходяща МВ може да се направи чрез следната алгоритмична реализация (А5b):

A5b: Algorithmic realization of intelligent MB choice

1. **get** $Price_{mb}$ $\leftarrow$ get entered maximal value for MB price
2. **for** $i = 0, k = 0$ **to** number of MBs **do** $\leftarrow$ create list of MBs fitting to this requirement
3. **if** $MB-Data[i,3] \leq Price_{mb}$ **then**
4. $MBc[i] = MB-Data[i,0]$; $\leftarrow$ selectable item of "drop-down" list
5. **else**
6. $MBc[i] = ""$; $\leftarrow$ empty (not selectable) item of "drop-down" list
7. $k = k + 1$; $\leftarrow$ counter of empty (not selectable) items
8. **if** $k = \text{number of MBs}$ **then** message **goto** 1
9. **else set** $i = m$; $\leftarrow$ choice of MB with price less or equal to $Price_{mb}$;
10. **continue** $\leftarrow$ continuing with next part of modules choice algorithm;

Резултатът от изпълнението на алгоритмичната реализация (A5b) ще бъде определяне на MB, чиято цена е по-малка или равна на зададената $Price_{mb}$.

При необходимост от определяне на граница за цената за общата конфигурация, рационалният избор на модули се реализира от друга алгоритмична процедура (A6):

A6: Algorithmic realization of the design cost calculation

1. **get** m, c , and r $\leftarrow$ get indexes of chosen modules
2. $Price_{sum} = MB-Data[m,3] + CPU-Data[c,3] + RAM-Data[r,3]$
3. **end**

Използването на подобен подход към структуриране на данните и разработването на съответни алгоритми, може да бъде реализирано в автоматизирана система, подпомагаща конфигурирането на модулни инженерни системи и вземаща предвид разнообразни потребителски изисквания.

Описаните методи за оптимално, итеративно и рационално проектиране могат да бъдат използвани и за други инженерни системи. Формулираните едно- и многокритериални модели дават възможност за получаване на оптимално или Парето-оптимално предложение за проектиране на инженерни системи, отчитайки зависимостите между отделните модули. При формулирането на подобни задачи за други класове системи е необходимо да се вземат предвид специфичните особености на моделираната система.

Глава 3:

Оптимално проектиране на модулни инженерни системи, вземащи предвид съществуващите взаимовръзки между модулите и външните експлоатационни условия

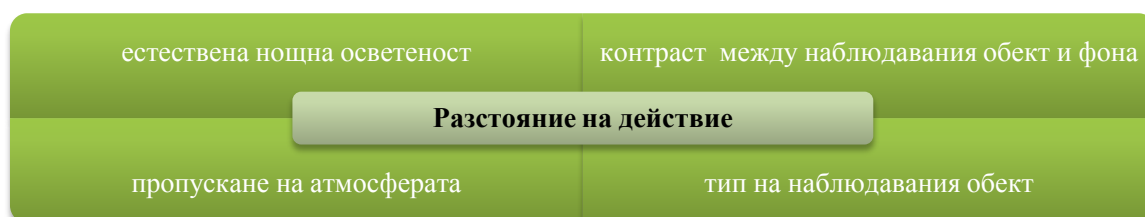
Съществуват класове модулни инженерни системи, за които е характерно, че параметрите им зависят както от параметрите на модулите, така и от външните условия, при които се използват. Предложеният в Глава 2 модел за оптимално проектиране на инженерни системи е разширен и модифициран, така че да бъде приложим и този клас системи. Един типичен пример за такава система са уредите за нощно виждане (УНВ). Изборът на УНВ като пример за такава система е мотивиран от факта за масовото им използване в различни области, не само за военни, но и за граждански цели. Военните приложения са традиционна област за УНВ и включват управление на бойни машини от различни типове (самолети, хеликоптери, танкове, морски съдове, транспортни машини), водене на бойни действия в нощни условия (стрелба, патрулиране, разузнаване) и др. За граждански цели УНВ намират приложение при спасителни операции, при бедствия и аварии, за навигация и управление на транспортни средства, за научни изследвания, за охрана, за туризъм и развлечения и др. УНВ се характеризират с добре структуриран модулен дизайн и с наличие на множества от модули, които могат да се използват за проектиране на уреди с различни функционални характеристики. Тъй като параметрите на УНВ са в пряка зависимост от външните условия на експлоатация, то те са достатъчно описателен пример за проблемите при проектиране на модулни инженерни системи, вземащи предвид съществуващите взаимовръзки и външните експлоатационни условия. Съществуват два основни класа уреди за нощно виждане, в зависимост от използваните технологии, базирани на топлинно излъчване и на усилване на светлината.

В тази глава са описани получените резултати при оптимално проектиране на модулни инженерни системи, вземащи предвид както съществуващите взаимовръзки, така и външните експлоатационни условия. За да се тества практическата приложимост

на оптималното проектиране за този клас модулни системи е използван един типичен представител, а именно уредите за нощно виждане (очила, прицели и бинокли), базирани на технологията на усилване на светлината. Описани са: 1) детерминиран, стохастичен и обобщен математически модели за проектиране на модулни инженерни системи, вземащи предвид съществуващите взаимовръзки между модулите и външните експлоатационни условия, 2) математически модели, определящи външните експлоатационни условия, при които се постига една и съща стойност на зададен параметър на системата, 3) методи за проектиране на системи за този клас – итеративно, рационално и оптимално проектиране, 4) обобщен алгоритъм за оценка на параметрите на проектираните системи. Получените резултати са отразени в публикации D5, D7, D22, D23, D29, D30, D31 и D33.

3.1. Влияние на външните експлоатационни условия, на примера на проектиране на уреди за нощно виждане

Като достатъчно представителен пример, илюстриращ особеностите на проектирането на системи от този клас, са избрани уредите за нощно виждане. Масово разпространените УНВ използват технология на усилване на отразената светлина от нощното небе, от отражението на различни светлинни източници на земната или водна повърхност и т.н. Основен и най-съществен за практическото използване на УНВ параметър е *разстоянието на действие*, което зависи, както от характеристиките на уреда, така и от външните условия на наблюдение – естествена нощна осветеност, пропускане на атмосферата, контраст между наблюдавания обект и фона, и типа на наблюдавания обект (Фиг. 3.1):

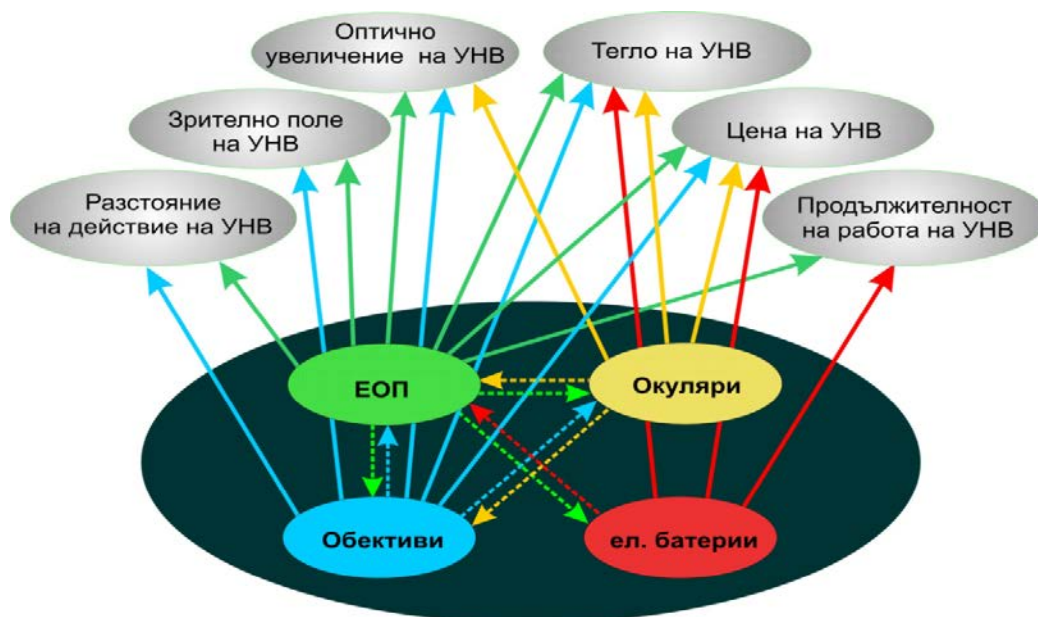


Фиг. 3.1. Външни условия, влияещи върху разстоянието на действие на УНВ

Има много различни променливи, които влияят на разстоянието на действие на УНВ. Колкото по-голям е обектът, толкова по-лесно е видим. Друга променлива, влияеща на разстоянието на действие е нощната осветеност. Колкото повече е нощната осветеност (звездна светлина, лунна светлина, инфрачервена светлина) толкова по-добре и по-голямо ще бъде разстоянието на действие. Не без значение е и контрастът

между наблюдавания обект и фона, като по-големият контраст обуславя и по-голямо разстояние на действие. Пропускливостта на атмосферата също оказва влияние.

Основен компонент на УНВ е т. нар. *оптоелектронен канал* (обектив, ЕОП, окуляр, захранване). Големото разнообразие от съществуващи основни модули за оптоелектронния канал – обективи, ЕОПи, окуляри, захранвания, поставя въпроса кои от тях да бъдат избрани, така че проектираният УНВ да отговаря на зададени параметри. При това, трябва да се вземат предвид функционалните зависимости между отделните модули на оптоелектронния канал. Например, отношението на фокусното разстояние на обектива и окуляра определя увеличението на УНВ, което за бинокли и прицели е по-голямо от единица, а за очила за нощно виждане това увеличение е равно на единица. Екранът на ЕОП трябва да се вижда целия през окуляра, тъй като това осигурява пълноценното използване на ЕОП. Това от своя страна определя зависим избор между диаметрите на ЕОП и окуляра. Зрителното поле на обектива зависи от диаметъра на ЕОП и фокусното разстояние на обектива. Батерийното електрическо захранване пряко определя продължителността на работата на уреда с избрания тип батерии и влияе върху теглото на уреда. Разстоянието на действие на УНВ зависи от усилването на ЕОП и от параметрите на обектива. Всички тези зависимости трябва да се вземат предвид при формулирането на комбинаторния проблем за определяне на най-добрата комбинация от модули, като същевременно се вземат предвид и изискванията към параметрите на проектирания уред (Фиг. 3.2).



Фиг. 3.2 Зависимости между модулите и параметрите на УНВ

3.2. Детерминиран оптимизационен модел на уреди за нощно виждане

При разработването на оптимизационни модели трябва да се определи критерий за оптималност на проектирането. Като обобщен и достатъчен за практически потребности оптимизационен критерий може да се разглежда т.нар. “качество” на УНВ. Най-често поставяните изисквания към качеството на УНВ могат да се обобщят чрез следните компоненти на критерия за качество на УНВ:

- разстояние на действие (разстояния на откриване, разпознаване и идентификация),
- зрително поле,
- относителен отвор на обектива,
- фокусировка на обектива,
- отдалеченост на изходната зеница,
- аберации на обектива и окуляра,
- продължителност на работата,
- тегло на оптоелектронния канал,
- цена на оптоелектронния канал.

Тези компоненти участват в критерий за качество на УНВ, който се използва при формулиране на оптимизационни модели на оптоелектронния канал на УНВ.

Детерминираният оптимизационен модел е формулиран при максимизиране на разстоянието на действие, зрителното поле, относителния отвор на обектива и отдалечеността на изходната зеница и минимизиране на разстоянието за фокусировка, аберациите на обектива и окуляра, както и теглото и цената на използваните модули, а което параметрите на външните условия на наблюдение (τ_a – пропускане на атмосферата, E – естествена нощна осветеност, K – контраст между наблюдавания обект и фона, $A^*_{об}$ – приведена площ на наблюдавания обект) се приемат за детерминирани величини с известни стойности:

$$\begin{aligned} \max \{R, W_{об}, (1/k), ОИЗ\} \\ \min \{FR, AC_{об}, AA_{об}, AD_{об}, AK_{об}, AC_{ок}, AA_{ок}, AD_{ок}, AK_{ок}, T, C\} \end{aligned} \quad (3.1)$$

$$R = \sqrt{0.07 \sum_j y_j (f_{об}^j \tau_{об}^j D_{вх}^j) \sum_i x_i \left(\frac{S^i \delta^i}{M \Phi_{пр.фк}} \right) \tau_a E K A^*} \quad (3.2)$$

$$\sum_{i=1}^m x_i = 1, x_i \in \{0, 1\} \quad (3.3)$$

$$\sum_{j=1}^n y_j = 1, y_j \in \{0, 1\} \quad (3.4)$$

$$W_{o\bar{o}} = \sum_{j=1}^n y_j W_j^{o\bar{o}} \quad (3.5)$$

$$\frac{1}{k} = \sum_{j=1}^n y_j \frac{1}{k_j} \quad (3.6)$$

$$OИЗ = \sum_{k=1}^l z_k OИЗ_k \quad (3.7)$$

$$\sum_{k=1}^l z_k = 1, z_k \in \{0, 1\} \quad (3.8)$$

$$\sum_{j=1}^n y_j F_j^{o\bar{o}} = \gamma \sum_{k=1}^l z_k F_k^{ok} \quad (3.9)$$

$$AC_{o\bar{o}} = \sum_{j=1}^n y_j AC_j^{o\bar{o}} \quad (3.10)$$

$$AA_{o\bar{o}} = \sum_{j=1}^n y_j AA_j^{o\bar{o}} \quad (3.11)$$

$$AD_{o\bar{o}} = \sum_{j=1}^n y_j AD_j^{o\bar{o}} \quad (3.12)$$

$$AK_{o\bar{o}} = \sum_{j=1}^n y_j AK_j^{o\bar{o}} \quad (3.12)$$

$$FR_{ob} = \sum_{j=1}^n y_j FR_j^{ob} \quad (3.14)$$

$$AC_{ok} = \sum_{k=1}^l z_k AC_k^{ok} \quad (3.15)$$

$$AA_{ok} = \sum_{k=1}^l z_k AA_k^{ok} \quad (3.16)$$

$$AD_{ok} = \sum_{k=1}^l z_k AD_k^{ok} \quad (3.17)$$

$$AK_{ok} = \sum_{k=1}^l z_k AK_k^{ok} \quad (3.18)$$

$$W_{ok} = \sum_{k=1}^l z_k W_k^{ok} \quad (3.19)$$

$$T = T_{EOП} + T_{o\bar{o}} + T_{ok} + T_{\bar{o}am} \quad (3.20)$$

$$T_{EOП} = \sum_{i=1}^m x_i T_i^{EOП} \quad (3.21)$$

$$T_{o\bar{o}} = \sum_{j=1}^n y_j T_j^{o\bar{o}} \quad (3.22)$$

$$T_{ok} = \sum_{k=1}^l z_k T_k^{ok} \quad (3.23)$$

$$T_{\delta am} = N \left(\sum_{p=1}^t a_p \left(s_p \sum_{q=1}^{k_p} b_q^p H_B^q + \sum_{p=1}^t t_p \right) \right) \quad (3.24)$$

$$\sum_{p=1}^t a_p = 1 \quad (3.25)$$

$$\sum_{p=1}^t \left(a_p - \sum_{q=1}^{k_p} b_q^p \right) = 0 \quad (3.26)$$

$$BL = \frac{Q_{\delta am}}{I_{EOП}} = N \left(\sum_{p=1}^t a_p \left(s_p \sum_{q=1}^{k_p} b_q^p Q_{\delta am}^q \right) \right) \frac{1}{\sum_{i=1}^m x_i I_i^{EOП}} \quad (3.27)$$

$$C = C_{EOП} + C_{об} + C_{ок} + C_{\delta am} \quad (3.28)$$

$$C_{EOП} = \sum_{i=1}^m x_i C_i^{EOП} \quad (3.29)$$

$$C_{об} = \sum_{j=1}^n y_j C_j^{об} \quad (3.30)$$

$$C_{ок} = \sum_{k=1}^l z_k C_k^{ок} \quad (3.31)$$

$$C_{\delta am} = N \left(\sum_{p=1}^t a_p \left(s_p \sum_{q=1}^{k_p} b_q^p C_B^q + \sum_{p=1}^t k_p \right) \right) \quad (3.32)$$

Съществена роля в така формулирания оптимизационен модел играят двоичните целочислени променливи x_i , y_j , z_k , a_p и b_q^p . Те гарантират избор на единствен модул (обектив, ЕОП, окуляр, батерия) от предварително зададени множества.

Един от най-важните компоненти в предложения критерий за качество на УНВ е разстоянието на действие. Формула (3.2) определя разстоянието на действие на УНВ, като взема предвид всички съществени параметри на оптоелектронния канал, както и външните условия на наблюдение (Borissova и Mustakerov, 2006). Чрез въведения параметър “приведена площ” на наблюдавания обект A^* (Borissova и Mustakerov, 2006), могат да бъдат изчислявани различните разновидности на разстоянието на действие R , където: $D_{вх}$ – диаметър на входната зеница на обектива в m, $f_{об}$ – фокусно разстояние на обектива в mm, τ_o , τ_a – пропускане на оптиката, пропускане на атмосферата, $\Phi_{пр.фк}$ – прагова чувствителност на фотокатода на ЕОП в lm, δ – разделителната способност на ЕОП в lp/mm, S – интегрална чувствителност на фотокатода на ЕОП в A/lm, M – отношение сигнал/шум, E – естествена нощна осветеност в lx, K – контраст, A^* – приведена площ на наблюдавания обект в m².

Зрителните полета на обектива $W_{об}$ и окуляра $W_{ок}$ участват в критерия за качество, чрез (3.5) и (3.19), където променливите y_j и z_k определят избора на обектив и окуляр и са целочислени двоични променливи, $y_j, z_k \in \{0, 1\}$.

Изборът на обектив с определен относителен отвор, се осъществява чрез променливите y_j в (3.6), където: $(1/k)$ – относителен отвор на съответния обектив. За реализиране избор на обектив с определено разстояние за фокусиране FR се използва равеството (3.14). Аберациите на избрания обектив се определят от (3.10) – (3.12), а аберациите за окуляра се базират на зависимостите (3.15) – (3.18). Изборът на окуляр, с определена отдалеченост на изходната зеница, се осъществява чрез (3.7). Продължителността на работа на УНВ зависи както от капацитета на използвания тип батерии и от необходимия ток за захранване на ЕОП (Borissova, 2007).

Като оценка за теглото на оптоелектронния канал са използвани зависимостите (3.20), изразяващи съответно сумата от теглата на ЕОП, обектива, окуляра и електрическите батерии, където: N – брой на паралелните свързан батериите, t типа батерии с различно захранващо напрежение от 1,5V или 3V и всеки t тип батерия може да има k_p вида с различен капацитет, a_p – двоични променливи за избор на тип батерии, b_q^p – двоични променливи за избор на батерия от q -подтип на p -тип батерия, H_B^q – тегло на батерия от p -тип и q -вид електрически батерии, t_p – тегло на механичната конструкция за тип p електрическа батерия. Цената на оптоелектронния канал се определя от (3.28), където C_B^p – цена на батерия от тип p и вид q , k_p – цена на механичната конструкция за батерия от тип p , $s_p = 1$ за батерии от 3V и $s_p = 2$ за батерии от 1.5V. Единственият избор на вида на необходимите електрически батерии, без комбиниране на различните видове батерии и подтипове за всяко $p \in \{1, 2, \dots, t\}$ се осигурява чрез зависимостите (3.25) и (3.26).

3.3. Стохастичен оптимизационен модел на уреди за нощно виждане

При построяване на модел, адекватен в по-голяма степен на реалните условия на наблюдение, е необходимо да се отчита вероятностния характер на външните условия на наблюдение – естествена нощна осветеност, пропускане на атмосферата, контраст между наблюдавания обект и фона, и площ на наблюдавания обект. Това изисква тези величини да се разглеждат като случайни величини и да се използват за дефиниране на съответен стохастичен модел на оптоелектронния канал на УНВ.

Пропускане на атмосферата. Преминаването на оптично лъчение през въздушна среда е съпроводено от поглъщане и разсейване, при което лъчението отслабва. Разсейването на оптичното лъчение в атмосферата се дължи на време-пространствените нееднородности на нейното съдържание. За големите частици разсейването не зависи от дължината на вълната, докато за малки частици, съизмерими с дължината на вълната, отслабването на разсейването на лъчението е пропорционално на дължината на вълната λ^{-4} (Релеевско разсейване). На практика то слабо зависи от метеорологичните условия. Численото пресмятане на поглъщането в реална атмосфера изисква познаване на газовия състав на атмосферата (съотношения между газовите компоненти и големина на частиците), на съответната височина над земната повърхност, на конкретните метеорологични условия и на много други климатични фактори (Sospedra и др., 2004). Пропускането на атмосферата в даден географски регион е непрекъсната случайна величина със стойности в някакъв интервал, като максимум на прозрачността на атмосферата се наблюдава през зимата, а минимум през лятото. За видимия оптичен диапазон 400–700 nm, средният коефициент на прозрачност на атмосферата е измерен със стойности от 0.68 до 0.93. С преминаване към инфрачервената област от спектъра, прозрачността на атмосферата се подобрява. При метеорологична видимост от 10 km, пропускането на атмосферата до височина 1 km при дължина на вълната 600 nm, е 0.72, а в центъра на областта на спектъра 1400–1800 nm, то е 0.93 (Волков, 2001). Случайният характер на пропускането на атмосферата може да се вземе предвид в стохастичния оптимизационен модел, чрез използване на съответното ѝ математическо очакване:

$$T_a^{cm} = \sum_{i=1}^p P_{\tau_a}^i \tau_a^i, \text{ където: } \sum_{i=1}^p P_{\tau_a}^i = 1 \quad (3.34)$$

Контрастът между наблюдавания обект и фона при наблюдение също е случайна величина. Практическите стойности за контраста между наблюдавания обект и фона са в диапазона от 0.05 до 0.5 (Кошавцев и др., 1999). В стохастичния оптимизационен модел на оптоелектронния канал на ОНВ случайният характер на контраста между наблюдавания обект и фона се отчита чрез използване на математическото му очакване:

$$K^{cm} = \sum_{j=1}^t P_K^j K^j, \text{ където: } \sum_{j=1}^t P_K^j = 1 \quad (3.35)$$

Естествена нощна осветеност. Стойностите на реалната естествена нощна осветеност E за някои характерни случаи могат да бъдат представени като пълна луна

(0.1 lx), половин луна (0.5 lx), четвърт луна (0.01 lx), звездно небе (0.001 lx) и облачно небе (0.0001 lx) (Ryer, 1997). В стохастичният модел на опто-електронния канал на УНВ, случайната природа на естествената нощна осветеност се взема предвид чрез математическото ѝ очакване:

$$E^{cm} = \sum_{k=1}^h P_E^k E^k, \text{ където: } \sum_{k=1}^h P_E^k = 1 \quad (3.36)$$

Площ на наблюдавания обект. При формулирания детерминиран модел, площта на наблюдавания обект се приема за известна. С достатъчна за практиката точност, за случаите на наблюдение на *танк, товарен автомобил, джип,* и др., това може да се приеме за достоверно в голяма степен. За случая на наблюдение на човешка фигура трябва да се вземе предвид, че размерите ѝ трудно могат да бъдат приети за детерминирани. В най-общия случай те зависят от расата и пола (Steckel & Prince, 2001). Това се отразява в стохастичния модел като площта на наблюдавания обект участва с математическото си очакване:

$$A_{об}^{cm} = \sum_{m=1}^l P_{A_{об}}^m A_{об}^m, \text{ където: } \sum_{m=1}^l P_{A_{об}}^m = 1 \quad (3.37)$$

От изложеното по-горе за стохастичната природа на величините – пропускане на атмосферата, естествена нощна осветеност, контраст и размери на наблюдавания обект, формулираният детерминиран оптимизационен модел се преобразува в стохастичен, като вместо детерминиранияте стойности се използват математическите им очаквания (Borissova & Mustakerov, 2009):

$$\begin{aligned} & \max \{R, W_{об}, (1/k), ОИЗ\} \\ & \min \{F, AC_{об}, AA_{об}, AD_{об}, AK_{об}, AC_{ок}, AA_{ок}, AD_{ок}, AK_{ок}, T, C\} \end{aligned} \quad (3.38)$$

$$R = \sqrt{0.07 \sum_j y_j (f_{об}^j \tau_{об}^j D_{ex}^j) \sum_i x_i \left(\frac{S^i \delta^i}{M \Phi_{пр.фк}} \right) T_a^{cm} E^{cm} K^{cm} A^{cm} *} \quad (3.39)$$

$$\sum_{i=1}^m x_i = 1, x_i \in \{0, 1\} \quad (3.40)$$

$$\sum_{j=1}^n y_j = 1, y_j \in \{0, 1\} \quad (3.41)$$

$$W_{об} = \sum_{j=1}^n y_j W_j^{об} \quad (3.42)$$

$$\frac{1}{k} = \sum_{j=1}^n y_j \frac{1}{k_j} \quad (3.43)$$

$$OИЗ = \sum_{k=1}^l z_k OИЗ_k \quad (3.44)$$

$$\sum_{k=1}^l z_k = 1, z_k \in \{0, 1\} \quad (3.45)$$

$$\sum_{j=1}^n y_j F_j^{o\delta} = \gamma \sum_{k=1}^l z_k F_k^{ok} \quad (3.46)$$

$$AC_{o\delta} = \sum_{j=1}^n y_j AC_j^{o\delta} \quad (3.46)$$

$$AA_{o\delta} = \sum_{j=1}^n y_j AA_j^{o\delta} \quad (3.48)$$

$$AD_{o\delta} = \sum_{j=1}^n y_j AD_j^{o\delta} \quad (3.49)$$

$$AK_{o\delta} = \sum_{j=1}^n y_j AK_j^{o\delta} \quad (3.50)$$

$$FR_{ob} = \sum_{j=1}^n y_j FR_j^{ob} \quad (3.51)$$

$$AC_{ok} = \sum_{k=1}^l z_k AC_k^{ok} \quad (3.52)$$

$$AA_{ok} = \sum_{k=1}^l z_k AA_k^{ok} \quad (3.53)$$

$$AD_{ok} = \sum_{k=1}^l z_k AD_k^{ok} \quad (3.54)$$

$$AK_{ok} = \sum_{k=1}^l z_k AK_k^{ok} \quad (3.55)$$

$$W_{ok} = \sum_{k=1}^l y_k W_k^{ok} \quad (3.56)$$

$$T = T_{EOИ} + T_{o\delta} + T_{ok} + T_{\delta am} \quad (3.57)$$

$$T_{EOИ} = \sum_{i=1}^m x_i T_i^{EOИ} \quad (3.58)$$

$$T_{o\delta} = \sum_{j=1}^n y_j T_j^{o\delta} \quad (3.59)$$

$$T_{ok} = \sum_{k=1}^l z_k T_k^{ok} \quad (3.60)$$

$$T_{\delta am} = N \left(\sum_{p=1}^t a_p \left(s_p \sum_{q=1}^{k_p} b_q^p H_B^q + \sum_{p=1}^t t_p \right) \right) \quad (3.61)$$

$$\sum_{p=1}^t a_p = 1 \quad (3.62)$$

$$\sum_{p=1}^t \left(a_p - \sum_{q=1}^{k_p} b_q^p \right) = 0 \quad (3.63)$$

$$BL = \frac{Q_{\bar{o}am}}{I_{EOП}} = N \left(\sum_{p=1}^t a_p \left(s_p \sum_{q=1}^{k_p} b_q^p Q_{\bar{o}am}^q \right) \right) \frac{1}{\sum_{i=1}^m x_i I_i^{EOП}} \quad (3.64)$$

$$C = C_{EOП} + C_{o\bar{o}} + C_{ok} + C_{\bar{o}am} \quad (3.65)$$

$$C_{EOП} = \sum_{i=1}^m x_i C_i^{EOП} \quad (3.66)$$

$$C_{o\bar{o}} = \sum_{j=1}^n y_j C_j^{o\bar{o}} \quad (2.67)$$

$$C_{ok} = \sum_{k=1}^l z_k C_k^{ok} \quad (3.68)$$

$$C_{\bar{o}am} = N \left(\sum_{p=1}^t a_p \left(s_p \sum_{q=1}^{k_p} b_q^p C_B^q + \sum_{p=1}^t k_p \right) \right) \quad (3.69)$$

$$T_a^{cm} = \sum_{i=1}^p P_{\tau_a}^i \tau_a^i \quad (3.70)$$

$$E^{cm} = \sum_{k=1}^h P_E^k E^k \quad (3.71)$$

$$K^{cm} = \sum_{j=1}^t P_K^j K^j \quad (3.72)$$

$$A_{o\bar{o}}^{cm} = \sum_{m=1}^l P_{A_{o\bar{o}}}^m A_{o\bar{o}}^m \quad (3.73)$$

В така дефинирания стохастичен модел на оптоелектронния канал на УНВ, се отчита вероятностният характер на външните условия на наблюдение – естествена нощна осветеност, пропускане на атмосферата, контраст между наблюдавания обект и фона, и площ на наблюдавания обект (3.70) – (3.73).

3.4. Обобщен модел за оптимално проектиране на модулни инженерни системи, вземащи предвид съществуващите взаимовръзки между модулите и външните експлоатационни условия

Основната идея на обобщения модел за оптимално проектиране на модулни инженерни системи, вземащи предвид съществуващите взаимовръзки между модулите и външните експлоатационни условия, е да се определи комбинация от модули, удовлетворяващи както зависимостите между модулите, така и зададени потребителски изисквания. Обобщеният математически модел за оптимално проектиране има вида (Mustakeroov & Borissova, 2007):

$$\max F(P) = (f_1(P), f_2(P), \dots, f_q(P)) \quad (3.74)$$

при ограничения

$$P = \sum_{j_i=1}^{J_i} P_{j_i, k_i}^i x_{j_i}^i \quad (3.75)$$

$$g(P) = (g_1(P), g_2(P), \dots, g_m(P)) \quad (3.76)$$

$$\sum_{j_i} x_{j_i}^i = 1, x \in \{0,1\} \quad (3.77)$$

$$P_{j_i, k_i}^{Li} \leq P_{j_i, k_i}^i \leq P_{j_i, k_i}^{Ui}, i = 1, n \quad (3.78)$$

В тази формулировка $f_1(P), f_2(P), \dots, f_q(P)$ описват q целеви функции (критерии за изпълнение) на вектора на променливите $P = \{ P_{j_i, k_i}^i \mid i=1, \dots, n, j_i \in \{J_i\}, k_i \in \{K_i\} \} \in R^n$, представляващ съвкупността от всички разновидности на модулите, необходими за проектирането на системата. R^n е параметричното пространство от модулите за реализиране на избор, j_i изразява разновидностите на модулите от i -ия тип, k_i са параметрите на i -ия модул от тип j_i и P_{j_i, k_i}^i е k_i -ия параметър на i -ия модул от тип j_i . $P' = \{ P_{s,r}^t \mid t \in \{i\}, r \in \{k_i\}, s \in \{j_i\} \}$ е векторът на решението с избраните параметри на модулите в резултат на оптималния комбинаторен избор. Оптималният избор се реализира чрез използване на ограниченията (3.75) като се използват някои двоични целочислени променливи $X = \{ x_{j_i}^i \}$, при ограничения (3.77). Реалистичното оптимално проектиране включва не само целеви функции, но също така и ограничения, които определят граници в пространството на променливите. Например, с P_{j_i, k_i}^{Li} и P_{j_i, k_i}^{Ui} са означени долната и горната граница на променливите и функциите на ограниченията $(g_1(P), g_2(P), \dots, g_m(P))$ описващи специфичните зависимости, технически и потребителски изисквания на проектираната система. Това са зависимостите, при които изборът на даден модул i от тип j_i ограничава избора на другите типове модули от общото множество на модулите.

3.5. Многокритериални модели за определяне на външните условия на наблюдение, съвместими с параметрите на уреди за нощно виждане

В процеса на проектиране на УНВ могат да се използват детерминирани или стохастични величини за параметрите на външните условия на наблюдение. След изработване на устройството и определяне на параметрите му, представлява интерес определянето на външните условия на наблюдение, при които това устройство ще има

същото разстояние на действие. Такава оценка е интересна и за предлаганите на пазара УНВ, когато параметрите на УНВ се дават в каталожни спецификации, но не винаги се указва при какви външни условия е определено специфицираното разстояние на действие. От потребителска гледна точка е важно да бъдат определени диапазоните на изменение на външните условия на наблюдение, при които уредът ще има специфицираното разстояние на действие.

3.5.1. Граници на изменение на външните условия на наблюдение, съвместими с определено разстояние на действие на уреда

Параметрите, които влияят на разстоянието на действие са както вътрешни, така и от външни параметри (Borissova & Mustakerov, 2009): 1) разделителна способност на ЕОП – мярка за това колко линии с различна интензивност (светло до тъмно) могат да бъдат различени в рамките на милиметър площ на екрана; 2) отношение сигнал-шум – мярка за светлинния сигнал, достигащ окото, разделен на шума, който се вижда с просто око (Higginbotham, 2006; Riegler и др., 1991); 3) фотокатодна чувствителност на ЕОП – способността на фотокатодния материал за производството на електрони, при облъчване с фотони (Task, 1992); 4) диафрагмено число f-число – изразява отношение от фокусното разстояние на обектива към диаметъра на входната зеница; 5) естествена нощна осветеност – пасивните УНВ използват налична околната светлина от звезди и луна и т.н. (Marasco & Task, 2003); 6) пропускане на атмосферата – зависи от температурата на въздуха; атмосферно налягане; относителна влажност; брой, размер и разпределение на атмосферните аерозоли; концентрация на необичайни атмосферни съставки като дим, прах, изгорели газове, химически отпадни води, както и коефициентите на пречупване на всички видове аерозоли в оптичния път (Indiso, 1970; Ohkawara, 2012); 7) контраст между фона и наблюдавания обект – монохроматична разлика между наблюдавания обект и фоновия интензитет; 8) тип на наблюдавания обект (човешка фигура, джип, танк, и т.н.) (Borissova & Mustakerov, 2006; Russell & Lombardo, 1998).

Функционалната зависимост на разстоянието на действие на УНВ от вътрешните параметри и външни условия за наблюдение се изразява аналитично чрез предложена формула (Borissova & Mustakerov, 2006; Borissova & Mustakerov, 2009):

$$R^2 = \left(\frac{0.07 D_{in} f_{ob} \tau_{ob} \tau_a S \delta E K A_{target}}{M \Phi_{min,ph}} \right) \quad (3.79)$$

където: R – разстояние на действие, D_{in} – диаметър на входната зеница на обектива в m , f_{ob} – фокусно разстояние на обектива в mm , τ_a – пропускане на атмосферата, τ_{ob} – пропускане на обектива, S – интегрална чувствителност на фотокатода на ЕОП в A/lm , δ – разделителна способност на ЕОП в lp/mm , E – естествена нощна осветеност в lx , K – контраст, A_t^* – редуцирана площ на наблюдавания обект m^2 , Φ_{min} – прагова чувствителност на ЕОП в lm , M – отношение сигнал-шум.

Пропускането на атмосферата варира в тесен интервал от (0.712 - 0.804) (Indiso, 1970). От 1933 г. до края на 1940 г., коефициентът на пропускане остава стабилен на нива около 0.74 до 0.75, като в средата на 1980 г. достига 0.69 и след това се увеличава до 0.71 в началото на 2000 г. (Ohkawara, 2012). Поради това, пропускането на атмосферата може да се счита за постоянна величина.

Уравнение (3.79) може да се използва за намиране на различни комбинации от външни условия за наблюдение, съответстващи на дадено разстояние на действие на уреда. Минималната стойност на естествената нощна осветеност и максималната стойност на контраста между обекта и фона и обратното представляват две гранични условия за разглежданото разстояние на действие на уреда. Те не могат да бъдат определени от (3.79), тъй като се получава едно уравнение с две неизвестни. Теоретичните минимални или максимални стойности на осветеност и контраст в (3.79), в общия случай може да не бъдат достигнати за конкретно разглеждан уред. Тези две гранични стойности за осветеността и контраста при конкретен обект за наблюдение и разстояние на действие могат да бъдат определени с помощта на формулиране на подходящ многокритериален проблем.

Едно от граничните условия съответства на максимална естествена нощна осветеност и минимален контраст между наблюдавания обект и фона, което може да бъде изразено като Задача 1 (Borissova и др., 2014):

$$\begin{cases} \max E = \left(\frac{R^2 M \Phi_{min,ph}}{0.07 D_{in} f_{ob} \tau_{ob} S_{\Sigma} \delta \tau_a K A_{target}} \right) \\ \min K = \left(\frac{R^2 M \Phi_{min,ph}}{0.07 D_{in} f_{ob} \tau_{ob} S_{\Sigma} \delta \tau_a E A_{target}} \right) \end{cases} \quad (3.80)$$

при ограничения:

$$E^l \leq E \leq E^u \quad (3.81)$$

$$K^l \leq K \leq K^u \quad (3.82)$$

$$A^l \leq A_{target} \leq A^u \quad (3.83)$$

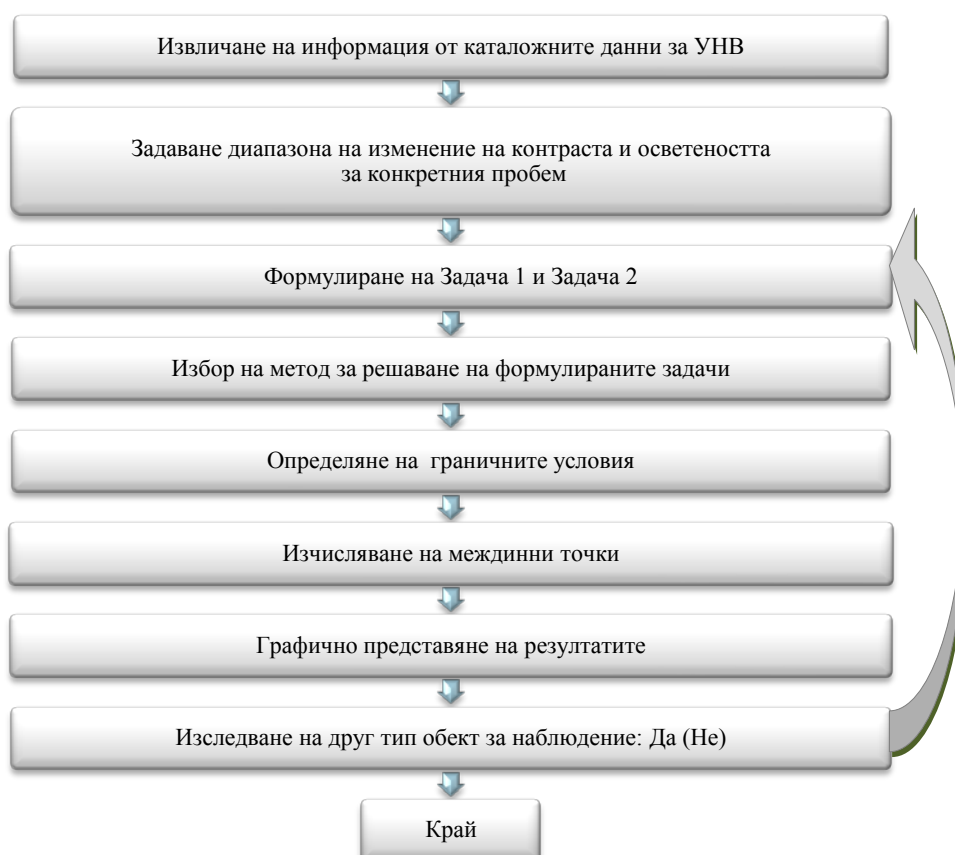
където с E^u , K^u , A^u и E^l , K^l , A^l са означени горните и долните граници на естествената нощна осветеност, контрастът и редуцираната площ на наблюдавания обект; R е зададеното разстояние на откриване в метри; M , $\Phi_{min.ph}$, D_{in} , f_{ob} , τ_{ob} , S , и δ са константи, зависещи от конкретния УНВ.

Другото гранично условие съответства на минимална естествена нощна осветеност и максимален контраст между наблюдавания обект и фона, изразено като Задача 2 (Borissova и др., 2014):

$$\begin{cases} \max K = \left(\frac{R^2 M \Phi_{min.ph}}{0.07 D_{in} f_{ob} \tau_{ob} S_{\Sigma} \delta \tau_a E A_{target}} \right) \\ \min E = \left(\frac{R^2 M \Phi_{min.ph}}{0.07 D_{in} f_{ob} \tau_{ob} S_{\Sigma} \delta \tau_a K A_{target}} \right) \end{cases} \quad (3.84)$$

при ограничения (3.81) – (3.83).

Така формулираните оптимизационни задачи са използвани в разработена методология за определяне на диапазона на изменение на външните условия на наблюдение, съответстващи на каталожните данни на УНВ. Етапите на предложената методология са илюстрирани на Фиг. 3.3 (Borissova и др., 2014).



Фиг. 3.3 Методология за определяне на условията на наблюдение, съвместими с каталожните данни на УНВ

Първият етап от методологията се състои в определяне на вътрешни параметри на УНВ от наличната спецификация. На следващия етап се задават стойностите на горните и долните граници на нощната осветеност и контраста. Определя се типа на обекта за наблюдение, например – стоящ човек, джип, танк и т.н. След това се формулира Задача 1 (за максимум нощната осветеност и минимум на контраста между обекта и фона) и Задача 2 (за минимална нощната осветеност и максимален контраст) и се избира метод за решаване. Изчислява се една или повече междинни точки, като се задава стойност на единия параметър в определените допустими граници и с помощта на (3.79) се изчислява стойността на другия параметър. Получените междинни точки се използват за начертване на съответна графика, която визуализира изменението на параметрите при зададеното разстояние на действие. При необходимост от определяне на диапазон на изменение на външните условия за различен наблюдаван обект, процедурата се повтаря.

Така описаната методологията позволява да бъдат определени различни комбинации на външните условия за наблюдение, съответстващи на параметрите на конкретния уред.

За тестване на предложената методологията са използвани два типа УНВ – очила за нощно виждане и прицел с каталожни данни, показани в Таблица 3.1.

Таблица 3.1. Каталожни данни на УНВ

Разделителна способност lp/mm	Отношение сигнал-шум, $dimensionless$	Фотокатодна чувствителност, A/lm	Диам. на входната зеница, m	Фокусно разстояние, mm	Разстояние на откриване, m
Очила за нощно виждане – MVP-MV14BGP*					
64	21	0.001350	0.018	25	300
Нощен прицел – MV-740**					
64	24	0.001800	0.018	100	425

* http://www.morovision.com/night_vision_goggles/MVP-MV-14BGP.htm

** http://morovision.com/weapons_sights/MVPA-MV-740-3P.htm

И за двете устройства е прието, че пропускането на обектива е равно на 0.80, минималната фотокатодна чувствителност е 3.4×10^{-12} A/lm, а атмосферното пропускане е 0.71. Границите на изменение на външните условия наблюдение са както следва: нощна осветеност E , променяща се в интервала $0.00013 \leq E \leq 0.013$ lx (от звездно небе до пълнолуние); контраст между наблюдавания обект и фона K , променящ се в интервала $0.1 \leq K \leq 0.5$ и редуцирана площ на наблюдавания обект, съгласно критериите на Джонсън (Lombardo, 1998) за стоящ мъж ($A_{\text{мъж}} = 0.72 \text{ m}^2$), за джип ($A_{\text{джип}} = 2.47 \text{ m}^2$), за камион ($A_{\text{камион}} = 5.9 \text{ m}^2$) и за танк ($A_{\text{танк}} = 10 \text{ m}^2$).

Използван е методът на претеглената сума за решаване на формулираните многокритериални задачи 1 и 2. В резултат на използвания метод, трансформираната еднокритериална оптимизационна задача 1 има вида:

$$\max(w_1 E' + w_2 K') \quad (3.85)$$

при ограничения (3.81) – (3.83) и $\sum_i w_i = 1$, а E' и K' са нормализираните критерии (Marler & Arora, 2010).

Трансформираната еднокритериална оптимизационна задача задача 2 има вида:

$$\max(w_1 E' + w_2 K') \quad (3.86)$$

при ограничения (3.81) – (3.83) и $\sum_i w_i = 1$, а K' и E' са нормализираните критерии.

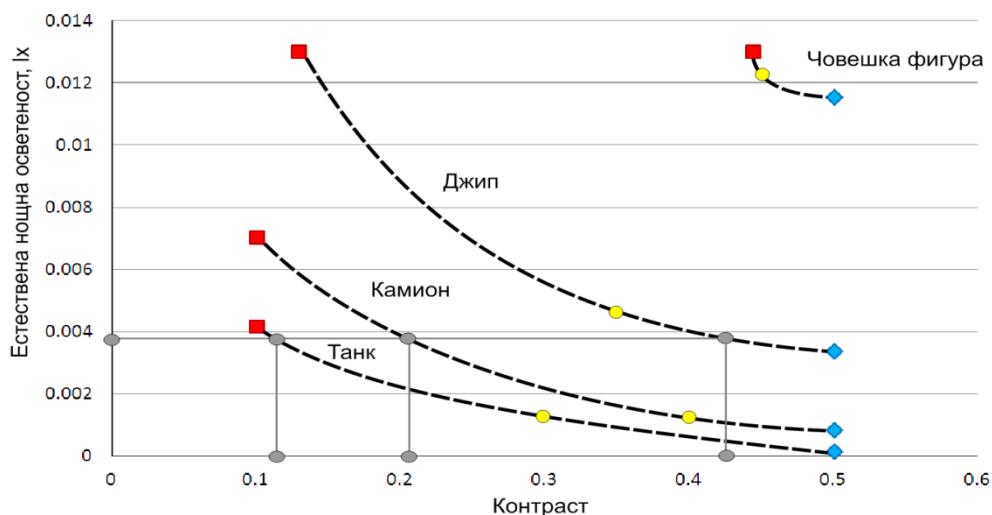
Резултатите от решаването на Задача 1 и Задача 2, определящи гранични условия за нощна осветеност и контраст за четири различни типа наблюдавани обекта, са показани в Таблица 3.2.

Таблица 3.2. Резултати за гранични точки на осветеността и контраста

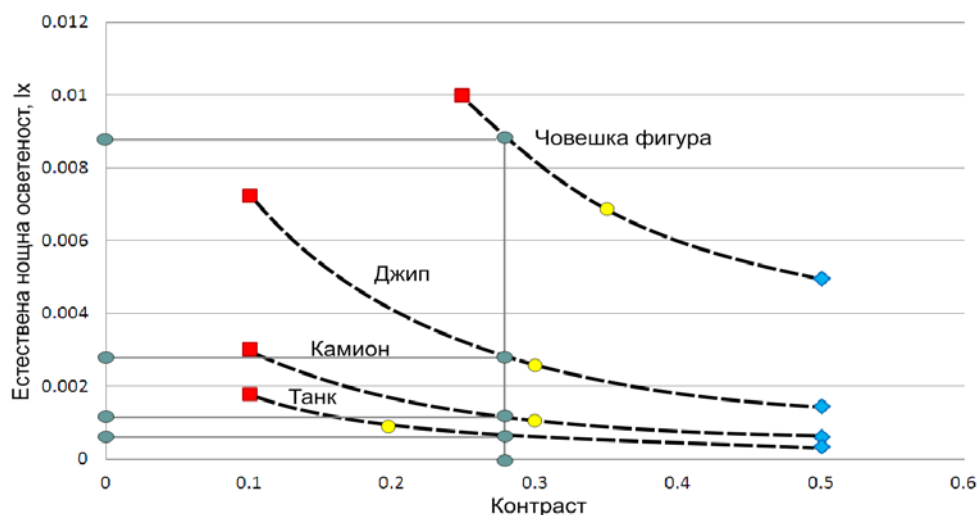
УНВ	Тип на наблюдавания обект	Осветеност, lux	Контраст, dimensionless	Разстояние на откриване, m
Очила за нощно виждане MVP-MV14BGP	Задача 1: (max E, min K, w ₁ =0.50, w ₂ =0.50)			
	(A _{man} = 0.72)	0.0130000	0.44	300
	(A _{jeep} = 2.47)	0.0130000	0.13	300
	(A _{truck} = 5.9)	0.0007045	0.10	300
	(A _{tank} = 10)	0.0041568	0.10	300
	Задача 2: (min E, max K, w ₁ =0.50, w ₂ =0.50)			
	(A _{man} = 0.72)	0.011547	0.50	300
	(A _{jeep} = 2.47)	0.003365	0.50	300
	(A _{truck} = 5.9)	0.001409	0.50	300
	(A _{tank} = 10)	0.000831	0.50	300
Нощен прицел MV-740	Задача 1: (max E, min K, w ₁ =0.50, w ₂ =0.50)			
	(A _{man} = 0.72)	0.001000	0.25	425
	(A _{jeep} = 2.47)	0.007237	0.10	425
	(A _{truck} = 5.9)	0.003030	0.10	425
	(A _{tank} = 10)	0.001787	0.10	425
	Задача 2: (min E, max K, w ₁ =0.50, w ₂ =0.50)			
	(A _{man} = 0.72)	0.004965	0.50	425
	(A _{jeep} = 2.47)	0.001447	0.50	425
	(A _{truck} = 5.9)	0.000606	0.50	425
	(A _{tank} = 10)	0.000357	0.50	425

Диапазоните на изменение на естествената нощна осветеност и контраста за двата типа уреди за нощно виждане и за четирите различни типа наблюдавани обекта са показани графично на Фиг. 3.4 и Фиг. 3.5.

Вижда се (Фиг. 3.4 и Фиг. 3.5), че съществува повече от една комбинация на естествената нощна осветеност и контраста за даден тип наблюдаван обект, съответстващ на едно и също разстояние на откриване за дадено устройство. От тези криви е възможно да се оцени ефективността на конкретното УНВ по отношение на типа на наблюдавания обект и диапазона на изменение на нощната осветеност и контраста.



Фиг. 3.4. Диапазон на изменение на естествената нощна осветеност и контраста в случая на очила за нощно виждане



Фиг. 3.5. Диапазон на изменение на естествената нощна осветеност и контраста в случая на нощен прицел

Например, при нощна осветеност от 0.0038 lx (Фиг. 3.4) разстояние на откриване от 300 m при ОНВ може да бъде постигнато при контраст от 0.12 (за танк), при контраст от 0.21 (камион) и при контраст от 0.42 (джип), но човешка фигура на разстояние от 300 m при зададената осветеност не може да бъде открита. В случая на

прицел с разстояние на откриване от 425 m, при стойност на контраста от 0.28, откриването на танк може да бъде постигнато при осветеност от 0.0005, за камион при осветеност от 0,001, за джип при осветеност от 0.0024 и за човешка фигура в цял ръст при осветеност от 0.0088.

3.5.2. Минимално допустими външни условия, при които се постига една и съща стойност за параметъра разстояние на действие

Техническите спецификации на УНВ почти винаги включват разстояние на действие, определено за фиксиран обект на наблюдение (напр. човешка фигура в цял ръст), но в повечето случаи не специфицират стойности на осветеността, контраста и пропускането на атмосферата, за които това разстояние е измерено. От потребителска гледна точка тези стойности са необходими, за да се определи дали уредът е подходящ за дадена област на приложение. За да се определят различни комбинации от минимално допустимите стойности на осветеността, контраста и пропускането на атмосферата, удовлетворяващи зададеното разстояние на действие, може да бъде използван следния многокритериален оптимизационен модел (Borissova & Mustakerov, 2009)

$$\begin{cases} \min E \\ \min K \\ \min \tau_a \end{cases} \quad (3.87)$$

при ограничения:

$$\sqrt{\frac{0.07 D_{in} f_{ob} \tau_o S_{\Sigma} \delta E \tau_a K A'_{ob}}{M \Phi_{\min, ph}}} = R^* \quad (3.88)$$

$$E^l \leq E \leq E^u \quad (3.89)$$

$$\tau_a^l \leq \tau_a \leq \tau_a^u \quad (3.90)$$

$$K^l \leq K \leq K^u \quad (3.91)$$

където R^* е зададеното в каталога разстояние на действие в метри, E^u , τ_a^u , K^u са горните и долните E^l , τ_a^l , K^l граници за осветеността, пропускането на атмосферата и контраста между обекта и фона.

За числена проверка на предложения модел са използвани очила за нощно виждане със следните каталожни данни: ЕОП от тип Gen. 3 US с разделителна способност $\delta = 68$ lp/mm, фотокатодна чувствителност $S = 0.0019$ A/lm, отношение сигнал-шум $M = 25$ и прагова фотокатодна чувствителност $\Phi_{\min} = 4.10^{-13}$, обектив с

диаметър на входната зеница $D_{in} = 0.018$ m, фокусно разстояние $f_{ob} = 26$ mm и пропускане $\tau_o = 0.8$, разстояние на откриване $R^* = 325$ m. Допустимите граници на външните условия на наблюдение са: естествена нощна осветеност $0.0001 \leq E \leq 0.01$, пропускане на атмосферата $0.65 \leq \tau_a \leq 0.80$, контраст между наблюдавания обект и фона $0.1 \leq K \leq 0.5$, и редуцирана площ на човешка фигура в цял ръст за съответното разстояние $A_{man} = 0.72$ m². Използвайки метода на претеглената сума, съответната трансформирана оптимизационна задача са свежда до:

$$\min(w_1 * E' + w_2 * K' + w_3 * \tau'_a) \quad (3.92)$$

при ограничения:

$$0.0001 \leq E \leq 0.01 \quad (3.93)$$

$$0.1 \leq K \leq 0.5 \quad (3.94)$$

$$0.65 \leq \tau_a \leq 0.80 \quad (3.95)$$

$$\sum_i w_i = 1 \quad (2.96)$$

където E' , K' и τ'_a са нормализираните критерии. Относителната важност за всеки критерии се взема предвид чрез тегловите коефициенти.

Разгледани са три различни случая, при които пропускането на атмосферата е използвано като константа с известна стойност, а чрез различните тегловни коефициенти са отразени потребителите изисквания по отношение важността на параметрите естествена външна осветеност и контраст.

В Таблица 3.3 са показани резултатите за получените стойности на естествената нощна осветеност (E), контраста (K) и пропускането на атмосферата (τ_a) при използване на различни комбинации на тегловите им коефициенти.

Таблица 3.3. Теглови коефициенти и резултати за външните условия

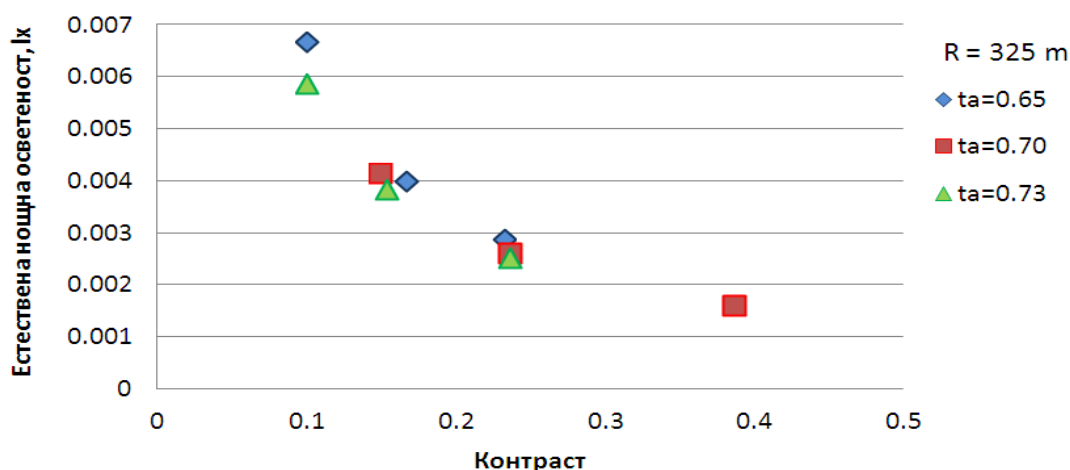
No	w_1	w_2	w_3	E, lux	K	τ_a	R, m
1	0.34	0.33	0.33	0.00400	0.166	0.65	325
2	0.60	0.30	0.1	0.00287	0.232	0.65	
3	0.10	0.60	0.30	0.00666	0.100	0.65	
4	0.30	0.10	0.60	0.00261	0.237	0.70	
5	0.10	0.10	0.80	0.00413	0.150	0.70	
6	0.10	0.80	0.10	0.00413	0.150	0.70	
7	0.80	0.10	0.10	0.00159	0.387	0.70	

В Таблица 3.4 са показани резултатите за получените стойности на естествената нощна осветеност (E) и контраста (K) при използване на различни комбинации на тегловите им коефициенти и при една и съща стойност на пропускането на атмосферата ($\tau_a = 0.73$).

Таблица 3.4. Използвани теглови коефициенти и получени резултати

No	w_1	w_2	E, lux	K	τ_a	R, m
1	0.5	0.5	0.00383	0.153	0.73	325
2	0.7	0.3	0.00251	0.236		
3	0.3	0.7	0.00585	0.100		

На Фиг. 3.6 са показани получените комбинации на различни външни условия на наблюдение, осигуряващи разстояние на откриване 325 m.



Фиг. 3.6. Комбинации на нощната осветеност и контраста при различно пропускане на атмосферата, осигуряващи разстояние на откриване 325 m

Възможно е да се използват фиксирани стойности не само за пропускането на атмосферата, но и за другите параметри на външните условия на наблюдение, за да се изследват възможните комбинации, отговарящи на заденото разстояние на действие на УНВ.

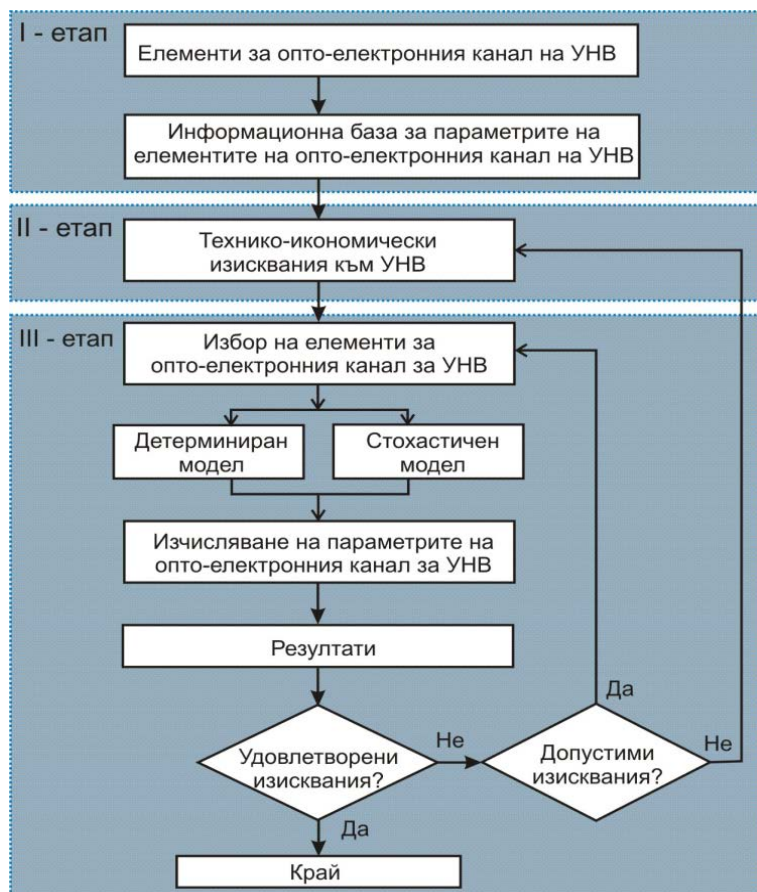
3.6. Методи за проектиране на модулни инженерни системи

Формулираните детерминиран и стохастичен модели в т. 3.1 са използвани за разработване на методи за проектиране на УНВ чрез избор от множества елементи. Формулирани са три метода – метод на итеративно проектиране, метод на рационално проектиране и метод на оптимално проектиране.

3.6.1. Метод на итеративно проектиране

Методът на итеративното проектиране на модулни системи използва итеративни процедури за симулиране на различни конфигурации за проектираната система, отчитайки съществуващите взаимовръзки между модулите и експлоатационните

условия. На Фиг. 3.7 са показани етапите на итеративния метод за проектиране на УНВ. Първият основен етап на метода е определяне на необходимите конструктивни елементи за УНВ и изграждане на информационна база, включваща съответните параметри.



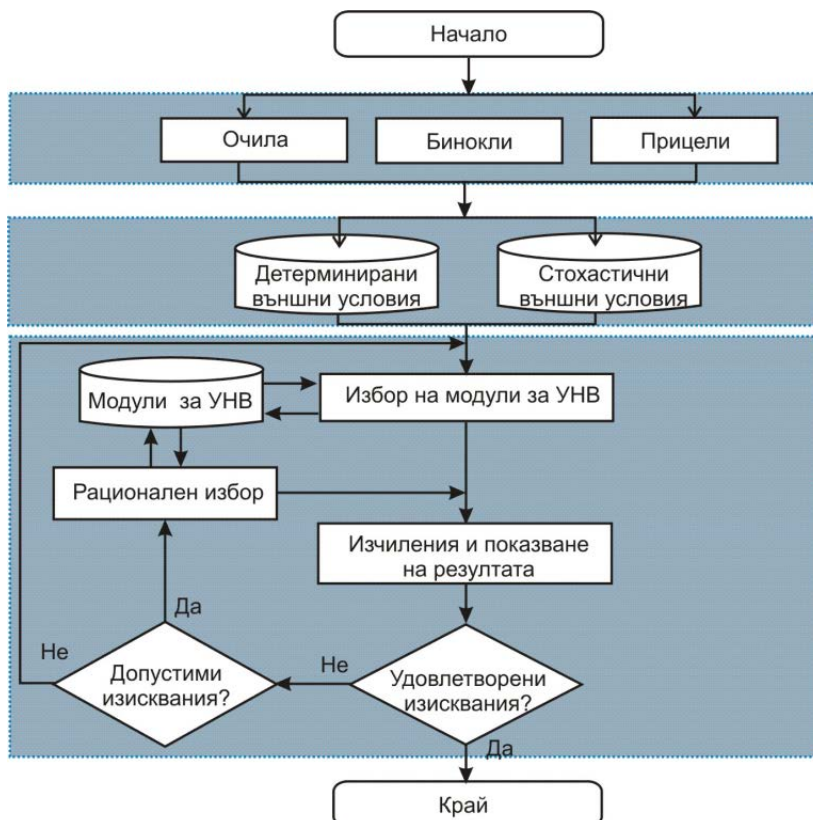
Фиг. 3.7. Етапи на итеративния метод за проектиране

Вторият основен етап се състои в определяне на технико-икономическите изисквания към разработваните уреди. На този етап се установяват потребителските изисквания към параметрите на УНВ – разстояние на действие, зрително поле, тегло, цена и т.н. На третия етап се реализира избора на конструктивните елементи за опто-електронния канал – обектив, ЕОП, окуляр, ел. батерии. Този избор се осъществява с помощта на лицето, вземащо решение (ЛВР). При избран конкретен тип на УНВ и стойности на параметрите на външните условия на наблюдение, параметрите на проектираното устройство се изчисляват посредством детерминирания модел, а при използване на математическите очаквания на външните условия на наблюдение се реализира стохастичния модел. След оценка на определените параметри на УНВ, ако те не съответстват на поставените изисквания, се извършва промяна в направения избор на някои от елементите и отново се сравняват получените резултати. По такъв начин,

чрез последователно изпълнение на процеса на избор и изчисляване параметрите на опто-електронния канал на УНВ, се търси удовлетворяване на изискванията към параметрите на проектираното устройство.

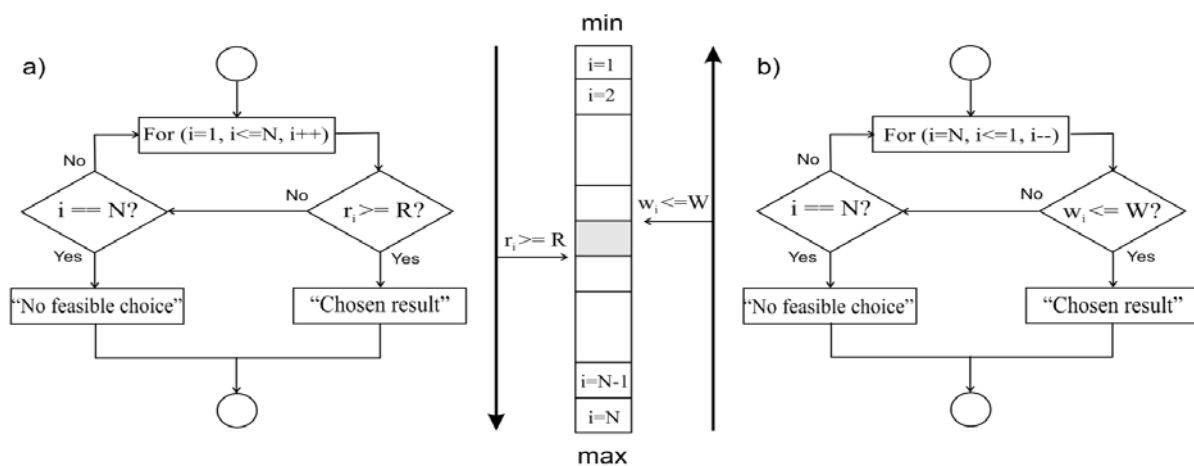
3.6.2. Метод на рационално проектиране

Концепцията на метода на рационален избор съвпада в голяма степен с използваната при многокритериалната оптимизация, рационална (удовлетворителна) оценка. Рационалното вземане на решения означава, че ЛВР не оптимизира някаква ценностна функция, а се старее да достигне определени удовлетворителни нива на критериите. В най-общ случай, получените решения не са оптимални, и се разглеждат като рационални или удовлетворителни решения. Итеративният процес на избор на УНВ модули и изчисляване параметрите на проектираното устройство се свежда до удовлетворяване на зададени гранични стойности за параметрите. ЛВР задава тези гранични стойности като горни или долни граници за параметрите на проектираното устройство. Схематичното представяне на процеса на проектиране на УНВ чрез използване на метода на рационалния избор е показано на Фиг. 3.8 (Borissova & Mustakerov, 2007).



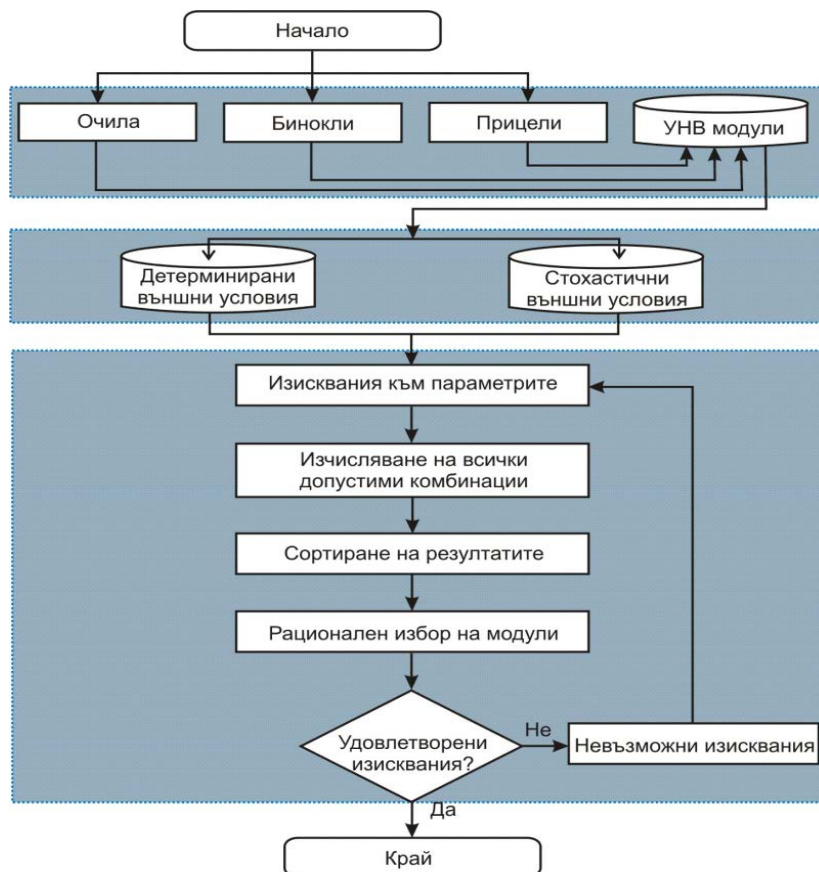
Фиг. 3.8. Етапи на метода на рационалното проектиране

За определяне удовлетворяването на тези граници се използват сортирани масиви и търсене в тях, както е илюстрирано на Фиг. 3.9 (Mustakerov & Borissova, 2007). При задаване на ограничения за теглото и цената, процесът на търсене започва от големите към малките стойности в съответните сортирани масиви, докато бъде достигната стойност по-малка или равна на зададена.



Фиг. 3.9. Реализация на търсенето в сортираните масиви на стойности за:
а) разстоянието на действие и увеличението; б) теглото и цената

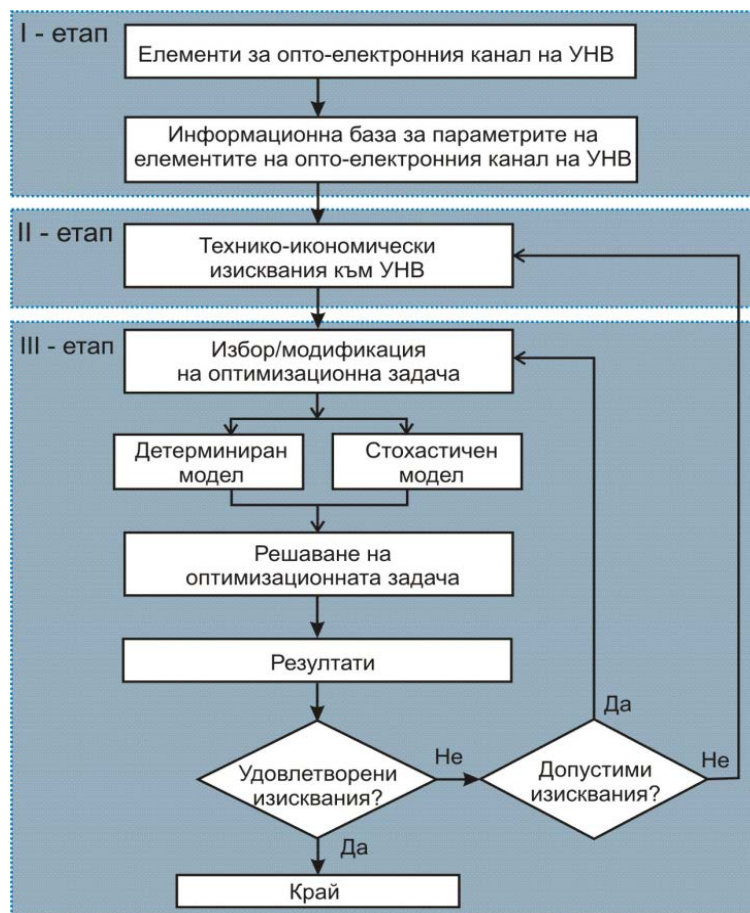
Една обобщена схема за реализация на метода на рационалното проектиране е показана на Фиг. 3.10 (Mustakerov & Borissova, 2007).



Фиг. 3.10. Алгоритмичната реализация на метода на рационалното проектиране

3.6.3. Метод на оптимално проектиране

Методът на оптималното проектиране се отличава от описаните по-горе методи с формулирането и решаването на съответни оптимизационни задачи. На Фиг. 3.11 са показани етапите на метода за оптимален избор на елементите на опто-електронния канал на УНВ.

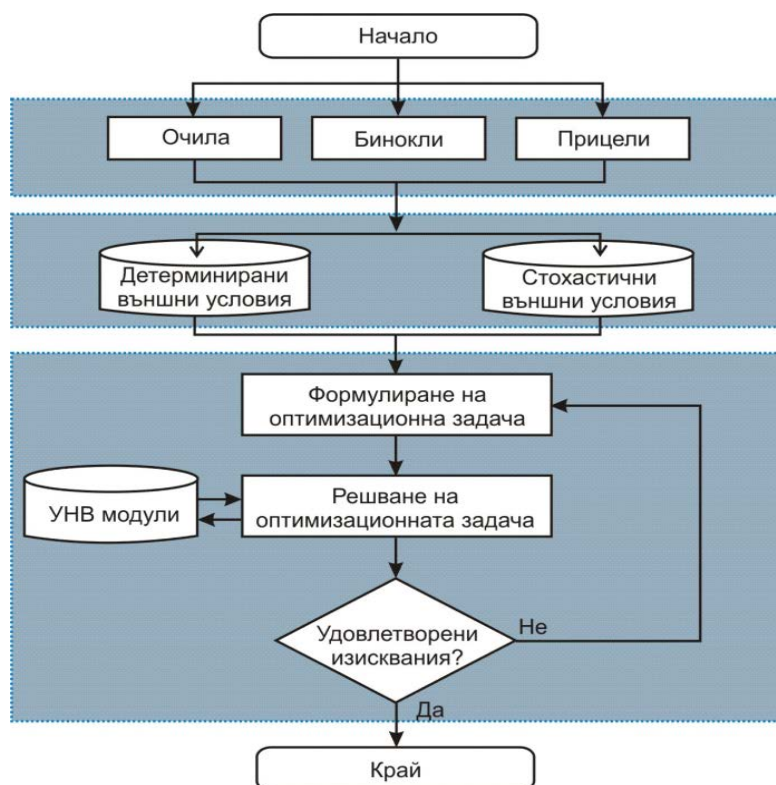


Фиг. 3.11. Етапи на метода на оптималното проектиране

Първите два етапа съвпадат с етапите в итеративния и рационалния метод за избор на елементи за опто-електронния канал на УНВ. На третия етап потребителят (ЛВР) избира (или модифицира) една от формулираните в предишните точки оптимизационни задачи, използвайки детерминирания или стохастичния модел на външните условия на наблюдение. Чрез съответен програмен модул за оптимизация се решава формулираната задачата. Полученото решение се анализира от ЛВР и ако резултатите не удовлетворяват изискванията, може да се избере друга оптимизационна задача или същата да се модифицира и да се реши отново. Този процес се повтаря до получаване на резултати, удовлетворяващи поставените изисквания. И тук ЛВР има възможност за предварително определяне на желаните граници на някои параметри,

участващи в съответния модел. Изпълнението на отделните етапи може да се автоматизира чрез разработване на специализиран софтуер, улесняващ интерактивното взаимодействие на потребителя при реализацията на методите.

Една обобщена алгоритмична реализация на метода на оптималното проектиране е показана на Фиг. 3.12 (Borisova & Mustakerov, 2007).



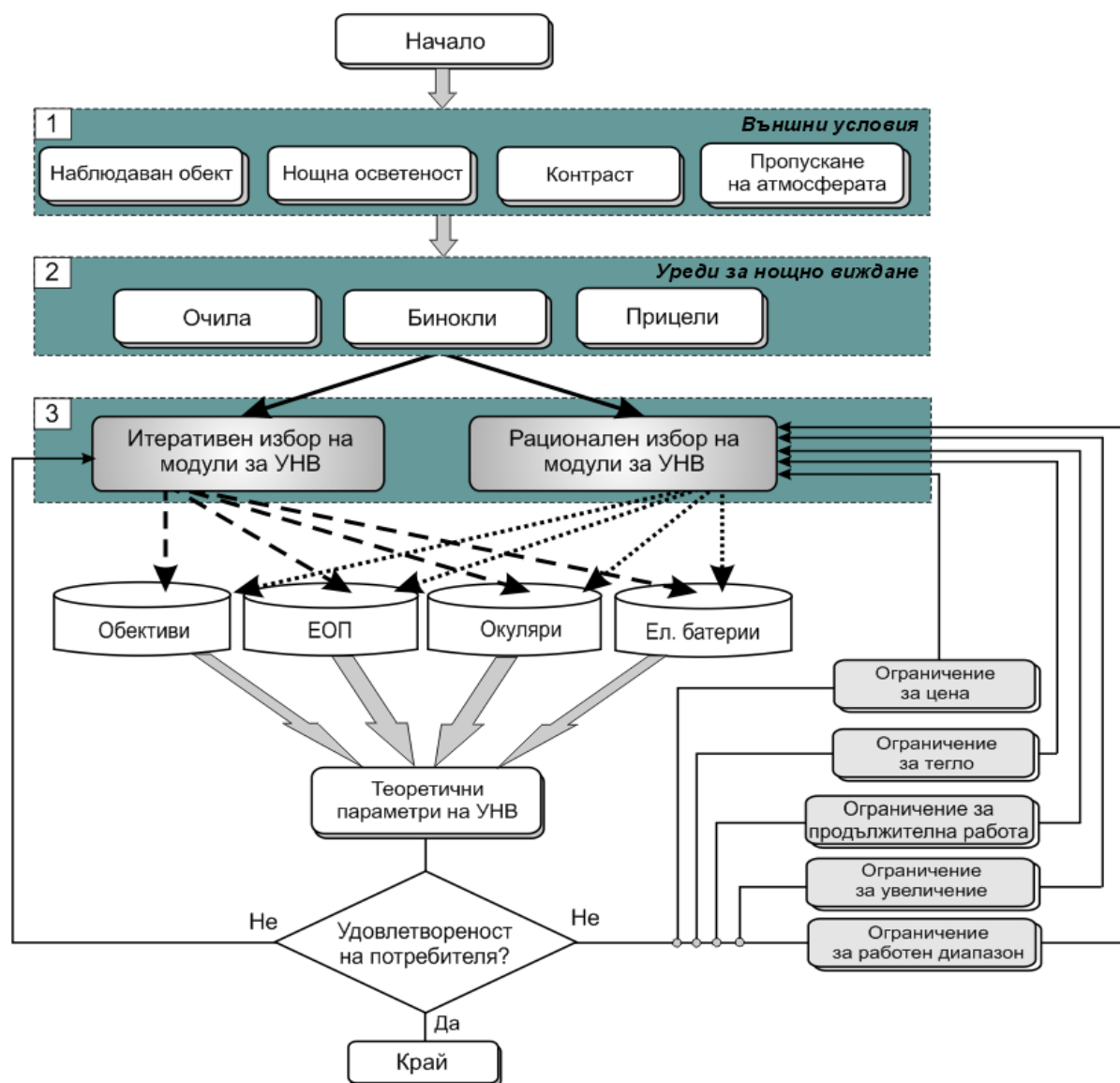
Фиг. 3.12. Алгоритмична реализация на метода на оптималното проектиране

Описаният метод, реализиращ оптимален избор на елементи за проектиране на опто-електронния канал на УНВ, също използва итеративни процедури за определяне на параметрите на УНВ, под контрола на ЛВР. Съществена характеристика на този метод е, че получените решения са оптимални (или Парето-оптимални за случая на многокритериални задачи), съгласно зададените от ЛВР изисквания и ограничения в оптимизационната задача.

3.6.4. Обобщен алгоритъм за оценка на параметрите на проектираните уреди за нощно виждане

Алгоритмите са в основата на всяко компютърно приложение. Поради това е важно да се предложи подходящ алгоритъм, който най-добре да отразява процеса на проектиране на УНВ. В тази връзка е предложен обобщен алгоритъм за проектиране на различни

УНВ, вземащ предвид както зависимостите между отделните модули и разновидностите на УНВ и външните условия на наблюдение, така и изискванията на ЛВР. Предложеният алгоритъм прави възможна интеграцията между определени групи от модули, обработва данните чрез описаните формули от детермирования и стохастичен модели и предоставя като изход получената информация за параметрите на проектираното устройство. Алгоритъмът включва два компонента – компонент за итеративен дизайн и компонент за рационален дизайн. Обобщената диаграма на предложения алгоритъм е показана на Фиг. 3.13 (Borissova и др., 2013).



Фиг. 3.13. Алгоритъм за определяне на предварителна теоретична оценка на параметрите на проектираните УНВ

Алгоритъмът започва с определяне на външни условия за наблюдение. Поради спецификата на УНВ е важно да се вземат предвид различните условия на наблюдение

при процеса на проектиране. Най-важният параметър УНВ, разстоянието на действие, се влияе както от параметрите на отделните модули на устройството, така и от външни условия за наблюдение. На втория етап се избира предназначението на уреда за нощно виждане (очила, прицел, бинокъл). Третият етап на алгоритъма се разделя на два клона за реализация на итеративния или рационалния избор на модули. Итеративният клон на алгоритъма позволява на потребителя да направи собствен избор от съвместими модули за избрания тип устройство и да определи параметрите на проектирания уред като цяло, вземайки предвид избраните външни условия. Ако в потребителя не е доволен от изчислените параметри на проектираното устройство, той може да избере други модули. Този интерактивен процес на проектиране приключва когато потребителят е удовлетворен от получените параметри на симулираното устройство.

Вторият клон на алгоритъма позволява на потребителя да зададе някои предварителни изисквания относно параметрите на проектираното устройство. Основната идея на този рационален избор на модули е да се намери такава допустима комбинация от модули, която да отговаря на зададените изисквания на потребителя.

Алгоритмичната реализация на рационалния избор се основава на определяне на параметрите на всички възможни комбинации от модули, респективно устройства и да покаже точно тази комбинация, която най-добре отразява зададените предпочитания на потребителя. Ако такава комбинация не съществува, на екрана се показва подходящо съобщение, така че потребителят да получи допълнителна информация за параметъра, на който трябва да се отслаби ограничението. Получените масиви се сортират във възходящ ред. Ако е зададено ограничение за стойността на разстоянието на действие (откриване или разпознаване или идентификация) или ограничение за стойността на увеличението, то процесът на търсене започва от малките към големите стойности в съответните сортирани масиви докато бъде достигната стойност по-голяма или равна на зададената. При задаване на ограничения за теглото и цената, процесът на търсене започва от големите към малките стойности в съответните сортирани масиви докато бъде достигната стойност по-малка или равна на зададена. Ако са въведените недопустими ограничения за проектираното устройство, които не могат да бъдат удовлетворени, то съответно съобщение се показва на екрана, за да подпомогне потребителя да направи съответните корекции.

Глава 4:

Оптимално проектиране на инженерни системи, вземащи предвид съществуващите взаимовръзки, външните експлоатационни условия и икономическата ефективност

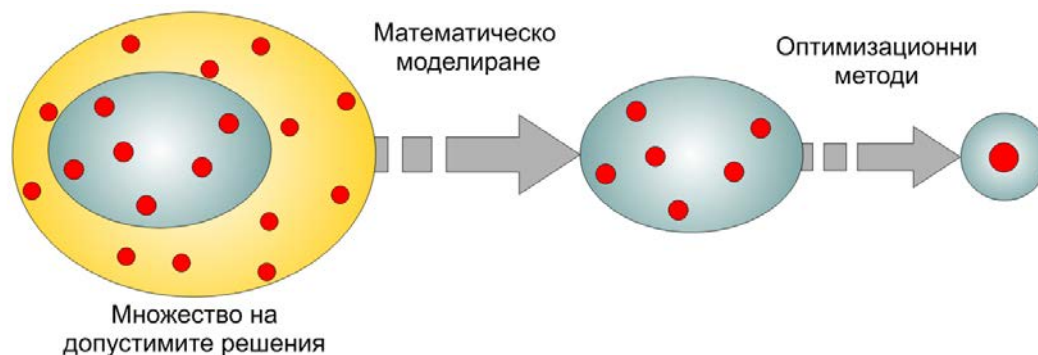
В тази глава е описан обобщен метод за проектиране инженерни системи, вземащи предвид съществуващите взаимовръзки, външните експлоатационни условия и икономическата ефективност. Този клас системи се характеризират не само с избор на съвместими модули и съобразяване с външните експлоатационни условия, но и с удовлетворяване на изискване за икономическа ефективност. Един достатъчно представителен пример за проектиране на този клас системи са ветроенергийните паркове (ВЕП). На база на предложения модел са формулирани съответни едно- и многокритериални оптимизационни задачи за оптимално проектиране. Описан е алгоритъм за приложение на многокритериален подход при проектирането на системи от този тип. Получените резултати са публикувани в статии D3, D19, D20 и D21.

Изборът на ВЕП като илюстративен пример за този клас системи е обоснован от факта, че вятърната енергия, като енергиен източник, бързо заема челно място в групата на възобновяемите енергийни източници (вятърът е безплатен, чист и неизчерпаем източник на енергия). От друга страна, развитието на възобновяеми енергийни източници е в духа европейските програми за „*Интелигентна Енергия за Европа*” и включената в нея кампания „*Устойчива енергия за Европа*”. С Директива 2009/28/ЕО от 23 април 2009 г., по отношение на възобновяемите източници на енергия, се въвежда цел за дял от 20 %, която да бъде постигната до 2020 г. Новата енергийна политика е в подкрепа на дългосрочната цел за намаляване на емисиите на парникови газове на ЕС с 80-95 % до 2050 г. За постигането на тази цел са определяни задължителни национални цели и задължения за всички страни-членки. Удовлетворяването на тези задължения включва изграждане на ветроенергийни паркове (ВЕП), като елемент от групата на възобновяемите източници на енергия за

промишлено производство на електроенергия. ВЕП се характеризират със зависимости между модулите (тип, брой и разположение на турбините), наличие на експлоатационни условия (площ на ВЕП и посока на вятъра) и осигуряване на определено ниво на икономическа ефективност.

4.1. Специфика на проектирането на инженерни системи, отчитащи вътрешни и външни зависимости, и икономическата ефективност – ветроенергийни паркове

Прилагането на математическото моделиране и оптимизационните подходи към сложни комплексни системи води до подобряване на тяхната производителност. Такива приложения се нуждаят от разработването на подходящи математически модели, на база на които да се формулират съответни оптимизационни задачи. Проектирането на ВЕП изисква вземане на ключови решения за избор на типа и броя на ВТ, за избор на разположението на ветровите турбини (ВТ) във ВЕП и на избор на терен за ВЕП (Mustakeroov & Borissova 2010). Всяко от тези решения трябва да е съобразено с различни изисквания и ограничения. Формулирането на оптимизационен математически модел цели редуциране на множеството от допустимите решения и чрез подходящи методи за решаване за определяне на оптималното решение (Фиг. 4.1).



Фиг. 4.1. Етапи за определяне на оптималното решение

Проблемите, свързани с проектирането на ВЕП могат да се разглеждат като част от приложната математика, при които се търси подобряване на ефективността на ВЕП чрез подобряване разположението на ВТ. Разработването на модели и методи за оптимално проектиране на ВЕП изисква използване на достоверни данни за ветровия потенциал, заедно с други реални данни за съществуващи ограничения, както физически и технически, така и екологични.

Финансирането на конкретен проект за ВЕП изисква детайлна преценка на различни аспекти и на свързаните с това правни и търговски споразумения. Преди отпускането на каквито и да било инвестиции, всеки финансов кредитор би искал да е наясно е рисковете при погасяване на кредита. Всички тези съображения при проектирането и реализирането на ВЕП са представени като 4 различни аспекта, включващи определяне на понтенциала на ветровия ресурс, математическо моделиране и оптимизация на ВЕП, анализ на риска и реализиране на подходящи програмни средства, както е показано на Фиг. 4.2 (Borissova & Mustakerov, 2010).



Фиг. 4.2. Основни аспекти при проектирането на ветрови парк

Географското разположение на ВЕП трябва да се избере внимателно, тъй като енергията, която ще се произвежда и най-вече рентабилността на инвестицията, са в зависимост от скоростта и постоянството на вятъра:

- получената енергия е пропорционална на скоростта на вятъра на 3^{-та} степен, т.е. малка разлика в скоростта може да доведе до голяма разлика в получената енергия.
- постоянството на вятъра е важно за непрекъснатото генериране на енергия; спирането ВТ може да създаде проблеми в собствените нужди на ВЕП. От своя страна, твърде силните пориви на вятъра могат да подложат на риск инсталацията на ветровите турбини.

Необходимостта от оптимално проектиране на ВЕП е продиктувано от основната цел за максимално производството на електрическа енергия и минимални капиталови и оперативни разходи, при наличието на ограничения. При проектирането на ВЕП е необходимо да се определи броят и типът на ВТ, като се вземат предвид

размерите на ветровото поле, ориентацията на вятъра, височината на турбините, роторните им диаметри, и разстоянието между турбините, за да се избегнат загубите от затихването на вятъра след турбините (т. нар. *ветрова следа*).

Математическият модел трябва да вземе предвид планирания капацитет за ВЕП и евентуално бъдещо разширяване. Това е от особена важност при изграждането на индустриални ВЕП, където трябва да се направят значителни инвестиции за модернизация на преносната система, за да се осигури доставяне на получената енергия от ВЕП. При определяне на капацитета на ВЕП трябва да се вземе предвид и видът на турбините, които могат да бъдат транспортирани и инсталирани във ВЕП. Капацитетът на ВЕП се влияе от географското местоположение на конкретното ветрово поле, като районите с повече изолирани хълмове предлагат по-малко подходящи места за разположение на турбини.

Типът на ВТ е един от ключовите фактори, който да се вземе предвид по време на процеса на проектиране на ВЕП. Турбините са разработени така, че да са съобразени с различни специфични параметри на вятъра. Класовете на ВТ се определят от стандарта на Международната електротехническа комисия, въз основа на три параметъра: средната скорост на вятъра (на нивото на височината на турбината), екстремни 50-годишни пориви и наличието на турбуленция във ветровото поле, както е показано в Таблица 4.1

Таблица 4.1 Класове ВТ

Класове ВТ	I	II	III	IV
Средногодишна скорост на вятъра	10 m/s	8.5 m/s	7.5 m/s	6.0 m/s
Екстремни моментни пориви на вятъра	70 m/s	59.5 m/s	52.5 m/s	42.0 m/s
Турбулентност Class A	18%			
Турбулентност Class B	16%			

Източник: <http://windwire.blogspot.com/2009/05/iec-classification-of-turbines.html>

Заб: Турбулентността е стандартното отклонение на скоростта на вятъра, измерена при 15 м / сек скорост на вятъра.

Изследванията на различни индустриална ВЕП показват, че 60 – 80% от общите инвестиционни разходи представляват разходи за покупка и инсталиране на ВТ. От друга страна, стойностите на комерсиалните ВТ варират от 1 до 2 милиона долара за MW инсталирана мощност. По-малките турбини (по-малки от 100 KW) струват по-малко, но са по-скъпи по отношение на инсталирана електрическа мощност. Погрешният избор на: типа ВТ, височината на кулата, броя и разположението им, могат да доведат до неефективност на ВЕП и до значителни икономически загуби.

Използването на по-високи кули за ВТ води до увеличаване на произвежданата енергия, но изграждане им струва повече, така че е необходим компромис и оптимизиране по време на процеса на проектиране на конкретния ВЕП.

Изборът на тип на ВТ трябва да е такъв, че най-добре да отразява конкретните условия на ветровото поле. ВТ обикновено са подредени в редици, перпендикулярна на преобладаващите ветрове. Едно от критичните ограничения е свързано с разстоянието между турбините в парка. Ако вятърът достигне следващата турбина преди скоростта на вятъра е била възстановена от сблъсъка с предходната турбина, то производство на енергия от втората турбина ще бъде намалено – *ефект на ветровата следа*.

За да се сведат до минимум инвестиционните разходи и за да се постигне максимално оползотворяване на енергията от вятъра, е предложен обобщен оптимизационен модел за проектиране на ВЕП. Основната идея на предложения модел на проектиране на ВЕП се базира на използването на дискретна комбинаторна оптимизация. Нека $a \in \{a_1, a_2, a_3, \dots, a_n\}$ са управляващи променливи за избор на конкретен тип ВТ от зададено множество. Всяка от тези променливи има свои параметри, които могат да бъдат обобщени в матрица. В тази матрица, различните типове турбини са представени чрез редовете на матрицата, а колоните на матрицата представляват конкретните параметри на турбините. Ако чрез a_i^1 се изрази номиналната мощност на i -тата турбина, а a_i^2 е роторния диаметър на витлото на ВТ, a_i^3 е височината на кулата, a_i^4 е шумовият фактор на турбината и т.н., то съответната матрица, определяща пространството на променливите a_i^m за $i = 1, \dots, n$ може да се представи като:

$$\begin{pmatrix} a_1^1 & a_1^2 & \dots & a_1^m \\ a_2^1 & a_2^2 & \dots & a_2^m \\ \cdot & \cdot & \dots & \cdot \\ \cdot & \cdot & \dots & \cdot \\ a_n^1 & a_n^2 & \dots & a_n^m \end{pmatrix} \quad (4.1)$$

Друга управляваща променлива може да бъде променливата, чрез която да се реализира избор на конкретно ветрово поле за изграждане на ВЕП от налични k такива ветрови полета. Параметрите на ветровото поле също могат да бъдат изразени чрез подходяща матрица. Ако означим с b_j^1 размера на j -тото ветрово поле, с b_j^2 формата на ветровото поле, с b_j^3 означим неравността, и т.н. до b_j^k за $j=1, \dots, s$, то съответната матрица може да се представи като:

$$\begin{pmatrix} b_1^1 & b_1^2 & \dots & b_1^k \\ b_2^1 & b_2^2 & \dots & b_2^k \\ \cdot & \cdot & \dots & \cdot \\ \cdot & \cdot & \dots & \cdot \\ b_s^1 & b_s^2 & \dots & b_s^k \end{pmatrix} \quad (4.2)$$

По този начин, могат да бъдат въведени и други управляващи променливи, отразяващи спецификата на проектиране на ВЕП. Основната идея на предложението обобщен комбинаторен оптимизационен подход използва двоични целочислени променливи за избор на съответните управляващи променливи. За целта са въведени вектори от двоични променливи. Например, стойностите на параметрите, съответстващи на управляващата променлива a се избират от вектора X за който е в сила:

$$X = \begin{pmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ x_m \end{pmatrix}, x_i \in \{0, 1\}, \sum_{i=1}^m x_i = 1 \quad (4.3)$$

Аналогично, стойностите на b се избират чрез вектора Y :

$$Y = \begin{pmatrix} y_1 \\ y_2 \\ \cdot \\ \cdot \\ y_k \end{pmatrix}, y_j \in \{0, 1\}, \sum_{j=1}^k y_j = 1 \quad (4.4)$$

Изборът на конкретна променлива (респективно параметър) се реализира чрез умножение на матрицата от параметрите и двоичните вектори:

$$a = \begin{pmatrix} a_1^1 & a_1^2 & \dots & a_1^m \\ a_2^1 & a_2^2 & \dots & a_2^m \\ \cdot & \cdot & \dots & \cdot \\ \cdot & \cdot & \dots & \cdot \\ a_n^1 & a_n^2 & \dots & a_n^m \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ x_m \end{pmatrix}^T \quad (4.5)$$

$$b = \begin{pmatrix} b_1^1 & b_1^2 & \dots & b_1^k \\ b_2^1 & b_2^2 & \dots & b_2^k \\ \cdot & \cdot & \dots & \cdot \\ \cdot & \cdot & \dots & \cdot \\ b_s^1 & b_s^2 & \dots & b_s^k \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ \cdot \\ \cdot \\ y_k \end{pmatrix}^T \quad (4.6)$$

Реалистичният процес на проектиране изисква да бъдат взети и различни функционални ограничения от типа:

$$g = f(a, b) \quad (4.7)$$

От друга страна, към параметрите също могат да се поставят различни изисквания, напр. чрез съответни долни и/или горни граници:

$$\begin{aligned} L_p &\leq a_i^p \leq U_p, p \in \{1, 2, \dots, m\} \\ L_r &\leq b_j^r \leq U_r, r \in \{1, 2, \dots, k\} \end{aligned} \quad (4.8)$$

Оптималното проектиране на ВЕП трябва да се подчинява на съответни критерии, които се задават като функция от параметрите и променливите:

$$f_i(a_i, b_j) \quad (4.9)$$

Някои примери на известни критерии, които се използват при проектирането на ветрови паркове са:

- минимизиране на разходите за единица енергия (Mosetti и др, 1994)

$$costs = N \left(\frac{2}{3} + \frac{1}{3} \exp(-0.00174N^2) \right) \quad (4.10)$$

където N е броят на инсталираните турбини;

- минимизиране на усреднената цена на енергията (Manwell и др, 2002)

$$LCOE = \frac{C_c + FCR + C_{O\&M}}{AEP} \quad (4.11)$$

където C_c са общите капиталови разходи за инсталацията на ВЕП (турбини, инфраструктура и разходи за пренос на енергията), FCR са разходи, зависещи от данъците, застраховките, лихвите и степента на възвръщаемост, $C_{O\&M}$ са годишните разходи за експлоатация и поддръжка, AEP е годишното производство на енергия;

- максимизиране на инсталираната мощност

$$P = NP_{wt} \quad (4.12)$$

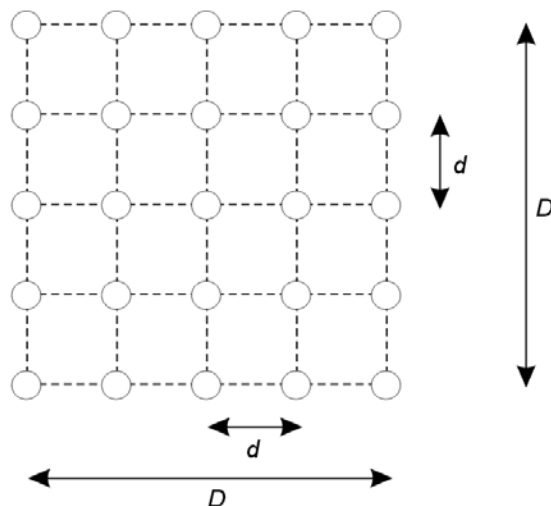
където N – брой на инсталираните турбини, а P_{wt} е номиналната мощност на турбината.

Подобни критерии или комбинация от тях могат да се използват за формулиране на едно- или многокритериални оптимизационни задачи от типа $\max$ (или $\min$) на (4.9), при ограничения (4.1) – (4.8). Решенията на тези задачи ще определят оптимални или Парето-оптимални решения за проектиране на ВЕП.

4.2. Разположение на ветровите турбини във ветроенергийни паркове

Броят на ВТ в един парк обикновено е ограничен в определена площ. Разположението на ВТ в парка не може да бъде произволно. Съществуват различни ограничения, които трябва да бъдат взети предвид. Едно от основните ограничения е свързано с факта, че зад всяка ВТ се формира пространство, където скоростта на вятъра намалява и интензивността на турбулентността на въздуха се увеличава. Следователно, проектирането на разположението на турбините във ВЕП е проблем, чиято сложност

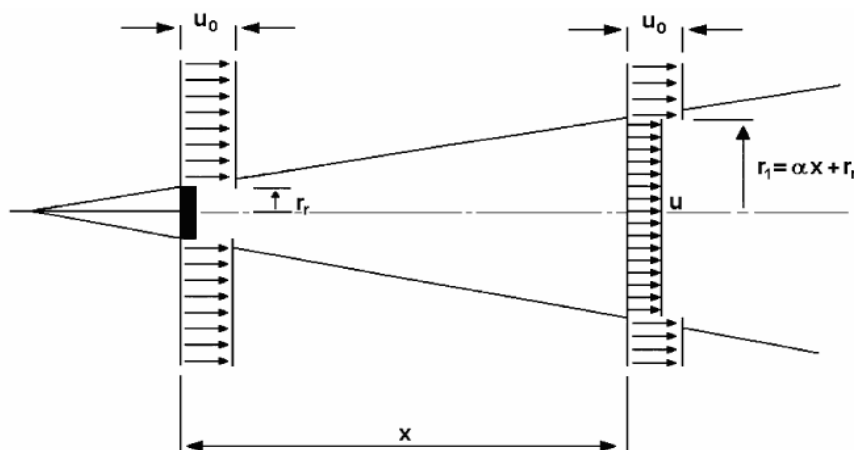
нараства с увеличаване броя на турбините и взаимното им влияние, особено при екстремни скорости на вятъра. На Фиг.4.3 е показано едно от възможните разположения на турбините във ВЕП с n турбини във всеки ред и всяка колона.



Фиг. 4.3. Разположения на турбините във ВЕП

Разстоянието d между турбините е избрано едно и също в двете посоки и се измерва като множител на диаметъра на роторите на турбините. Общата площ на ветровия парк ($D \times D$) – често е ограничена до предварително известна стойност. На практика разположението на ВТ няма да бъде в квадрат, а ще зависи от конкретните ветрови условия. В парка турбулентността се увеличава значително след всяка турбина, а скоростта на вятъра намалява незначително. Може да се приеме, че всички турбини в парка са изложени на турбулентност, зависеща от разстоянието между отделните турбини D , разпределението на посоките на вятъра и мястото на всяка турбина в парка.

Турбулентността или ветровата следа в близост до турбината има радиус, равен на радиуса на витлото r_r , като радиусът на ветровата следа се увеличава линейно и пропорционално на отдалечаването от генератора, Фиг. 4.4 (Grady, и др., 2005)



Фиг. 4.4. Ветрова следа зад ветрова турбина

За определяне на скоростта на вятъра непосредствено зад ротора, се използва следната формула (Marmidis и др., 2008):

$$u = u_o \left[1 - \frac{2a}{1 + \alpha \frac{x}{r_l}} \right] \quad (4.13)$$

където u_o е средната скорост на вятъра, a е аксиалният индукционен фактор, x е разстоянието от ВТ, r_l е радиуса на следата и α е константа на завихрянето.

Радиусът на следата е функция от радиуса на ротора на ВТ (Marmidis и др., 2008):

$$r_l = r_r \sqrt{\frac{1-a}{1-2a}} \quad (4.14)$$

Кинетичната енергия на следите, при наличие на множество турбини в парка, може да се приеме равна на сумата от дефицитите на кинетичната енергия, което води до следния израз за скоростта на вятъра след преминаването на N турбини (Grady, и др., 2005; Marmidis и др., 2008):

$$u_i = u_o \left[1 - \sqrt{\sum_{i=1}^N \left(1 - \frac{u}{u_o} \right)^2} \right]. \quad (4.15)$$

Следователно, оползотворената мощност от ветровите турбини може да се разглежда като функция от локалната скорост на вятъра. Посоката, интензивността и вероятността за поява на вятър, характеризират локалното ветрово поле. Височината на кулата и диаметърът на ротора също влияят на получавана ветрова мощност. При използването на ветровата енергия, представлява интерес интервала 40 – 150 m, в който да се разполагат ВТ. За тази област зависимостта на скоростта на вятъра от неравностите на терена се представя чрез следната зависимост (Ettoumi и др., 2008):

$$u = u_o \left(\frac{h}{h_o} \right)^\gamma \quad (4.16)$$

където: u – скорост на височина h , u_o – известна скорост на височина h_o , γ – експериментално определен коефициент, чиято стойност зависи от височината, часа от денонощието, сезона, температурата и характера на терена. В Таблица 4.2 са дадени типични, зависими от терена, стойности за γ (Ettoumi и др., 2008).

Таблица. 4.2 Типични стойности на γ

Тип на терена	γ
Морска, езерна, гладка земна повърхност	0.10
Висока трева върху равен терен	0.15
Високи посеви, храсталаци	0.20
Гориста местност с много дървета	0.25
Малки селища с дървета, храсти	0.30
Градове с високи сгради	0.40

4.3. Еднокритериален оптимизационен модел за оптимално проектиране на ветроенергийни паркове

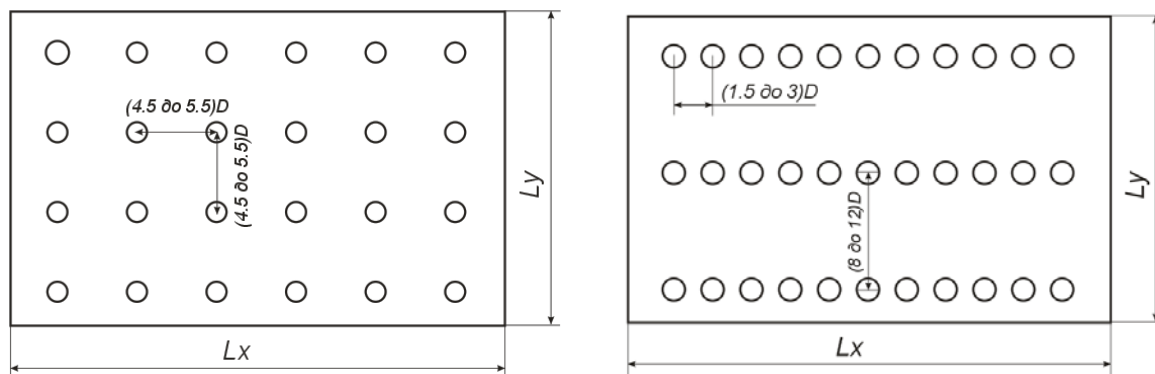
Проектирането на ВЕП е итеративен процес. Необходимо е в ранен етап да се установи, че има достатъчен ресурс, за да направи проекта жизнеспособен. Успоредно с това, е необходимо да се оцени 1) планирането на различни въпроси, включително: околна среда, екология, визуално въздействие, звук, електромагнитни смущения и 2) технически въпроси – терен и условия за достъп, които могат да повлияят на проекта.

Разстоянието между турбините във ВЕП, зависи от терена, от посоката и скоростта на вятъра, и от размерите на ВТ. Турбините обикновено се разполагат в редове, перпендикулярни на преобладаващата посока на вятъра. Ако вятърът достигне втората турбина преди скоростта на вятъра да е възстановена от предишната турбина, то произведената енергия от втората турбина ще бъде по-малко в сравнение с първата. Размерът на това намаление е функция на вятъра и неговата турбулентност, както и на терена. Тези загуби могат да бъдат намалени в диапазона от 5 до 10 %, като по посоката на вятъра разстоянието между ВТ се избере със стойност около 10 роторни диаметъра. Интервалите между ВТ могат допълнително да се увеличат, за да произвежда повече енергия, но за сметка на повече площ, повече пътища и електрически кабели. Като оптимално разстояние между турбините може да се разглежда интервала от 8 до 12 роторни диаметъра по посока на вятъра и от 1.5 до 3 роторни диаметъра за перпендикулярната посока (Emami & Noghreh 2010).

Тук ще се разглежда ветрово поле с предварително установени ветрови характеристики (посока на вятъра и площ). Целта на модела е да бъде определена оптимална конфигурация на ВТ в т. ч. броя и вида на ВТ, така че общата мощност на турбините в полето да е максимална, а инвестиционните разходи да са минимални, при зададени ограничения.

Предизвикателството при оптималното проектиране на ВЕП е да се увеличи енергията, уловена в границите на ветровото поле при зададени ограничения. Минималното възможното разстояние между турбините трябва да гарантира физически да не се получи сблъсък между роторите на ВТ. Близостта между турбините определя областта, непосредствено свързана с генератора, в която няма други турбини, които да бъдат разположени в нея. Точките на мрежата, които са в рамките на тази област са функция на радиуса на ротора на ВТ и посоката на вятъра. При наличие на вятър с еднаква скорост във всички посоки, необходимото разстояние между ВТ може да е между 4.5 – 5.5 роторни диаметъра, за което експериментално е установено, че осигурява достатъчно добро възстановяване на ветровата енергия (Фиг. 4.5а).

Разположението на ВТ във ВЕП при наличие на вятър с преобладаваща посока е показано на Фиг. 4.5б (Grady, и др, 2005).



Фиг. 4.5. Разположение на ВТ:

а) при равностоеен вятър от всички посоки б) при наличие на вятър с преобладаваща посока

Разходите при изграждането на ВЕП са свързани с първоначална инвестиция за закупуване на съответните ВТ, изграждане на електрическа инфраструктура, на инфраструктура за свързване към мрежата и на прилежаща (гражданска) инфраструктура.

При оптималното проектиране трябва да определи какъв тип турбини (размери на ротора, мощност, цена и др.) е целесъобразно да се използват и колко точно трябва да е разстоянието помежду им, така че да се оползотвори цялото налично пространство и при използването на минимални разходи. За определяне разположението на ВТ най-често се използват генетични алгоритми (Grigorios и др., 2008; Grady и др., 2005; González и др., 2010). Друг подход за определяне разположението на ВТ е използването на точни методи за оптимизация като смесено цялочисленото програмиране (Donovan; 2005). В този смисъл, за оптималното разположение на ВТ могат да се използват различни алгоритми, позволяващи да се направи оценка на ВЕП по отношение на неговата ефективност. При оптималното проектиране на ВЕП като целеви функции могат да участват: отношението на получената енергия и инвестиционните разходи, за което се търси максимизиране или получаването на максимална мощност при минимални разходи, т.е. възможно е да се зададе една или повече целеви функции:

$$\max\left(\frac{E}{C}\right) \text{ или } \begin{cases} \max E \\ \min C \end{cases} \quad (4.17)$$

Броят ВТ един ред N_x при наличие на ветрово поле с дължина L_x , отчитайки и необходимото разстояние между отделните турбини се определя чрез израза:

$$N_x = \frac{L_x}{k_x \cdot D} + 1, \quad (4.18)$$

$$k_x^{\min} \leq k_x \leq k_x^{\max} \quad (4.19)$$

където k_x – коефициент, отразяващ разстоянието между отделните турбини.

Важно е да се отбележи, че така полученият брой турбини N_x трябва да бъде цяло число. Броят на турбините по колони във ВП N_y се определя по подобен начин:

$$N_y = \frac{L_y}{k_y \cdot D} + 1, \quad (4.20)$$

$$k_y^{\min} \leq k_y \leq k_y^{\max} \quad (4.21)$$

Диаметърът на ротора е управляваща променлива и за яснота разстоянието между турбините се дава спрямо диаметъра на ротора D . Това означава, че ако диаметърът на ротора е по-голям, то и разстоянието между турбините се увеличава. Изборът на ВТ с определен роторен диаметър се реализира с помощта на двоични променливи x_i както следва:

$$D = \sum_i^m x_i D_{wg}^i \quad (4.22)$$

Обикновено се изисква да се използва цялата налична площ, тъй като и тя е обект на инвестиционни разходи при изграждане на ВЕП. Максималният брой на ВТ в парка N може да се определи чрез равенството:

$$N = N_x N_y \quad (4.23)$$

Общата инсталирана мощност на ВЕП P_{wp} се получава като сума от мощностите на индивидуалните турбини P_{wg} :

$$P_{wp} = NP_{wg} \text{ или } P_{wg} = \sum_i^N P_{wg}^i \quad (4.24)$$

Получената енергия от ВЕП може да бъде определена като произведението от коефициента, отчитащ произвеждането на енергия (η), броя часове на работа на ВЕП, броя инсталирани ВТ и мощността на всяка ВТ (MustakeroV & Borissova, 2010):

$$E = \eta 8760 N P_{wg} \quad (4.25)$$

Инвестиционните разходи и общата мощност, извлечена от турбините участват като управляващи променливи. Цената на електрическата енергия, произведена от вятъра, зависи от разположението на ВТ и обуславя инвестиционните разходи и рентабилността на инсталацията. Общите годишни разходи ВЕП могат да се изразят като функция от общия брой на турбините в парка (Grady и др., 2005, Grigorios и др., 2008):

$$cost = N \left(\frac{2}{3} + \frac{1}{3} e^{-0.00174 N^2} \right) \quad (4.26)$$

Изборът на ВТ с определено напрежение на произвежданата електрическа енергия се осъществява също с помощта на двоичните променливи x_i :

$$U = \sum_i^m x_i U_{wg}^i \quad (4.27)$$

при ограничение по напрежение

$$U^l \leq U \leq U^u \quad (4.28)$$

Тъй като съществуващите турбини имат възможност за еднофазно или трифазно напрежение, то изборът може да бъде предварително ограничен, за да се избере ВТ със необходимата фаза на напрежението:

$$Ph = \sum_i^m x_i Ph_{wg}^i \quad (4.29)$$

Предлаганите ВТ се различават и по честота на ел. енергията – 50 Hz за европейския стандарт и 60 Hz при американския стандарт. За да бъде моделът адекватен на тези стандарти, се въвежда ново ограничение за избор на ВТ, съобразен с тази честота:

$$Hz = \sum_i^m x_i Hz_{wg}^i \quad (4.30)$$

Заедно с всички технологични параметри на ВТ е необходимо да се отчита и влиянието им върху околната среда чрез използване на ограничение за избор на ВТ с определени (допустими) нива на шума:

$$Noise = \sum_i^m x_i Noise_{wg}^i \quad (4.31)$$

$$Noise^l \leq Noise \leq Noise^u \quad (4.32)$$

Всички описани параметри на ВТ се избират с помощта на използването на двоични целочислени променливи, за които е в сила равенството:

$$\sum_i^m x_i = 1, \quad x_i - \text{двоични променливи}, \quad (4.33)$$

където m – множество на ВТ, от които ще се реализира избора.

Основно предимство на подобна формулировка на оптимизационния модел е възможността лесно да се добавят допълнителни потребителски ограничения, с цел точно удовлетворяване на изискванията към проектирания ВП.

Съществува голямо разнообразие от ВТ, произвеждани от различни производители. ВТ могат да се класифицират по различни признаци – мощност, диаметър на ротора, напрежение и честота на ел. ток, шум и др. На пазара могат да се намерят турбини с ротори с диаметри от 0.5 m до над 90 m. Най-често се използва

класификация по мощност, а именно: микротурбини от 50 до 250W, минитурбини от 250W до 1kW), домашни ВТ с мощност 1 – 50 kW, ВТ със средна мощност 50 – 750kW и индустриални ВТ >750kW.

Чрез описаните по-горе зависимости (4.17) – (4.33) се формулира следният еднокритериален оптимизационен модел (Mustakerov & Borissova, 2010):

$$\min\left(\frac{costs}{P}\right) \quad (4.34)$$

при ограничения

$$costs = N\left(\frac{2}{3} + \frac{1}{3}\exp(-0.00174N^2)\right) \quad (4.35)$$

$$P = h_y \eta N P_{wt} \quad (4.36)$$

$$N = N_{row} N_{col}, \text{ integer} \quad (4.37)$$

$$N_{row} = \frac{L_x}{k_{row} D} + 1 \quad (4.38)$$

$$N_{col} = \frac{L_y}{k_{col} D} + 1 \quad (4.39)$$

$$k_{row}^{\min} \leq k_{row} \leq k_{row}^{\max}, k_{row} > 0 \quad (4.40)$$

$$k_{col}^{\min} \leq k_{col} \leq k_{col}^{\max}, k_{col} > 0 \quad (4.41)$$

$$P_{wt} = \sum_i^m x_i P_{wt}^i \quad (4.42)$$

$$D = \sum_i^m x_i D_{wt}^i \quad (4.43)$$

$$H = \sum_i^m x_i H_{wt}^i \quad (4.44)$$

$$\sum_i^m x_i = 1, x \in \{0,1\} \quad (4.45)$$

Целевата функция (4.34) минимизира отношението между инвестиционните разходи и инсталираната мощност. Следователно, решението на този клас оптимизационни задачи ще търси компромис между мощността на ВЕП и общите инвестиционни разходи за неговото изграждане, тъй като и двата параметра (мощност и разходи) зависят от вида и броя на ВТ и от тяхното разположение. От друга страна, броят на ВТ зависи от размера на ВЕП и от преобладаващата посока на вятъра. Произвежданата енергия от ВЕП пряко зависи от броя на ВТ, тяхната номинална мощност и скоростта на вятъра. Ветровите условия за конкретния ВЕП се вземат предвид чрез използване на статистически определен номинален коефициент η на използване на ветровата енергия. В резултат на решаване на съответна оптимизационна задача ще се определя както вида и броя на ВТ, така и тяхното разположение.

Съществена роля имат стойностите на коефициентите k_{row} , k_{col} , чрез които се определят съответните разстояния между ВТ. Броят на ВТ заедно с разстоянията между тях дават оптималното разположение на ВТ във ВЕП.

4.3.1. Числено тестване

Така описаният модел (4.34) – (4.45) е тестван числено, като са разгледани два варианта на ориентацията на вятъра спрямо ветровото поле – наличие на вятър с преобладаваща посока и наличие на равностоеен вятър от всички посоки. Стойностите на коефициентите h_y и η са както следва: $h_y = 8760$ часа (365 дни x 24 часа) и $\eta = 0,3$ (30%). Използвани са параметри на реални ВТ, показани в Таблица 4.3.

Таблица 4.3. Параметри на ВТ

i	ВТ		Номинална мощност, MW	Диаметър на ротора, m	Височина на кулата, m
	тип	Модел			
1	1.1	Enercon E-33	0.330	33.4	37
2	1.2				44
3	1.3				50
4	2.1	Enercon E-48	0.800	48.0	50
5	2.2				60
6	2.3				76
7	3.1	Enercon E-53	0.800	52.9	60
8	3.2				73
9	4.1	Vestas V52	0.850	52.0	49
10	4.2				65
11	4.3				74
12	5.1	Enercon E-44	0.900	44.0	45
13	5.2				55
14	6.1	Vestas V82	1.650	82.0	78
15	7.1	Vestas V80	2.000	80.0	60
16	7.2				78
17	7.3				100
18	8.1	Enercon E-82	2.000	82.0	78
19	8.2				138
20	9.1	Enercon E-70	2.300	71.0	58
21	9.2				113
22	10.1	Vestas V90	3.000	90.0	95
23	10.2				105
24	10.3				125

Източници: <http://www.enercon.de/en-en/>, <http://www.vestas.com/>.

Моделът е тестван за ветрово поле с една и съща площ, но с различни размери, за да се определи влиянието на формата на ветровото поле.

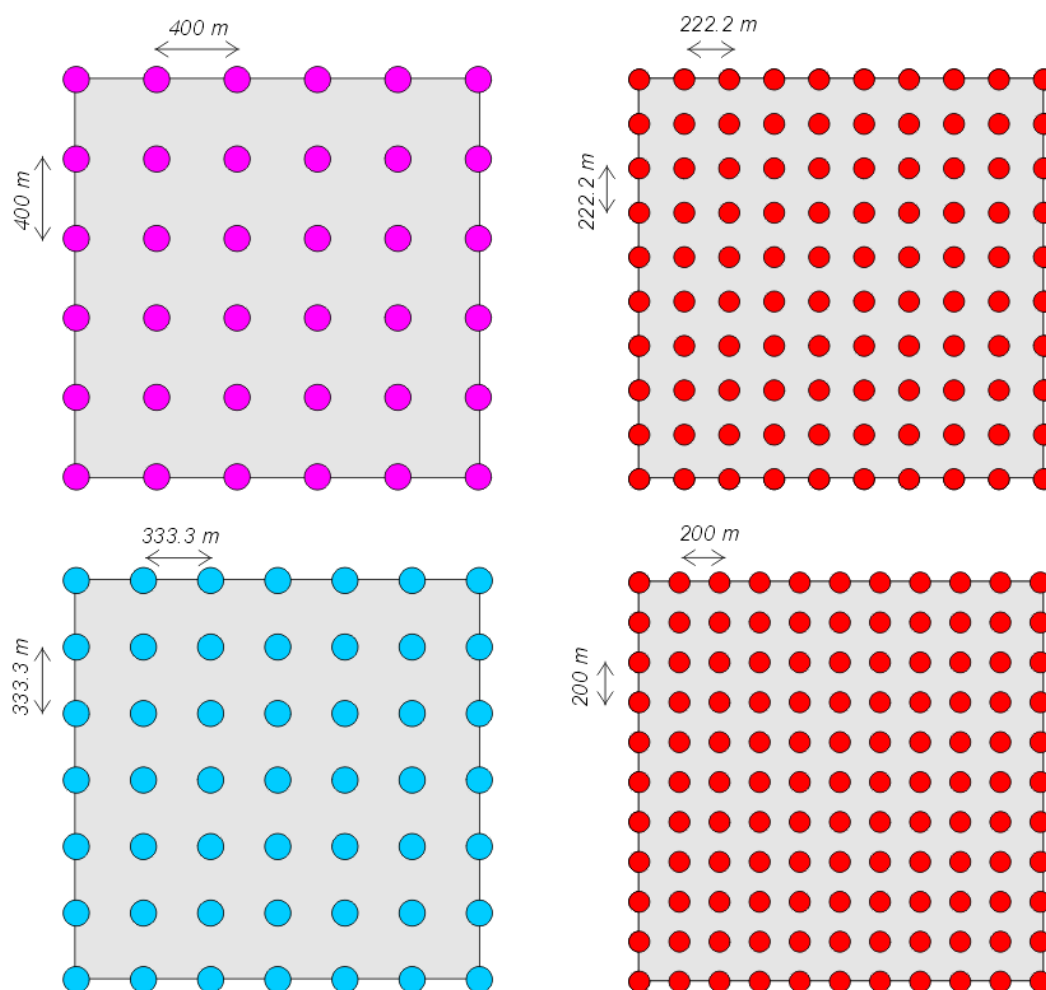
А) случай на равностоеен вятър от всички посоки

За този случай са решени 8 смесено-целочислени дискретни нелинейни оптимизационни задачи,. Четири от тях разглеждат ветровото поле във формата на

квадрат с размери $L_x = 2 \text{ km}$, $L_y = 2 \text{ km}$, а останалите четири разглеждат ветровото поле във формата на правоъгълник със същата площ и размери $L_x = 4 \text{ km}$, $L_y = 1 \text{ km}$.

- **Задача 1а.** Това е задачата (4.34) – (4.45), в която ограниченията (4.40) и (4.41) имат следните стойности $k_{row}^{\min} = k_{col}^{\min} = 4.8$ и $k_{row}^{\max} = k_{col}^{\max} = 5.2$
- **Задача 1б.** Тази задача се отличава от задача 1а по добавеното допълнително ограничение за диаметъра на турбината $D < 70 \text{ m}$. Това допълнително ограничение измества избора на вятърни турбини към по-малките диаметри на вятърните турбини.
- **Задача 1с и Задача 1д** са еквивалентни на задачите 1а и 1б, но за да илюстрира влиянието на разстоянията между ВТ върху решението, имат увеличени граници за коефициентите $k_{row}^{\min} = k_{col}^{\min} = 4.5$ и $k_{row}^{\max} = k_{col}^{\max} = 5.5$.

Решенията на задачите са показани в Таблица 4.4, а съответното разположение на турбините е показано на Фиг. 4.6.



Фиг. 4.6. Разположение на ВТ при равностоеен вятър от всички посоки и квадратна форма на ВЕП

Таблица 4.4. Решения на задачите при ВП с размери $L_x = L_y = 2000$ m

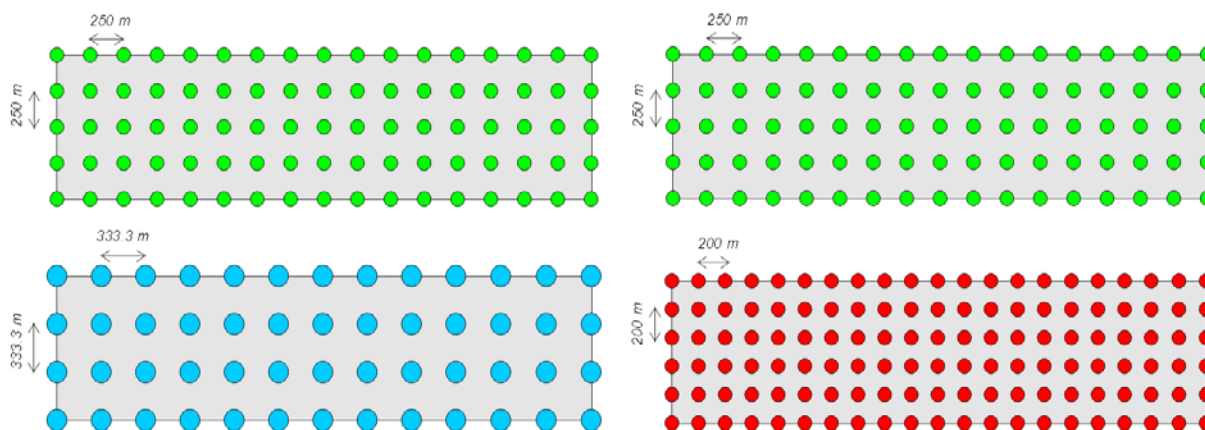
Задача	Избран тип ВТ	Брой ВТ	Инсталирана мощност, MW	Разделителни коефициенти	Разстояния между ВТ, m	Разходи за единица енергия
Задача 1a	Enercon E-82 $H = 78$ m	$N = 36$ $N_x = 6, N_y = 6$	72	$k_{row} = 4.88$ $k_{col} = 4.88$	$SD_x = 400$ $SD_y = 400$	0.000133
Задача 1b	Enercon E-44 $H = 45$ m	$N = 100$ $N_x = 10, N_y = 10$	90	$k_{row} = 5.05$ $k_{col} = 5.05$	$SD_x = 222.2$ $SD_y = 222.2$	0.000282
Задача 1c	Enercon E-70 $H = 58$ m	$N = 49$ $N_x = 7, N_y = 7$	112.7	$k_{row} = 4.7$ $k_{col} = 4.7$	$SD_x = 333.3$ $SD_y = 333.3$	0.000111
Задача 1d	Enercon E-44 $H = 45$ m	$N = 121$ $N_x = 11, N_y = 11$	108.9	$k_{row} = 4.54$ $k_{col} = 4.54$	$SD_x = 200$ $SD_y = 200$	0.000282

Интересно е да се оцени влиянието на геометрична форма на ветровото поле по отношение на избора и разположението на ВТ. За целта, описаните по-горе задачи (Задача 1a, Задача 1b, Задача 1c и Задача 1г) са решени при наличие на една и съща ветрова площ, но за правоъгълна форма с размери $L_x = 4$ km, $L_y = 1$ km.

Резултатите от решаването на модифицираните задачи (Задача 2a, Задача 2b, Задача 2c и Задача 2г) са показани в Таблица 4.5, а разположението на турбините - на Фиг. 4.7.

Таблица 4.5. Решения на задачите при ВП с размери $L_x = 4000$ m, $L_y = 1000$ m

Задача	Избран тип ВТ	Брой ВТ N	Инстали-рана мощност, MW	Разделителни коефициенти	Разстояния между ВТ, m	Разходи за единица енергия
Задача 2a	Vestas V52 $H = 74$ m	$N = 85$ $N_x = 17, N_y = 5$	72.25	$k_{row} = 4.8$ $k_{col} = 4.8$	$SD_x = 250$ $SD_y = 250$	0.000298
Задача 2b	Vestas V52 $H = 49$ m	$N = 85$ $N_x = 17, N_y = 5$	72.25	$k_{row} = 4.8$ $k_{col} = 4.8$	$SD_x = 250$ $SD_y = 250$	0.000298
Задача 2c	Enercon E-70 $H = 58$ m	$N = 52$ $N_x = 13, N_y = 4$	119.6	$k_{row} = 4.69$ $k_{col} = 4.69$	$SD_x = 333.3$ $SD_y = 333.3$	0.000111
Задача 2d	Enercon E-44 $H = 45$ m	$N = 126$ $N_x = 21, N_y = 6$	113.4	$k_{row} = 4.54$ $k_{col} = 4.54$	$SD_x = 200$ $SD_y = 200$	0.000282

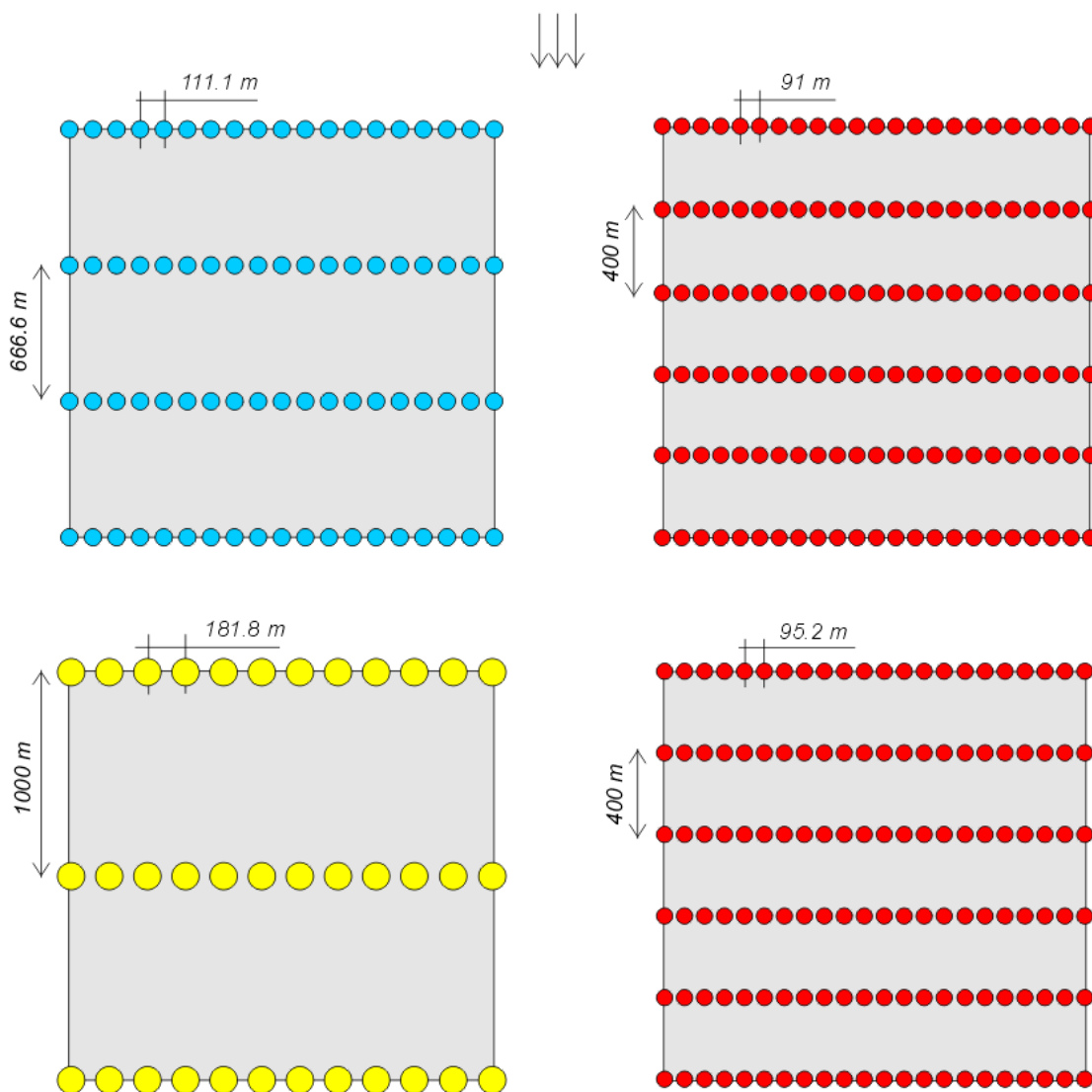


Фиг. 4.7. Разположение на ВТ при равностоеен вятър от всички посоки и правоъгълна форма на ВЕП

Задачите са решени и в случая на различна ориентация на правоъгълното ветрово поле с размери $L_x=1\text{ km}$, и $L_y=4\text{ km}$. Резултатите от решенията показват, че промяната на ориентацията на правоъгълното ветровото поле не влияе на избора на турбини, а само сменя броя на турбините по колони и редове (стойностите на N_{row} и N_{col}).

Б) случай на преобладаващ вятър за една посока

Формулираните по-горе задачи са трансформирани така, че да отразяват наличието на вятър с преобладаваща посока. Задачи **3а** и **3б** използват следните диапазони на изменение на коефициентите: $k_{row}^{\min} = 1.5$, $k_{row}^{\max} = 3$, $k_{col}^{\min} = 9$, $k_{col}^{\max} = 11$. Задачи **3с** и **3д** използват по-големи диапазони на изменение на коефициентите $k_{row}^{\min} = 2$, $k_{row}^{\max} = 4$, $k_{col}^{\min} = 8$, $k_{col}^{\max} = 12$. Решенията на тези задачи са показани в Таблица 4.6, а на Фиг. 4.8 е графично илюстрирано разположението на турбините.



Фиг. 4.8. Разположение на ВТ при вятър с преобладаваща посока и квадратна форма на ВЕП

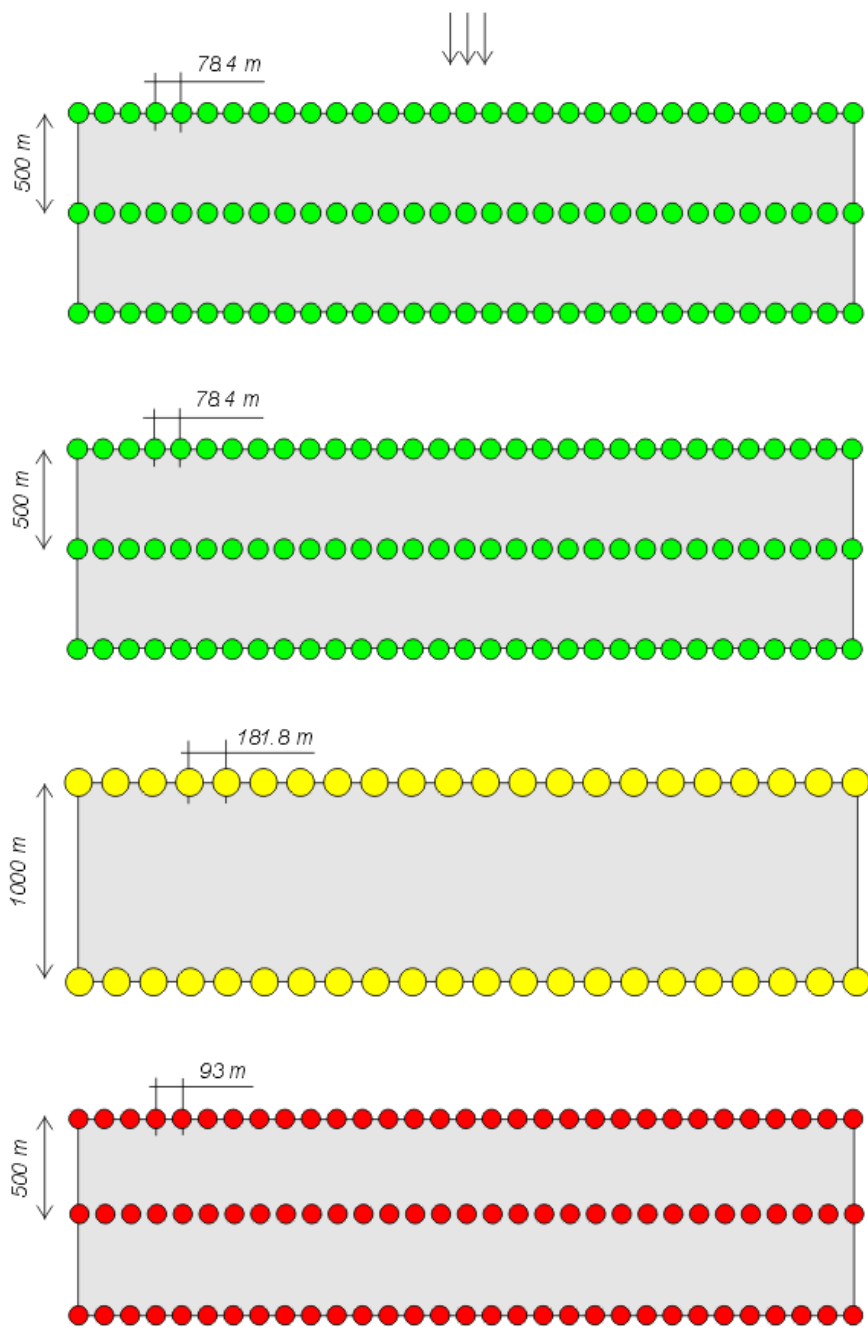
Таблица 4.6. Решения на задачите за преобладаваща посока на вятъра
при ВП с размери $L_x=L_y=2000$ m

Задача	Избран тип ВТ	Брой ВТ N	Инсталирана мощност, MW	Разделителни коэффициенти	Разстояния между ВТ, m	Разходи за единица енергия
Задача 3a	Enercon E-70 $H = 58$ m	$N = 76$ $N_x = 19, N_y = 4$	174.8	$k_{row} = 1.56$ $k_{col} = 9.39$	$SD_x = 111.1$ $SD_y = 666.6$	0.000110
Задача 3b	Enercon E-44 $H = 45$ m	$N = 138$ $N_x = 23, N_y = 6$	124.2	$k_{row} = 2.07$ $k_{col} = 9.09$	$SD_x = 91$ $SD_y = 400$	0.000282
Задача 3c	Vestas V90 $H = 105$ m	$N = 36$ $N_x = 12, N_y = 3$	108	$k_{row} = 2.01$ $k_{col} = 11.11$	$SD_x = 181.8$ $SD_y = 1000$	0.000089
Задача 3d	Enercon E-44 $H = 45$ m	$N = 132$ $N_x = 22, N_y = 6$	118.8	$k_{row} = 2.16$ $k_{col} = 9.09$	$SD_x = 95.2$ $SD_y = 400$	0.000282

В случая на преобладаваща посока на вятъра, геометричната форма на ветровото поле влияе върху избора на вида и броя на вятърните турбини. Оптимизационните задачи са решени и в случая на правоъгълно ветрово поле със следните размери $L_x = 4$ km и $L_y = 1$ km. Получените резултати от решаването на задачите са показани в Таблица 4.7 и са графично онагледени на Фиг. 4.9.

Таблица 4.7. Решения на задачите за преобладаваща посока на вятъра
при ВП с размери $L_x = 4$ km, $L_y = 1$ km

Задача	Избран тип ВТ	Брой ВТ N	Инсталирана мощност, MW	Разделителни коэффициенти	Разстояния между ВТ, m	Разходи за единица енергия
Задача 4a	Vestas V52 $H = 74$ m	$N = 156$ $N_x = 52, N_y = 3$	132.6	$k_{row} = 1.51$ $k_{col} = 9.61$	$SD_x = 78.43$ $SD_y = 500$	0.000298
Задача 4b	Vestas V52 $H = 49$ m	$N = 156$ $N_x = 53, N_y = 3$	132.6	$k_{row} = 1.51$ $k_{col} = 9.61$	$SD_x = 78.4$ $SD_y = 500$	0.000298
Задача 4c	Vestas V90 $H = 105$ m	$N = 46$ $N_x = 23, N_y = 2$	138	$k_{row} = 2.02$ $k_{col} = 11.11$	$SD_x = 181.8$ $SD_y = 1000$	0.000086
Задача 4d	Enercon E-44 $H = 45$ m	$N = 132$ $N_x = 44, N_y = 3$	118.8	$k_{row} = 2.11$ $k_{col} = 11.36$	$SD_x = 93$ $SD_y = 500$	0.000282

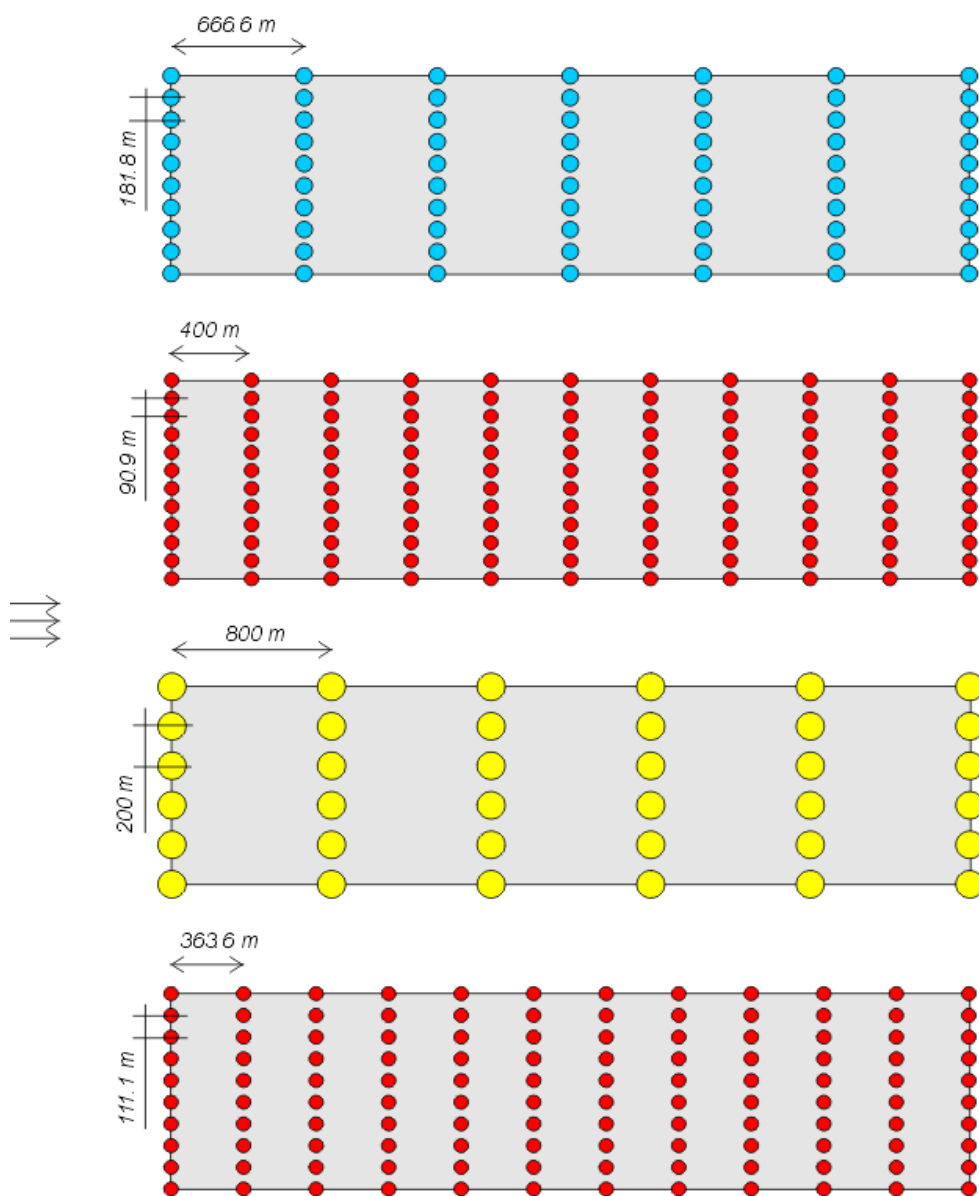


Фиг. 4.9. Разположение на ВТ при вятър с преобладаваща посока и правоъгълна форма на ВЕП

За да се изследва влиянието на ориентацията на ветровото поле спрямо преобладаващата посока на вятъра са формулирани Задача 5а, Задача 5б, Задача 5в и Задача 5г, за размери ветровото поле $L_x=1 \text{ km}$, и $L_y=4 \text{ km}$. Съответните решения и разположения на вятърните турбини са показани в Таблица 4.8 и Фиг. 4.10 съответно.

Таблица 4.8. Решения на задачите при ВП ($L_x = 1 \text{ km}$, $L_y = 4 \text{ km}$)

Задача	Избран тип ВТ	Брой ВТ, N	Инсталирана мощност, MW	Разделителни коефициенти	Разстояния между ВТ, m	Разходи за единица енергия
Задача 5a	Enercon E-70 $H = 58 \text{ m}$	$N = 70$ $N_x = 10, N_y = 7$	161	$k_{row} = 1.56$ $k_{col} = 9.39$	$SD_x = 111.1$ $SD_y = 666.6$	0.000110
Задача 5b	Enercon E-44 $H = 45 \text{ m}$	$N = 132$ $N_x = 12, N_y = 11$	118.8	$k_{row} = 2.07$ $k_{col} = 9.09$	$SD_x = 90.9$ $SD_y = 400$	0.000282
Задача 5c	Vestas V90 $H = 105 \text{ m}$	$N = 36$ $N_x = 6, N_y = 6$	108	$k_{row} = 2.22$ $k_{col} = 8.88$	$SD_x = 200$ $SD_y = 800$	0.000089
Задача 5d	Enercon E-44 $H = 45 \text{ m}$	$N = 120$ $N_x = 10, N_y = 12$	108	$k_{row} = 2.52$ $k_{col} = 8.26$	$SD_x = 111.1$ $SD_y = 363.6$	0.000282



Фиг. 4.10. Разположение на ВТ при вятър с преобладаваща посока и при друга ориентация на правоъгълната форма на ВЕП

Увеличаването на диапазона на изменение на коефициентите k_{row} и k_{col} , определящи разстоянието между ВТ, води до получаването на различни избори на ВТ и респективно до различен дизайн на ВЕП. Най-общо казано, при наличие на равностоеен вятър от всички посоки и по-голям диапазон на изменение на коефициентите k_{row} и k_{col} , определят по-добър избор на ВТ по отношение на инсталирана мощност и разходи за единица енергия (Задача 1а и Задача 1в, Задача 1а-1 и Задача 1в-1). При наличие на преобладаващ вятър за една посока, такава тенденция не се наблюдава (Задача 2а и Задача 2в, Задача 2а-1 и Задача 2в-1, Задача 2а-2 и Задача 2в-2).

Промяната на формата на ВЕП от квадратна към правоъгълна оказва влияние върху избора на типа на турбините. Решенията на някои задачи определят избор на един и същ тип ВТ, но с различен брой и разположение. Промяната на ориентацията на ВЕП с правоъгълна форма в случая на равностоеен вятър за всички посоки не влияе на избора на ВТ, а само сменя стойностите на N_{row} с N_{col} съответно. При наличие на вятър с преобладаваща посока, изборът на ВТ зависи от ориентацията на правоъгълното ветрово поле по отношение на посоката на вятъра. Резултатите от изчисленията показват, че формата и ориентацията на ветровото поле трябва да бъдат взети под внимание при оптималното проектиране на ВЕП. Ограничаването на роторния диаметър на ВТ ограничава избора на турбини и в някои случаи води до избор на един и същ тип ВТ, но с различен брой турбини. Масово използваният и в други изследвания критерий за оптималност изисква минимизиране на стойността на разходите за единица енергия. Сравнението на тези стойности в последните колони на Таблицы 4.4, 4.5, 4.6, 4.7 и 4.8 показват, че използването на по-малък брой мощни ВТ е икономически по-изгодно отколкото по-голям брой маломощни ВТ. Този критерий е един възможните критерии за оптимизация. Предложеният комбинаторен оптимизационен подход за проектиране на ВЕП с помощта на еднокритериални задачи може да се използва и с други критерии.

4.4. Многокритериален оптимизационен модел за оптимално проектиране на ветроенергийни паркове

Многокритериалният модел за проектиране ВЕП взема предвид двата най-важни критерия за оптималност на всеки ветрови парк – разходи и мощност. Като вземем предвид описания комбинаторен оптимизационен подход на моделиране (Mustakerov & Borissova, 2010; Mustakerov & Borissova, 2011), то многокритериалният модел може да бъде формулиран като:

$$\begin{aligned} \max & \text{Power} \\ \min & \text{Costs} \end{aligned} \quad (4.46)$$

при ограничения

$$\text{Power} = h_y \eta N P_{wt}, \quad (4.47)$$

$$\text{Costs} = N \left(\frac{2}{3} + \frac{1}{3} e^{-0.00174 N^2} \right), \quad (4.48)$$

$$SD_x = k_x D_{wt} \quad SD_y = k_y D_{wt} \quad (4.49)$$

$$SD_y = k_y D_{wt} \quad (4.50)$$

$$N = N_x N_y \quad (4.51)$$

$$N_x = (L_x / SD_x) + 1, \quad N_x - \text{integer} \quad (4.52)$$

$$N_y = (L_y / SD_y) + 1, \quad N_y - \text{integer} \quad (4.53)$$

$$k_y^{\min} \leq k_y \leq k_y^{\max}, \quad k_y > 0 \quad (4.54)$$

$$k_x^{\min} \leq k_x \leq k_x^{\max}, \quad k_x > 0 \quad (4.55)$$

$$P_{wt} = \sum_i^m x_i P_{wt}^i \quad (4.56)$$

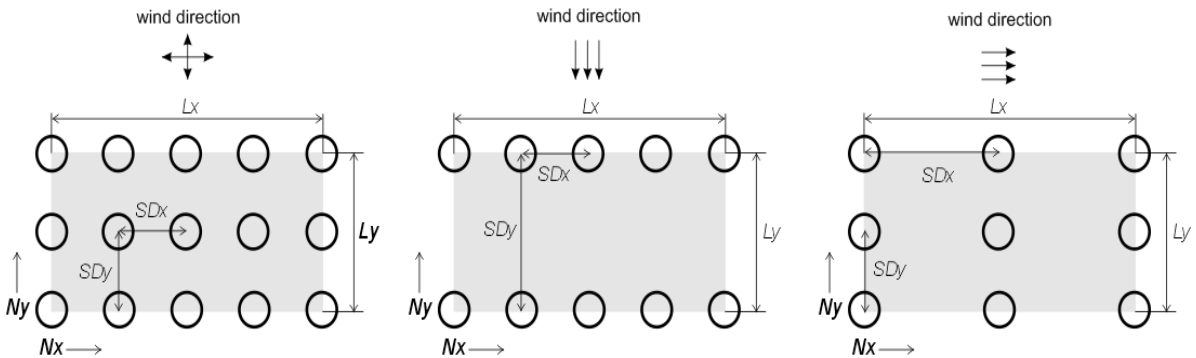
$$D_{wt} = \sum_i^m x_i D_{wt}^i \quad (4.57)$$

$$\sum_i^m x_i = 1, \quad x \in \{0,1\} \quad (4.58)$$

където N – брой на ВТ, N_x – брой ВТ в ред, N_y – брой ВТ в колона, P_{wt}^i – номинална мощност на ВТ, D_{wt}^i – диаметър на ротора на ВТ, L_x, L_y – размери на ВЕП, x_i – двоични целочислени променливи, k_x, k_y, SD_x, SD_y – коефициенти и съответни разстояния между ВТ.

4.4.1. Числено тестване

За численото тестване на предложения многокритериален модел за проектиране на ВЕП и описания алгоритъм са използвани като входни данни, показаните на Фиг. 4.12.



Фиг. 4.12. Размери на ВЕП ($L_x = 4 \text{ km}$, $L_y = 1 \text{ km}$) и посока на вятъра

Разположението на ВТ във ВЕП, зависи от разстоянията SD_x и SD_y , които се определят от диаметъра на ротора D_{WT} на избраната ВТ, умножено по съответните коефициенти k_x и k_y , за трите варианта, показани на Фиг. 2.32: а) $k_x^{\min} = k_y^{\min} = 4.5$, $k_x^{\max} = k_y^{\max} = 5.5$; б) $k_x^{\min} = 1.5, k_y^{\min} = 8$, $k_x^{\max} = 3, k_y^{\max} = 12$; в) $k_x^{\min} = 8, k_y^{\min} = 1.5$, $k_x^{\max} = 12, k_y^{\max} = 3$. Годишната енергия, генерираната от ВЕП се изчислява брой часове за година $h_y = 8760$ часа и за коефициент на използване на вятърната енергия $\eta = 0.3$, т.е. $Power = 2628NP_{wt}$. Параметрите на ВТ са показани в Таблица 4.9.

Таблица 4.9. Параметри на ВТ

#	ВТ	Номинална мощност, kW	Диаметър на ротора, m
1	Enercon E-33	330	33.4
2	Enercon E-48	800	48
3	Enercon E-53	800	52.9
4	Vestas V52	850	52
5	Vestas V60	850	60
6	Enercon E-44	900	44
7	GAMESA AE-61-1320	1320	61
8	Vestas V82	1650	82
9	M Torres	1650	77
10	Vestas V100	1800	100
11	Vestas V80	2000	80
12	Enercon E-82	2000	82
13	Enercon E-70	2300	71
14	SWT-2.3-82 VS	2300	82.4
15	Enercon E-82 E2	2300	82
16	SWT-2.3-113	2300	113
17	SWT-2.3-108	2300	108
18	SWT-2.3-93	2300	93
19	Enercon E-92	2350	92
20	C96 Clipper Windpower	2500	96
21	Vestas V100	2600	100
22	Enercon E-82	3000	82
24	Vestas V90	3000	90
25	Vestas V112	3000	112
23	Enercon -101	3050	101
26	SWT-3.6-120	3600	120
27	SWT-3.6-107	3600	107
28	SWT-6.0-154	6000	154
29	Vestas V164-7.0	7000	164
30	Enercon E-126	7580	127

Всички тези данни се използват за решаване на формулирания проблем (4.46) – (4.58) чрез метода на претеглената сума и лексикографския метод, за да се оценят параметрите от проектирания ВЕП.

Метод на претеглената сума

Използването на метода на претеглената сума води до следната трансформирана задача:

$$\max \left(w_p \left(\frac{Power - Power^{min}}{Power^{max} - Power^{min}} \right) + w_c \left(\frac{Costs^{max} - Costs}{Costs^{max} - Costs^{min}} \right) \right) \quad (4.59)$$

при ограничения (4.47) – (4.58) и допълнителното ограничение

$$w_p + w_c = 1 \quad (4.60)$$

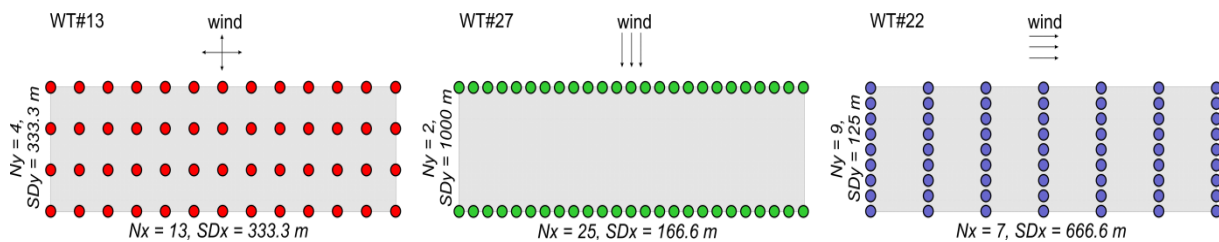
където $Power^{max}$, $Power^{min}$, $Costs^{max}$, $Costs^{min}$ са минималните и максималните стойности на критериите $Power$ и $Costs$.

За да изследва възможността за използване на многокритериалната оптимизация като инструмент за симулация и дизайн на ВЕП, трансформираната задача е решавана за различни предпочитания на ЛРВ и за различни посоки на вятъра, както е показано в Таблица 4.10

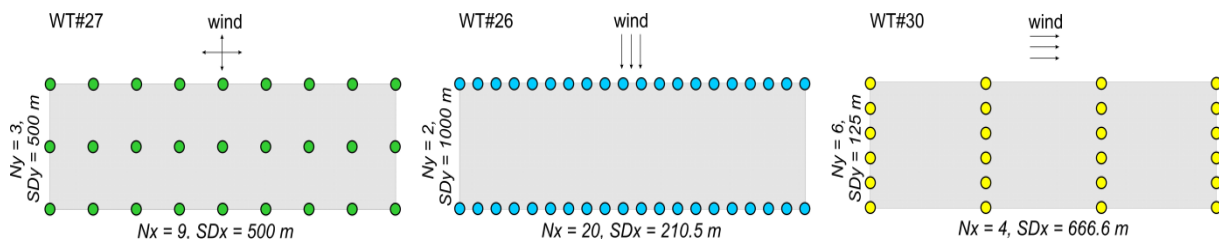
Таблица 4.10. Резултати от използване метода на претеглената сума

Посока на вятъра	Теглови коефициенти	Избрана ВТ (#Таблица 1)	Брой ВТ	Енергия MWh/year	Разходи за единица енергия Costs	Разделителни коефициенти
от всички посоки (Фиг. а)	$w_p = 0.9$ $w_c = 0.1$	ВТ # 13	52	314309	34.824	$k_x = k_y = 4.695$
	$w_p = 0.1$ $w_c = 0.9$	ВТ # 27	27	255 442	20.531	$k_x = k_y = 4.673$
преобладаваща посока (Фиг. б)	$w_p = 0.9$ $w_c = 0.1$	ВТ # 27	50	473 040	33.548	$k_x = 1.558$ $k_y = 9.346$
	$w_p = 0.1$ $w_c = 0.9$	ВТ # 26	46	435 197	31.053	$k_x = 1.515$ $k_y = 8.333$
преобладаваща посока (Фиг. в)	$w_p = 0.9$ $w_c = 0.1$	ВТ # 22	63	496 692	42.021	$k_x = 1.524$ $k_y = 8.130$
	$w_p = 0.1$ $w_c = 0.9$	ВТ # 30	24	478 086	18.936	$k_x = 1.575$ $k_y = 10.499$

Поставянето на по-голяма тежест открито изходна мощност на ВЕП ($w_p=0.9$, $w_c=0.1$) определя разположение на ВТ за различните посоки на вятъра, както е показано на Фиг. 4.13, а използването на противоположните предпочитания ($w_p=0.1$, $w_c=0.9$) води разположение на ВТ, показани на Фиг. 4.14.



Фиг. 4.13. Разположение на ВТ за различни посоки на вятъра и $w_c=0.1$, $w_p=0.9$



Фиг. 4.14. Разположение на ВТ за различни посоки на вятъра и $w_c=0.9$, $w_p=0.1$

Както може да се види от Фиг. 4.13 и Фиг. 4.14, различните тегла, определени от ЛВР водят до определяне на различни разположения на ВТ, като по-голямото теглото води до избор на ВТ с по-малък диаметър. Тъй като диаметърът на ВТ е в пряко влияние на разстоянията между ВТ, то по-малкият диаметър води до използването на по-голям брой ВТ и респективно на по-голяма инсталирана мощност. При противоположния случай (Фиг. 4.14), предпочитанията на ЛВР се отразява в избор на по-мощни турбини и с по-голям диаметър. Това се отразява в определяне на по-малък брой турбини, респективно намаляване на разходите. Ориентацията на ВЕП по отношение посоката на вятъра също влияе на избора на ВТ и на параметрите на проектирания ВЕП (Фиг. 4.13б vs. Фиг. 4.14б и Фиг. 4.13в vs. Фиг. 4.14в).

Използване на лексикографския метод

Използвани са две подредби, за да се изследва тяхното влияние.

- а) В първия случай е избран критерия за мощност като по-важен спрямо разходите за ВЕП. Съответната оптимизационна процедура е както следва:

Стъпка 1: Решаване на оптимизационната задача:

$$\max Power \quad (4.61)$$

при ограничения (4.47) – (4.58)

Стъпка 2: Решаване на оптимизационната задача

$$\min Costs \quad (4.62)$$

при ограничения (4.47) – (4.58) и допълнително ограничение за мощността

$$Power \geq \varepsilon Power^{max} \quad (4.63)$$

където коефициентът ε определя степенята на близост до оптималната стойност $Power^{max}$ определена на първата стъпка. Използвани са две стойности на този коефициент $\varepsilon = 0.7$ и $\varepsilon = 0.9$ за да се сравни на тяхното влияние върху решенията.

- б) При втория случай, като по-важен е избран критерия за разходите $Costs$, което води до следната изчислителна процедура:

Стъпка 1: Решаване на оптимизационната задача

$$\min \left\{ N \left(\frac{2}{3} + \frac{1}{3} e^{-0.00174 N^2} \right) \right\} \quad (4.64)$$

при ограничения (4.47) – (4.58).

Стъпка 2: Решаване на оптимизационната задача

$$\max Power \quad (4.65)$$

при ограничения (4.47) – (4.58) и допълнително ограничение за разходите

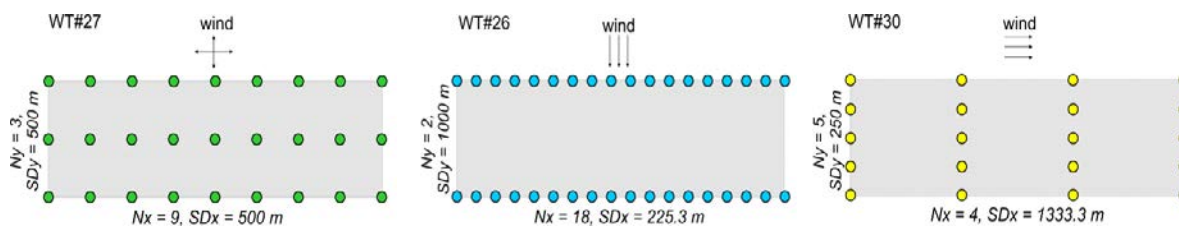
$$Costs \leq \varepsilon Costs^{min} \quad (4.66)$$

И тук коефициентът ε определя степенята на близост до оптималната стойност $Costs^{min}$ от първата стъпка и също така са използвани две стойности $\varepsilon = 1.3$ и $\varepsilon = 1.1$. Получените решения за избор на ВТ, брой ВТ, както и тяхното разположение чрез използване на лексикографския метод са показани в Таблица 4.11.

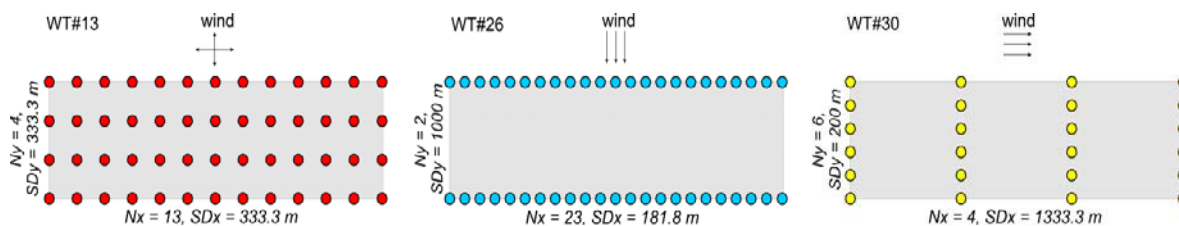
Таблица 4.11. Резултати от използване на лексикографския метод

Посока на вятъра	ε	ВТ (# от Табл. 4.9)	Брой ВТ	Енергия MWh/year	Разходи за единица енергия	Разделителни коефициенти
а) лексикографска подредба: $Power, Costs$						
от всички посоки (Фиг. а)	0.7	ВТ # 27	27	255 442	20.531	$k_x = k_y = 4.673$
	0.9	ВТ # 13	52	314 309	34.823	$k_x = k_y = 4.695$
преобладаваща посока (Фиг. б)	0.7	ВТ # 26	36	340 589	25.253	$k_x = 1.961$ $k_y = 8.333$
	0.9	ВТ # 26	46	435 197	31.053	$k_x = 1.515$ $k_y = 8.333$
преобладаваща посока (Фиг. в)	0.7	ВТ # 30	20	398 405	16.657	$k_x = 1.968$ $k_y = 10.499$
	0.9	ВТ # 30	24	478 086	18.936	$k_x = 1.575$ $k_y = 10.499$
б) лексикографска подредба $Costs, Power$						
от всички посоки (Фиг. а)	1.3	ВТ # 27	27	255 442	20.531	$k_x = k_y = 4.673$
	1.1	ВТ # 27	27	255 442	20.531	$k_x = k_y = 4.673$
преобладаваща посока (Фиг. б)	1.3	ВТ # 27	36	340 589	25.258	$k_x = 2.199$ $k_y = 9.346$
	1.1	ВТ # 27	28	264 902	21.052	$k_x = 2.876$ $k_y = 9.346$
преобладаваща посока (Фиг. в)	1.3	ВТ # 30	24	478 086	18.936	$k_x = 10.499$ $k_y = 1.575$
	1.1	ВТ # 30	16	318 724	14.083	$k_x = 10.499$ $k_y = 2.625$

Разположението на ВТ във ВЕП при използването на лексикографска подредба *Power, Costs* за различните посоки на вятъра са показани на Фиг. 4.15 (при $\varepsilon = 0.7$) и Фиг. 4.16 ($\varepsilon = 0.9$).

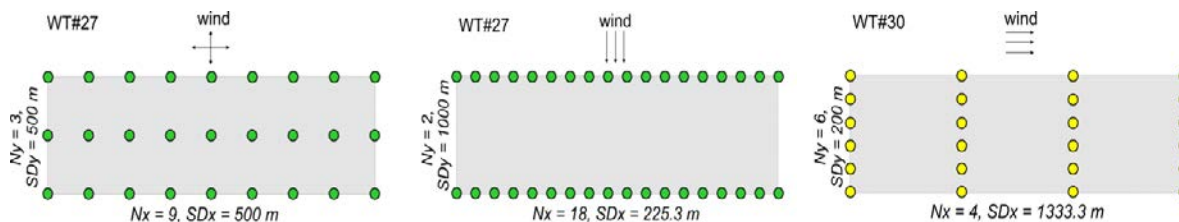


Фиг. 4.15. Лексикографска подредба *Power, Costs* и $\varepsilon = 0.7$

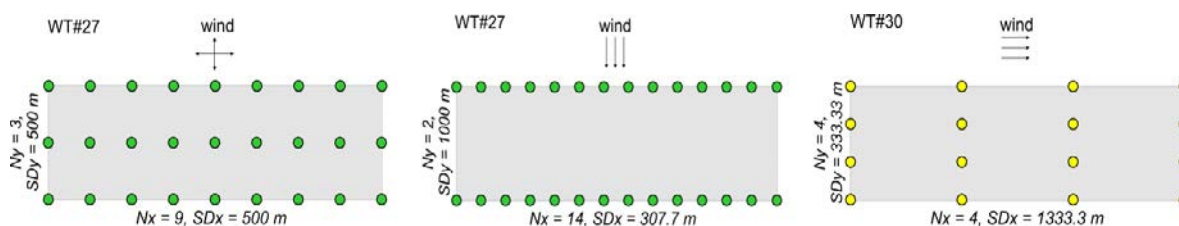


Фиг. 4.16. Лексикографска подредба *Power, Costs* и $\varepsilon = 0.9$

На Фиг. 4.17 ($\varepsilon = 1.3$) и Фиг. 4.18 ($\varepsilon = 1.1$) са показани разположението на ВТ във ВЕП при използването на лексикографска подредба *Costs, Power*.



Фиг. 4.17. Лексикографска подредба *Costs, Power* и $\varepsilon = 1.3$

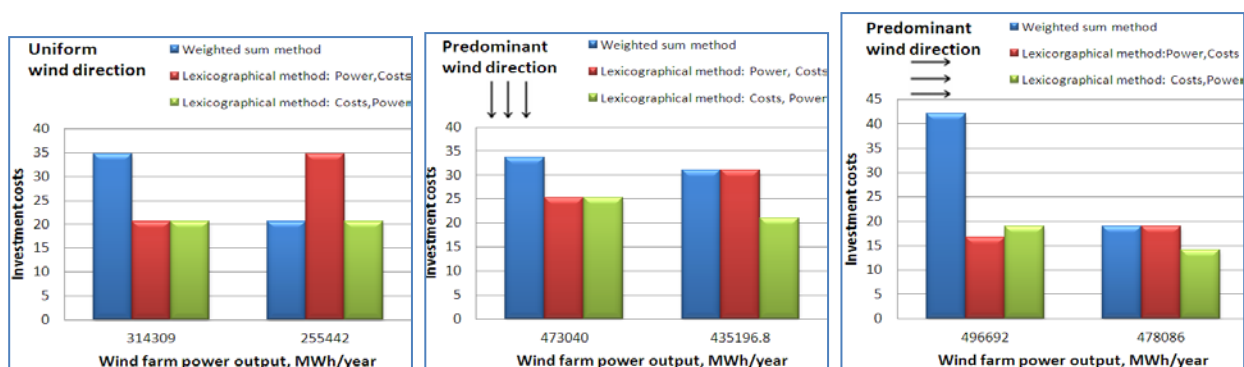


Фиг. 4.18. Лексикографска подредба: *Costs, Power* и $\varepsilon = 1.1$

Промяната на ε от 0.7 до 0.9 се отразява в увеличаване броя на турбините, така че мощността да достигне по-близо до своя оптимум (Фиг. 4.15 и Фиг. 4.16). Идентична тенденция, но в различна посока се наблюдава, когато лексикографската подредбата се смени, намаляване стойността на ε от 1.3 на 1.1 води до намаляване на разходите чрез намаляване на броя на турбините (Фиг. 4.17 и Фиг. 4.18). Използването на коефициента

е позволява гъвкава настройка на предпочитанията на ЛВР по отношение на критериите.

Ефективността на проектирането на ВЕП може да бъде оценено от инвестиционните разходи и от очакваната мощност. Използването и на двата метода за решаване (претеглената сума и лексикографския) осигуряват теоретични оценки на мощността и разходите при различни посоки на вятъра и различни предпочитания на ЛВР. На Фиг. 4.19 са показани получените резултати от решаваните многокритериални задачи и получените параметри на ВЕП – инвестиционни разходи и очаквана мощност.



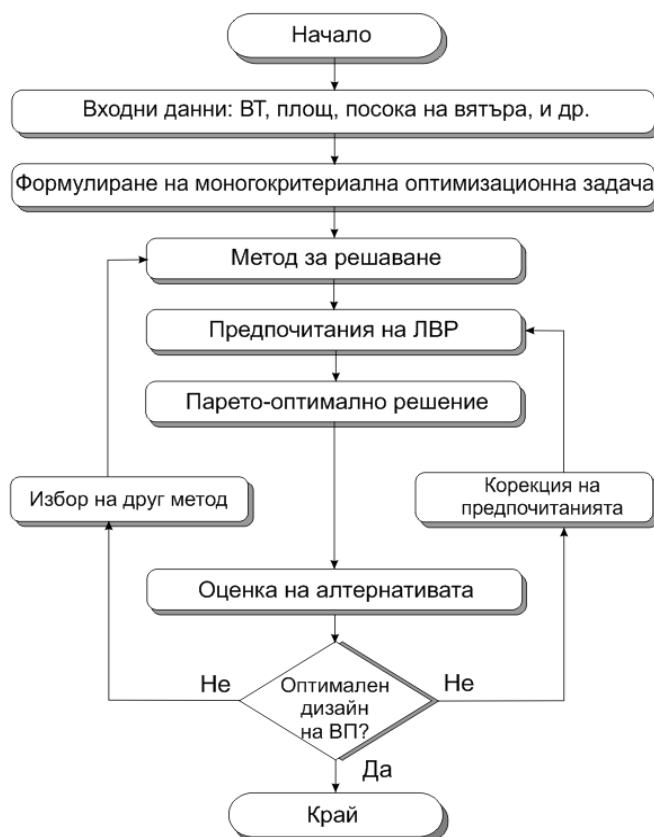
Фиг. 4.19. Инвестиционни разходи и очаквана мощност при различни предпочитания на ЛВР

Така получените резултати могат да се използват за формиране на крайното решение при проектирането на конкретен ВЕП.

4.4.2. Алгоритъм за приложение на многокритериалния подход при проектиране на ветроенергийни паркове

Многокритериалният модел за оптимално проектиране ветрови парк е в основата на алгоритъм за оценка на разположението на турбините на проектирани ВЕП. Обобщената диаграма на предложения алгоритъм е показана на Фиг. 4.20. За формулиране на многокритериалната задача за оптимално проектиране на ВЕП е необходимо да се определят входните данни – ВТ, от които да бъде направен избора, площ на ВЕП, посока на вятъра, разстояния между ВТ и т.н. След формулиране на задачата се избира метод за решаване, в зависимост от който ЛВР задава своите предпочитания към определените критерии. На база на зададените предпочитания се определя и Парето-оптималното решение. Резултатите от решенията се съхраняват като Парето-оптимални алтернативи и могат да бъдат оценени на по-късен етап от ЛВР. Ако ЛВР не е удовлетворен от полученото решение, той може да промени предпочитанията си относно критериите или да използва друг метод за решаване. В резултат на

изпълнението на алгоритъма се получава набор от Парето-оптимални алтернативни, на база на които ЛВР може да вземе крайното решение за дизайна на ВЕП.



Фиг. 4.20. Алгоритъм за приложение на многокритериалния подход за проектиране на ВЕП

Формулираните едно- и многокритериални оптимизационни задачи са решавани с LINGO вер. 12 на настолен компютър с Intel® Celeron® 2.93 GHz процесор, 2 GB оперативна памет и операционна система MS Windows XP ©. Времената за решаване на числените примери са от порядъка на секунди.

Описаните едно- и многокритериални оптимизационни модели за проектиране инженерни системи, вземащи предвид съществуващите взаимовръзки, външните експлоатационни условия и икономическата ефективност са базирани на примера на проектирането на ветроенергийни паркове. Формулираният модел отчита както параметрите на наличните ветрови ресурси, така и параметрите на наличните ВТ. Характерна особеност и на едно- и многокритериалният модел е, че едновременно с определяне на избора на типа ВТ се определя и необходимия брой и разположението на избрания тип ВТ. В резултат на формулирането и решаването на съответните оптимизационни задачи се получава оптимален или Парето-оптимален избор на типа на ВТ, броя и съответното им разположение в ВЕП. Алгоритъмът за използване на многокритериалния подход може да бъде реализиран в подходяща софтуерна система за подпомагане проектирането на ВЕП.

Глава 5:

Модели и алгоритми за оптимално планиране

5.1. Въведение

В тази глава са описани предложените модели и алгоритми за оптимално планиране при наличие на ограничени ресурси, касаещи формулирането на: 1) модели за определяне на оптимални разписания за зависима обработка на детайли върху множество машини, 2) модели и алгоритми за определяне на оптимални разписания при смесена (независима и зависима) обработка на детайли върху множество машини, 3) обобщен модел и алгоритъм за оптимално линейно разкрояване и 4) обобщен модел и алгоритъм за оптимално планиране на персонала и на графика за работата му. Получените резултати са публикувани в статии D1, D6, D9, D27.

Организацията на производствените процеси в съвременните предприятия осигурява не само постигането на производствените цели, но изисква и оптимално разпределение на материалните и човешки ресурси. Планирането на ресурсите на предприятието оптимизира управлението на компанията (Stoilov & Stoilova, 2008). В много случаи от практиката това води до сложни комбинаторни оптимизационни проблеми. При това, големината на разглежданите предприятия не е показател за сложността на тези проблеми. Възможно е малките и средните предприятия да имат по-сложни организационни проблеми от някои големи фирми и компании. Едно от обясненията е много по-изострената конкурентна борба, в резултат на която предприятията са принудени да разширяват спектъра от дейностите си. Друга особеност на съвременните предприятия е разширяване на спектъра на целите. Като основна цел се приема постигането на максимална печалба за определен период от време. Всеки сътрудник в организацията също има свои лични цели и задачи, които също не би следвало да се пренебрегват. Подобно многообразие на цели и задачи допълнително усложнява цялостната организация на дейността на съвременните предприятия. Трябва да се вземе предвид и факта, че съвременните предприятия не

функционират изолирано, а са част от определена икономическа среда, която влияе съществено – наличие на конкуренти, възможности за финансово подпомагане при изпълнението на определени условия и т.н. Това влияние постоянно расте и в съчетание с останалите фактори определя необходимостта от използване на научни подходи при решаване на подобни организационни проблеми.

В общия случай, управлението на производствения процес включва всички дейности, които трябва да бъдат изпълнени при коректно планирано разпределение на ресурсите и отчитане на целите му. Това включва използването на информация за времетраенето на производствения процес, времетраенето на отделните операции, наличните производствени ресурси (машини, специалисти, материали), съществуващи ограничения и т.н. Тази информация може се използва за определяне на моментите от време на започване и на завършване на всяка операция, последователността на операциите, маршрутите на детайлите между обработващите машини, т.е. необходимо е изготвянето на производствен план (разписание), съобразен с всички организационни изисквания и ограничения. Желателно е този производствен план да е най-добрия възможен (оптимален) в смисъла на определени критерии. Като се има предвид сложността на съществуващите взаимовръзки между операциите и ресурсите, горната задача не може да бъде решена по интуитивен път дори от много опитни специалисти. Развитието на съвременната компютърна техника и на научните постижения в областта на операционните изследвания са добра предпоставка за използването на математическите средства и методи за постигане на ефективно оползотворяване на наличните ресурси.

Разписанията могат да се разглеждат като форма на вземане на решения за управление на организационния процес, играеща важна роля в производството. Управлението на производствения процес включва всички дейности, които трябва да бъдат изпълнени, за да бъде планирано коректното разпределение на ресурсите и да бъдат постигнати целите.

Цел на теория на разписанията е създаване на математични методи за получаване на решение с използване на моделите за планиране, напр. в задачи за съгласуване или разпределение. При задачите за съгласуване е необходимо да се определи продължителността на набор от дейности, при зададено тяхно разпределение по машините и общото време за тяхното изпълнение, с цел да се минимизират сумарните разходи за дейностите. При тези задачи се предполага, че стойността на дейността зависи от нейната продължителност. Целта на задачите за разпределение е да

се намери оптималното разпределение на набор от дейности в набор от машини, подходящи за тяхното изпълнение. Трябва да се отбележи, че тези оптимизационни задачи силно се различават помежду си. Практически, това води до невъзможност за създаването на единна обобщена методология, селектиране и разработване на алгоритми, подходящи за цялостното им решаване. Към всеки отделен случай се изисква индивидуален подход с цел търсене на частно решение. Като изходни данни се разглеждат две множества: множеството на машините $M = \{M_1, M_2, \dots, M_m\}$ и множеството на работите $J = \{J_1, J_2, \dots, J_n\}$, като всяка работа се състои от определени операции $J_i = \{O_{i1}, O_{i2}, \dots, O_{in}\}$. Разписанието предоставя информация на кои машини и в кои моменти от време трябва да се изпълнят работите. Всяка операция O_{ij} има определено време за изпълнение t_{ij} и може да се изпълнява на една от множество машини $\mu_{ij} \subseteq \{M_1, \dots, M_m\}$. Ако $|\mu_{ij}| = 1$, за $\forall ij$, се получава модел с предписания. Ако $|\mu_{ij}| = m$, за $\forall ij$, се получава модел с паралелни машини. Според спецификата на определяното разписанието, проблемите могат се класифицират в 5 групи (Androvasko, 2011; Naderi & Azab, 2014; Tseng и др., 2004, Naderi & Zandieh, 2014; Koulamas & Kyparisis, 2015): 1) *job shop* – за всяка операция има определена машина $|\mu_{ij}| = 1$ и съществува линеен порядък за изпълнение на операциите, 2) *flow shop* – съществува подредба на машините $M_1, M_2, \dots, M_m$ и всяка работа се извършва на всички машини в този ред, $n_i = m$ и $\mu_{ij} = M_j$, за $\forall i$, 3) *open shop* – всяка работа се състои от m операции ($n_i = m$), но $\mu_{ij} = \{M_1, \dots, M_m\}$ и не съществува условие за подредба на множеството на операциите, 4) *mixed shop* – комбинация от *job shop* и *open shop*, 5) *general case* – произволен ред на работите и операциите на машините. Един от най-изследваните оптимизационни комбинаторни проблеми са проблемите за определяне на разписания при *job shop*. Въпреки това, този проблем остава все още труден за решаване. От изчислителна гледна точка, това е NP-труден проблем (Garey, и др., 1976; Sotskov & Shakhlevich, 1995). Най-простият случай за такова разписание е проблемът за последователна обработка на една машина (Baker & Keller, 2010). Една от основните цели на при определянето на оптимални разписания е минимизиране на общия престой на машините. Най-краткото време за обработка ще определи и оптималното разписание, ако живота на инструмента се счита за безкрайно дълъг (Akturk и др., 2003). В действителност, реалните проблеми за разписания обикновено включват много и различни машини.

Класическият проблем за определяне на разписание за обработка от тип *job shop* е един от най-типичните и сложни проблеми, характеризиращ се със следните

особености: 1) обработката се извършва на различни машини, с определени операции на всяка от тях; 2) всяка работа се състои от набор от операции и известен ред на изпълнение на работата по машините; 3) всяка операция се характеризира с необходимите машини и време за обработка на всяка машина. Целта на разписанието е да се присвоят n работи на m машини, така че да се оптимизира определена цел (Tavakkoli-Moghaddam и др., 2005). Възможни са следните оптимизационни цели: 1) да се максимизира използването на ресурсите или с други думи, да се минимизира времето на работите, 2) да се минимизира операционното време на всяка машина или времето за престой, причинено от преждевременно изпълнение на операция или от загуба на време при започване на работа на машините. Това са 2 паралелни цели и в някои случаи оптималното разписание се променя в зависимост от типа на целите. Това е очевидно в по-голяма степен, когато загубите са различни за различните машини. Възможен е и случай на комбинирането на горните цели.

През последните десетилетия множество техники са разработени за детерминирани класически проблеми за разписания – аналитични техники, използване на правила, мета-евристични алгоритми и алгоритми, които използват динамично програмиране (Pinedo, 1983; Weber и др., 1986; Golenko-Ginzburg & Gonik, 2002; Lawrence & Sewell, 1997; Gromicho и др., 2012). Традиционно най-често се използват подходи с правила, комбинаторна оптимизация и анализ на ограниченията (Dubois и др., 1995).

5.2. Оптимално разписание за зависима обработка на детайли върху множество машини

Определянето на разписания за зависима обработка на детайли върху множество машини и е сложен оптимизационен проблем. При зависимата обработка на детайли съществуват следните особености: 1) детайлите се обработват в определена последователност $D_{i1}, \rightarrow D_{i2}, \rightarrow \dots \rightarrow D_{in}$; 2) обработката на всеки детайл трябва да следва определен ред на изпълнение на операциите $O_{i1}, \rightarrow O_{i2}, \rightarrow \dots \rightarrow O_{in}$; 3) продължителността на всяка операция е известна; 3) всяка машина може да изпълнява в даден момент от време само една операция. В този смисъл, зависимата обработка на детайли може да се разглежда като вариант на проблем, познат като *job shop*. Целта е да се определи минималното общо време за обработка на всички детайли, както и началните времена на започване и завършване на всяка операция за детайлите на всяка от машините.

5.2.1. Обобщен модел за определяне на оптимални разписания за зависима обработка на детайли върху множество машини

При формулирането на обобщения модел за класа от проблеми, касаещи определяне на оптимално разписание за зависима обработка на детайли върху множество машини се въвеждат управляващи променливи $X_{i,j}$, определящи началните моменти от време за отделните операции, като индексът “ i ” се отнася за детайлите, а индексът “ j ” за машините, на които се обработват детайлите. Въвеждат се и променливи, изразяващи моментите от време на завършване на обработката на отделните детайли $X_{i,E}$. Цел на обобщеният модел за определяне на оптимални разписания за зависима обработка на детайли върху множество машини е оптимизиране на процеса на обработка на детайлите, по отношение на времето на обработка, т.е. целевата функция минимизира продължителността на обработката на всичките детайли (Mustakeroov & Borissova, 2008):

$$\min \sum x_{iE}, \quad i \in I, \quad (5.1)$$

при ограничения

$$x_{i,(j+1)} - x_{i,j} \geq T_{i,j}, \quad i \in I, j \in J \quad (5.2)$$

$$x_{(i+1),j} - x_{i,j} \geq T_{i,j}, \quad i \in I, j \in J \quad (5.3)$$

$$x_{ij} \geq 0, \text{ за } \forall i \in I, \forall j \in J, \quad (5.4)$$

$$x_{iE} > 0, \text{ за } \forall i \in I, \quad (5.5)$$

където: I е множеството на обработваните детайли, а J е множеството на машините, необходими за обработването на тези детайли.

Ограниченията (5.2) изразяват факта, че операцията за обработка на детайл i на машина $(j+1)$ може да започне след необходимото време $T_{i,j}$ за обработка на детайл i на машина j . Ограниченията (5.3) изразяват факта, че следващият детайл $(i+1)$ може да започне обработка на същата машина, само след като завърши обработката на предишния детайл, като се осигурява обработката само на един детайл, в даден момент от време на конкретна машина.

5.2.2. Числено тестване

За да се илюстрира числено описаният математически модел (5.1) – (5.5) за определяне на оптимално разписание на зависима обработка на детайли върху множество машини, е използван реален проблем от работата на металообработваща фирма, произвеждаща машини с ЦПУ. В производствения процес на фирмата се използват гама от специфични и универсални машини. При това, е необходимо да се обработват големи и

тежки корпусни детайли, чието транспортиране и съхраняване също трябва да бъде взето предвид. Това налага да бъде планирано и разпределено времето за обработката им така, че да бъдат спазени не само технологичните изисквания, но необходимата последователност при обработването на детайлите, отчитайки използването на едни и същи налични машини и минимизирайки времената за престой в склада. Технологичният маршрут на механичната обработка на детайли, необходими за сглобяването на металорежеща машина с ЦПУ тип M500, която е част от производството на фирмата, е показан в Таблица 5.1.

Таблица 5.1. Технологична последователност, операции и машини за обработка на детайли

Технологична последователност	Детайл	Операция	Време (часове)	Машина
1	Д1 (Основа)	O11 (фрезова 1)	8	M1 (Координатна машина Микромат)
		O12 (пробивно-фрезова 1)	6	M2 (Координатна машина VTEC)
		O14 (пробивна)	6	M4 (Борверг с CNC MicroCUT)
2	Д2 (Колона)	O21 (фрезова 1)	8	M1 (Координатна машина Микромат)
		O22 (пробивно-фрезова 1)	10	M2 (Координатна машина VTEC)
		O24 (пробивна)	6	M4 (Борверг с CNC MicroCUT)
3	Д3 (Кръстачка)	O31 фрезова 1	8	M1 (Координатна машина Микромат)
		O33 (фрезова 2)	8	M3 (Хоризонтална фреза с CNC)
		O32 (пробивно-фрезова 1)	8	M2 (Координатна машина VTEC)
		O34 (пробивна)	4	M4 (Борверг с CNC MicroCUT)
4	Д4 (Z Шейна)	O41 (пробивно-фрезова 3)	4	M1 (Координатна машина Микромат)
		O42 (фрезова 3)	1	M2 (Координатна машина VTEC)
		O43 (пробивно-фрезова 2)	2	M3 (Хоризонтална фреза с CNC)
5	Д5 (Маса)	O51 (фрезова 1)	4	M1 (Координатна машина Микромат)
		O52 (пробивно-фрезова 1)	12	M2 (Координатна машина VTEC)
		O53 (пробивно-фрезова 2)	4	M3 (Хоризонтална фреза с CNC)
		O55 (шлифовъчна)	8	M5 Шлайфхобел
6	Д6 (Шпинделна кутия)	O61 (пробивно-фрезова 3)	6	M1 (Координатна Машина Микромат)
		O63 (пробивно-фрезова 2)	8	M3 (Хоризонтална фреза с CNC)
7	Д7 (Винкел за магазин)	O73 (фрезова 2)	6	M3 (Хоризонтална фреза с CNC)
		O74 (пробивно-фрезова 4)	8	M4 (Борверг с CNC MicroCUT)

Въведени са променливи, изразяващи началните моменти от време за обработка на детайлите, като са използвани данните от Таблица 5.1:

- x_{11}, x_{12}, x_{14} – начални моменти на обработка на детайл Д1 на машини М1, М2, и М4,
- x_{21}, x_{22}, x_{24} – начални моменти за обработка на детайл Д2 на машини М1, М2 и М4,
- $x_{31}, x_{33}, x_{32}, x_{34}$ – начални моменти за обработка на детайл Д3 на машини М1, М3, М2 и М4,
- x_{41}, x_{42}, x_{43} – начални моменти за обработка на детайл Д4 на машини М1, М2 и М3,
- $x_{51}, x_{52}, x_{53}, x_{55}$ – начални моменти за обработка на детайл Д5 на машини М1, М2, М3 и М5,
- x_{61}, x_{63} – начални моменти за обработка на детайл Д6 на машини М1 и М3,
- x_{73}, x_{74} – начални моменти за обработка на детайл Д7 на машини М3 и М4,

а също и променливи за крайните моменти от време за обработка на детайлите:

- $x_{1E}, x_{2E}, x_{3E}, x_{4E}, x_{5E}, x_{6E}, x_{7E}$ – съответно за Д1, Д2, Д3, Д4, Д5, Д6 и Д7.

Като се вземе предвид информацията от Табл. 5.1 е формулирана следната еднокритериална оптимизационна задача при технологична последователност на обработка на детайлите: $Д1 \rightarrow Д2 \rightarrow Д3 \rightarrow Д4 \rightarrow Д5 \rightarrow Д6 \rightarrow Д7$:

$$\min = x_{7E} \quad (5.9)$$

при ограничения

$$x_{12} - x_{11} \geq 8 \quad (5.10)$$

$$x_{14} - x_{12} \geq 6 \quad (5.11)$$

$$x_{1E} - x_{14} \geq 6 \quad (5.12)$$

$$x_{22} - x_{21} \geq 8 \quad (5.13)$$

$$x_{24} - x_{22} \geq 10 \quad (5.14)$$

$$x_{2E} - x_{24} \geq 6 \quad (5.15)$$

$$x_{33} - x_{31} \geq 8 \quad (5.16)$$

$$x_{32} - x_{33} \geq 8 \quad (5.17)$$

$$x_{34} - x_{32} \geq 8 \quad (5.18)$$

$$x_{3E} - x_{34} \geq 4 \quad (5.19)$$

$$x_{42} - x_{41} \geq 4 \quad (5.20)$$

$$x_{43} - x_{42} \geq 1 \quad (5.21)$$

$$x_{4E} - x_{43} \geq 2 \quad (5.22)$$

$$x_{52} - x_{51} \geq 4 \quad (5.23)$$

$$x_{53} - x_{52} \geq 12 \quad (5.24)$$

$$x_{55} - x_{53} \geq 4 \quad (5.25)$$

$$x_{5E} - x_{55} \geq 8 \quad (5.26)$$

$$x_{63} - x_{61} \geq 6 \quad (5.27)$$

$$x_{6E} - x_{63} \geq 8 \quad (5.28)$$

$$x_{74} - x_{73} \geq 6 \quad (5.29)$$

$$x_{7E} - x_{74} \geq 8 \quad (5.30)$$

$$x_{21} - x_{11} \geq 8 \quad (5.31)$$

$$x_{31} - x_{21} \geq 8 \quad (5.32)$$

$$x_{41} - x_{31} \geq 8 \quad (5.33)$$

$$x_{51} - x_{41} \geq 4 \quad (5.34)$$

$$x_{61} - x_{51} \geq 4 \quad (5.35)$$

$$x_{22} - x_{12} \geq 6 \quad (5.36)$$

$$x_{32} - x_{22} \geq 10 \quad (5.37)$$

$$x_{42} - x_{32} \geq 8 \quad (5.38)$$

$$x_{52} - x_{42} \geq 1 \quad (5.39)$$

$$x_{43} - x_{33} \geq 8 \quad (5.40)$$

$$x_{53} - x_{43} \geq 2 \quad (5.41)$$

$$x_{63} - x_{53} \geq 4 \quad (5.42)$$

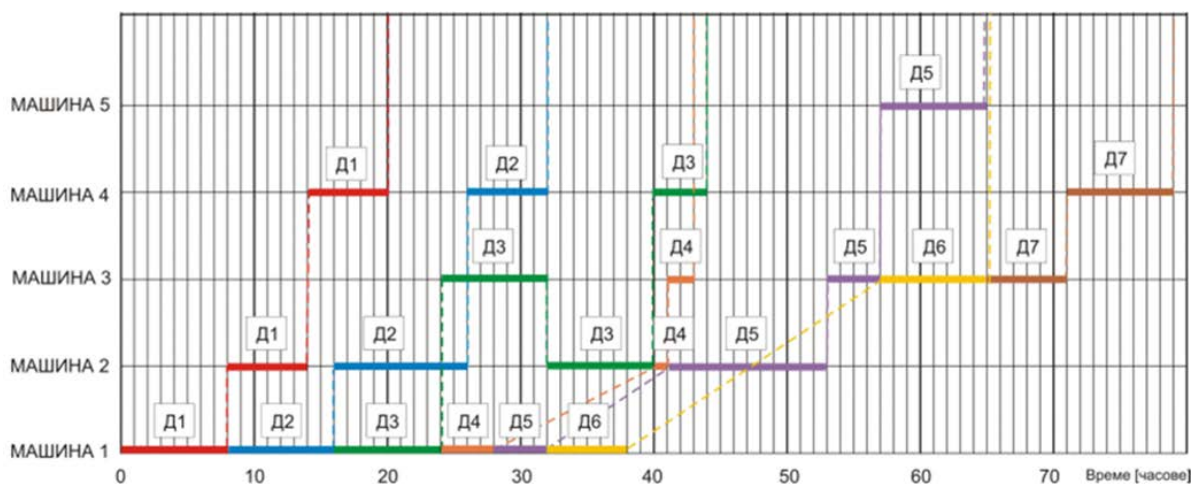
$$x_{73} - x_{63} \geq 8 \quad (5.43)$$

$$x_{24} - x_{14} \geq 6 \quad (5.44)$$

$$x_{34} - x_{24} \geq 6 \quad (5.45)$$

$$x_{74} - x_{34} \geq 4 \quad (5.46)$$

Получените резултати от решаването на задачата определят началните моменти на започване на обработката на отделните детайли на всяка машина, както и крайните моменти на завършване на обработката на всеки детайл (Фиг. 5.1).



Фиг.5.1. Оптимално разписание при последователност за обработка
 $D1 \rightarrow D2 \rightarrow D3 \rightarrow D4 \rightarrow D5 \rightarrow D6 \rightarrow D7$

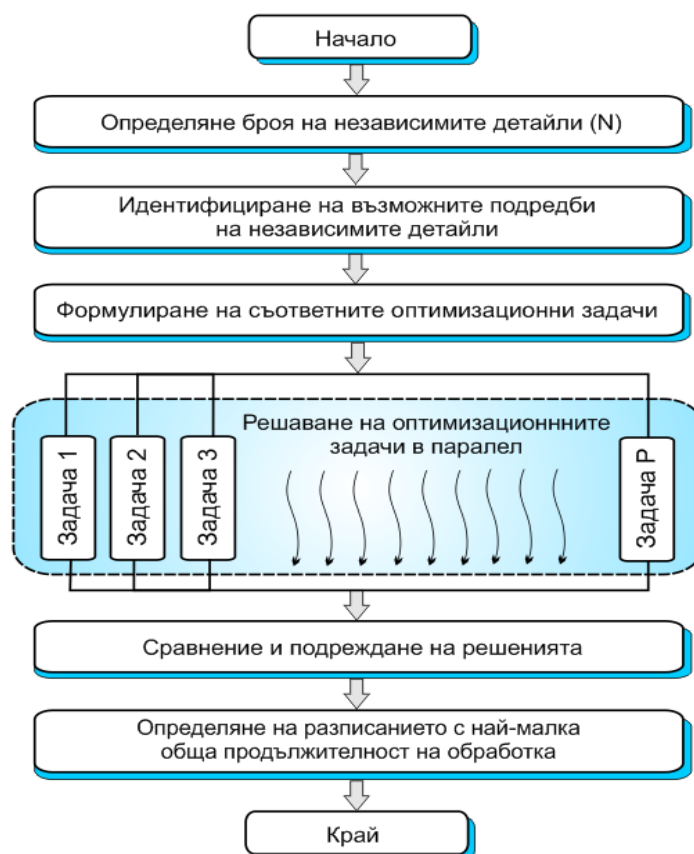
Общото време за обработката на тези детайли съвпада с крайния момент на завършване на операциите на последния детайл и се равнява на 79 часа. Иноформацията за моментите на започване и завършване на обработката на всеки детайл позволява да се оптимизира както времето за доставката им от леярната, така и престоя им в склада на фирмата.

5.3. Определяне на оптимални разписания за смесена обработка на детайли върху множество машини

При смесената (независима и зависима) обработка на детайли се предполага съществуването на определена последователност на обработката само на част от детайлите, докато за останалите детайли последователността за обработката не е фиксирана. В този смисъл, смесената обработка на детайли може да се разглежда като вариант на проблем, който е комбинация от познатите *job shop* и *open shop* проблеми. Оптималното разписание за този случай изисква определяне на оптималната последователност на обработка на всички детайли. Формулираният модел (5.1) – (5.5) може да се използва и в този случай, ако се приложи за всички възможни подредби на обработка на независимите детайли. При това се могат да се формулират отделни оптимизационни задачи за всяка възможна подредба, като решенията им са независими едно от друго. Този факт е използван при разработването на паралелен алгоритъм за определяне на възможните разписания при смесена обработка на детайлите, от които да се определи разписанието с минимална продължителност.

5.3.1. Паралелен алгоритъм за определяне на оптимално разписание с минимална продължителност

Предложен е паралелен алгоритъм за смесена обработка (независима и зависима) на детайли върху множество машини, илюстриран на Фиг. 5.2 (Borissova & Mustakerov, 2014). На първата стъпка от алгоритъма се определят независимите детайли. Ако съществува зависима обработка между някои детайли (напр. $D4 \rightarrow D5 \rightarrow D6$), то те се разглеждат като един независим детайл с обща продължителност и последователност, определена от зависимите детайли. На втората стъпка се определят всички възможни пермутации за обработка на независимите детайли. На следващата стъпка всяка възможна подредба на обработваните детайли се формализира в съответна оптимизиционна задача, съгласно описания модел (5.1) – (5.5).



Фиг. 5.2. Алгоритъм за определяне на оптимално разписание при смесена обработка на детайли

Тук е важно да се подчертае, че така формулираните оптимизационни задачи са независими една от друга, т.е. решението на всяка от тях не зависи от данните или от решението на друга задача. Това прави възможно тяхното решаване в паралел, както е показано на етапа 4. След решаване на задачите се прави сравняване на получените резултати и те се подреждат по времето за продължителност на обработката. На последната стъпка от алгоритъма, се избира разписанието, съответстващо на най-малкото време за обработка на детайлите.

5.3.2. Числено тестване

Разглежда се пример за обработка на 6 детайла с известни последователности и времена на операциите, които се извършват на четири различни машини. Последователностите на операциите за всеки детайл са зададени като: $D_1 \{O_{11}, O_{12}, O_{14}\}$, $D_2 \{O_{21}, O_{22}, O_{24}\}$, $D_3 \{O_{31}, O_{33}, O_{32}\}$, $D_4 \{O_{41}, O_{42}, O_{43}\}$, $D_5 \{O_{11}, O_{12}, O_{13}\}$ и $D_6 \{O_{61}, O_{63}\}$. Поради своята специфика, обработката на детайлите D_4 , D_5 и D_6 е зависима и трябва да се извърши в последователността $D_4 \rightarrow D_5 \rightarrow D_6$, докато останалите детайли могат да се обработват в произволна последователност. Детайлите не могат да се

припокриват на машините и един детайл не може да бъде обработван едновременно от две или повече машини. Всяка операция трябва да бъде изпълнена в непрекъснат интервал от време на определената машина. Целта е да се определи разписание, което да свежда до минимум общият престой на машините и определя минимално време за общата обработка на всички детайли.

Всички налични данни са обобщени в Таблица 5.2, където с O_{ij} са означени съответните операции за обработка на детайл i на машина j .

Таблица 5.2. Данни за обработваните детайли

Детайли	Операция	Време за обработка на М1	Време за обработка на М2	Време за обработка на М3	Време за обработка на М4
D_1	O_{11}	8			
	O_{12}		6		
	O_{14}				6
D_2	O_{21}	8			
	O_{22}		9		
	O_{24}				6
D_3	O_{31}	8			
	O_{33}			8	
	O_{32}		8		
D_4	O_{41}	4			
	O_{42}		2		
	O_{43}			2	
D_5	O_{51}	4			
	O_{52}		9		
	O_{53}			5	
D_6	O_{61}	6			
	O_{63}			4	

Като се използват данните от Таблица 5.2 и оптимизационния модел (5.1) – (5.5) и зададена последователост на обработка на детайлите в реда $D_1 \rightarrow D_2 \rightarrow D_3 \rightarrow (D_4 \rightarrow D_5 \rightarrow D_6)$, се формулира оптимизационна задача от вида:

$$\min(x_{1,end} + x_{2,end} + x_{3,end} + x_{4,end} + x_{5,end} + x_{6,end}) \quad (5.47)$$

при ограничения

- за последователност на операциите за всеки детайл

$$x_{1,2} - x_{1,1} \geq 8 \quad (5.48)$$

$$x_{1,4} - x_{1,2} \geq 6 \quad (5.49)$$

$$x_{1,end} - x_{1,4} \geq 6 \quad (5.50)$$

$$x_{2,2} - x_{2,1} \geq 8 \quad (5.51)$$

$$x_{2,4} - x_{2,2} \geq 9 \quad (5.52)$$

$$x_{2,end} - x_{2,4} \geq 6 \quad (5.53)$$

$$x_{3,3} - x_{3,1} \geq 8 \quad (5.54)$$

$$x_{3,2} - x_{3,3} \geq 8 \quad (5.55)$$

$$x_{3,end} - x_{3,2} \geq 8 \quad (5.56)$$

$$x_{4,2} - x_{4,1} \geq 4 \quad (5.57)$$

$$x_{4,3} - x_{4,2} \geq 2 \quad (5.58)$$

$$x_{4,end} - x_{4,3} \geq 2 \quad (5.59)$$

$$x_{5,2} - x_{5,1} \geq 4 \quad (5.60)$$

$$x_{5,3} - x_{5,2} \geq 9 \quad (5.61)$$

$$x_{5,5} - x_{5,3} \geq 4 \quad (5.62)$$

$$x_{5,end} - x_{5,5} \geq 5 \quad (5.63)$$

$$x_{6,3} - x_{6,1} \geq 6 \quad (5.64)$$

$$x_{6,end} - x_{6,3} \geq 4 \quad (5.65)$$

- за последователност на обработваните детайли

$$x_{2,1} - x_{1,1} \geq 8 \quad (5.66)$$

$$x_{3,1} - x_{2,1} \geq 8 \quad (5.67)$$

$$x_{4,1} - x_{3,1} \geq 8 \quad (5.68)$$

$$x_{5,1} - x_{4,1} \geq 4 \quad (5.69)$$

$$x_{6,1} - x_{5,1} \geq 4 \quad (5.70)$$

$$x_{2,2} - x_{1,2} \geq 6 \quad (5.71)$$

$$x_{3,2} - x_{2,2} \geq 9 \quad (5.72)$$

$$x_{4,2} - x_{3,1} \geq 8 \quad (5.73)$$

$$x_{5,2} - x_{4,2} \geq 2 \quad (5.74)$$

$$x_{4,3} - x_{3,3} \geq 8 \quad (5.75)$$

$$x_{5,3} - x_{4,3} \geq 2 \quad (5.76)$$

$$x_{6,3} - x_{5,3} \geq 5 \quad (5.77)$$

$$x_{2,4} - x_{1,4} \geq 6 \quad (5.78)$$

$$x_{i,j} \geq 0 \quad (5.79)$$

Приема се, че зависимата обработка на детайлите D_4 , D_5 и D_6 в последователност $D_4 \rightarrow D_5 \rightarrow D_6$ се свежда до независима обработка на един детайл, обобщен като D_4 . Тогава броят на подредбите на независимите детайли D_1 , D_2 , D_3 и D_4 е равен броя на пермутациите им $4! = 24$. За да се определи оптималното разписание, при независима обработка на детайлите D_1 , D_2 , D_3 и D_4 е необходимо да бъдат формулирани и решени 24 оптимизационни задачи. Например, за обработка на детайлите в последователност $D_1 \rightarrow D_3 \rightarrow D_2 \rightarrow (D_4 \rightarrow D_5 \rightarrow D_6)$ ограниченията (5.66) – (5.68) се трансформират като:

$$x_{3,1} - x_{1,1} \geq 8 \quad (5.66)$$

$$x_{2,1} - x_{3,1} \geq 8 \quad (5.67)$$

$$x_{4,1} - x_{2,1} \geq 8 \quad (5.68)$$

Целевата функция (5.47) и останалата част от ограничения остават същите.

За обработка на детайлите в последователност $D_2 \rightarrow D_1 \rightarrow D_3 \rightarrow (D_4 \rightarrow D_5 \rightarrow D_6)$ ограниченията (5.66) – (5.68) се трансформират като:

$$x_{1,1} - x_{2,1} \geq 8 \quad (5.66)$$

$$x_{3,1} - x_{1,1} \geq 8 \quad (5.67)$$

$$x_{4,1} - x_{3,1} \geq 8 \quad (5.68)$$

За обработка на детайлите в последователност $(D_4 \rightarrow D_5 \rightarrow D_6) \rightarrow D_1 \rightarrow D_2 \rightarrow D_3$ ограниченията (5.66) – (5.78) се трансформират като:

$$x_{5,1} - x_{4,1} \geq 4 \quad (5.66)$$

$$x_{6,1} - x_{5,1} \geq 4 \quad (5.67)$$

$$x_{1,1} - x_{6,1} \geq 6 \quad (5.68)$$

$$x_{2,1} - x_{1,1} \geq 8 \quad (5.69)$$

$$x_{3,1} - x_{2,1} \geq 8 \quad (5.70)$$

$$x_{5,2} - x_{4,2} \geq 2 \quad (5.71)$$

$$x_{1,2} - x_{5,2} \geq 9 \quad (5.72)$$

$$x_{2,2} - x_{1,2} \geq 6 \quad (5.73)$$

$$x_{3,2} - x_{2,2} \geq 9 \quad (5.74)$$

$$x_{5,3} - x_{4,3} \geq 2 \quad (3.75)$$

$$x_{6,3} - x_{5,3} \geq 5 \quad (5.76)$$

$$x_{3,3} - x_{6,3} \geq 4 \quad (5.77)$$

$$x_{2,4} - x_{1,4} \geq 6. \quad (5.78)$$

По подобен начин, всички възможни комбинации от последователности за

обработката на детайлите могат да бъдат изразени в 24 различни модификации на основната оптимизационна задача (5.47) – (5.79). Решаването на всички задачи, може да се направи паралелно, тъй като всички задачи са напълно независими една от друга. Резултатът от решаването на формулираните 24 задачи е илюстриран под формата на т. нар. *Гант-диаграми* на Фиг. 5.3.

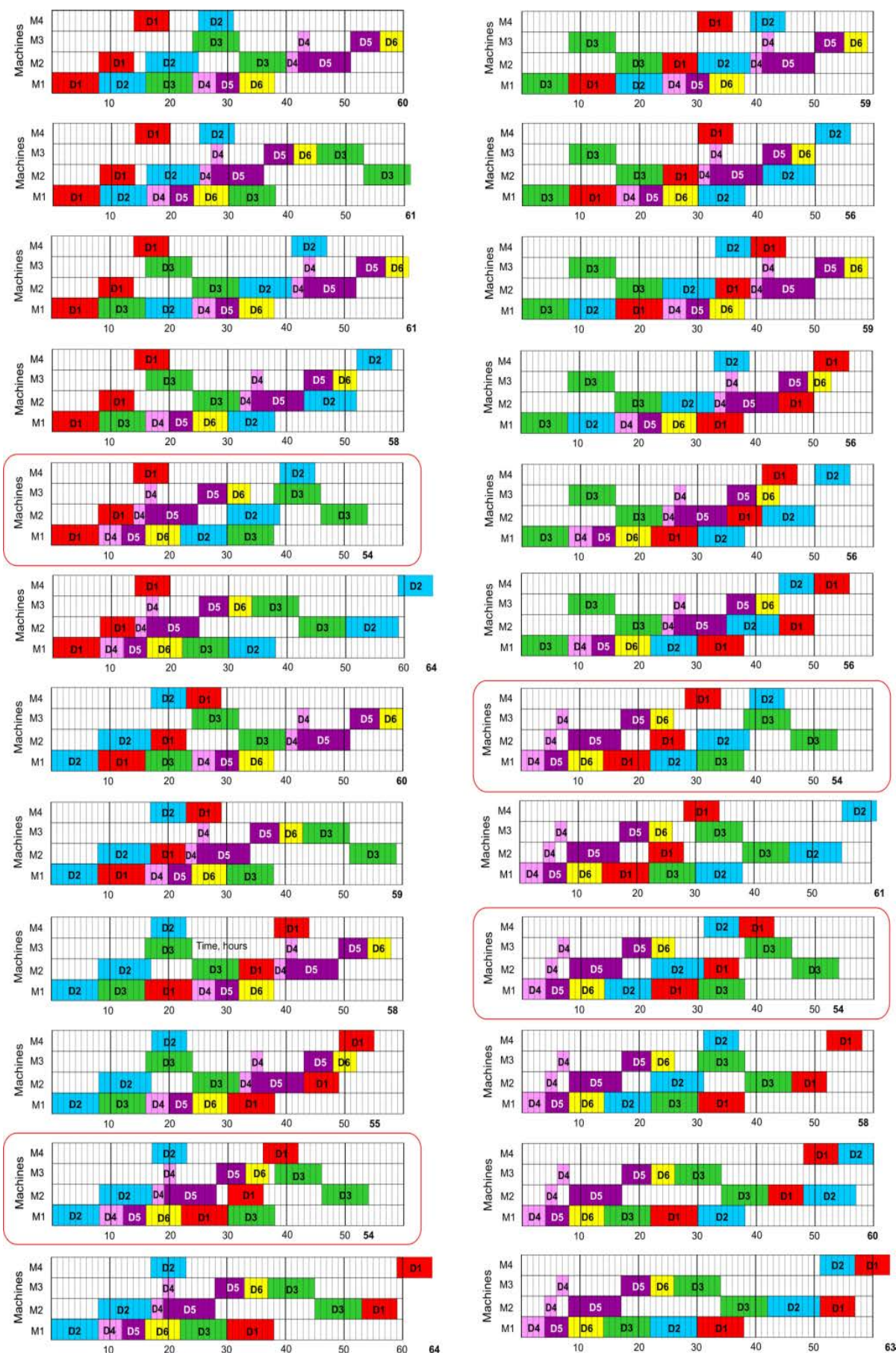
Формулираните оптимизационни задачи са решавани с LINGO вер. 12 на настолен компютър с Intel® Celeron® 2.93 GHz процесор, 2 GB оперативна памет и операционна система MS Windows XP ©, а времената за решаване са от порядъка на секунди. Резултатите от решаването на задачите показват, че съществува повече от една възможна последователност за обработка на детайлите, която води до една и съща минимална продължителност от 54 часа, както е показано в Таблица 5.3.

Таблица 5.3. Оптимални разписания с една и съща продължителност

Оптимална последователност на детайлите	Обща продължителност на обработката, часове
$D_1 \rightarrow (D_4 \rightarrow D_5 \rightarrow D_6) \rightarrow D_2 \rightarrow D_3$	54
$D_2 \rightarrow (D_4 \rightarrow D_5 \rightarrow D_6) \rightarrow D_1 \rightarrow D_3$	54
$(D_4 \rightarrow D_5 \rightarrow D_6) \rightarrow D_1 \rightarrow D_2 \rightarrow D_3$	54
$(D_4 \rightarrow D_5 \rightarrow D_6) \rightarrow D_2 \rightarrow D_1 \rightarrow D_3$	54

В такива случаи изборът на едно от тези разписания би могъл да се извърши по други обективни или субективни критерии.

Необходимостта от разглеждане на всички пермутации на възможните последователности за обработка на детайлите може да се разглежда като недостатък на предложения подход. Този недостатък може да се превърне в предимство. Изчислителните трудности, възникващи при използването на обобщена оптимизационна задача, разглеждаща всички комбинации на обработката едновременно могат да бъдат преодоляни като се използват съвременните технологии за паралелни изчислителни алгоритми – напр. CUDA технологията. При наличието на зависима обработка между част от детайлите, както е в повечето реални практически задачи, то много сложни (и на практика нерешими) проблеми могат да се декомпозират до голям брой, но лесно изчислими задачи.



Фиг. 5.3. Комбинации от възможни разписания за смесена обработка на детайлите

5.4. Модели за линейно разкрояване

Проблемът за оптимално разкрояване на материали има много приложения в индустрията. Например, често се налага да се изработят елементи с предварително зададени размери от стандартни заготовки материали, които след това се сглобяват някакви изделия – например, дограма за жилищни сгради. Постоянният ръст на цените, материалите и на енергията изисква минимизиране на производствените разходи за всеки елемент. С решаването на задачи за оптимално разкрояване на материали се цели да се постигне по-рационално използване на материалите. От математическа гледна точка те се свеждат до максимизиране използването на материалите или до минимизиране на отпадъците от материали. Тези проблеми са от съществено значение при планирането на производството в много отрасли като, металургия, пластмаси, хартия, стъкло, мебели, текстил, дограма и др. Интересът към тези проблеми е провокиран от много практически приложения и е предизвикателство за тяхното решаване (Mobasher & Ekici, 2013; Dikili и др., 2007; Cui & Lu, 2009; Cherri и др., 2009; Dikili и др., 2008). На пръв поглед тези проблеми са лесни за формулиране, но в същото време те са изключително трудни за решаване. Проблемите за разкрояването и опаковането принадлежат към класа на NP-трудните проблеми (Mukhacheva & Mukhacheva, 2006).

Повечето материали за случаите на линейно разкрояване се доставят на под формата на стандартни заготовки с определени дължини. На по-късен етап тези заготовки се разкрояват в зависимост от конкретната поръчка. Класическият проблем за линейно разкрояване разглежда разкрояването на стандартни заготовки на по-малки линейни отрязъци, като се цели минимизиране на отпадъците. Канторович първи формулира проблема за линейното разкрояване (Kantorovich, 1960), а Гилмор и Гомори (Gilmore & Gomory, 1961; Gilmore & Gomory, 1963) предлагат първата методология за решаването му. В повечето случаи, проблемът за ленийното разкрояване се формулира като целочислен линеен оптимизационен проблем, целящ намаляване на общото количество отпадъци, при удовлетворяване на зададено количество заготовки (Suliman, 2001). Известен е и подход, който разглежда проблема за линейното разкрояване на два етапа (Aktin & Ozdemir, 2009): на първия етап се извършва изчисляване на общия брой на елементи, които трябва да бъдат отрязани и се генерират комбинациите чрез евристична процедура, а на втория етап се определя оптималния план за разкрояване.

Всеки конкретен практически проблем за линейно разкрояване има своя собствена специфика. Това налага в много случаи известните модели да бъдат подходящо модифицирани или дори да бъдат разработени напълно нови методи, за да се решат поставените реални проблеми.

5.4.1. Обобщен модел за линейно разкрояване

Заготовките за едномерното линейно разкрояване обикновено се доставят с определена стандартна дължина L^* . В следствие тези заготовки се разкрояват, за да се получи необходимото количество елементи с определени дължини. За всеки конкретен проект елементите се различават по своя размер и количество. Въпросът, който възниква, е да се определи такава дължина на заготовките L , при която да се получат най-малки отпадъци, като се намерят и оптималните модели на разкрояване за всяка заготовка. Така описаният проблем може да бъде формулиран чрез подходящ математически модел, приложим за подобни класове задачи. За разлика от други подобни модели, с настоящия модел се цели да се определи не само оптималната дължина на заготовките, съобразно количеството на необходимите елементи, но и оптималните схеми за разкрояване за всяка отделна заготовка. Това означава, че е необходимо предварително да е известен броя N на заготовките. Броят на заготовките N може да се определи като се вземе предвид общата дължина за реализиране на поръчката и оптималната дължина L на заготовките. От друга страна, оптималната дължина L на заготовките ще стане известна едва след решаване на оптимизационната задачата. Този проблем може да бъде преодолян като се приеме, че търсената оптимална дължина L ще има някаква стойност, близка до стандартната дължина L^* в диапазона от $(L^* - \Delta_{min})$ до $(L^* + \Delta_{max})$. Имайки предвид този факт, броят на необходимите за поръчката заготовки N може да се изчисли като общата дължина на всички елементи, необходими за реализиране на поръчката, се раздели на стандартната дължина L^* и полученият резултат се закръгли до цяло число. Получената по този начин стойност на N , може да бъде използвана за формулирането на следния обобщен оптимизационен модел, приложим към класа от проблеми, касаещи линейното разкрояване (Mustakherov & Borissova, 2014):

$$\min \rightarrow \sum_{i=1}^N (L - L_i), i = 1, \dots, N \quad (5.80)$$

при ограничения

$$\forall i: L_i = \sum_{j=1}^J x_{ij} l_j, j = 1, \dots, J \quad (5.81)$$

$$\forall i: L_i \leq L \quad (5.82)$$

$$\forall j: \sum_{i=1}^N x_{ij} = k_{ij} \quad (5.83)$$

$$(L^* - \Delta_{min}) \leq L \leq (L^* + \Delta_{max}) \quad (5.84)$$

$$\forall j: x_{ij} = \begin{cases} \text{binary integer 0 or 1, if } N \leq k_{ij} \\ \text{integer, otherwise} \end{cases} \quad (5.85)$$

където N е броя на заготовките; L е оптималната дължина на заготовките; L_i е използваната дължина от всяка заготовка; l_j е дължината на елементите; x_{ij} са двоични целочислени променливи, присвоени на всеки елемент i и съответна заготовка j ; k_{ij} представлява необходимото количество за всеки елемент.

В така формулирания модел, целевата функция изисква намаляване на общите отпадъци от всички заготовки. Оптималните модели на разкрояване на всяка от заготовките се определят от променливите x_{ij} в (5.81). В зависимост от поръчката и броя на съответните елементи за разкрояване, както и от общия брой на използваните заготовки, променливите x_{ij} (5.85) могат да бъдат двоични целочислени променливи или цели числа. Например, ако броят на заготовките е по-малък от максимално допустимото количество за даден елемент, то тогава променливите x_{ij} трябва да се разглеждат като цели числа. Това представяне позволява да бъде определен повече от 1 елемент в рамките на една заготовка, за да се удовлетвори необходимия брой от елементи. Количеството от съответните необходими елементи се постига чрез (5.83). При това са зададени граници Δ_{min} и Δ_{max} за отклонението определената оптимална дължина на заготовките от стандартната дължина L^* .

5.4.2. Числено тестване

В (Korsemov и др, 2013) е направено проучване, в което е установено, че в практиката размерите на дограмите могат да се редуцират до определен брой унифицирани модули. Например, дограмата на среден по размер апартамент може да бъде реализирана от следните четири модула:

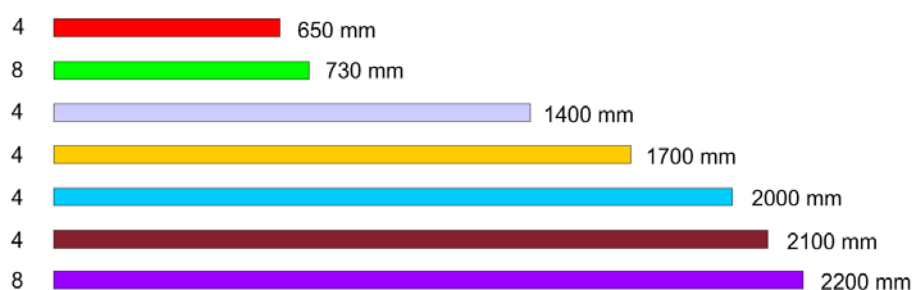
- модул 1 – за 4 врати с размери 2200 mm x 730 mm
- модул 2 – за 2 врати с размери 2000 mm x 650 mm
- модул 3 – за 1 прозорец с размери 1400 mm x 1400 mm
- модул 4 – за 2 прозореца с размери 1700 mm x 2100 mm

За разлика от класическия проблем за линейното разкрояване, при който се използват заготовки с определена дължина, тук проблемът се усложнява чрез необходимостта от едновременно определяне на оптималната дължина на заготовките и оптималните модели на разкрояване на всяка заготовка, при удовлетворяване на заявените количества елементи от поръчката. Разгледаният по-горе пример за производство на унифицирани модули от дограма определя необходимите елементи за тяхното реализиране, както и показано в Таблица 5.4.

Таблица 5.4. Количество и брой на необходимите елементи

Елементи, j	Дължина, l_j , mm	Количество, $k_{i,j}$
1	$l_1 = 650$	4
2	$l_2 = 730$	8
3	$l_3 = 1400$	4
4	$l_4 = 1700$	4
5	$l_5 = 2000$	4
6	$l_6 = 2100$	4
7	$l_7 = 2200$	8

Количеството и размера на елементите за производството на модулите са илюстрирани графично на Фиг. 5.4.



Фиг. 5.4. Необходими елементи

Данните от Табл. 5.4 са използвани за разработване на алгоритъм за определяне на оптималната дължина на заготовките и оптималните модели за разкрояване на всяка отделна заготовка. Алгоритъмът е съставен от следните стъпки:

1) Определяне на общата дължина на всички елементи за тази поръчка, отчитайки необходимия брой от всеки елемент: $L_{sum} = 54840 \text{ mm}$;

2) Определяне на необходимия брой на заготовките N . За конкретния пример той се определя като: $54840/6000 = 9.14$ и получената стойност се закръглява до $N = 9$. Задават се и допустимите отклонения на оптималната дължина на заготовките спрямо стандартната им дължина, като $\Delta_{min} = \Delta_{max} = 1000 \text{ mm}$.

3) Формулира се оптимизационната задача, чието решение ще определи оптималната дължина на заготовките и моделите за разкрояване на всяка една заготовка като:

$$\min \{(L - L_1) + (L - L_2) + (L - L_3) + (L - L_4) + (L - L_5) + (L - L_6) + (L - L_7) + (L - L_8) + (L - L_9)\} \quad (5.86)$$

при ограничения

$$x_{11}l_1 + x_{12}l_2 + x_{13}l_3 + x_{14}l_4 + x_{15}l_5 + x_{16}l_6 + x_{17}l_7 = L_1 \quad (5.87a)$$

$$x_{21}l_1 + x_{22}l_2 + x_{23}l_3 + x_{24}l_4 + x_{25}l_5 + x_{26}l_6 + x_{27}l_7 = L_2 \quad (5.87b)$$

$$x_{31}l_1 + x_{32}l_2 + x_{33}l_3 + x_{34}l_4 + x_{35}l_5 + x_{36}l_6 + x_{37}l_7 = L_3 \quad (5.87c)$$

$$x_{41}l_1 + x_{42}l_2 + x_{43}l_3 + x_{44}l_4 + x_{45}l_5 + x_{46}l_6 + x_{47}l_7 = L_4 \quad (5.87d)$$

$$x_{51}l_1 + x_{52}l_2 + x_{53}l_3 + x_{54}l_4 + x_{55}l_5 + x_{56}l_6 + x_{57}l_7 = L_5 \quad (5.87e)$$

$$x_{61}l_1 + x_{62}l_2 + x_{63}l_3 + x_{64}l_4 + x_{65}l_5 + x_{66}l_6 + x_{67}l_7 = L_6 \quad (5.87f)$$

$$x_{71}l_1 + x_{72}l_2 + x_{73}l_3 + x_{74}l_4 + x_{75}l_5 + x_{76}l_6 + x_{77}l_7 = L_7 \quad (5.87g)$$

$$x_{81}l_1 + x_{82}l_2 + x_{83}l_3 + x_{84}l_4 + x_{85}l_5 + x_{86}l_6 + x_{87}l_7 = L_8 \quad (5.87h)$$

$$x_{91}l_1 + x_{92}l_2 + x_{93}l_3 + x_{94}l_4 + x_{95}l_5 + x_{96}l_6 + x_{97}l_7 = L_9 \quad (5.87i)$$

$$L_1 \leq L \quad (5.88a)$$

$$L_2 \leq L \quad (5.88b)$$

$$L_3 \leq L \quad (5.88c)$$

$$L_4 \leq L \quad (5.88d)$$

$$L_5 \leq L \quad (5.88e)$$

$$L_6 \leq L \quad (5.88f)$$

$$L_7 \leq L \quad (5.88g)$$

$$L_8 \leq L \quad (5.88h)$$

$$L_9 \leq L \quad (5.88i)$$

$$x_{11} + x_{21} + x_{31} + x_{41} + x_{51} + x_{61} + x_{71} + x_{81} + x_{91} = 4 \quad (5.89a)$$

$$x_{12} + x_{22} + x_{32} + x_{42} + x_{52} + x_{62} + x_{72} + x_{82} + x_{92} = 8 \quad (5.89b)$$

$$x_{13} + x_{23} + x_{33} + x_{43} + x_{53} + x_{63} + x_{73} + x_{83} + x_{93} = 4 \quad (5.89c)$$

$$x_{14} + x_{24} + x_{34} + x_{44} + x_{54} + x_{64} + x_{74} + x_{84} + x_{94} = 4 \quad (5.89d)$$

$$x_{15} + x_{25} + x_{35} + x_{45} + x_{55} + x_{65} + x_{75} + x_{85} + x_{95} = 4 \quad (5.89e)$$

$$x_{16} + x_{26} + x_{36} + x_{46} + x_{56} + x_{66} + x_{76} + x_{86} + x_{96} = 4 \quad (5.89f)$$

$$x_{17} + x_{27} + x_{37} + x_{47} + x_{57} + x_{67} + x_{77} + x_{87} + x_{97} = 8 \quad (5.89g)$$

$$5 \leq L \leq 7 \quad (5.90)$$

$$x_{ij} - \text{двоични целочислени: } 0 \text{ или } 1 \quad (5.91)$$

Равенствата (5.87) в комбинация с неравенствата (5.88) определят оптималните модели за разкрояване за всяка отделна заготовка. Оптималните модели за разкрояване се определят по такъв начин, че да не се надвишава дължината на заготовките и същевременно да бъде удовлетворено необходимото количество от всеки елемент, изразено чрез равенствата (5.89).

Целевата функция (5.86) търси решение, което да свежда до минимум отпадъците от всички заготовки, вземайки предвид моделите за разкрояване за всяка една заготовка. В случая, оптималната дължина на заготовките се търси в интервала от 5 до 7 метра, изразено чрез неравенството (5.90). Променливите, определящи оптималните модели за разкрояване, са двоични цели числа, изразени чрез (5.91).

Решението на задачата (5.86) – (5.91) определя оптималната дължина на заготовките L за изпълнение на поръчката, която е 6550 mm . Общото количество на отпадъците е 4110 mm и се определя като сума от отпадъците за всяка една заготовка (Таблица 5.5), съобразно получените модели за разкрояване (Таблица 5.6).

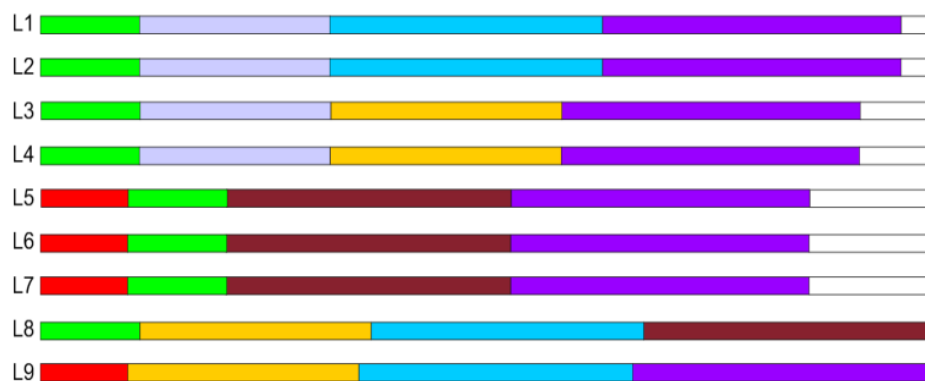
Таблица 5.5. Резултати за отпадъците за всяка заготовка

Оптималната дължина на заготовките L , mm	Общо количество на отпадъците, mm	Използвана дължина от всяка заготовка, mm	Отпадъци от всяка заготовка, mm
6550	4110	$L_1 = 6330$	220
		$L_2 = 6330$	220
		$L_3 = 6030$	520
		$L_4 = 6030$	520
		$L_5 = 5680$	870
		$L_6 = 5680$	870
		$L_7 = 5680$	870
		$L_8 = 6530$	20
		$L_9 = 6550$	0

Таблица 5.6. Оптимални модели за разкрояване за всяка заготовка

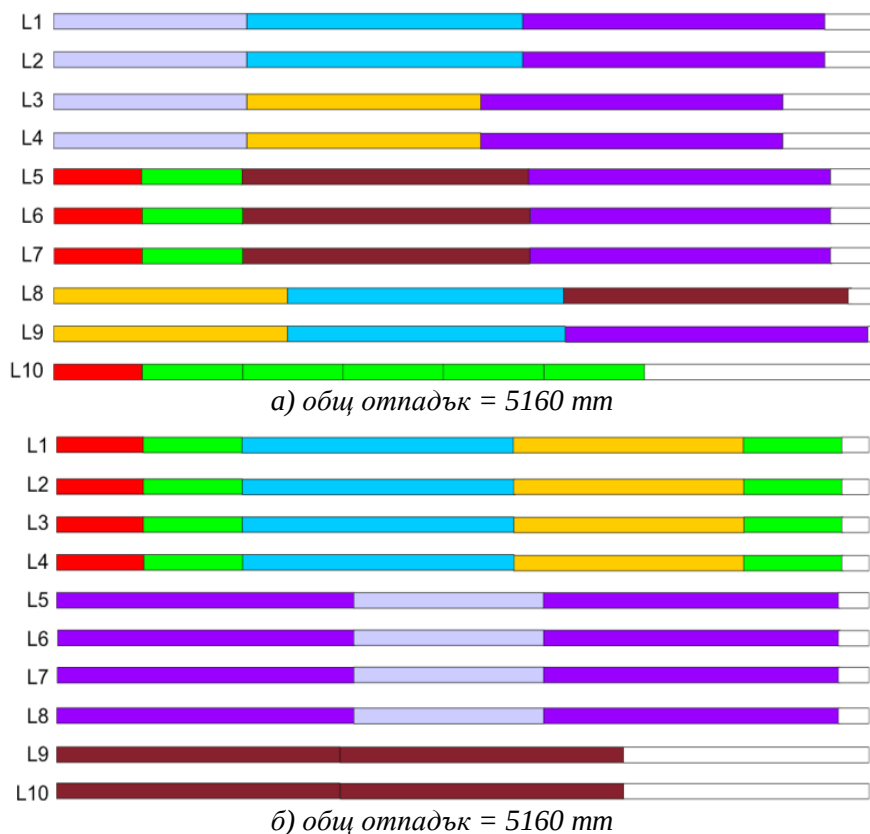
	Елемент1	Елемент2	Елемент3	Елемент4	Елемент5	Елемент6	Елемент7
L_1	0	1	1	0	1	0	1
L_2	0	1	1	0	1	0	1
L_3	0	1	1	1	0	0	1
L_4	0	1	1	1	0	0	1
L_5	1	1	0	0	0	1	1
L_6	1	1	0	0	0	1	1
L_7	1	1	0	0	0	1	1
L_8	0	1	0	1	1	1	0
L_9	1	0	0	1	1	0	1

На Фиг. 5.5. графично са илюстрирани оптималните модели за разкрояване за всяка от заготовките.



Фиг. 5.5. Оптимални модели за разкрояване на заготовките при използване на заготовки с оптимална дължина $L_{opt} = 6550\text{ mm}$

Така получените оптимални модели за разкрояване (Фиг. 5.5) са сравнени с възможни модели за разкрояване за същата поръчка при използване на стандартна дължина на заготовките от 6000 mm . На Фиг. 5.6 графично са показани два такива модела на разкрояване. Вижда се, че и в двата случая общите отпадъци са по-големи от отпадъците, получени при използването на предложени оптимизационен модел.



Фиг. 5.6. Модели за разкрояване при използване на заготовки със стандартна дължина $L = 6000\text{ mm}$

В резултат на използване на предложени модел и формулираната оптимизационна задача е определена оптимална дължина на заготовките от 6550 mm , което е с 550 mm увеличение, спрямо стандартната дължина. Така определената оптимална дължина води до намаляване на разходите, сравнено с използването на стандартна дължина на заготовките. Използване на стандартната дължина от 6000 mm не само увеличава общия отпадък, но също така увеличава броя на необходимите за доставка заготовки за изпълнение на поръчката. Това е особено важно при изпълнението на големи проекти, при които количеството на заготовките определя и съществени разходи за транспортиране.

Формулираната оптимизационна задача (5.86) – (5.91) използва 64 целочислени променливи и е решавана чрез средствата на LINGO solver на PC с 2.93 GHz Intel i3 процесор и 4 GB RAM. Времето за решаване е 1 час, 23 минути и 50 секунди.

В заключение може да се обобщи, че намаляването на загубите при разкрояването е един от основните проблеми при различните производства, включващи линейно разкрояване. Описаният подход може да допринесе не само за намаляване на общите загуби чрез оптимизиране на дължината на заготовките и моделите на разкрояване, но също така може да намали общото време за производството и свързаните с него разходи. Съществено предимството на предложения модел е възможността за едновременно определяне на оптималната дължина на заготовките и на оптималните модели на разкрояване за всяка заготовка.

5.5. Оптимален брой на персонал и график за работа

Управлението на човешките ресурси има ключова роля в управлението и организацията на работния процес. Реалните проблеми са свързани по един или друг начин с оптимизиране на ресурсите, независимо дали те са материали, финансови, или човешките ресурси. Ефективното управление на човешките ресурси е предпоставка за конкурентоспособност и цялостна организационна ефективност. Разработването на планове за човешките ресурси е в пряка връзка с възможностите за повишаване на производителността, тъй като по-ефективното им използване води до намаляване на потребностите. Неефективното управление на персонала, от друга страна, почти неизбежно ще доведе до по-ниски нива на качество и производителност (Griffin, 2007). Много често, определянето на необходимия персонал зависи от субективното решение на мениджъра, но по-добрия подход е то да се вземе с помощта на подходящи модели (Blochlinger, 2004). Управлението на човешките ресурси е свързано с проблемите на използването им за реализиране на набор от дейности за постигане на определена цел (Miwa & Takakuwa, 2010). Съществуват различни вариации на проблема за персонала, свързани със специфичните потребности на всяка организация. Един често срещан проблем е определянето на необходимия персонал за производства от типа *open-shop*. Поради тяхната широка гама от приложения, този тип производства са в основата на много проблеми, свързани с разписания, графици и оптимално използване на ресурси (машини или работни места), (Malapert и др., 2012; Błażewicza и др., 2004; Hasić и др., 2008). В класическият *open shop* проблем, набор от n работи трябва да бъдат извършени на M машини. Всяка работа се състои от операции, като всяка операция използва различна машина за определено време (Bai & Tang, 2013). Операциите могат да бъдат реализирани в произволен ред и работите са независими една от друга. При това, само една операция може да бъде изпълнена на всяка от машините.

Тук ще бъде разгледан проблема за определяне на оптимален брой персонал за реализиране на дадена задача, състояща се от изпълнение на n независими операции, извършвани на различни машини. Всяка операция с индекс j се извършва на машина със същия индекс. Целта е да се сведе до минимум времето на престой на персонала за реализирането на задачата. Този проблем се разглежда като комбинаторен оптимизационен проблем, за който е формулиран математически модел и алгоритъм за определяне на различни комбинации за броя на персонала и определяне на най-добрата от тях, спрямо зададен критерий за оптималност.

5.5.1. Обобщен модел за определяне броя на персонала

При формулирането на обобщения модел за определяне на оптимален брой на персонала се предполага наличието на една работа, състояща се от определен брой операции. Всяка операция се реализира на определена машина с известно време за обработка. Всяка машина се обслужва от различен брой оператори. Операциите могат да бъдат реализирани в произволен ред и последователност или паралелно, при наличието на незаета машина. Броят на машините от даден тип е равен на единица, а операторите са компетентни да обслужват всички машини.

С помощта на подходящ математически модел може да се направи оценка на различни кадрови политики, свързани с наличните машинни ресурси. За целта са въведени следните означения, използвани при формулирането на модела:

PT	– общо време за обработка,
PT_{min}	– минималното време за обработка на една дейност (операция),
PT_{max}	– максималното време за обработка на една дейност (операция),
M	– брой машини,
W	– брой оператори, обслужващи машина,
$SA = W*PT$	– площ в Гант-диаграма (заетост на операторите),
$J = \{1, 2, \dots, M\}$	– множество на индексите на операциите,
$h_j, j \in J,$	– време за обработка на j -тата машина,
$w_j, j \in J$	– брой оператори, обслужващи j -тата машина,
$W_{min} = \max \{w_j\}$	– минимален брой оператори, обслужващи машините,
$W_{max} = \sum w_j$	– максимален брой оператори за обслужване на машините (при случая на паралелна обработка),
$OA = \sum w_j h_j$	– обща площ на всички операции в Гант-диаграмата,
$SI = SA - OA$	– време на престой на персонала,
$l_{jm} = 0/1$	– променлива равна на 1 ако j -тия правоъгълник е разположен отляво на m -тия правоъгълник или 0, ако не е така,

- $b_{jm} = 0/1$ – променлива равна на 1 ако j -тия правоъгълник е разположен отдолу на m -тия правоъгълник или 0, ако не е така,
- x_j, y_j – координати на левия горен ъгъл на j -тия правоъгълник, представящ j -тата операция на j -тата машина.

С помощта на въведените означения е формулиран следния обобщен оптимизационен модел (Borissova & Mustakerov, 2013):

$$\min PT \quad (5.92)$$

при ограничения

$$PT \geq PT_{\min} \quad (5.93)$$

$$PT \leq PT_{\max} \quad (5.94)$$

$$x_j \geq 0, \text{ integer}, j \in J \quad (5.95)$$

$$y_j \geq 0, \text{ integer}, j \in J \quad (5.96)$$

$$x_j + h_j \leq PT \quad (5.97)$$

$$y_j + w_j \leq W \quad (5.98)$$

$$l_{jm} + l_{mj} + b_{jm} + b_{mj} = 1 \quad (5.99)$$

$$x_j + h_j \leq x_m + (1 - l_{jm})PT_{\max} \quad (5.100)$$

$$y_j + w_j \leq y_m + (1 - b_{jm})W \quad (5.101)$$

$$SA = W * PT \quad (5.102)$$

$$SI = SA - OA \quad (5.103)$$

$$PT_{\min} = \max\{h_j\} \quad (5.104)$$

$$PT_{\max} = \max \sum_{j=1}^J h_j \quad (5.105)$$

$$OA = \sum_{j=1}^J w_j h_j \quad (5.106)$$

В така представения модел всяка операция се интерпретира като правоъгълник в диаграма от тип Гант, с време за обработка по оста x в часове и брой оператори по оста y , необходими за изпълнението на оперецията. Целевата функция (5.92) минимизира общото време за реализиране на всички дейности. Променливата PT е ограничена от минималното (за сценарий на паралелна обработка) и максималното (за сценарий на последователна обработка) време (5.93) и (5.94), необходимо за обработката. Координатите x_j, y_j на горният ляв ъгъл на j -ия правоъгълник представлява началото на j -тата операция върху j -тата машина. Неравенства (5.97) и (5.98) не позволяват на правоъгълниците в Гант-диаграмата да излязат извън допустимата област, докато равенство (5.99) изразява факта, че всеки правоъгълник може да бъде само в една позиция спрямо друг правоъгълник – от ляво, от дясно, от горе или от долу. Чрез

използването на неравенства (5.100) и (5.101) не се допуска припокриване на правоъгълниците помежду им. Така формулирания оптимизационен модел (5.92) – (5.103) цели определяне на минималното време за обработка при зададен брой оператори. Тъй като общата площ на операциите OA (5.103) е постоянна и общата площ на диаграмата SA може да бъде изчислена от (5.102), то непроизводителния престой на персонала SI за даден брой на операторите W може да бъде изчислен чрез (5.103).

За да се определи оптималния брой оператори по отношение на минималния престой на персонала, оптимизационният проблем (5.92) – (5.103) трябва да бъде решен за различен брой оператори, като се вземе предвид минимално и максимално допустимия брой оператори. След това може да бъде определена, комбинацията, съответстваща на минималния престой на персонала. За целта се използва алгоритъм, в който е интегриран така описания математически модел (5.92) – (5.103).

5.5.2. Алгоритъм за определяне на оптимален брой оператори

Реализирането на алгоритъма за оптимален брой на оператори по отношение на минимален непроизводителен престой следва следните стъпки (Фиг. 5.7):

Стъпка 1: Определяне на изходните данни:

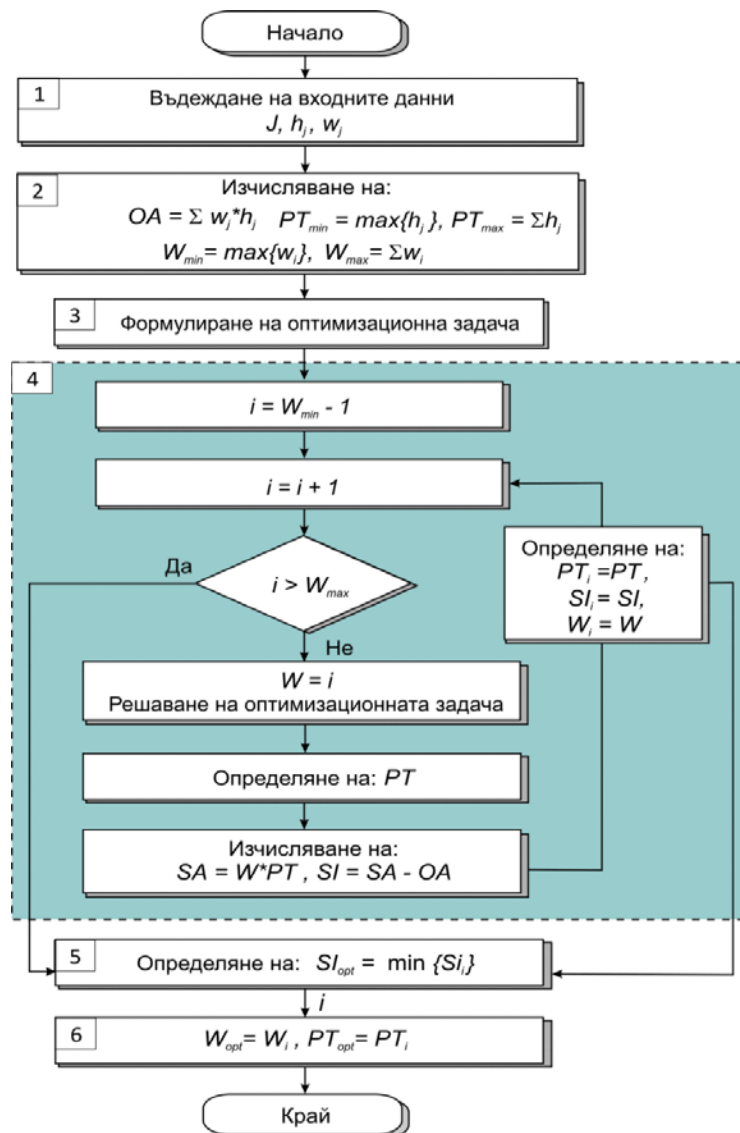
- наличен брой оператори, обслужващ машините,
- времената за обработка на операцията на съответната машина,
- брой на необходимите оператори за обслужване на всяка машина.

Стъпка 2: Изчисляване на:

- общия брой на необходимите човекочасове за всички операции $OA = \sum w_j h_j$,
- минимално време за обработка $PT_{min} = \max\{PT_j\}$ (при паралелна обработка),
- максималното време за обработка $PT_{max} = \sum PT_j$ (при последователна обработка),
- минималния брой оператори $W_{min} = \max\{w_j\}$ (при последователна обработка),
- максималния брой оператори $W_{max} = \sum w_j$ (при паралелна обработка).

Изчисляването на минималните и максималните стойности на PT и W се базира на две гранични условия за реализиране на задачата – обработка на операциите в паралел или последователна обработка на операциите.

Стъпка 3: Дефиниране на оптимизационната задача, използвайки формулирания модел (5.92) – (5.106).



Фиг. 5.7. Алгоритъм за определяне на оптимален брой оператори

Стъпка 4: Решаване на формулираната оптимизационна задача за различен брой оператори, започвайки от W_{min} и увеличавайки броя им с 1 до достигане на W_{max} . Определяне на SA_i и SI_i за всяка от задачите и изчисляване стойността на PT_i .

Стъпка 5: Определяне на:

- $SI_{opt} = \min \{ SI_i \}$,
- индекса i , съответстващ на SI_{opt}

Стъпка 6: Определяне на:

- $W_{opt} = W_i$
- $PT_{opt} = PT_i$

Крайният резултат от изпълнението на алгоритъма е получаване на оптимален брой оператори W_{opt} за конкретния проблем и определяне на оптималното време за обработка PT_{opt} , при определения оптимален брой оператори.

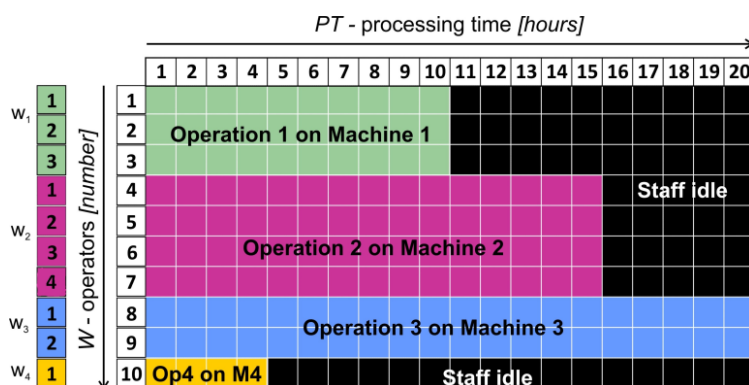
5.5.3. Числено тестване

Описаният модел (5.92) – (5.106) и алгоритъм (Фиг. 5.7) за определяне на оптимален брой оператори е тестван числено за реален проблем, чиито входни данни са показани в Таблица 5.7.

Таблица 5.7. Описание на проблема за оптимален брой оператори

	Операция 1 на машина 1	Операция 2 на машина 2	Операция 3 на машина 3	Операция 4 на машина 4
Време за обработка	10	15	20	4
Брой оператори, обслужващи машина	3	4	2	1
Брой машини	1	1	1	1

За производството на определено изделие е необходимо изпълнение на операции, всяка от които се извършва на определена машина. Операциите са независими една от друга. Всяка машина се обслужва от определен брой оператори, като всички оператори имат компетентността да работят на всички машини. От всяка машина има по един брой. Разпределението на броя оператори по съответните машини и необходимата продължителност за работа на всяка машина е графично илюстрирана на Фиг. 5.8.



Фиг. 5.8. Гант диаграма на изследвания проблем

Описаният модел и алгоритъм са използвани за формулиране на съответни оптимизационни задачи, използващи различен брой оператори W , както следва:

$W = W_{min}$, $W = W_{min} + 1$, $W = W_{min} + 2$, до $W = W_{max}$, където $W_{min} = 4$ и $W_{max} = 10$.

Получените стойности за общата продължителност на обработката, заедно с необходимия брой оператори и техния престой, са показани в Таблица 5.8.

Чрез Гант диаграмите са илюстрирани графично резултатите от решаването на съответните задачи (Фиг. 5.9). Фиг. 5.9а илюстрира случая на последователно изпълнение на операциите с минимален брой оператори $W_{min} = 4$, при което се получава максимално време за обработка $PT_{max} = 45$, а общото време на непроизводителния престой възлиза на 46 часа.

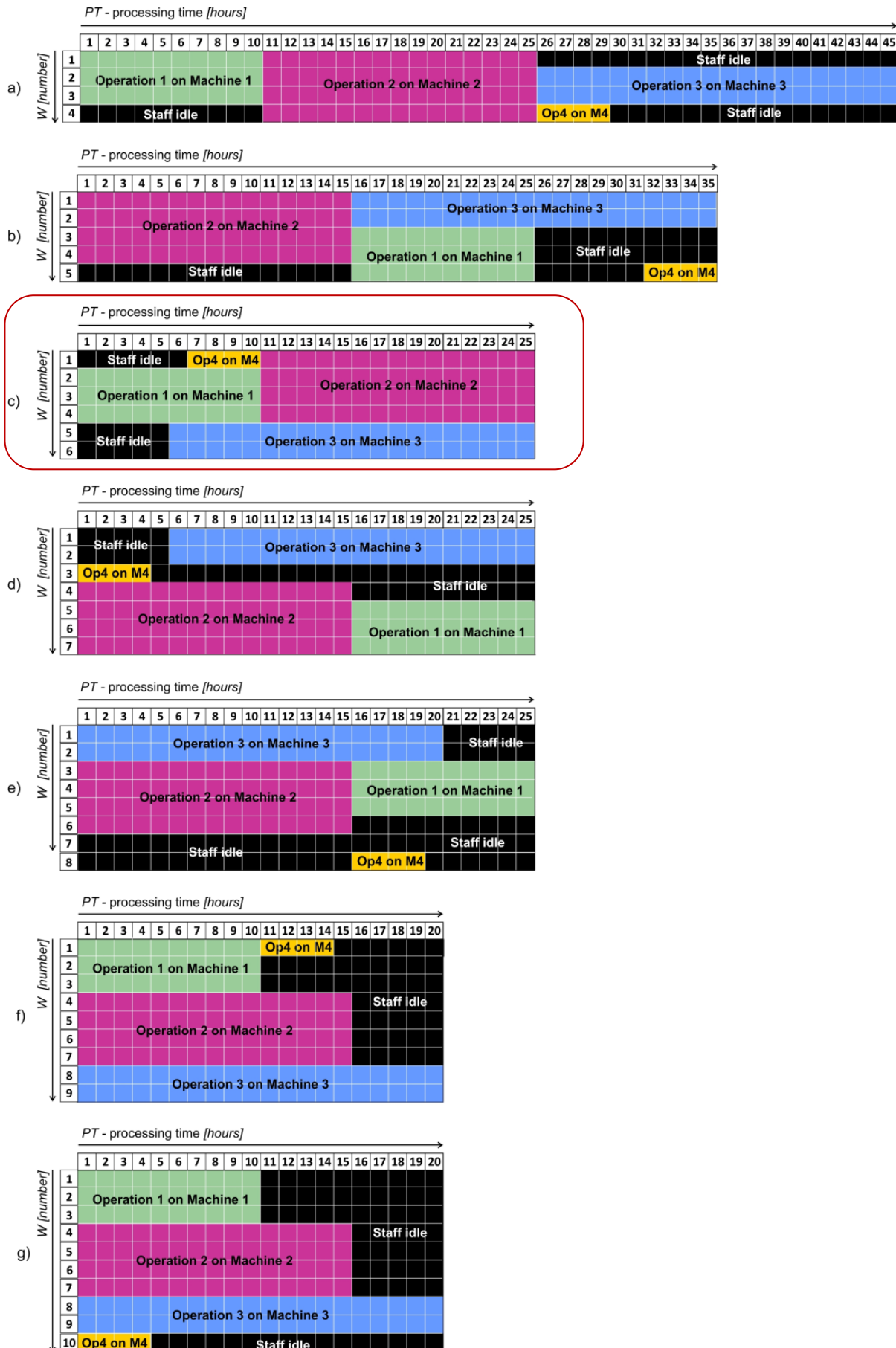
Таблица 5.8. Резултати от изпълнението на алгоритъма

Брой оператори W	Общо време за обработка PT , часове	Престой на персонала SI , човеко-часа
4	45	46
5	35	41
6	25	16
7	25	41
8	25	66
9	20	46
10	20	66

На Фиг. 5.9g е показан случай на паралелна обработка с максимален брой оператори $W_{max} = 10$, при което се получава минимално време за обработка $PT_{min} = 20$, а общото време на непроизводителен престой на операторите е 66 часа. Това са двата гранични случая на проблема за определяне на оптималния брой оператори. Оптималният брой оператори се получава като се търси минималното време за непроизводителен престой на операторите, което се постига при наемането на 6 оператора (Таблица 5.8 и Фиг. 5.9c), където общото време за непроизводителен престой възлиза на $SI = 16$ часа. Изчислителните времена на оптимизационните задачи са от порядъка на няколко секунди, при използване на LINGO v.12 на PC с Intel® Celeron® 2.80 GHz CPU, 1.96 GB оперативна памет за операционна система MS Windows XP ©.

Трябва да се отбележи, че използвания подход за моделиране може да доведе до решения, при които повече от една комбинация на разположение на правоъгълниците водят до идентични стойности на целевата функция. Например, на Фиг. 5.9 правоъгълникът, съответстващ на операция 4 на машина 4 може да има друго местонахождение в областта на незаеия персонал, без това да променя стойността на целевата функция. В някои случаи това може да бъде предимство, тъй като дава свобода за комбиниране с други графици въз основа на други субективни критерии.

При определянето на оптималния брой оператори могат да бъдат използвани други критерии, като например минимално време за обработка. Също така може да времето за време за обработка да е фиксирано и да се търси оптималния брой оператори за реализиране на задачата. Използваният икономически критерий за минимален простой на операторите може да бъде модифициран с подходящи корекции, така че вземе предвид и други изисквания при определянето на оптималния брой оператори.



Фиг. 5.9. Графично представяне на получените резултати

Глава 6:

Модели и алгоритми за оптимално управление при предсказващо поддържане

6.1. Въведение

В тази глава са описани предложените математически модели и алгоритми за управление на предсказващо поддържане на инженерни системи при: 1) определяне на оптимална стратегия, базирана на оценки за приходите и разходите, и определяща ремонт или замяна на машина като цяло или на отделни нейни компоненти, 2) определяне на оптимална стратегия, отчитаща условията на неопределеност, 3) определяне на оптималното разположение на зададен брой сензори и за едновременно определяне на разположението и броя на сензорите за нуждите на мониторинга при предсказващо поддържане. Получените резултати са публикувани в статии D4, D12, D13, D17, D18.

Технологичното развитие води до повишаване сложността на производствените машини и системи. Съвременната индустрия все повече се нуждае от висока надеждност, ниска степен на риск за околната среда и по-голяма безопасност на човешкия живот, при осигуряване на максимална продуктивност. Следователно, превенцията на аварии и ранното откриване на зараждащите се повреди на машините, би подпомогнало увеличаването на полезния срок на експлоатация на оборудването. Напоследък използването на съвременни информационни и комуникационни технологии за горната цел се изразява чрез термина *e-поддръжка*. Е-поддръжката се стреми да избегне повредите чрез навременно откриване на потенциални повреди, на базат на постоянен мониторинг на състоянието. Това позволява в реално време да се диагностицира предстояща повреда и да се прогнозира бъдещото състояние на оборудването, с цел вземане на решение за извършване на техническо обслужване (Nielsen и др., 2012). Този подход се използва, за да се намали несигурността при дейностите по поддръжка и поддръжката да се осъществява в съответствие с реалната необходимост от ремонт на оборудването. Решенията за техническа поддръжка зависят

от действително измерените отклонения на параметрите на системата и прогнозиране на тенденцията за влошаване на техническото ѝ състояние (Mahadevan и др., 2010). Основната идея е предварително да се открие възможността за поява на неизправност, с което да се предотвратят критични повреди на цялата система и по този начин да се намалят общите разходи за поддръжка. Това става чрез наблюдение на съответни параметри и техники за анализ на сигналите, и сравняване на стойностите с тези при нормални условия (Ferreira и др., 2009; Bouvard и др., 2011). Също така, е-поддръжката трябва да бъде съгласувана с правилата за стратегически решения на предприятието (Moble, 2010). Те определят не само основните тенденции на поддръжка, но и нейните цели, и следователно са доминиращи за всички процеси по поддържане, интегрирани в процесите на предприятието.

Съвременните технологии, свързани със събиране, пренос и обработка на данните, с цел последващо използване за поддръжка на различни системи, стават все по-важни (Gandhare & Akarte, 2012; Akhshabi, 2011; Cavalcante и др., 2010). Е-поддръжката може да се разглежда по четири различни начина: 1) стратегия за техническо обслужване (т.е. метод на управление), 2) план за техническото обслужване, 3) тип на техническото обслужване и 4) подпомагане на техническото обслужване (Ginevicius, 2011). Е-поддръжката като вид техническо обслужване описва заместването на традиционната поддръжка с технологична поддръжка, базирана на състоянието на системата, получено от мониторинга и на прогностични функции (Rigopoulos и др., 2010). Чрез мониторинга на системата могат да бъдат прогнозирани предстоящи проблеми и аварийни ситуации, като по този начин се избегнат потенциално възможни аварии. Като информационна база за прогнозиране на бъдещи повреди могат да бъдат използвани и статистическите данни за оборудването, но получаването на актуалната информация за състоянието води до получаване на по-точни прогнози (Peng и др., 2010).

6.2. Определяне на оптимална стратегия, базирана на оценките на приходите и разходите

Предсказващото поддържане определя състоянието на техническа система в процеса на експлоатация и предсказва необходимостта от извършване на ремонтни дейности. Постоянният мониторинг на състоянието на системите позволява ранно откриване на възможни повреди и е пряко свързано с експлоатационните разходи и с необходимостта от инвестиции. При непрекъснатото производство тези разходи биха

могли да се намалят чрез минимален престой на машините, свързан със замяна само на отделни дефектирали части и недопускане на аварийни ситуации, водещи до по-големи щети и съответни разходи (Nameed и др., 2009). Използването на сензори за мониторинг играе ключова роля в контрола и откриването на повреди на системите. Използването на голям брой сензори за мониторинг осигурява голям обем данни, но това е свързано с разходи за закупуване и монтаж на тези сензори. Поради това, от съществено значение е разположението на оптимален брой сензори за реализиране на ефективен мониторинг на системата (Li и др., 2011). Това води до увеличен интерес към разработване на методи за определяне на броя и местоположението на сензорите. Оптимизацията на разположението на сензорите има съществен ефект върху системата за наблюдение. Тъй като структурното наблюдение на състоянието на системата се основава на данните, получени от сензорите, то крайната цел е да се използват минимален брой сензори, поставени на правилните места, така че загубата на данни да бъде минимална и наличието на структурни дефекти да може да се идентифицира ефективно.

Основната цел на мониторинга е да превърне наличните данни за състоянието на системата в полезна информация. Това изисква данните да се събират и оценяват посредством подходящи методи и да се избират най-подходящите алтернативни стратегии за управление на поддръжката. Технологичният напредък в управлението на информацията прави изпълнима и целта за цялостната поддръжка.

6.2.1. Еднокритериални модели за определяне на оптимална стратегия, базирана на оценката за приходи и разходи за предсказващо поддържане

Откриването на повреди и диагностиката в ранните стадии на увреждане е необходимо условие за предотвратяване неправилното функциониране и възможните аварийни ситуации по време на работа. За значително увеличаване на ползите могат да се използват различни техники за оптимизация на работата на предприятието чрез подобряване използването на ресурсите на различни равнища на процеса на вземане на решения. Мониторингът на състоянието позволява предотвратяване на повреди, допринася за предотвратяването на непланирани спирания и дава възможност за оптимизиране на ресурсите за поддръжка, като поддръжката се планира когато е необходимо, на базата на анализ на данните за състоянието. Всички тези дейности помагат да се предскажат потенциалните откази на оборудването, преди те да се появят. С други думи, поддръжката е стратегия за вземане на решения в реално време

въз основа на диагноза за предстоящи проблеми и прогнозиране на бъдещото състояние на оборудването (Groba и др., 2007). Различни подходи за вземане на решения са предложени за избор на стратегия за поддръжка (Mathew др., 2006; Chowdhury & Akram, 2011; Li и др., 2005; Muller и др., 2008; Lee и др., 2006). За да се осигури оптималност при вземането на решение за това, кога да се извърши техническа поддръжка на оборудването, е необходимо да има съгласуваност с цялостната работа на производство. Ето защо е необходима допълнителна информация за наблюдение и събиране на данни за текущото състояние на оборудването. Необходимо е интегрирането на данни от различни източници, за да бъде направено правилното решение за подходящото техническо обслужване. Изборът на подходящо техническо обслужване е сложен проблем за вземане на решения. Програмата за техническо обслужване трябва да се съобрази с поддръжката както на различните части на системата, така и на системата като цяло. Следователно, оптималното техническо обслужване може да бъде разглеждано като комбинаторен оптимизационен проблем.

При проблемите, свързани с вземането на решения следва да се отбележи, че във всички случаи съществува лице, вземащо решение (ЛВР). Търсенето на оптимално решение изисква съществуването на функция, която трябва да бъде оптимизирана. Ако $\mathbf{x}$ е вектор, съдържащ променливите, определящи решението, то оптимизационният проблем може да бъде представен като:

$$\min/\max f_i(\mathbf{x}), i = (1, 2, \dots, n) \quad (6.1)$$

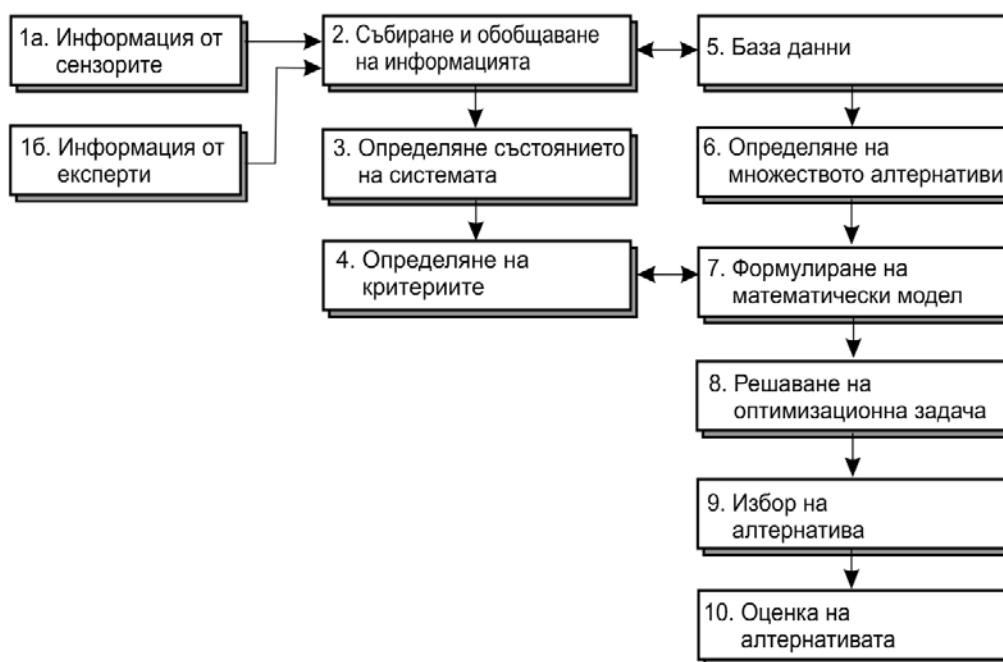
при ограничения

$$g_j(\mathbf{x}) \leq 0, j = (1, 2, \dots, J). \quad (6.2)$$

където $\mathbf{x} = \{x_1, x_2, \dots, x_p\}$ – вектор на променливите.

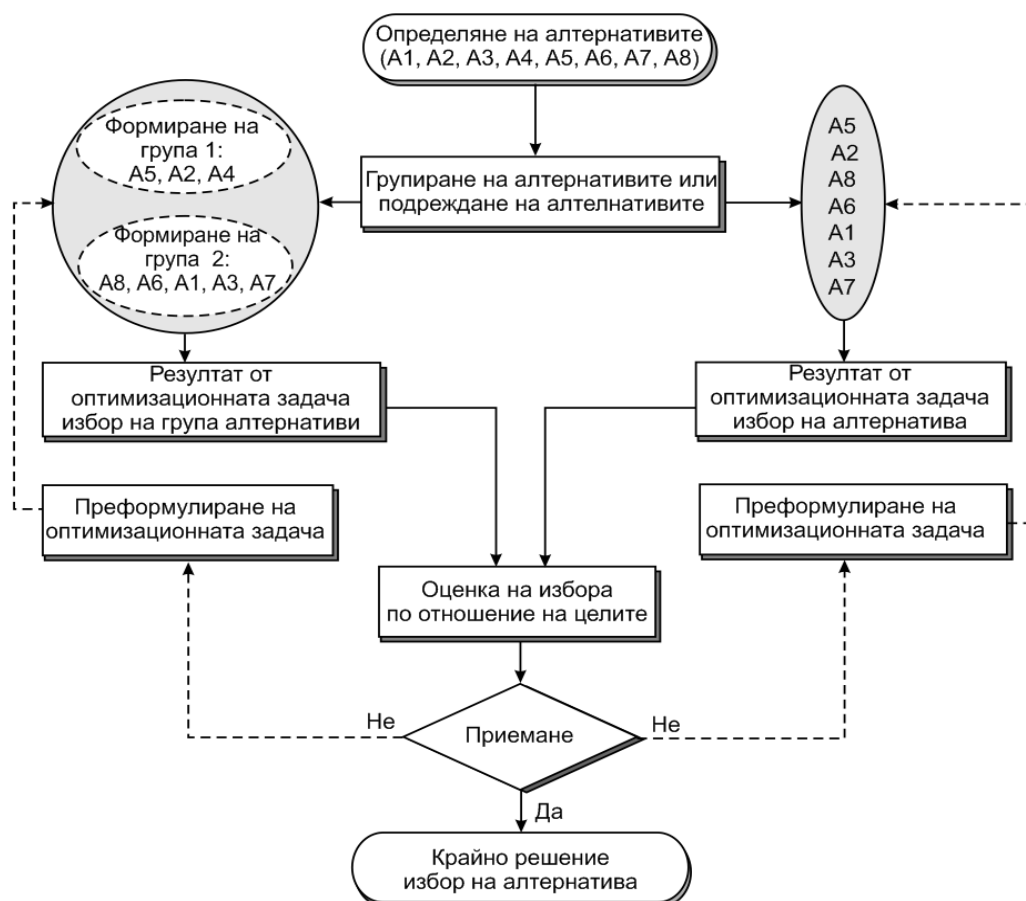
Техниките за предсказващо поддържане могат да се разглеждат като неделима част от вземането на решения. За реализиране на предсказващо поддържане е необходима актуална информация за състоянието на системата, определяне на целите, съставяне на съответна задача за вземане на решение, решаване на тази задача, оценка за ефективността на решението и участие на ЛВР на различните етапи на процеса. Това може да бъде илюстрирано, както е показано на Фиг. 6.1 (Mustakerov & Borissova, 2013).

При реализацията на алгоритъма на етапите 1а и 1б се събира информация за състоянието на системата, като етапът 1а осигурява обективна, количествена информация от апаратни средства (сензори), а на етап 1б се реализира събиране на информация от специалистите, обслужващи системата.



Фиг. 6.1. Обобщена схема на предсказващо поддържане

На етап 2 се извършва обобщаване и обработка на получената информация за целите на реализирането на етап 3 и 4, а именно – съставяне на описание на системата и определяне на целите и критериите за ефективност. Обобщената и обработена информация на етап 2 се използва също на етап 5 за създаване на база от данни за управляемата система. Резултатите от изпълнението на етап 4 и 5 се използват за дефиниране на математически модел на системата и формулиране на съответна многокритериална задача за вземане на решение (етап 6). Решаването на тази задача на етап 7 чрез интерактивно участие на ЛВР, определя Парето-оптимални алтернативи. На етап 8 ЛВР избира една от тези алтернативи. На етап 9 избраната алтернатива (решение) се изпълнява и на етап 10 се извършва оценка на изпълнението на решението по отношение на очакваните цели и резултати от управлението на системата. Ако получените резултати се разминават с очакваните, се извършва настройка на математическия модел на системата, преформулира се съответната многокритериална задача за вземане на решение и се повтарят етапите 7, 8, 9, и 10. Поради априорно непълната информация за управляемата система, в някои случаи постигането на желаните цели за управление на системата изисква и преформулиране на целите и критериите от етап 4. Реализирането на етапите, свързани с определяне на алтернативите и избор на най-подходяща алтернатива (етапи 6, 7, 8, 9 и 10) са илюстрирани на Фиг. 6.2.



Фиг. 6.2. Определяне на алтернативите и избор на най-подходяща алтернатива

В резултат на решаването на съответни оптимизационни задачи, се избира група от алтернативи или една алтернатива и се извършва оценка на избора по отношение на поставените цели. За решаване на подобни проблеми могат да бъдат използвани различни подходи, реализирани в подходящи програмни средства. Един широко използван подход в областта на икономиката и управлението на ресурсите е *анализът на разходите и ползите* (*cost-benefit analysis*) (Campbell & Brown, 2005). Този анализ дава оценка на ползата от увеличаване на инвестициите за ремонт/замяна и неговото намаляване при увеличаване на разходите за превенция от повреди.

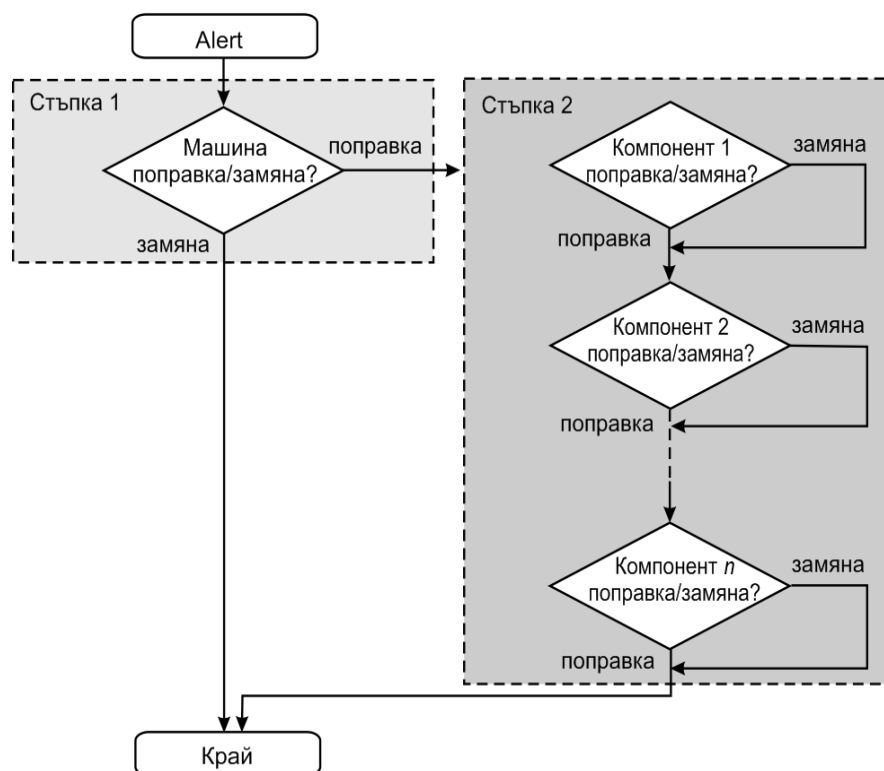
Тъй като предсказващото поддръжане е стратегия, която може значително да намали разходите за поддръжка, то използването на подходяща оценка за разходите и ползите ще допринесе за определяне на най-подходящата стратегия. За целите на предсказващото поддръжане са предложени два типа оценки за разходите и ползите (Mustakerov & Borissova, 2013):

$$CBE_{repair} = \frac{\alpha C_{repair_component}}{\beta P_{remaining}} \quad (6.3)$$

$$CBE_{new} = \frac{C_{new_component}}{P_{remaining}} \quad (6.4)$$

където $C_{repair_component}$ е цената за ремонт на компонента, $C_{new_component}$ е цената на замяната с нов компонент, α е коефициент, показващ колко пъти компонентът е повреден, $P_{remaining}$ е печалбата от оставащото полезно време на работа на компонента, β е коефициент, който показва дали компонентът е ремонтиран ($0 < \beta < 1$) или е заменен с нов ($\beta = 1$).

За определянето на оптимална стратегия за техническо обслужване (замяна или ремонт) могат да се формулират подходящи оптимизационни модели като се вземат предвид предложените оценки за разходите и ползите (6.3) и (6.4). Тези математически модели ще се базират на максимизиране на оценките за разходите и ползите, защото максималните стойности на разходите и ползите определят най-лошия случай (много повече разходи отколкото ползи) за съответното техническо обслужване. Въпросът, който възниква при появата на алармиращ сигнал при мониторинга, е какво решение трябва да се вземе. Процесът на вземане на решение може да се реализира на два етапа. Алгоритъмът на този процес на вземане на решения е показан на Фиг. 6.3 (Mustakerov & Borissova, 2013).



Фиг. 6.3. Алгоритъм на вземане на решения при предсказващо поддръжане

На първия етап се разглеждат две алтернативни – да се ремонтира машината или да смени машината като цяло. Ако решението от първия етап е да смени машината с нова, то процесът за вземане на решение свършва. Ако решението от етап 1 е ремонт на машината, следващият въпрос е – какво да се прави с всеки отделен компонент на машината – да се ремонтира или да се замени?

Етап 1: Отговорът на въпроса, дали да се поправи или да смени машината като цяло се получава в резултат на решаване на следната оптимизационна задача (Mustakerov & Borissova, 2013):

$$\max \left(\sum_{i=1}^n x CBE_i^{repair} + \sum_{i=1}^n y CBE_i^{new} \right) \quad (6.5)$$

при ограничения

$$x + y = 1, \quad x, y \in \{0, 1\} \quad (6.6)$$

където CBE_i^{repair} представлява оценката на разходите и ползите за ремонт на i -ия компонент, CBE_i^{new} е оценката на разходите и ползите за смяна на i -ия компонент, x, y са двоични целочислени променливи присвоени към всяка алтернатива.

Етап 2: Стратегията за ремонт/замяна за всеки отделен компонент се определя от решаването на следния оптимизационен проблем (Mustakerov & Borissova, 2013):

$$\max \left(\sum_{i=1}^n x_i CBE_i^{repair} + \sum_{i=1}^n y_i CBE_i^{new} \right) \quad (6.7)$$

при ограничения

$$\forall i \in \{1, 2, \dots, n\}: \sum_{i=1}^n x_i + y_i = 1, \quad (6.8)$$

$$x_i, y_i \in \{0, 1\} \quad (6.9)$$

където CBE_i^{repair} представлява оценката на разходите и ползите при ремонт на i -ия компонент, CBE_i^{new} е оценката на разходите и ползите при смяна на i -ия компонент, x_i, y_i са двоични целочислени променливи присвоени към всеки компонент на машината.

При този етап, стратегията за техническо обслужване разглежда всеки компонент като независима променлива за разлика от предходния етап, където техническото обслужване на машината се разглежда като едно цяло.

6.2.2. Числено тестване

Предложеният подход е илюстриран на базата на реален пример за предсказващо поддържане, включващ 4 компонента. Оценките за разходите и ползите при ремонт или замяна на компонентите, съгласно условията (4.3) и (4.4) са представени в Таблица 6.1.

Таблица 6.1. Оценки на алтернативите

Алтернатива	Costs-Benefit Estimations				
	Компонент 1	Компонент 2	Компонент 3	Компонент 4	Компонент 5
ремонт	0.85	0.70	0.65	0.90	0.80
замяна	0.75	0.80	0.75	0.85	0.70

Дефинираните оптимизационни модели (6.5) – (6.6) и (6.7) – (6.9) са използвани за формулиране на оптимизационни задачи, съответстващи на описаните етапи 1 и 2 от алгоритъма, за определяне на най-добрата алтернатива.

Етап 1: Оптималната стратегия за ремонт или за смяна на машината като цяло се определя от оптимизационния модел (6.5) – (6.6), в който са използвани определените оценки на разходите и ползите, показани в Таблица 5.1. Формулираната задача за конкретния пример има вида:

$$\max \{x(0.85 + 0.70 + 0.65 + 0.90 + 0.80) + y(0.75 + 0.80 + 0.75 + 0.85 + 0.70)\} \quad (6.10)$$

при ограничения

$$x + y = 1, \quad x, y \in \{0, 1\} \quad (6.11)$$

Резултатите от решаването на оптимизационна задача (6.10) – (6.11) са показани в Таблица 6.2.

Таблица 6.2. Резултати от решението на етап 1

Стойност на целевата функция	х (ремонт)	у (замяна)	Решение
3.90	1	0	ремонт

Съгласно получените резултати от решаването на съответната задача (Таблица 4.2), оптималната стратегия за техническо обслужване е ремонт на машината. Това означава, че трябва да бъде изпълнен етап 2 от предложения алгоритъм.

Етап 2: Като се използват определените оценки на разходите и ползите от Таблица 6.1 и математическият модел (6.7) – (6.9), се формулира оптимизационна задача за конкретния пример от вида:

$$\max \{(0.85x_1 + 0.75y_1) + (0.70x_2 + 0.80y_2) + (0.65x_3 + 0.75y_3) + (0.90x_4 + 0.85y_4) + (0.80x_5 + 0.70y_5)\} \quad (6.12)$$

при ограничения

$$\forall i \in \{1, 2, \dots, 5\}: \sum_{i=1}^5 x_i + y_i = 1, \quad x_i, y_i \in \{0, 1\} \quad (6.13)$$

Резултатите от решаването на оптимизационна задача (6.12) – (6.13), определящи съответните стратегии за всеки компонент, са показани в Таблица 6.3.

Таблица 6.3. Резултати от решението на етап 2

Алтернатива	Costs-Benefit Estimations				
	Компонент 1	Компонент 2	Компонент 3	Компонент 4	Компонент 5
x_i (ремонт)	1	0	0	1	1
y_i (замяна)	0	1	1	0	0

Резултатите от изпълнението на първия етап от предложения алгоритъм показват, че оптималната поддръжка е ремонт на машината (Табл. 6.2). Това е определено на базата на формулираната и решена оптимизационна задача (6.10) – (6.11). Формулираният математически модел (6.7) – (6.9) и съответната оптимизационна задача (6.12) – (6.13) се решава на етап 2, в зависимост от полученото решението, определено на етап 1. Алтернативата за подходяща поддръжка (ремонт или подмяна) може да бъде оценена чрез използване на концепцията за анализ на разходите и ползите, и посредством подходящо формулирани оптимизационни модели. Формулирани са два различни оптимизационни модела за определяне на подходяща стратегия за обслужване 1) определяне на стратегия за машината като цяло; 2) определяне на стратегия за всеки отделен компонент на машината. Тези оптимизационни модели са използвани в разработения алгоритъм на вземане на решения на подходяща поддръжка.

Числените резултати показват приложимостта на предложения подход и ефективността на описаните математически модели за определяне на оптимална стратегия за обслужване. Използването на такава оптимални стратегии допринася за максимизиране на доходността чрез намаляване на оперативните и производствените разходи. Определянето на оптимална стратегия за поддръжка на базата на наблюдение на състоянието, като се отчитат и условията на неопределеност, подобрява нивото на надеждност и води до намаляване на ненужните разходи за техническа поддръжка.

6.3. Определяне на оптимална стратегия, отчитаща условията на неопределеност

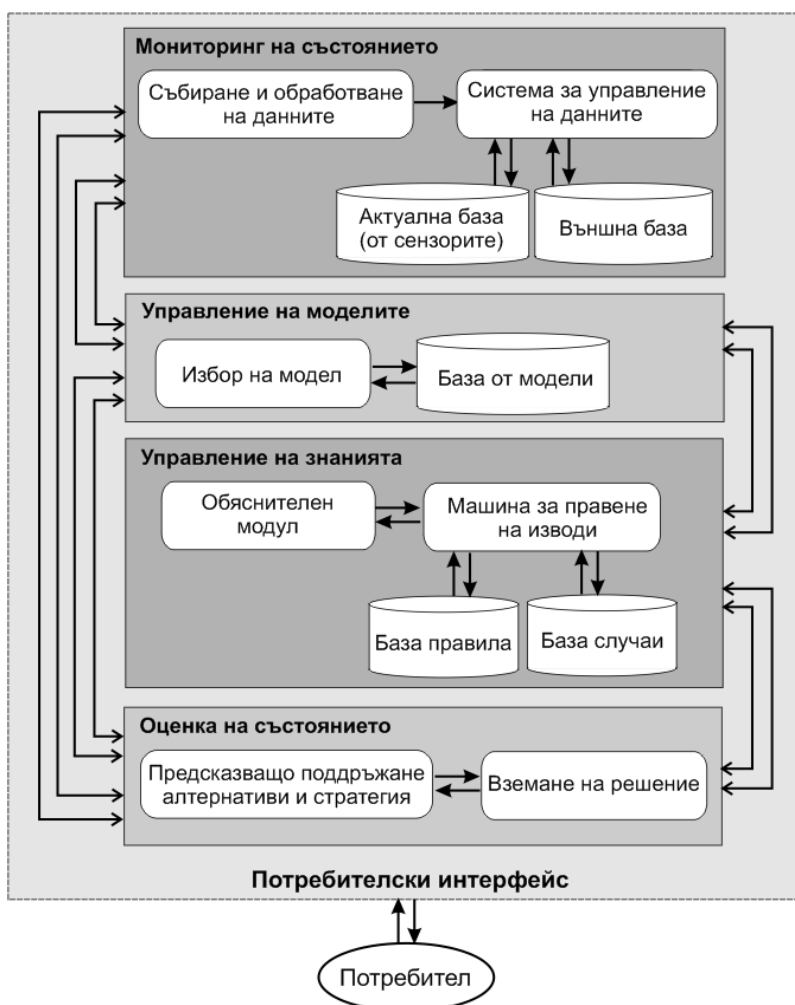
Всяка инженерна система подлежи на износване с течение на времето. Този естествен процес на влошаване на технологичните параметри в процеса на експлоатация може да бъде ускорен от външни фактори или да бъде забавен чрез мониторинг, диагностика и вземане на съответни мерки. За целта са необходими стратегии за вземане на адекватни решения в условията на неопределеност или непълна информация за състоянието на системата.

Системата за е-поддръжка има за цел да препоръча действия по поддръжката, основавайки се на информацията от мониторинга – събиране на данни, обработка на данни и подпомагане вземането на решения за поддръжката. Това означава, че за да получат коректни данни за състоянието на системата, събраните данни трябва да бъдат обработени чрез използването на подходящи методи и алгоритми (Гарванов, 2014). Събраните от мониторинга данни представляват измервания, които имат отношение към състоянието на машината (напр. данни от вибрации). Мониторингът на състоянието е подходящ за управление на инженерни съоръжения, тъй като той е в състояние да идентифицира потенциални симптоми на повреди и могат да се предприемат съответни действия, преди да се наложи аварийно прекъсване. В зависимост от състоянието на отделните елементи може да бъде взето решение, дали да се извърши ремонт или да се подмени съответния елемент.

Използването на подходяща система за е-поддръжка може да улесни анализа и да ускори вземането на решения. Едно съвременно направление в тази област е създаването на интелигентни системи, които извличат голямо количество данни и могат да анализират и да предлагат подходящи решения (Sgurev & Hadjiski, 2003; Sgurev & Kojnov, 2008).

Първата стъпка при обработката на данни е верифициране на данните, така че да бъдат ефективни при използването им за анализ и моделиране. Грешките в данните от наблюдение на състоянието могат да бъдат причинени и от самия сензор, ако има неизправност в сензора. Диагностика и прогнозиране са двете основни техники, използвани за извършване на ефективни политики за подпомагане вземането на решения при поддръжката. Диагностика използва както статистически подходи и изкуствен интелект, така и подходи, базирани на модели. Основната цел на поддръжката е да се направи оценка на състоянието на оборудването и да се прогнозира предстоящи повреди чрез събиране, обработка и анализ на данните.

Предложена е архитектура на система за подпомагане вземането на решения за целите на предсказващото поддръжане, реализирана от четири основни слоя – Фиг. 6.4 (Borissova & Mustakerov, 2012).



Фиг. 6.4. Архитектура на система за подпомагане вземането на решения за целите на предсказващо поддръжане

Първият слой представлява модул за контрол на състоянието, който събира и обработва данни от сензорите, като ги сравнява с подобни данни. Обработката на сигналите от сензорите е една от най-важните функции на системи за мониторинг с цел предсказващо поддръжане. Първичните входни данни се коригират (отстраняват се шумове и се нормализират при необходимост), за да се извлече полезната информация. Модулът за наблюдение на състоянието има за цел да събира данни за състоянието от сензорите и да ги сравнява с други подобни данни за състоянието. Резултатите от обработка на сигналите се събират в база от данни за допълнителна обработка. За сравняване на получените резултати с подобни такива, могат да се използват външни източници за данни. Вторият слой съдържа модул за избор на модел и база с модели. Някои модели могат да оценят времето до повреда на машина, както и да прогнозират евентуална повреда. Прогностичните модели се фокусират върху търсенето на комбинация от фактори, тясно свързани с крайния резултат. Точните прогностични

модели са базирани на алгоритми, които са способни да оценят честотата на бъдещи повреди или скоростта на деградация. Разработването на стратегии за моделиране на прогностични оценки включва различни подходи, включително надеждност на системата, вероятностна оценка въз основа на жизнения цикъл и поддръжката, оптимизация на много критерии при неопределеност, както и интеграция на данните от мониторинга с цел управление на жизнения цикъл на системата. Подсистемата за управление на знанията е разположена в третия слой на системата за подпомагане на предсказващото поддръжане. Предсказващото поддръжане, базирано на мониторинг е специфична област, която се характеризира с наличието на минали информация за възможните проблеми, тяхната диагностика, оценка на ситуациите и ефективни решения. Поради това е удачно да бъде използвана експертна система, която да предлага съответни решения и начини за действия. Това се реализира от система за управление на знанията, база от знания (модели, базирани на правила и модели, базирани на случаи), машина за правене на изводи и подсистема за обяснения. Поради това, експертната система трябва да включва: експертни знания за специфичната предметна област, обикновено чрез съхраняването и представянето им като правила или под други форми (бази от знания); възможност за отделяне на общото знание от подробностите за конкретния разглеждан случай, към който се прилага базата от знания (конкретен контекст); възможност за ясно отделяне знанията от механизма за правене на заключения (машина за изводи); интерактивен (диалогов) потребителски интерфейс за осигуряване на възможности за обяснения, оценки и въпроси на потребителя и да осигурява съвет или решения в областта на експертизата. Правенето на изводи, на база на предишен опит, решава нови проблеми на база на съществуващи подобни случаи чрез сравняване и адаптиране към новите ситуации. Тези характеристики са подходящи за диагностика, прогнозиране на предсказващото поддръжане. Заедно с това, експертната система трябва да бъде в състояние да обработва и непълна информация, както и да се обучава, т.е. да модифицира базата от знания или машината за изводи.

Оценката на състоянието в зависимост от използваните данни е разположена в четвъртия слой на системата. Основната идея за оценка на състоянието се фокусира върху определяне състоянието на компонентите на системата и подпомагане определянето на подходяща поддържаща стратегия. Данните от всички слоеве се използват, за да бъде оценено състоянието на оборудването.

Прогнозирането на влошаване на параметрите на инженерната система във времето и вземането на съответни решения, е свързано със значителна неопределеност (Yang и др., 2004). В условия на неопределеност ЛВР има представа за целите, които трябва да бъдат постигнати, но информацията за алтернативите и за бъдещите събития е непълна и са необходими други специфични подходи за използване на неопределените (размити) възприятия (Jotsov & Sgurev, 2004). В подобни случаи се използва теорията на размитите множества в комбинация с многокритериално вземане на решения (Peneva & Popchev, 2009; Peneva & Popchev, 2010). При разглеждането на проблема на вземане на решения в условията на неопределеност, би следвало да се изброят изходните предпоставки:

- във всички случаи съществува ЛВР,
- вземането на оптимално решение предполага съществуването на функция f , която трябва да бъде оптимизирана:

$$x_{opt} = \max f(x) \quad (6.14)$$

където: x е множеството от анализирани алтернативи, а x_{opt} е оптималното решение.

Предпочитанието на ЛВР се разглежда като функция, наричана *функция на предпочитание на полезността* $f(x)$, върху множеството от възможни алтернативни решения:

$$x = \{x_1, x_2, \dots, x_m\} \quad (6.15)$$

При състояние на околната среда $s = \{s_1, s_2, \dots, s_n\}$, изборът на решение в условия на неопределеност се налага в случай на липса на информация за заобикалящата среда и обекта или при недостиг на информация за тях. Ако се приеме, че ЛВР може да оцени полезността E_{ij} на отделните алтернативи x_i ($i = 1, 2, \dots, m$), в някакви условни единици, то могат да се използват някои от известните подходи, използващи следните принципи.

Принцип на Валд: За всяко алтернативно решение x_i ($i = 1, \dots, m$) се намира най-лошият изход, равен на $\min E_{ij}$ ($j = 1, \dots, n$). След това се определя това алтернативно решение, за което величината $\min E_{ij}$ ($j = 1, \dots, n$) е максимална (Harrell, 2001):

$$x_{opt} \Rightarrow \max \min E_{ij} (i = 1, 2, \dots, m), (j = 1, 2, \dots, n) \quad (6.16)$$

Този принцип, наричан още *най-добрият от лошите*, се определя като презастрахован или песимистичен и се използва, когато на околната среда се гледа като на активен, злонамерен противник.

Принцип на Сейвидж: Този принцип е известен, като стремеж да се сведат до минимум възможните *загуби*, т.е. търси се най-малка загуба на ефективност. Под загуби се имат предвид загубите, в резултат на пропуснати възможности, които се изчисляват по формулата:

$$R_{ij} = |E_{ij} - \max E_{ij}| \quad (6.17)$$

т.е. за всяка загуба се намират най-благоприятните решения и неговото значение се изважда от всичките елементи на стълба, съответстващ на това състоянието на средата (Леготин, 2008). Оптимална ще бъде минималната стойност на загубите R_i сред всички алтернативи, т.е.

$$x_{opt} \Rightarrow \min \max R_{ij} (i = 1, 2, \dots, m), (j = 1, 2, \dots, n) \quad (6.18)$$

Според този критерий, в условията на неопределеност би било добре да се избира тази стратегия, при която *загубата има най-малка стойност в най-неблагоприятната ситуация*.

Принцип на Лаплас: В основата на този принцип лежи схващането, че щом няма информация за определяне на едно състояние като по-вероятно от друго, то оптималното решение може да се определи по формулата:

$$x_{opt} \Rightarrow \max_i \left(\sum_{j=1}^n \frac{E_{ij}}{n} \right) \quad (6.19)$$

Според този принцип се приема, че *всички възможни състояния са равновероятни*, когато не разполагаме с друга информация.

Принцип на Хурвиц: Този принцип представлява опростен вариант на принципа на Лаплас. При известни вероятности на отделните състояния, взема се средно-аритметичното на резултатите от най-добрите решения (Дубров, и др., 2000). Ако съществува възможност да се определят теглата на най-лошото и на най-доброто решение, се използва тяхната претеглена средноаритметична стойност. Този принцип отчита както минималната, така и максималната печалба (съответно $\min E_{ij}$ и $\max E_{ij}$). Оптималното решение може да се определи по формулата:

$$x_{opt} \Rightarrow \max \{ \alpha \max E_{ij} + (1 - \alpha) \min E_{ij} \} \quad (6.20)$$

където: α – коефициент на оптимизъм ($0 < \alpha < 1$).

Коефициентът $\alpha = 0$ съответства на напълно антагонистична среда, $\alpha = 0.5$ съответства на равностойна среда по отношение на антогонистичността и дружелюбността. При стойност $\alpha = 1$ това съответства на максимално благоприятна

околна среда. При $\alpha=1$, принципът на Хурвиц може да бъде наречен принцип на антагонистичния оптимист

$$x_{opt} \Rightarrow \max_i \max_j E_{ij} . \quad (6.21)$$

За различните стойности на α , съответната полезност се изчислява като:

$$E_{ij} = \alpha \max_j E_{ij} + (1 - \alpha) \min_j E_{ij} . \quad (6.22)$$

6.3.1. Числено тестване

За да се илюстрира приложимостта на предложената система за подпомагане вземането на решения за целите на предсказващото поддръжане при отчитане условията на неопределеност, е използван числен пример. Производствена фирма е оборудвана с различни машини, които имат различни характеристики на надеждност, нивото на безопасност, възможност за повреда, както и различно влияние върху печалбите на компанията. Предоставени са три алтернативи на техническото обслужване (ремонт, замяна, не се прави нищо) за всяка от машините. Решението за избор на подходяща стратегия за поддръжка зависи от оценката на състоянието на машините (висока, средна, ниска). Оценките на полезността E_{ij} за съответните алтернативи са показани в Таблица 6.4 (Borissova & Mustakerov, 2013).

Таблица 6.4. Оценки на полезността за съответните алтернативи

Алтернатива	Оценки на полезността		
A1	60	30	65
A2	40	20	70
A3	80	10	-5

Използвайки критерия на Валд се избира минималната стойност от всеки ред на таблицата, а след това се определя максималната стойност измежду тях, което води до следното решение:

$$x_{opt} = \max(30, 20, -5) = 30 \quad (6.23)$$

Загубите в резултат на пропуснати възможности, според принципа на Савидж, се изчисляват както следва:

$$r_{i1} = \max(60, 40, 80) - E_{i1} \quad (6.24)$$

$$r_{i2} = \max(30, 20, 10) - E_{i2} \quad (6.25)$$

$$r_{i3} = \max(65, 70, -5) - E_{i3} \quad (6.26)$$

Резултатите от изчисленията са представени в Таблица 6.5.

Таблица 6.5. Загуби и мини-макс по Савидж

Алтернатива	Загуби			Мини-макс
A1	20	0	5	20
A2	40	10	0	40
A3	0	20	75	75

Оптималната стратегия, според критерия на Савидж, се реализира от алтернатива A1, тъй като за нея се получава минимум на загубите в резултат на пропуснати възможности:

$$x_{opt} = \min(20; 40; 75) = 20 \quad (6.27)$$

Ако се приложи критерия на Лаплас, като решение се получава:

$$\begin{aligned} x_{opt} &= \max\left\{\left(\frac{1}{3}(60 + 30 + 65)\right), \left(\frac{1}{3}(40 + 20 + 70)\right), \left(\frac{1}{3}(80 + 10 - 5)\right)\right\} = \\ &= \max(51.6; 43.3; 28.3) = 51.6 \end{aligned} \quad (6.28)$$

Следователно и според критерия на Лаплас оптималната стратегия е също реализиране на алтернатива A1 (с очаквана полза от 51.6).

Използваният коефициент на оптимизъм, съгласно критерия на Хурвиц, има стойност 0.25. Решението определя като оптимална алтернатива отново алтернативата A1 (с очаквана полза от 37):

$$\begin{aligned} x_{opt} &= \max\{(0.2 * 65 + 0.8 * 30), (0.2 * 70 + 0.8 * 20), \\ &(0.2 * 80 + 0.8 * (-5))\} = \max(37; 30; 12) = 37 \end{aligned} \quad (6.29)$$

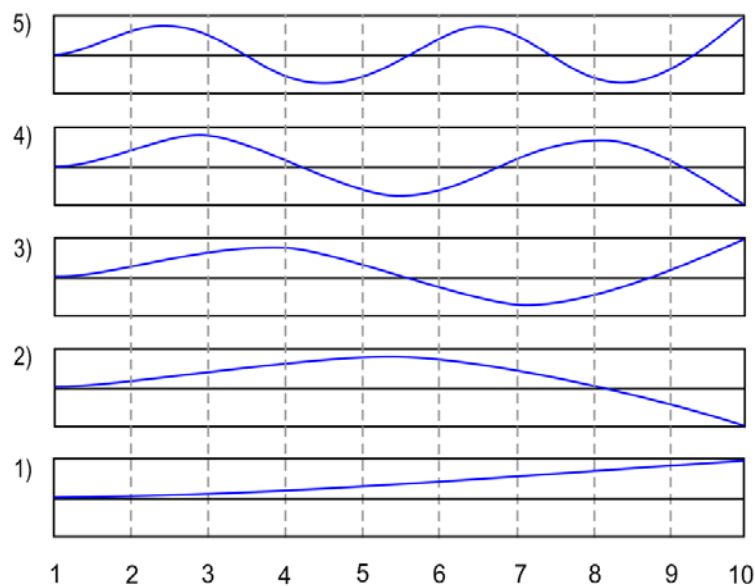
Всички тези примери показват, че с използването на подходящ критерий може да бъде определена най-добрата стратегия, която да подпомага вземането на решения, за целите на предсказващото поддръжане, при отчитане условията на неопределеност.

Същността на предложената концепция на система за подпомагане вземането на решения за целите на предсказващото поддръжане, е възможността за интегриране на текущата информация от мониторинга с подходящо моделиране и управление на знанията. Решенията за подходящо техническо обслужване се базират на актуалните данни от сензорите, подходите за отчитане на условията на неопределеност и с активното участие на ЛВР. За целта ЛВР оценява полезността на различните алтернативи, като се използват оптимизационни подходи. Използването на предложената концепция води до намаляване на производствените разходи, като техническото обслужване се извършва само тогава, когато това е необходимо, т.е осигурява производствени мощности и планирани спирания, с цел ремонт или замяна на дефектирали части. Всичко това води до значително повишаване на производителността и същевременно намаляване на разходите за поддръжка.

6.4. Определяне на оптималното разположение на сензори за мониторинг на състоянието

Проблемът за определяне на оптималния брой сензори, заедно с техните най-добри възможни места за разположение, играе важна роля при предсказващото поддържане. При оптималното разположение на сензорите е важно не само да се намерят най-добрите позиции на сензорите за конкретната задача, но също така да се оцени необходимия брой сензори за получаване на добри резултати от мониторинга (Li и др., 2004). Различни техники за оптимизация са разработени през последните десетилетия, включително евристични подходи и комбинаторна оптимизация. Състоянието на технологията за откриване на структурни дефекти са описани в (Doebbling и др., 1996; Farrar & Worden, 2007). За оптималното разположение на сензори може да се използва и неопределеност (Guratzsch & Mahadevan, 2006) или генетични алгоритми (Liu et al, 2008). Много от предложените методи третират и разположението на сензорите за мониторинг, свързан с откриването на дефекти (Kaveh и др., 2014; Wang и др., 2012; Yi и др., 2011; Worden & Burrows, 2001). Проблемът за оптималното разположение и проблемът за оптималния брой сензори са два отделни проблема. Знанията и опита на инженерите се комбинират с обработка на сигналите за правилното определяне на разположението на сензорите. При това, проблемът за определяне на оптималния брой сензори разчита изключително много на усъвършенстваните техники за обработка на сигнали (Staszewski & Worden, 2001).

Предсказващото поддържане, използващо мониторинг, се основава на динамична информация за откриване на възникнали дефекти. Измерванията на структурните вибрации могат да се използват за определяне разположението на сензорите, като се отчитат съответните динамични натоварвания. Структурата може да бъде опростена като система с определен брой стерени на свобода. Целта е да определят възможно най-много местоположения за позициониране на сензори, като колкото повече локации са определени, толкова по-детайлна информация за структурата на изследвания обект ще бъде получена. Използването на повече сензори е свързано с натоварване на самата структура и на практика е необходимо редуциране на броя сензори, които да бъдат разположени на подходящите места, така че най-добре да отчитат динамичните натоварвания. В резултат на подходящ анализ могат да се определят режимите на динамичните натоварвания. Например, за греда, закрепена неподвижно в единия край и подложена на променливи във времето натоварвания Фиг. 6.5 (Tejada, 2009).



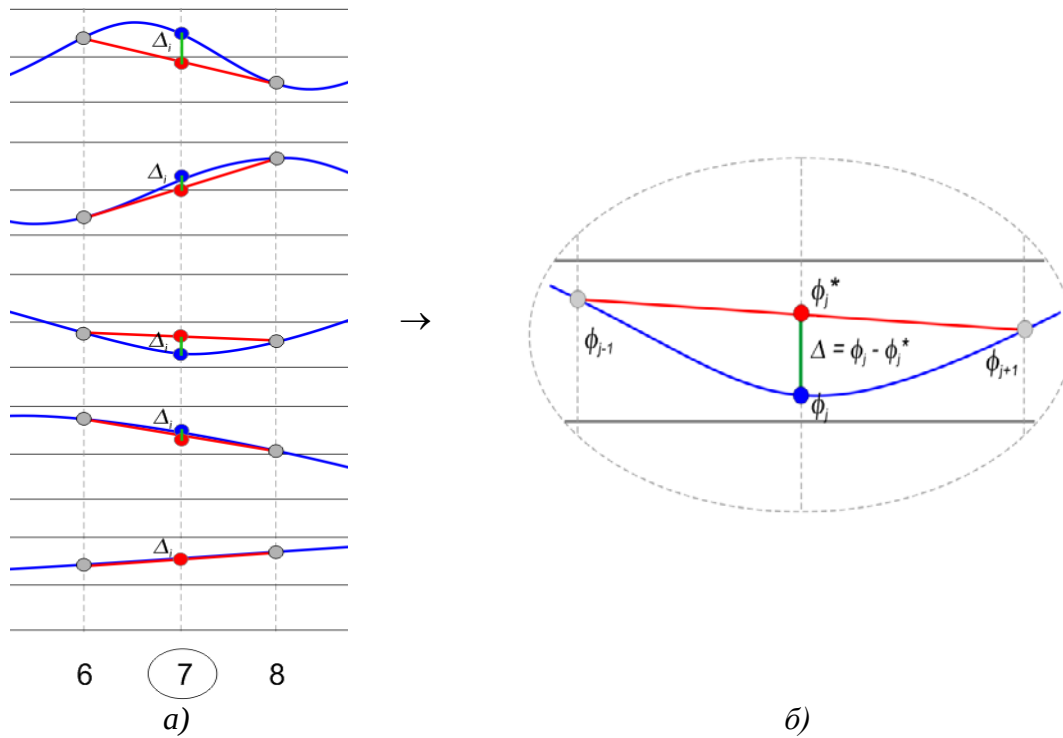
Фиг. 6.5. Криви на динамични натоварвания на греда

Проблемът за оптималното разположение на сензорите може да бъде описан като: кои сензори да отпаднат, така че да се минимизира загубата от информация и точност, като се вземат предвид едновременно всички режим на натоварванията. За да се илюстрира идеята за оптималното разположение на сензорите, ще бъдат използвани кривите на динамичните натоварвания с предварително определни и равномерно разположени 10 сензора. На Фиг. 6.6а показан разрез на кривите на динамично натоварване около разположените сензори 6, 7 и 8. Ако в позиция #7 сензорът бъде пропуснат, то неговите данни могат да бъдат заменени с линейна интерполация на данните от двата съседни за него сензора (Фиг. 6.6б). Отклонението на данните е пренебрежимо малко в случая, когато в позиция # 7 няма сензор, за режими на динамичните натоварвания от 1, 2 и 4, но е значително за режимите 3 и 5. Като се вземат предвид едновременно всички режими на динамичните натоварвания и всички местоположения на сензорите, проблемът за намиране на местоположението, където може да бъде пропуснат сензор, при запазване на минимално отклонение в данните $\Delta_j = \phi_j - \phi_j^*$, се свежда до проблем на комбинаторната оптимизация.

В общия случай, матрицата с информацията от динамичните натоварвания на структурата за p режима може да бъде представена като (Li и др., 2004):

$$\Phi = [\phi_1, \phi_2, \phi_3, \dots, \phi_p] \quad (6.30)$$

Целта е да се определи оптималното разположение на сензорите, като едновременно се отчитат едновременно всичките p режими на динамичните натоварвания на структурата.



Фиг. 6.6. Линейна интерполация при премахнат сензор #7 (a) и съответно отклонение в данните за 3-тата динамична крива (b)

6.4.1. Обобщен еднокритериален оптимизационен модел

За да се определи оптималния брой на сензорите, някои от тях трябва да отпаднат без значително да влияят на загубата на информация. Ако j -ия сензор за i -ия режим липсва, то неговите данни могат да бъдат изчислени като линейна интерполация от данните на неговите съседни сензора като $\phi_j^{i*} = \frac{(\phi_{j-1}^i + \phi_{j+1}^i)}{2}$. Отклонението на данните за липсващия сензор j за i -ия режим се изчислява като $\Delta_j^i = \phi_j^i - \phi_j^{i*}$. Изборът за присъствие или отсъствие на сензор, се извършва чрез въвеждането на двоични целочислени променливи x_j присвоени към всяко потенциално местоположение на сензори. Тази идея се реализира чрез следния еднокритериален оптимизационен модел за определяне разположението на сензори (Borissova и др., 2012):

$$\sum_{i=1}^p \sum_{j=2}^{n-1} x_j \left| \phi_j^i - \frac{\phi_{j-1}^i + \phi_{j+1}^i}{2} \right| \rightarrow \max \quad (6.31)$$

$$\sum_{j=1}^n x_j = m, \quad x_j \in \{0,1\} \quad (6.32)$$

където ϕ_j^i е стойността от i -ия режим в j -тата позиция на сензора, $(\phi_{j-1}^i - \phi_{j+1}^i)/2$ са редуцираните данни в случая на липсващ сензор и линейна интерполация на данните от съседните за него сензори, x_j са двоични целочислени променливи, присвоени към всяко потенциално местоположение на сензори.

В тази формулировка се предполага, че на първото и последното местоположение винаги има разположен сензор. Изискването за установяване на m сензора на подходящите местоположения се реализира от ограничението (6.32). Отпадането на сензорите с малката девиация Δ_i означава, че стойностите на функцията на структурната динамика ще бъдат най-близки до функцията, когато всички сензори са използвани. Очевидно е, че за да получим оптималните местоположения на сензорите, целевата функция (6.31) трябва да се максимизира, а оптималният избор на местоположения на сензорите се получава като решение на оптимизационната комбинаторната задача (6.31) при ограничения (6.32). Трябва да се отбележи, че функцията (6.31) се прилага едновременно за всичките режими на динамични натоварвания, като по този начин определянето дали определено местоположение на сензор да бъде заето или не взема предвид отчитайки всички режими на натоварване.

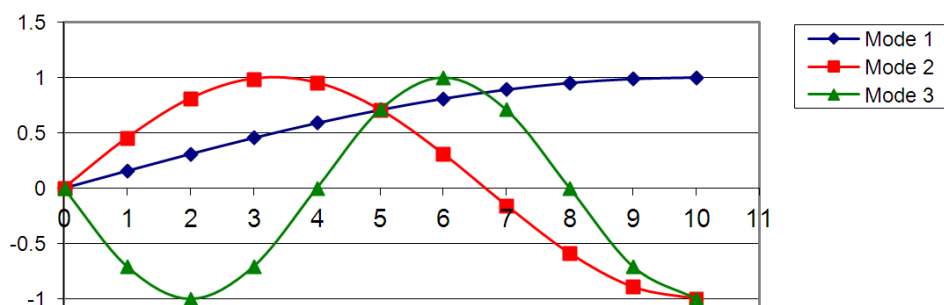
6.4.2. Числено тестване

За да се илюстрира предложения модел, са използвани данните за три режима на динамични натоварвания и 10 сензорни местоположения, получени на базата на анализа на вибрациите, както показано в Таблица 6.6. Данните за всеки режим са нормализирани така, че максималните стойности да са равни на 1.

Таблица 6.6 Данни за първите 3 режима на натоварване

Местоположение	Режим 1	Режим 2	Режим 3
1	0.15643	0.45400	-0.70706
2	0.30902	0.80901	-0.99993
3	0.45400	0.98770	-0.70706
4	0.58779	0.95107	0.00004
5	0.70711	0.70711	0.70701
6	0.80901	0.30902	1.00000
7	0.89103	-0.15643	0.70704
8	0.95107	-0.58779	0.00005
9	0.98770	-0.89103	-0.70713
10	1.00000	-1.00000	-0.99986

За по-голяма нагледност кривите на динамично натоварване за първите три режима са илюстрирани на Фиг. 6.7.



Фиг. 6.7. Графика на динамичните натоварвания при наличие на 10 сензора

Формулираната еднокритериална оптимизационна задача за определяне разположението на зададен брой сензори е формулирана като (Borissova и др., 2012).

$$\sum_{i=1}^p \sum_{j=2}^{n-1} x_j \left| \phi_j^i - \frac{\phi_{j-1}^i - \phi_{j+1}^i}{2} \right| \rightarrow \max \quad (6.33)$$

$$\sum_{j=1}^n x_j = m, \quad x_j \in \{0,1\} \quad (6.34)$$

$$x_1 = 1 \quad (6.35)$$

$$x_{10} = 1 \quad (6.36)$$

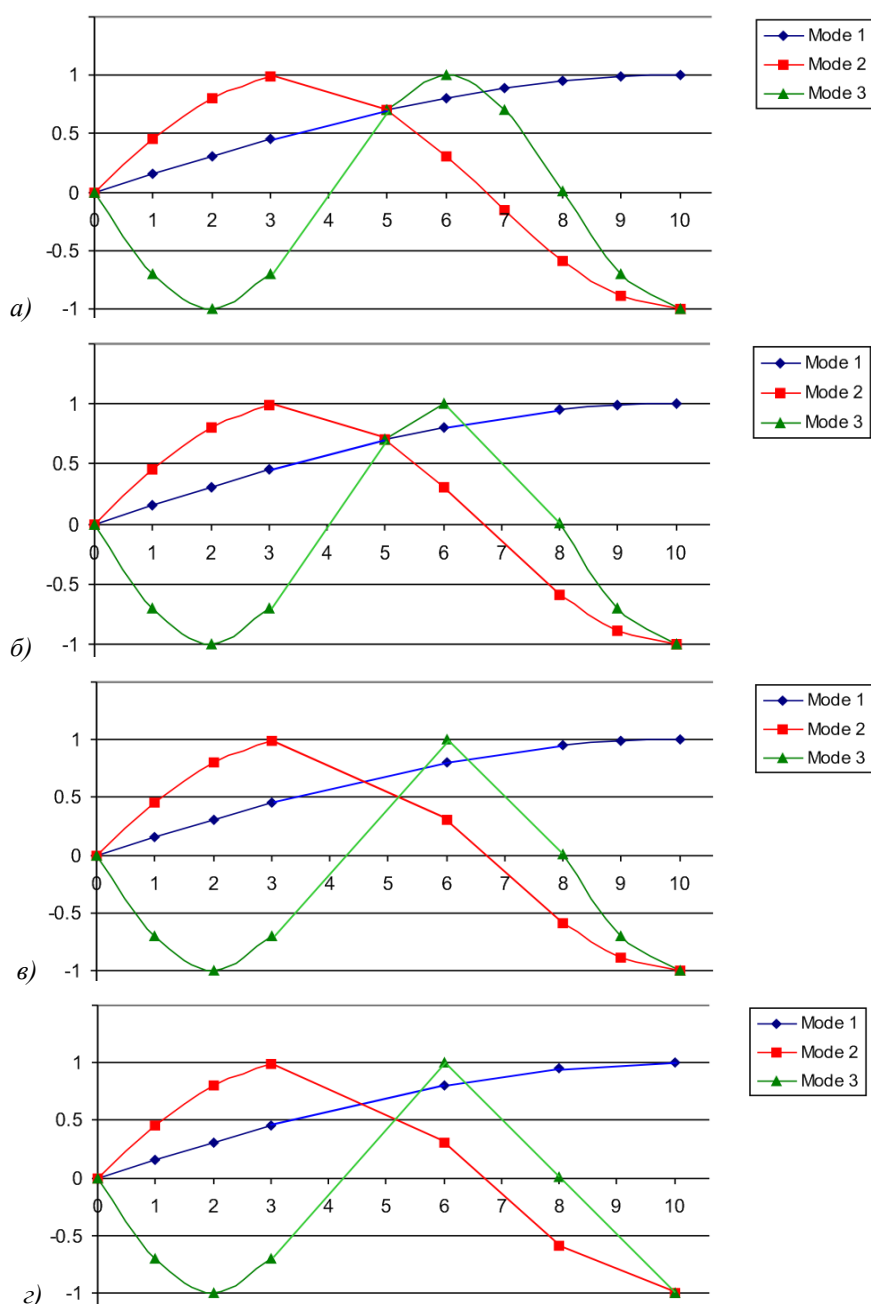
Целта на задачата е да се определят оптималните местоположения на 9, 8, 7 и 6 сензора ($m = 9, 8, 7, 6$) върху потенциални 10 местоположения и да се оцени общата грешка в данните. Формулираните оптимизационни задачи са решавани с помощта на LINGO Solver, а получените резултатите са показани в Таблица 6.7 и Фиг. 6.8.

Таблица 6.7. Резултати от решаването на задачите

Задача	Брой сензори за разполагане	Местоположение без сензор	Режим 1, грешка $\Sigma\Delta_i$	Режим 2, грешка $\Sigma\Delta_i$	Режим 3, грешка $\Sigma\Delta_i$
1	9	4	0.007235	0.10366	0.000065
2	8	4, 7	0.018225	0.12071	0.207080
3	7	4, 5, 7	0.026935	0.197775	0.414070
4	6	4, 5, 7, 9	0.039100	0.294910	0.621295

Решението на задачата за оптимално разположение на 9 сензора (Фиг. 6.4 а) и използваната апроксимация за липсващия сензор дава доста добро приближение в сравнение с разполагането на по-малък брой сензори (Фиг. 6.4 б, в, г). Както би следвало да се очаква, в резултат на използването на по-малък брой сензори се

наблюдава увеличаване на грешката в точността на данните Δ_i , показани в последните три колони на Табл. 6.7 и илюстрирани на Фиг. 6.8.



Фиг. 6.8. Криви на отчетените динамични натоварвания при наличие на:
а) 9 сензора, б) 8 сензора, в) 7 сензора, г) 6 сензора

Получените резултати показват, че предложеният подход може да бъде използван за определяне на оптималните местоположения на сензорите. Използването на линейната интерполация за липсващи сензори, позволява да не се губи цялата информация за конкретното местоположение. Оптималните местоположения на зададения брой сензори са определени при едновременно отчитане на всички режими.

6.4.3. Обобщен многокритериален оптимизационен модел

Проблемът за оптималното определяне на местоположението на сензори за целите на мониторинга, може да бъде формулиран и като проблем за определяне едновременно на броя и разположението на сензорите. Този проблем може да бъде математически представен чрез следния многокритериален оптимизационен модел (Mustakeroov & Borissova, 2014):

$$\begin{aligned} \max \sum_{i=1}^p \sum_{j=1}^n |x_j \phi_{i,j} + y_j \phi_{i,j}^*| \\ \min \sum_{j=1}^n x_j \end{aligned} \quad (6.37)$$

при ограничения

$$\phi_{i,j}^* = \left| \frac{(\phi_{i,j-1} + \phi_{i,j+1})}{2} \right| \quad (6.38)$$

$$x_j + y_j = 1 \quad (6.39)$$

$$x_j \in \{0,1\} \quad (6.40)$$

$$y_j \in \{0,1\} \quad (6.41)$$

където x_j са двоични целочислени променливи, присвоени към всяко потенциално местоположение на сензор; $\phi_{i,j}^*$ е стойността на линеината интерполация ако сензор j не е поставен; y_j са двоични целочислени променливи, присвоени към $\phi_{i,j}^*$; p е брой на режимите на динамичните натоварвания; n е брой на потенциалните местоположения на сензорите. Зависимостта (6.39) изразява факта, че ако местоположението на сензор j е заето, то не се налага интерполация на данните и обратно.

Съществуват различни подходи за решаване на многокритериалните проблеми. Наличието на няколко критерия при вземането на решения се характеризира с използването на специфични математически техники и методи за вземане на решения. В повечето многокритериални методи за вземане на решения, ЛВР играе основна роля при задаването на информация относно неговите предпочитания, които влияят върху крайното решение. Често използваните методи се основават на предварително изразяване на предпочитанията на ЛВР. Спецификата на изследвания проблем изисква да се намали броя на сензорите, като същевременно информацията за динамичното натоварване на структурата да е възможно най-близо до максималната (при наличие на всички сензори). За решаване на този проблем е избран като най-подходящ за конкретния случай *лексикографския метод*. При този метод критериите се подреждат

от ЛВР по степен на важност и последователно се решават еднокритериални оптимизационни задачи.

6.4.4. Числено тестване

За да се провери приложимостта на модела (6.37) – (6.41) за определяне на броя и разположението на сензори за мониторинг са използвани данните за първите 5 режима на динамични натоварвания, получени на базата на анализа на вибрациите (Tejada, 2009) и 10 сензорни местоположения, както показано в Таблица 6.8. Данните за всеки режим са нормализира така, че максималните стойности да са равни на 1.

Таблица 6.8. Данни за първите 5 режима на натоварване

Местоположение, j	Режим 1, $(\phi_{1,j})$	Режим 2, $(\phi_{2,j})$	Режим 3, $(\phi_{3,j})$	Режим 4, $(\phi_{4,j})$	Режим 5 $(\phi_{5,j})$
1	0.102	0.294	0.370	0.576	0.610
2	0.202	0.581	0.766	0.820	0.722
3	0.335	0.770	0.872	0.370	-0.090
4	0.467	0.870	0.604	-0.384	-0.724
5	0.590	0.807	-0.020	-0.810	-0.020
6	0.720	0.579	-0.670	-0.430	0.726
7	0.810	0.269	-0.854	0.410	0.050
8	0.890	-0.187	-0.465	0.820	-0.735
9	0.950	-0.603	0.251	0.230	-0.037
10	1.000	-1.000	1.000	-1.000	1.000

Формулирания могокритериален модел (6.37) – (6.41) използва данните от Табл. 6.8., като се приема задължително наличие на сензори в първото и последното местоположение. Като първи по важност критерий е определен критерия за запазване на информацията, касаеща мониторинга на структурата. Този критерий се използва като целева функция в следната оптимизационна задача (Mustakeroov & Borissova, 2014):

$$\max \sum_{i=1}^5 \sum_{j=1}^{10} |x_j \phi_{i,j} + y_j \phi_{i,j}^*| \quad (6.42)$$

при ограничения

$$\phi_{i,j}^* = \left| \frac{(\phi_{i,j-1} + \phi_{i,j+1})}{2} \right| \quad (6.43)$$

$$x_j + y_j = 1, j = 1, 2, \dots, 10 \quad (6.44)$$

$$x_j \in \{0,1\} \quad (6.45)$$

$$y_j \in \{0,1\} \quad (6.46)$$

$$x_1 = 1, x_{10} = 1 \quad (6.47)$$

Решението на тази задача представлява идеалния случай, когато всичките 10 сензора са налице, което ще осигури максималната стойност на целвата функция.

На втория етап от изпълнението на лексикографския метод, се избира втория по важност критерий от многокритериалната задача. В случая това е критерия за минимален брой сензори, които да бъдат използвани при мониторинга. Този критерий е използван като целева функция, а на база на получения резултат от предишната стъпка се определя подходяща допустима граница за първия критерий. Съответната еднокритериална оптимизационна задача има вида (Mustakeroov & Borissova, 2014):

$$\min \sum_{j=1}^{10} x_j, \forall x_j \in \{0,1\} \quad (6.48)$$

при ограничения

$$\sum_{i=1}^5 \sum_{j=1}^{10} |x_j \phi_{i,j} + y_j \phi_{i,j}^*| \geq \alpha \left(\sum_{i=1}^5 \sum_{j=1}^{10} |x_j \phi_{i,j} + y_j \phi_{i,j}^*| \right)^{optimum} \quad (6.49)$$

$$\phi_{i,j}^* = \left| \frac{(\phi_{i,j-1} + \phi_{i,j+1})}{2} \right| \quad (6.50)$$

$$x_j + y_j = 1 \quad (6.51)$$

$$x_j \in \{0,1\} \quad y_i \in \{0,1\} \quad (6.52)$$

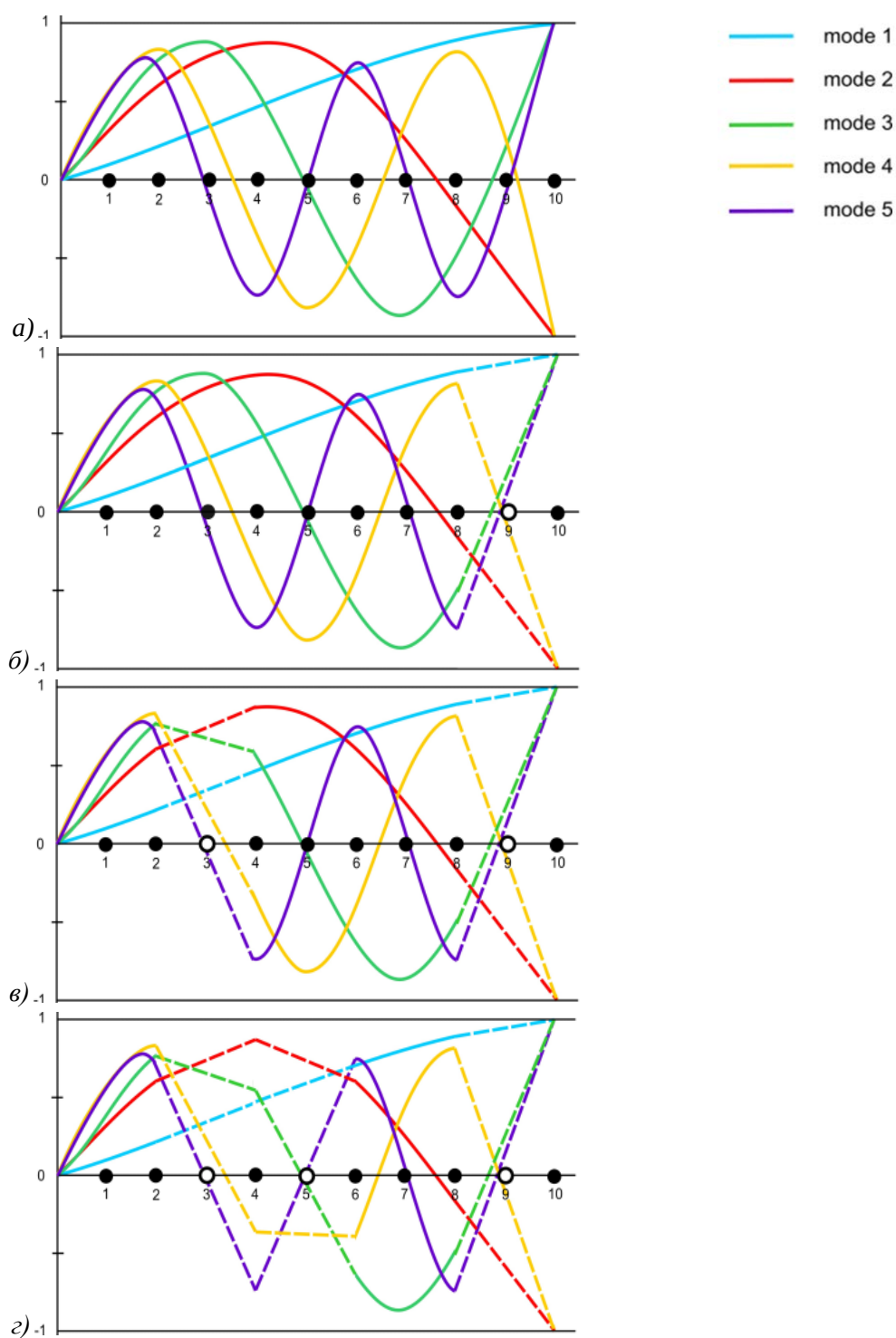
$$x_1 = 1, x_{10} = 1 \quad (6.53)$$

където $\left(\sum_{i=1}^5 \sum_{j=1}^{10} |x_j \phi_{i,j} + y_j \phi_{i,j}^*| \right)^{optimum}$ е оптималната стойност, определена от първия етап на лексикографския метод. Чрез коефициента α се определя с колко е допустимо да се отличават данните, когато се използват по-малко сензори за мониторинга.

Получените резултатите за определяне на оптимален брой и разположение на сензорите са показани на Фиг. 6.9. С пунктирна линия е показана интерполацията на данните за местоположение без сензор. На Фиг. 6.9а е показан резултата от използването на многокритериалния модел при използване на всичките 10 сензора.

На Фиг. 6.9б, 6.9в и 6.9г са показани резултатите от използването на многокритериалния модел за определяне на броя и разположението на сензори при зададени определени стойности на коефициента α ($\alpha = 98\%$, $\alpha = 97\%$ и $\alpha = 95\%$). Задавайки стойност на коефициента $\alpha = 98\%$ води до редуциране на общия брой сензори до 9, като в конкретния случай местоположението на сензор 9 остава незаето. С пунктирна линия е показана интерполацията при отсъствие на сензор. Ако се промени

стойността на коефициента като $\alpha = 97\%$, в решението на задачата, общият брой сензори се редуцира до 8, като в случая позиции #3 и #9 остават без сензори (Фиг. 6.9в). При $\alpha = 95\%$ броят сензори вече е намален до 7, като позиции #3, #5 и #9 остават без сензори (Фиг. 6.9г).



Формулираните оптимизационни задачи са решавани чрез LINGO вер. 12 на настолен компютър с Intel® Celeron® 2.93 GHz процесор, 2 GB оперативна памет и операционна система MS Windows XP ©. Времената за решаване на числените примери са от порядъка на секунди, но очевидно те зависят от размера на формулираните задачи.

За разлика от еднокритериалния модел, при формулирания многокритериален модел се определя както броя на сензорите за мониторинга, така и техните местоположения. Това се постига чрез въведения коефициент α , определящ отклонението на данните от идеалните криви на динамичните натоварвания. И в двата модела данните за местоположение без сензор се определят чрез линейна интерполация на данните от двата съседни за него сензора.

Глава 7:

Модели и алгоритми за обоснован избор на алтернативи

7.1. Въведение

В тази глава са описани модели и алгоритми на за обоснован избор на алтернативи за: 1) избор чрез класиране, 2) избор на k -най добри алтернативи, 3) избор на алтернатива с отчитане на околната среда, 4) избор на алтернатива чрез групово вземане на решения. Получените резултати са публикувани в статии D2, D11, D14, D15, D25, D29.

В последните години се наблюдава голямо разнообразие на предлагани типове и модели на различни системи и устройства. Целта е задоволяването на различните потребители по отношение на предлаганата функционалност на системите и устройствата. При наличието на такава богата гама от системи и устройства, вземането на решение за избор на най-удовлетворителна алтернатива в условията на множество конкурентни предложения е задача, която възниква както в организациите, така и в личния живот на всеки потребител. За подпомагане на този избор са разработени различни подходи и алгоритми. Важно условие за тяхната приложимост е вземането предвид на множество критерии, отразяващи качествата на системата/устройството и съобразяване с личното отношение на потребителя (лицето, вземащо решение – ЛВР). Една от насоките за изследвания в това направление е изучаване поведението на ЛВР при вземане на решение за избор. Това трябва да бъде отразено в разработваните подходи и алгоритми, без ЛВР да бъдат натоварвани с изискване за задаване на трудно оценима информация. Най-популярни са многокритериалните подходи за вземане на решения, които включват модели като *scoring models* (Nelson, 1986), *AHP* (Garcia-Cascales & Lamata, 2009), *ANP* (Saaty, 2005; Wu & Lee, 2007), *ELECTRE* (Wang & Triantaphyllou, 2008), *PROMETHEE* (Behzadian и др., 2010), *utility models* (Kalouptsidis и др., 2007), *TOPSIS* (Behzadian и др., 2012) и *axiomatic design* (Kulak и др., 2010). Характерна особеност на този клас задачи за вземане на решения е, че те са неформализирани или слабо формализирани задачи, за решаването на които е

необходимо участието на ЛВР. При това, получаваните решения са до голяма степен субективни и зависят от предпочитанията на ЛВР. От своя страна, задачите за вземане на решения при много критерии се разделят на два класа (Vassilev и др., 2004): задачи на многокритериалната оптимизация и задачи на многокритериалния анализ. При задачите на многокритериалната оптимизация краен брой аналитично зададени ограничения във вид на функции, определят безкраен брой алтернативи. При задачите на многокритериалния анализ явно се задават краен брой алтернативи в таблична форма, която може да бъде описана чрез матрица на алтернативите (Vassilev и др., 2004; Peneva & Popchev, 2009) – Таблица 7.1

Таблица 7.1 Матрицата на алтернативите

k_j a_i	$k_1(.)$	$k_2(.)$	...	$k_j(.)$	...	$k_k(.)$
a_1	a_{11}	a_{12}	...	a_{1j}	...	a_{1k}
a_2	a_{21}	a_{22}	...	a_{2j}	...	a_{2k}
...	...	...	...	...	...	...
a_i	a_{i1}	a_{i2}	...	a_{ij}	...	a_{ik}
..	...	...	...	...	...	...
a_n	a_{n1}	a_{n2}	...	a_{nj}	...	a_{nk}

където: a_i е алтернативата с индекс $i, i=1,...,n$; $k_j(.)$ е критерия с индекс $j, j=1,...,k$.

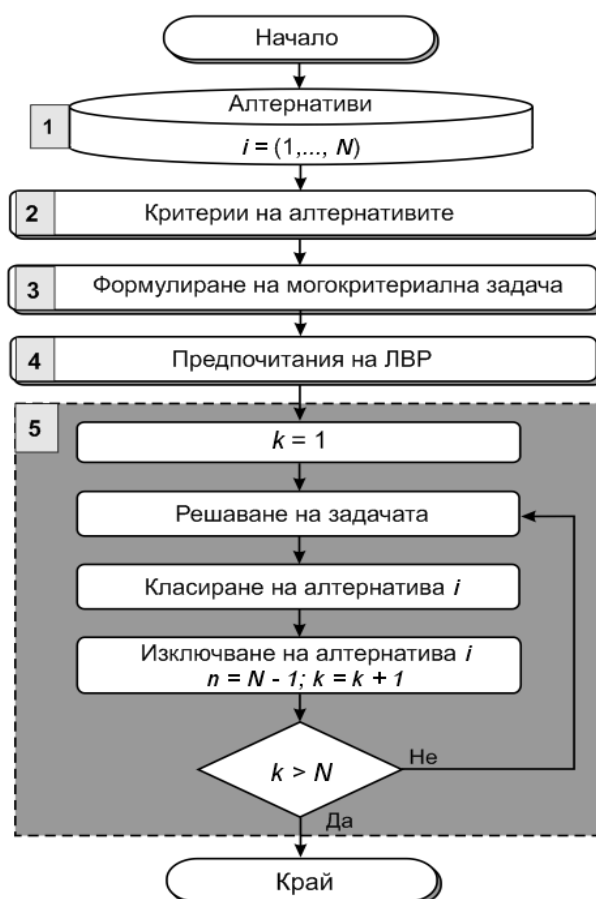
7.2. Обобщен алгоритъм за избор чрез класиране

Идеята на предложения подход е да се симулира на поведението на ЛВР в условията на многокритериален избор, а именно чрез последователно решаване на множество многокритериални задачи. Получените решения са в съответствие с предпочитанията на ЛВР и решенията позволяват алтернативите да се подредят в един оценъчен списък. Този списък се използва от потребителя за вземане на крайното решение. Често в практиката ЛВР има свой субективни предпочитания към определена алтернатива, които трудно могат да бъдат формализирани. Наличието на такъв подреден списък от алтернативи, ще позволи не само избор, но и оценка до каква степен интуитивният избор би бил достатъчно обоснован, когато се взема предвид влиянието на всички атрибути.

За целта е формулиран многокритериален оптимизационен модел, който е включен в алгоритъм за подредба на всички алтернативи, съгласно предварително

(априорно) заявени предпочитания от ЛВР. На всяка итерация на алгоритъма се формулира и решава многокритериална оптимизационна задача, в която алтернативата, определена като Парето-оптимална на предишната стъпка, е изключена, т.е. размерността на всяка следваща задача последователно намалява. Броят на итерациите е равен на броя на зададените алтернативи. На последната стъпка от цикъла, когато само една алтернатива е останала, изборът е очевиден, но решението на съответната задача ще предостави информация за стойността на целевата функция при избор на тази алтернатива. Стойностите на целевата функция могат да се използват като оценка на това доколко всяка алтернатива е отдалечена от най-добрата алтернатива, определена на първата стъпка на алгоритъма.

Предложеният обобщен алгоритъм за класиране на алтернативите чрез използване на многокритериална оптимизация, е илюстриран графично на Фиг. 7.1.



Фиг. 7.1. Обобщен алгоритъм за класиране на алтернативи

Алгоритъмът започва с определяне на броя на алтернативите N , които ще участват при избора. Всяка алтернатива се описва с набор от критерии (параметри), а ЛВР определя кои от тях да се максимизират и кои да се минимизират. На база на

определените критерии се формулира съответна многокритериална оптимизационна задача. На следващия етап от алгоритъма ЛВР взема решение за относителната важност между критериите. Списъкът с класираните устройства се създава на етап 5 чрез последователно решаване на множество многокритериални задачи, следвайки описаната процедура:

- 1) установяване на $k = 1$.
- 2) решаване на многокритериалната задачата и включване на определената Парето-оптимална алтернатива в списъка на класирането.
- 3) Премахване на i -ия ред от матрицата на алтернативите, като по този начин се изключва, Парето-оптимална алтернатива от предходната стъпка и се намалява броя на алтернативите, които трябва да се вземат предвид, ($n = N - 1$). Заедно с това се увеличава с единица брояча $k = k + 1$.
- 4) Проверка дали ($k > N$) – ако условието не е изпълнено се преминава към т. 2, в противен случай, алгоритъмът завършва изпълнението си.

Ако ЛВР не е удовлетворен от полученото класиране, то използваните входните данни могат да се коригират: 1) чрез добавяне или премахване някои критерии (параметри на алтернативите); 2) чрез промяна на предпочитанията; 3) чрез промяна на множеството от алтернативи. Поради факта, че многокритериалната оптимизация е основното ядро на алгоритъма, то може да се използва всеки метод с априорно задаване на информацията за предпочитанията на ЛВР, за решаване на многокритериалната задача. Така предложеният алгоритъм може да служи като инструмент за симулация при вземането на решения, при наличие на краен брой алтернативи и множество критерии.

7.2.1. Числено тестване

За да демонстрира приложимостта на предложеният алгоритъм е направено числено тестване за избор на устройство с много параметри, които са съществени за потребителския избор. Използвани са реални данни за 15 мобилни телефона, показани в Таблица 7.2. Чрез тези данни е формулирана следната многокритериална задача:

$$\begin{cases} \max(f^1, f^2, f^3, f^4, f^5, f^6) \\ \min(\varphi^1, \varphi^2) \end{cases} \quad (7.1)$$

при ограничения

$$f^1 = \sum_{i=1}^{15} f_i^1 x_i \quad (7.2)$$

$$f^2 = \sum_{i=1}^{15} f_i^2 x_i \quad (7.3)$$

$$f^3 = \sum_{i=1}^{15} f_i^3 x_i \quad (7.4)$$

$$f^4 = \sum_{i=1}^{15} f_i^4 x_i \quad (7.5)$$

$$f^5 = \sum_{i=1}^{15} f_i^5 x_i \quad (7.6)$$

$$f^6 = \sum_{i=1}^{15} f_i^6 x_i \quad (7.7)$$

$$\varphi^1 = \sum_{i=1}^{15} \varphi_i^1 x_i \quad (7.8)$$

$$\varphi^2 = \sum_{i=1}^{15} \varphi_i^2 x_i \quad (7.9)$$

$$\sum_{i=1}^{15} x_i = 1, \forall x_i \in \{0,1\} \quad (7.10)$$

където параметрите на устройствата са означени с f^j и φ^k , а променливите x_i са двоични целочислени променливи, присвоени на алтернативите. Стойностите на параметрите за всяка алтернатива са показани в Таблица 7.2.

Таблица 7.2. Параметри на алтернативите (устройствата)

#	Модел мобилен телефон	Camera resolution MP	Integrated memory, GB	Card slot capacity, GB	Standby time, hrs	GPS yes/no	Talk time, hrs	Тегло gr	Цена лв
		(f^1)	(f^2)	(f^3)	(f^4)	(f^5)	(f^6)	(φ^1)	(φ^2)
1	BlackBerry 9000 Bold	2.0	1.0	32	310	1 (yes)	5	133	949
2	Samsung Omnia HD	8.0	16.0	32	430	1 (yes)	10	148	919
3	Sony-Ericsson U1i Satio	12.0	0.128	32	350	1 (yes)	10	126	879
4	Apple iPhone 3G	2.0	8.0	0	250	1 (yes)	5	133	825
5	Nokia N-97 Mini	5.0	8.0	16	310	1 (yes)	6	138	785
6	HTC Google G1	3.15	0.192	16	400	1 (yes)	5	158	659
7	Sony-Ericsson W-995	8.1	0.118	8	370	1 (yes)	9	113	605
8	LG GD900 Crystal	8.0	1.5	32	300	0 (no)	4	127	599
9	Nokia E-75	3.15	0.050	16	250	1 (yes)	4	139	539
10	Samsung S8300	8.0	0.080	16	350	1 (yes)	4	105	485
11	BlackBerry Curve 8520	2.0	0.256	32	408	0 (no)	4.5	106	480
12	Nokia 5530	3.15	0.070	16	336	0 (no)	5	107	459
13	Sony-Ericsson C-903	5.0	0.105	8	350	1 (yes)	4	96	435
14	LG KM-900 Arena	5.0	8.0	16	300	1 (yes)	3.3	105	435
15	Samsung M-7600	3.15	0.050	16	350	0 (no)	5	99.7	419

Изявените от ЛВР предпочитания за важността на отделните параметри са взети предвид чрез съответни тегловни коефициенти, показани в Таблица 7.3.

Таблица 7.3. Тегловни коефициенти за важността на параметрите (критериите)

ЛВР	w_1	w_2	w_3	w_4	w_5	w_6	w_7	w_8
ЛВР-1	0.125	0.125	0.125	0.125	0.125	0.125	0.125	0.125
ЛВР-2	0.100	0.100	0.100	0.100	0.100	0.100	0.150	0.250
ЛВР-3	0.150	0.100	0.100	0.150	0.050	0.050	0.050	0.350

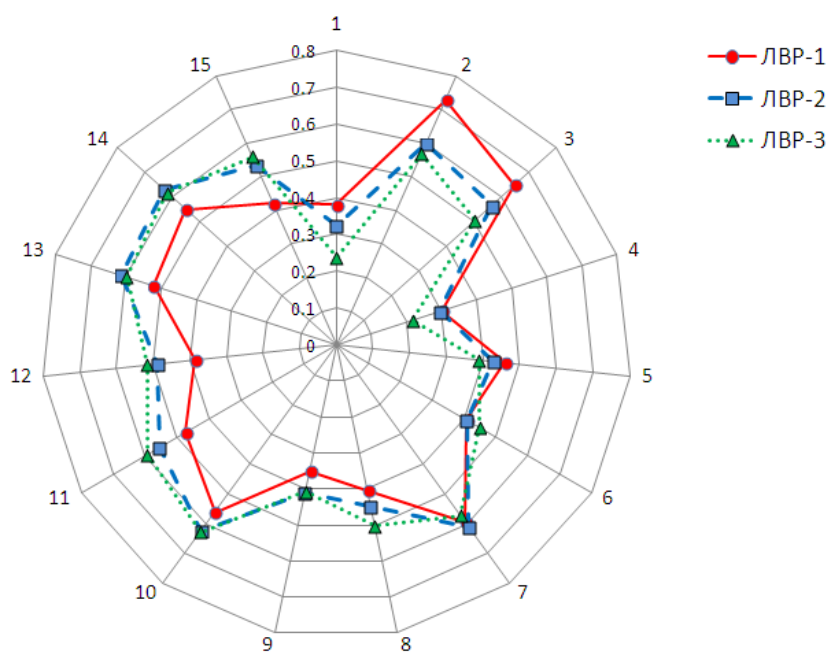
Оптимизационната задача (7.1) – (7.10) е решена чрез *метода на претеглената сума* и в зависимост от предпочитанията на ЛВР (Табл. 7.3) са получени три различни подредени списъка на устройствата. Получените резултати са показани в Таблица 7.4 и са илюстрирани на Фиг. 7.2.

Таблица 7.4. Класирани устройства, според предпочитанията на ЛВР

Класация	ЛВР-1			ЛВР-2			ЛВР-3		
	стойност на Ц.Ф.	Устройство от Табл. 1		стойност на Ц.Ф.	Устройство от Табл. 1		стойност на Ц.Ф.	Устройство от Табл. 1	
		#	модел		#	модел		#	модел
1	0.7272	2	Samsung Omnia	0.6283	14	LG KM-900 Arena	0.6279	10	Samsung S8300
2	0.6511	3	Sony-Ericsson U1i Satio	0.6233	10	Samsung S8300	0.6187	14	LG KM-900 Arena
3	0.5946	7	Sony-Ericsson W-995	0.6138	13	Sony-Ericsson C-903	0.5983	13	Sony-Ericsson C-903
4	0.5615	10	Samsung S8300	0.6093	7	Sony-Ericsson W-995	0.5936	11	BlackBerry Curve 8520
5	0.5501	14	LG KM-900 Arena	0.5983	2	Samsung Omnia	0.5729	7	Sony-Ericsson W-995
6	0.5229	13	Sony-Ericsson C-903	0.5665	3	Sony-Ericsson U1i Satio	0.5679	2	Samsung Omnia
7	0.4742	11	BlackBerry Curve 8520	0.5540	11	BlackBerry Curve 8520	0.5603	15	Samsung M-7600
8	0.4583	5	Nokia N-97 Mini	0.5335	15	Samsung M-7600	0.5164	12	Nokia 5530
9	0.4206	15	Samsung M-7600	0.4893	12	Nokia 5530	0.5059	3	Sony-Ericsson U1i Satio
10	0.4073	6	HTC Google G1	0.4474	8	LG GD900 Crystal	0.5021	8	LG GD900 Crystal
11	0.4042	8	LG GD900 Crystal	0.4292	5	Nokia N-97 Mini	0.4473	6	HTC Google G1
12	0.3869	12	Nokia 5530	0.4113	9	Nokia E-75	0.4086	9	Nokia E-75
13	0.3812	1	BlackBerry 9000 Bold	0.4079	6	HTC Google G1	0.3894	5	Nokia N-97 Mini
14	0.3499	9	Nokia E-75	0.3251	1	BlackBerry 9000 Bold	0.2388	1	BlackBerry 9000 Bold
15	0.3012	4	Apple iPhone 3G	0.2962	4	Apple iPhone 3G	0.2176	4	Apple iPhone 3G

Описаните модел и алгоритъм могат да се използват за обоснован избор на различни устройства и обекти, характеризиращи се с множество и често противоречиви параметри. При вграждането им в съответни софтуерни инструменти, те могат да бъде прилагани от потребители с различна квалификация в различни области, без да са необходими специални познания в областта на операционните изследвания. При това,

чрез въвеждането и използването на предпочитания на потребителя, лесно могат да бъдат симулирани различни стратегии за избор на най-подходяща алтернатива.



Фиг. 7.2. Стойности на целевите функции на класираните устройства

Друг числен пример, който може да служи като доказателство за приложимостта на описания алгоритъм в различни области, е класирането на 24 държави от Източна Европа и Централна Азия за правене на бизнес по определени индикатори. Броят на използваните икономически индикатори за подобна класация е определен от Световната Банка (*Economy rankings*, The World Bank, 2013, <http://www.doingbusiness.org/rankings>). В зависимост от броя на държавите, участващи в класацията, се обределят и възможните стойности за икономическите показатели. Следователно, икономическите индикатори за всяка държава от Източна Европа и Централна Азия, се оценяват със стойности от 1 до 24, като не се допуска оценяване на две или повече държави по едни и същи икономически индикатори да имат еднакви стойности. При това, колкото е по-малка стойността на индикатора, толкова е по-лесно реализирането на бизнес-дейности, т.е. регулаторната среда е по-благоприятна за стартиране и функциониране на бизнеса. Използваните икономическите индикатори от Световната Банка и съответните стойности са показани в Таблица 7.5.

В Таблица 7.5 съответните държави са подредени по азбучен ред, като крайната цел е те да се подредят в списък, показващ възможностите за правене на бизнес, като се вземат предвид едновременно всички стойности на критериите (икономическите индикатори).

Таблица 7.5. Икономически показатели на държави от Източна Европа и Централна Азия

Икономически показатели		Стартиране на бизнес	Строителни разреше-ния	Достъп до електричество	Регистриране на имот	Получаване на кредит	Защита на инвеститорите	Данъци	Трансгранична търговия	Изпълнение на договори	Разрешаване на несъстоятелност
№	Държава	c ₁	c ₂	c ₃	c ₄	c ₅	c ₆	c ₇	c ₈	c ₉	c ₁₀
1	Албания	14	24	14	22	6	3	20	9	17	10
2	Армения	4	3	10	3	11	5	14	14	19	9
3	Азербайджан	6	19	21	5	17	6	10	20	5	18
4	Беларус	3	2	20	2	21	18	17	18	2	7
5	Босна и Херцеговина	24	16	15	18	18	19	16	12	22	14
6	България	11	9	13	14	13	12	13	10	18	17
7	Хърватия	18	12	2	20	12	23	5	13	14	19
8	Кипър	8	7	9	19	15	9	3	2	21	1
9	Грузия	2	1	1	1	3	4	4	4	8	13
10	Казахстан	7	15	7	8	20	1	1	22	7	6
11	Косово	23	13	12	16	8	20	6	15	24	15
12	Киргизка република	5	6	22	6	5	2	23	21	13	23
13	Латвия	13	8	8	9	1	13	7	1	4	2
14	Литва	22	4	5	4	16	14	8	3	3	3
15	Македония	1	5	10	13	7	7	2	7	15	8
16	Молдова	20	17	16	7	14	17	15	16	6	16
17	Черна гора	12	18	4	21	2	10	12	5	23	4
18	Румъния	15	10	19	15	4	11	18	6	16	20
19	Руска Федерация	21	20	24	12	22	22	9	19	1	5
20	Сърбия	9	21	6	10	10	16	19	11	20	21
21	Таджикистан	17	22	23	17	24	8	24	23	11	12
22	Турция	16	11	3	11	19	15	11	8	9	22
23	Украйна	10	23	17	24	9	21	22	17	10	24
24	Узбекистан	19	14	18	23	23	24	21	24	12	11

Използвани са 3 различни предпочитания на ЛВР към представените икономически индикатори (Таблица 7.6).

Таблица 7.6. Предпочитания на ЛВР за важността на индикаторите

w_i	w_1	w_2	w_3	w_4	w_5	w_6	w_7	w_8	w_9	w_{10}
ЛВР-1	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10
ЛВР-2	0.15	0.05	0.05	0.05	0.05	0.10	0.15	0.10	0.05	0.25
ЛВР-3	0.05	0.25	0.05	0.25	0.05	0.05	0.10	0.05	0.10	0.05

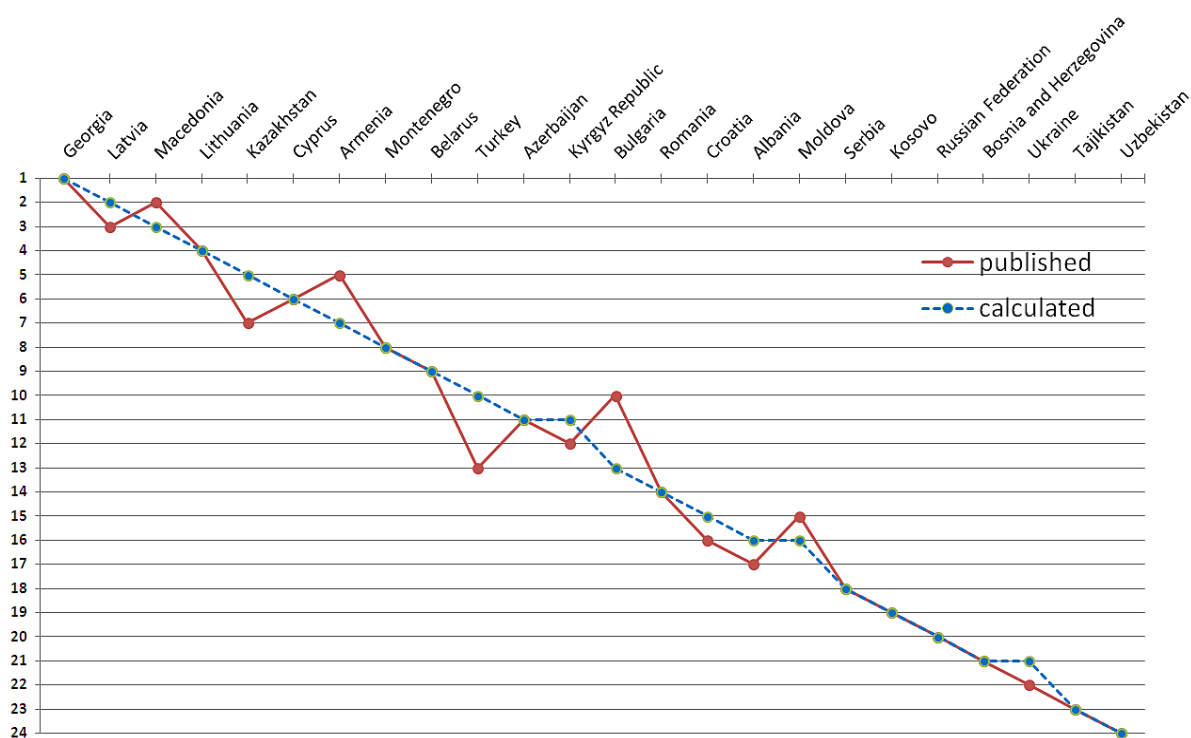
Предпочитанията, означени като ЛВР-1 съответстват на еднаква значимост на всички икономически показатели. Световната Банка е публикувала своя класация на държавите, като е приела именно такава еднаква значимост на показателите. Предпочитанията ЛВР-2 отразяват доминиращо значение на показателя за *разрешаване на производство по несъстоятелност*, а ЛВР-3 определя като най-съществени

показателите *получаване на разрешение за строеж* и *регистриране на имот*. Използването на предложените модел и алгоритъм за класиране на алтернативите (държавите), съобразени с така изразените предпочитания на ЛВР, определят 3 подредби на държавите, както е показано в Таблица 7.7 (Mustakero & Borissova, 2013).

Таблица 7.7. Класация на държавите

Класация	ЛВР-1		Класация	ЛВР-2		Класация	ЛВР-3	
	Държава	Стойност на Ц.Ф.		Държава	Стойност на Ц.Ф.		Държава	Стойност на Ц.Ф.
1	Грузия	0.12608	1	Грузия	0.19782	1	Грузия	0.08478
2	Латвия	0.24347	2	Македония	0.21739	2	Литва	0.22391
3	Македония	0.26521	3	Латвия	0.23478	3	Беларус	0.26739
4	Литва	0.30434	4	Кипър	0.24130	4	Латвия	0.27173
5	Казахстан	0.36086	5	Казахстан	0.29565	5	Македония	0.30434
6	Кипър	0.36521	6	Литва	0.32173	6	Армения	0.33478
7	Армения	0.37826	7	Черна гора	0.35652	7	Казахстан	0.37608
8	Черна гора	0.41739	8	Армения	0.36739	8	Киргизката република	0.41086
9	Беларус	0.42608	9	Беларус	0.41304	9	Кипър	0.43913
10	Турция	0.49130	10	Азербайджан	0.51086	10	Турция	0.45869
11-12	Азербайджан	0.50000	11	Албания	0.51956	11	Азербайджан	0.46956
	Киргизката република	0.50000	12	България	0.52826	12	България	0.49782
13	България	0.50434	13	Руска Федерация	0.55000	13	Молдова	0.51304
14	Румъния	0.53913	14	Турция	0.57826	14	Румъния	0.53913
15	Хърватия	0.54782	15	Хърватия	0.59565	15	Хърватия	0.57173
16-17	Албания	0.56086	16	Киргизката република	0.60000	16	Руска Федерация	0.58913
	Молдова	0.56086	17	Румъния	0.60217	17	Косово	0.59782
18	Сърбия	0.57826	18	Косово	0.61086	18	Черна гора	0.60217
19	Косово	0.60434	19	Молдова	0.61956	19	Сърбия	0.62173
20	Руска Федерация	0.6217391	20	Сърбия	0.63043	20	Босна и Херцеговина	0.71304
21-22	Босна и Херцеговина	0.7130435	21	Таджикистан	0.69130	21	Албания	0.73913
	Украйна	0.7130435	22	Босна и Херцеговина	0.69782	22	Узбекистан	0.75869
23	Таджикистан	0.7347826	23	Узбекистан	0.73695	23	Таджикистан	0.76086
24	Узбекистан	0.7739130	24	Украйна	0.76521	24	Украйна	0.81304

При използване на предпочитанията ЛВР-1, целевите функции за държавите Азербайджан и Киргизстан, за Албания и Молдова и за Босна и Херцеговина и Украйна имат едни и същи стойности (Табл. 7.7). Поради това в получената класация и трите двойки страни делят едно и също място. На Фиг. 7.3 получените резултати за ЛВР-1 (*с прекъсната линия*) са сравнени с резултатите, публикуваните от Световната Банка (*с непрекъсната линия*) при случая на еднаква важност на всички икономически показатели.



Фиг. 7.3. Сравнение на резултатите

Обяснението на наблюдаваните разлики при сравнение на резултатите от използването на предложения алгоритъм за класиране с резултатите, публикувани от Световната Банка е, че в класирането на Световната Банка (*Economy rankings*, The World Bank, 2013, <http://www.doingbusiness.org/rankings>) се използват средноаритметични стойности, които се закръгляват до цяло число. Например, средноаритметичните стойности за Македония и Латвия са 7.1 и 6.6 и когато тези стойности са закръглят до цяло число, се получава 7. Използвайки предложения алгоритъм за класация определя стойности на целевата функция от 0.2434783 за Латвия и 0.2652174 за Македония. Това означава, че Латвия има по-добри икономически показатели и следва да се класира преди Македония.

7.3. Многокритериален оптимизационен модел за избор на k най-добри алтернативи

Идеята за k най-добри алтернативи е да се редуцира броя на алтернативите до предварително зададен брой k на най-добрите алтернативи, като се вземат предвид потребителските изисквания относно важността на параметрите на всяка алтернатива. За целта е предложен оптимизационен модел, който за разлика от методите, базирани на сравнение и определяне на отклонението между оценките на две алтернативи по даден критерий, позволява определянето на k най-добри устройства, чрез решаване на

една единствена многокритериална оптимизациона задача. При това, за разлика от съществуващите модели, предложеният подход за класиране и избор на k най-добри устройства, отчитат както различните атрибути на алтернативите, така и потребителските изисквания, относно важността на тези атрибути.

Тази идея може да се илюстрира чрез пример за избор на k най-добри уреди за нощно виждане (УНВ), където потребителят действа като ЛВР и иска да вземе предвид всички аспекти на съществуващите алтернативи за избор на устройства. Предпочитаните k броя УНВ трябва най-добре да отразяват всички предпочитания на ЛВР. В този пример като най-съществените параметри на УНВ се разглеждат следните параметри - *разстояние на действие на УНВ; зрително поле; диапазон на фокусировка на обектива; живот на работа на УНВ от батерийното захранване; тегло и цена на УНВ*. На базата на зададено множество УНВ с известни параметри и при зададени външни условия на наблюдение е разработен съответен многокритериален модел за определяне на k най-добри УНВ (Borissova & Mustakerov, 2013):

$$\begin{aligned} & \text{maximize } \{R, FOV, BL\} \\ & \text{minimize } \{FR, Weight, Price\} \end{aligned} \quad (7.11)$$

при ограничения

$$R = \sum_{i=1}^n x_i \sqrt{0.07 E \tau_a K A_t^* \frac{D_{in}^i f_{ob}^i \tau_{ob}^i \delta^i S^i}{\Phi_{min}^i M^i}} \quad (7.12)$$

$$FOV = \sum_{i=1}^n FOV_i x_i \quad (7.13)$$

$$BL = \sum_{i=1}^n BL_i x_i \quad (7.14)$$

$$FR = \sum_{i=1}^n FR_i x_i \quad (7.15)$$

$$Weight = \sum_{i=1}^n Weight_i x_i \quad (7.16)$$

$$Price = \sum_{i=1}^n Price_i x_i \quad (7.17)$$

$$\sum_{i=1}^n x_i = k, x_i \in \{0, 1\} \quad (7.18)$$

$$1 < k < n \quad (7.19)$$

където: R – разстояние на действие, FOV – зрително поле, FR – минимално фокусно разстояние на обектива, BL – живот на батериите (времето на работоспособност на уреда), тегло и цена на УНВ. Двоичните целочислени променливи x_i са присвоени на всяко устройство, k е целочислена променлива, определяща броя на търсените най-добрите устройства, а n е броят на устройствата, от които се реализира изборът.

7.3.1. Числено тестване

Като входни данни за формулираната оптимизационна задача (7.11) – (7.19) са използвани параметрите на 10 реални ОНВ (очила за нощно виждане), които са показани в Таблица 7.8.

Таблица 7.8. Параметри на ОНВ

#	ОНВ	δ , lp/mm	f_{ob} mm	D_{in} , mm	τ_{ob}	M	S, A/lm	FOV, $degree$	BL $hours$	FR cm	Тегло gr	Цена $\$$
1	Night Optics D-2MV	40	26	21.67	0.78	12	0.00024	40	40	25	482	650
2	Rigel 3250	30	35	35	0.78	12	0.00022	30	30	25	430	699
3	ATN Cougar 2	32-40	35	25	0.78	16	0.00031	30	10-20	100	500	3071
4	ATN Cougar CGTI	45-54	35	25	0.78	15	0.00035	30	10-20	100	500	3696
5	ATN Night Cougar-3	64	35	25	0.78	20	0.00087	30	10-20	100	500	4884
6	ATN Night Cougar-4	64-72	35	25	0.80	25	0.00115	30	10-20	100	500	9932
7	ATN PS23-2	36-45	24	20	0.80	13	0.00070	40	60	25	700	3550
8	ATN PS23-CGT	45-54	24	20	0.80	17	0.00110	40	60	25	700	4195
9	ATN PS23-3	55-72	24	20	0.80	22	0.00160	40	35	25	700	5895
10	ATN PS23-4	64-72	24	20	0.80	24	0.00190	40	35	25	700	12995

Използвайки метода на претеглената сума, се получава следната трансформирана задача, определяща k най-добри ОНВ:

$$\max \left\{ \begin{aligned} &w_1 \frac{R - R_{\min}}{R_{\max} - R_{\min}} x_1 + w_2 \frac{FOV - FOV_{\min}}{FOV_{\max} - FOV_{\min}} x_2 + w_3 \frac{BL - BL_{\min}}{BL_{\max} - BL_{\min}} x_3 \\ &+ w_4 \frac{FR_{\max} - FR}{FR_{\max} - FR_{\min}} x_4 + w_5 \frac{Weight_{\max} - Weight}{Weight_{\max} - Weight_{\min}} x_5 + w_6 \frac{Price_{\max} - Price}{Price_{\max} - Price_{\min}} x_6 \end{aligned} \right\} \quad (7.20)$$

$$R = \sum_{i=1}^{10} x_i \sqrt{0.07 E \tau_a C A_t^* \frac{D_{in}^i f_{ob}^i \tau_{ob}^i \delta^i S^i}{\Phi_{\min}^i M^i}} \quad (7.21)$$

$$FOV = \sum_{i=1}^{10} FOV_i x_i \quad (7.22)$$

$$BL = \sum_{i=1}^{10} BL_i x_i \quad (7.23)$$

$$FR = \sum_{i=1}^{10} FR_i x_i \quad (7.24)$$

$$Weight = \sum_{i=1}^{10} Weight_i x_i \quad (7.25)$$

$$Price = \sum_{i=1}^{10} Price_i x_i \quad (7.26)$$

$$\sum_{i=1}^{10} x_i = k, \quad x_i \in \{0, 1\} \quad (7.27)$$

$$1 < k < 10 \quad (7.28)$$

$$\sum_{i=1}^6 w_i = 1, 0 \leq w_i \leq 1 \quad (7.29)$$

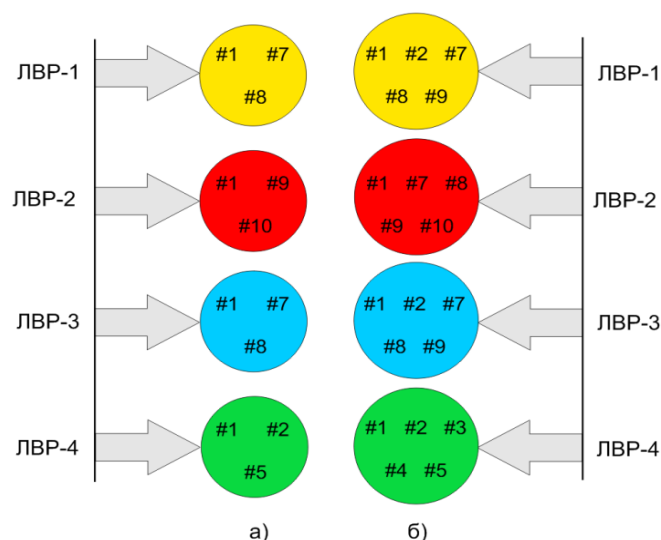
където w_i са тегловните коефициенти, изразяващи предпочитанията на ЛВР за важността на отделните параметри, разглеждани като критерии за оптимизация.

Използвани са четири различни набора от теглови коефициенти, изразяващи 4 различни предпочитания на ЛВР за относителната важност между критериите, както е показано в Таблица 7.9.

Таблица 7.9. Предпочитания на ЛВР

Предпочитания на ЛВР	Тегловни коефициенти					
	w_1	w_2	w_3	w_4	w_5	w_6
ЛВР-1	0.17	0.17	0.16	0.16	0.17	0.17
ЛВР-2	0.30	0.03	0.04	0.03	0.30	0.30
ЛВР-3	0.20	0.20	0.10	0.10	0.20	0.20
ЛВР-4	0.30	0.00	0.00	0.10	0.30	0.30

Първата комбинация от теглови коефициенти (ЛВР-1) изразява еднаква важност на всички критерии, другите комбинации на коефициентите (ЛВР-2/3/4) отразяват други предпочитания на ЛВР. Решенията на задачата (7.20) – (7.29), използвайки тегловните коефициенти от Табл. 7.6 и $k = 3$ и $k = 5$ (3 най-добри устройства и 5 най-добри устройства), са илюстрирани на Фиг. 7.4.

Фиг. 7.4. k най-добри ОНВ при различни предпочитания на ЛВР: а) $k = 3$; б) $k = 5$

От формална гледна точка, всяко Парето-оптимално решение е еднакво приемливо, тъй като е решение на многокритериалния оптимизационен проблем. На практика, само едно решение се избира като крайно решение и това се става чрез

субективния избор на ЛВР. За подпомагане на ЛВР при вземане на окончателното решение е предложен алгоритъм за оценяване на всяка от тези k най-добри алтернативи на устройства, спрямо едно “идеално” устройство (Borissova & Mustakeroov, 2013):

- *Стъпка 1:* Определя се “идеално” устройство с “идеални” параметри, т.е. устройство, чиито стойности на параметрите имат оптимални - съответно максимални или минимални стойности. Като се има предвид схемата за нормализация, то стойността на целевата функция на оптимизационния проблем (7.20) – (7.29) за това “идеално” устройство е равна на 1.
- *Стъпка 2:* Изчислява се стойността на целевата функция за всяка от получените k добри алтернативи на устройства.
- *Стъпка 3:* Определя се разликата между стойностите на целевите функции на “идеалното” устройство и на всяка алтернатива от определените k добри алтернативи и се изчислява относителната отдалеченост на устройствата до “идеалното” устройство в проценти.

Резултатите от изпълнението на описаната процедура за избраните k най-добри алтернативи, според различните ЛВР са показани Табл. 7.10, Табл. 7.11, Табл. 7.12 и Табл. 7.13.

Таблица 7.10. Сравнение на 5 най-добри устройства за ЛВР-1 с “идеалното” устройство

Избрани 5 най-добри ОНВ	Стойност на целевата функция	Относителна отдалеченост в %
#1	0.629	37.10
#8	0.589	41.01
#7	0.584	41.56
#9	0.494	50.60
#2	0.457	54.27

Таблица 7.11. Сравнение на 5 най-добри устройства за ЛВР-2 с “идеалното” устройство

Избрани 5 най-добри ОНВ	Стойност на целевата функция	Относителна отдалеченост в %
#1	0.4545	54.55
#9	0.3567	64.33
#10	0.3541	64.59
#8	0.3538	64.62
#7	0.3300	67.00

Таблица 7.12. Сравнение на 5 най-добри устройства за ЛВР-3 с “идеалното” устройство

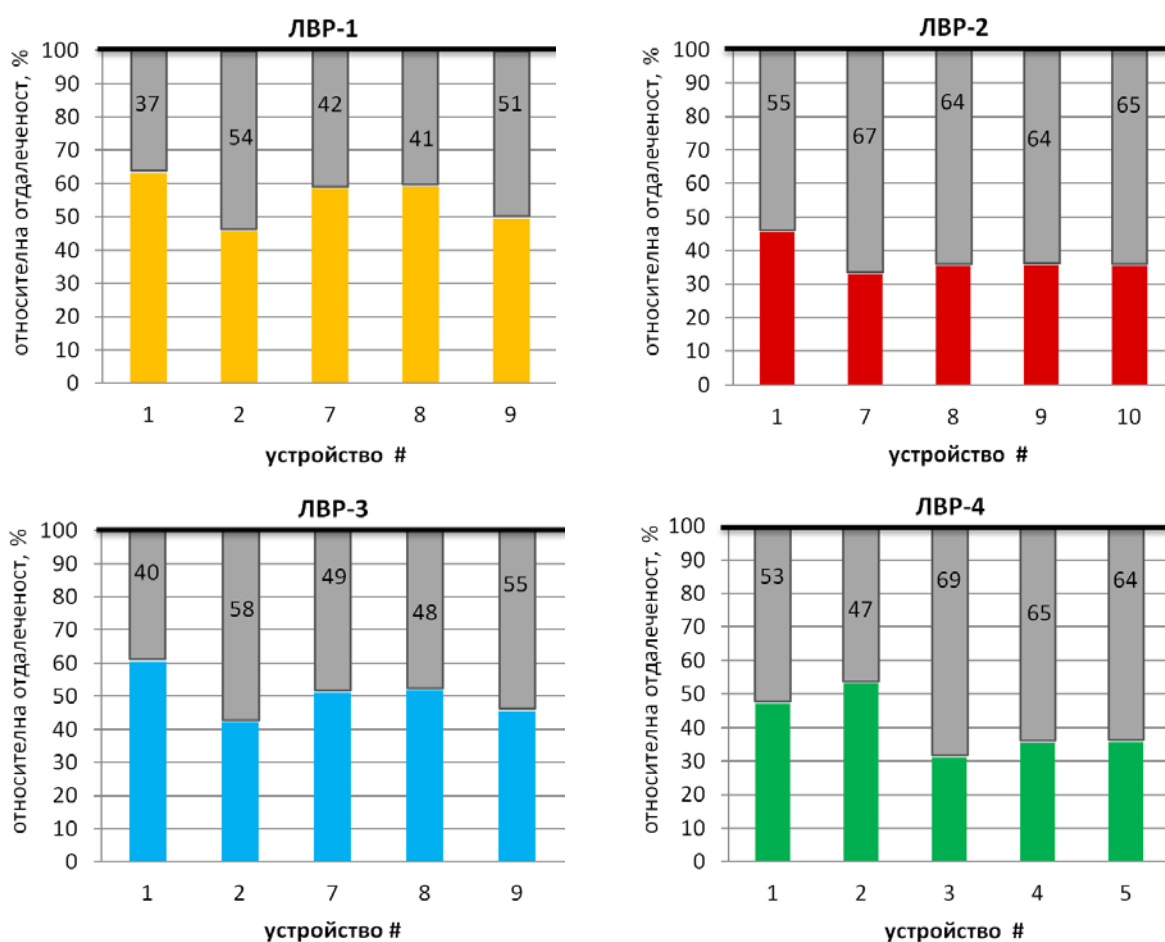
Избрани 5 най-добри ОНВ	Стойност на целевата функция	Относителна отдалеченост в %
#1	0.6037	39.63
#8	0.5175	48.25
#7	0.5110	48.90
#9	0.4539	54.61
#2	0.4204	57.96

Таблица 7.13. Сравнение на 5 най-добри устройства за ЛВР-4 с “идеалното” устройство

Избрани 5 най-добри ОНВ	Стойност на целевата функция	Относителна отдалеченост в %
#2	0.5306	46.94
#1	0.4723	52.77
#5	0.3578	64.22
#4	0.3550	64.50
#3	0.3116	68.84

Отдалечеността от “идеалното” устройство на определените чрез предложения алгоритъм 5 най-добри устройства, при различните предпочитания на ЛВР, са илюстрирани графично на Фиг. 7.5.

Както се вижда от Фиг. 7.5, близостта на устройството до “идеалното” устройство силно зависи от предпочитанията на ЛВР. Тази информация може да се използва от ЛВР при окончателния избор на конкретно устройство, което може да се приеме като един добър и рационален избор на устройство. Формулираният оптимизационен модел позволява получаване на числени резултати за времена на решаване от порядъка на няколко секунди.



Фиг. 7.5. Отдалеченост на 5 най-добри устройства при различни предпочитания на ЛВР

7.4. Избор на алтернатива с отчитане влиянието на околната среда

Често пъти възниква проблем за оценка и избор, който избор е зависим не само от параметри на алтернативата, но и от външни условия. При този избор се търси такава алтернатива, която да удовлетворява както определени параметри на системата/устройството, така и да функционира при зададени външни условия. Всичко това води до нетривиален процес на вземане на решения. Поради комбинаторния характер на този избор, включващ отчитане на множество параметри, както и на външни условия, е обосновано той да се базира на математически количествен анализ. Така описаният проблем за избор на алтернатива с отчитане влиянието на околната среда може да бъде илюстриран на примера за избор на УНВ, чиито параметри могат да бъдат представени като функция на параметрите на използваните модули и на външните условия на наблюдение. При това е необходимо е да се вземат предвид и предпочитанията на потребителя по отношение на важността на отделните параметри на УНВ. Известно е, че в каталожните данни на устройствата са параметрите са тествани за определени (лабораторни) условия на наблюдение. От друга страна реалните външни условия за наблюдение – естествена нощна осветеност, пропускане на атмосферата, контраст между фона и наблюдавания обект, и тип на самия наблюдаван обект, пряко влияят на разстоянието на действие на УНВ (Borissova и др., 2006). От друга страна, различните разновидности на разстоянието на действие (откриване, разпознаване и идентификация) съществено зависи от размерите на наблюдаваните обекти. Всичко това изисква, при избора на УНВ, да бъдат взети под внимание и външните условия на наблюдение, при които той ще бъде използван.

7.4.1. Многокритериален модел за избор на уреди за нощно виждане с отчитане на външните условия на наблюдение

Най-общо казано, оценката на ефективността на избора на дадена алтернатива (инженерна система, техническо устройство или уред) се базира на множество функционални характеристики. Сложността на този избор може да бъде преодоляна чрез използване на многокритериалните методи за вземане на решения. Вземайки предвид предвид подобни съображения, е формулиран математически модел, в който разстоянието на действие на УНВ се разглежда като функция на параметрите на конкретното устройство и едновременно с това се отчитат и стойностите на предполагаемите външни условия на наблюдение (Borissova, 2008; Borissova & Mustakerov, 2008):

$$\text{maximize } P(x) = (P_1(x), P_2(x), \dots, P_J(x))^T, \quad (7.30)$$

$$\text{minimize } N(x) = (N_1(x), N_2(x), \dots, N_K(x))^T,$$

при ограничения

$$R(x) = \sum_{i=1}^I x_i \sqrt{0.07 E \tau_a K A_t^* \frac{D_{in}^i f_{ob}^i \tau_{ob}^i \delta^i S^i}{\Phi_{min}^i M^i}} \quad (7.31)$$

$$P_j(x) = \sum_{i=1}^I P_{ij} x_i, j = 1, 2, \dots, J, \quad (7.32)$$

$$N_k(x) = \sum_{i=1}^I N_{ik} x_i, k = 1, 2, \dots, K, \quad (7.33)$$

$$\sum_{i=1}^I x_i = 1, x_i \in \{0, 1\} \quad (7.34)$$

където R – разстояние на действие, $P_1(x), \dots, P_J(x)$ – параметри, които се максимизират; $N_1(x), N_2(x), \dots, N_K(x)$ – параметри, които се минимизират; P_{ij} и N_{ik} са стойности на параметрите на всяко едно устройство; $x = (x_1, x_2, \dots, x_I)$ е вектор от двоични целочислени променливи, присвоени към всяко устройство, участващо в избора. Така формулираният модел позволява да се добавят и други изисквания към параметрите на избраното устройство, както и други допълнителни критерии или параметри, за да се отразят по-точно специфичните изисквания на отделните потребители.

7.4.2. Числено тестване

Формулираният многокритериален модел за избор на УНВ (7.30) – (7.34) е тестван на базата на данни за реални УНВ, показани в Таблица 7.14, при два варианта на външните условия на наблюдение, показани Таблица 7.15.

Таблица 7.14. Параметри на УНВ

No	УНВ	Разделителна способност, lp/mm	Обектив	Фокусно разстояние на обектива, mm	Пропускане на обектива	Сигнал/шум	Чувствител- ност, A/lm	Зрително поле, deg	Тегло gr	Цена \$
1	Night Optics D-2MV	40	1:1.2	26	0.78	12	0.00024	40	482	650
2	Rigel 3250	30	1	35	0.78	12	0.00022	30	430	699
3	ATN Cougar-2	32-40	1:1.4	35	0.78	16	0.00031	30	800	3071
4	ATN Cougar CGTI	40-51	1:1.4	35	0.78	15	0.00035	30	800	3696
5	ATN Night Cougar-3	64	1:1.4	35	0.78	20	0.00087	30	800	4884
6	ATN Night Cougar-4	68	1:1.4	35	0.8	25	0.00115	30	800	9932
7	ATN PS-23-2	36-45	1:1.2	24	0.8	13	0.0007	40	700	3550
8	ATN PS-23- CGT	45-54	1:1.2	24	0.8	17	0.0011	40	700	4195
9	ATN PS-23-3	55-72	1:1.2	24	0.8	22	0.0016	40	700	5895
10	ATN PS-23-4	64-72	1:1.2	24	0.8	24	0.0019	40	700	12995

Таблица 7.15. Външни условия на наблюдение

Външни условия на наблюдение	Естествена осветеност, lx	Пропускане на атмосферата	Контраст	Редуцирана площ на наблюдавания обект, m ²
(1)	0.01 (¼ луна)	0.75	0.20	0.758
(2)	0.001 (звездно небе)	0.80	0.30	

Многокритериалният модел (7.30) – (7.34) и данните от Табл. 7.14 и Табл. 7.15 са използвани за формулиране на съответни многокритериални оптимизационни задачи. Като метод за решаване на многокритериалните задачи е използван *методът на претеглената сума* с тегловни коефициенти, изразяващи предпочитанията на ЛВР към параметрите на УНВ, както са показани в Таблица 7.16.

Таблица 7.16: Теглови коефициенти

ЛВР	Разстояние на действие, w ₁	Зрително поле, w ₂	Цена, w ₃	Тегло, w ₄
ЛВР-1	0.50	0.10	0.30	0.10
ЛВР-2	0.25	0.20	0.50	0.05

Комбинацията от коефициенти на ЛВР-1 изразява теглата на критериите, подредени по важност в следната последователност: разстояние на действие; цена; зрително поле и тегло на уреда. Втората комбинацията от коефициенти (ЛВР-2) определя друга подредба по важност на параметрите на УНВ, а именно: цена; разстояние на действие; зрително поле и тегло. Други конкретните предпочитания на различните потребители към важността на параметрите на УНВ, разглеждани като критерии, могат да бъдат взети предвид чрез задаване на други стойности на тегловните коефициенти.

Трансформираната многокритериална задача, съгласно *метода на претеглената сума*, има вида (Borisova, 2008):

$$\max (w_1 R^*(x) + w_2 FOV^*(x) + w_3 C^*(x) + w_4 W^*(x)) \quad (7.35)$$

при ограничения

$$R^*(x) = \left(\sum_{i=1}^{10} x_i \sqrt{0.07 E \tau_a K A_t^* \frac{D_{in}^i f_{ob}^i \tau_{ob}^i \delta^i S^i}{\Phi_{min}^i M^i}} - R_{min} \right) (R_{max} - R_{min})^{-1} \quad (7.36)$$

$$FOV^*(x) = \left(\sum_{i=1}^{10} FOV_i x_i - FOV_{min} \right) (FOV_{max} - FOV_{min})^{-1} \quad (7.37)$$

$$C^*(x) = \left(C_{max} - \sum_{i=1}^{10} C_i x_i \right) (C_{max} - C_{min})^{-1} \quad (7.38)$$

$$W^*(x) = \left(W_{max} - \sum_{i=1}^{10} W_i x_i \right) (W_{max} - W_{min})^{-1} \quad (7.39)$$

$$\sum_{i=1}^{10} x_i = 1, \quad x_i \in \{0, 1\} \quad (7.40)$$

$$\sum_{i=1} w_i = 1, \quad 0 \leq w_i \leq 1, \quad (7.41)$$

където w_i са тегловните коефициенти, а $R^*(x)$, $FOV^*(x)$, $C^*(x)$ и $W^*(x)$ са нормализираните критерии.

Получените резултати от решаването на оптимизационните задачи са показани в Таблица 7.17.

Таблица 7.17. Получени резултати за избраното устройство

Външни условия на наблюдение	Тегловни коефициенти	Разстояние, m	Зрително поле, $degree$	Цена, \$	Тегло, gr	Избран УНВ от Табл. 4.12
(1)	ЛВР-1	748	40	5895	700	No 9
	ЛВР-2	335	40	650	482	No 1
(2)	ЛВР-1	187	40	5895	700	No 9
	ЛВР-2	84	40	650	482	No 1

В дадения случай, различните комбинации от външни условия на наблюдение не влияят върху избора на устройство, но определят различни разстояния на действие. Ако потребителят не е удовлетворен от изчислената теоретична оценка на разстоянието на действие, може да въведе долна граница за този параметър и задачата да бъде решена отново. Например, при избрана долна граница равна на 500 m, ще се добави ограничение от вида

$$R \geq 500 \quad (7.42)$$

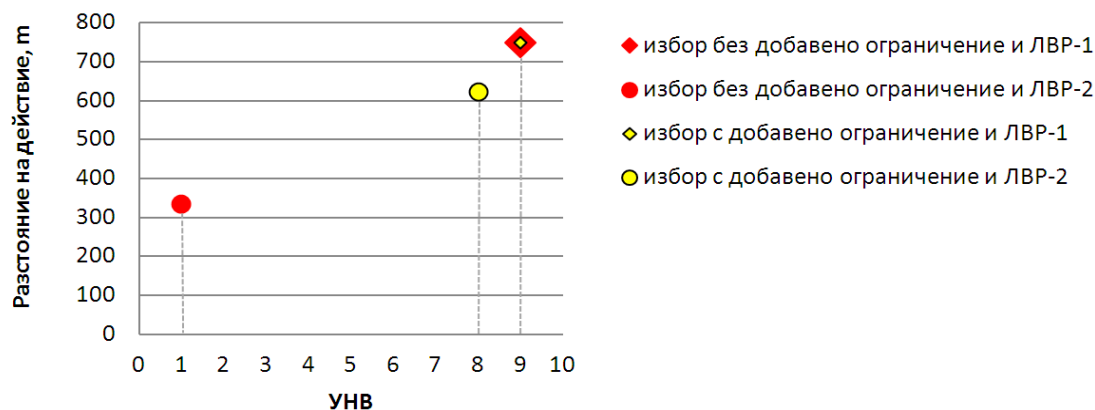
и резултатите от решаването на задачата с това ограничение ще бъдат, както са показани в Таблица 7.18.

Таблица 7.18. Резултати при добавено ограничение за разстоянието на действие

Външни условия на наблюдение	Теглови коефициенти	$R \geq 500, m$	Зрително поле, $degree$	Цена, \$	Тегло, gr	Избран УНВ от Табл. 5.6
(1)	ЛВР-1	748	40	5895	700	No 9
	ЛВР-2	623	40	4195	700	No 8

Използването на предпочитанията на ЛВР-1 при комбинация от външни условия на наблюдение (1) удовлетворява ограничение (7.42) и решението е избор на устройство No 9. Използването на предпочитанията на ЛВР-2 при същата комбинация от външните условия на наблюдение (1) и допълнителното ограничение (7.42) вече

водят до различен избор на устройство (No 8). Избраните устройства, показани в Табл. 7.17 и Табл. 7.18, при отсъствие и наличие на допълнително ограничение за разстоянието на действие и при различни потребителски предпочитания (ЛВР-1 и ЛВР-2) са показани графично на Фиг. 7.6.



Фиг. 7.6. Избор на УНВ при отсъствие и наличие на допълнително ограничение за разстоянието на действие и при различни предпочитания на ЛВР

От Фиг. 7.6 се вижда как различните външни условия на наблюдение заедно с потребителските предпочитания влияят върху избора на конкретно устройство. Например, сравнявайки избраното устройство в случай на предпочитания ЛВР-2, за избор без дабавени допълнителни ограничения, води до избор на устройство # 1, докато модифицираният проблем с добавено допълнително ограничение за разстоянието на действие по-голямо от 500 m, при същите външни условия на наблюдение, води до избор на устройство # 8. От формална гледна точка, всяко Парето-оптимално решение е еднакво приемливо, тъй като е решение на многокритериалната задача, но на практика, ЛВР избира само едно решение като крайно решение.

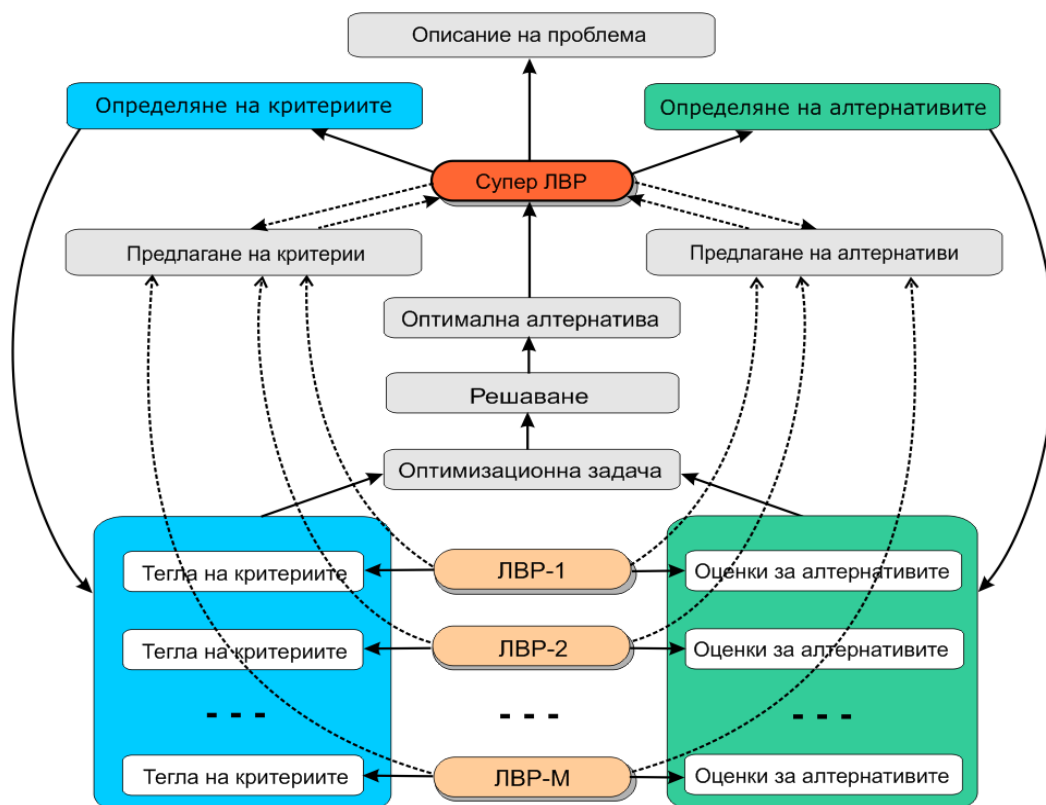
Поради естеството на задача за дискретен комбинаторен избор на УНВ е възможно да има ситуация, при която да не е възможно удовлетворяването на някои ограничения. В тези случаи се налага да бъдат модифицирани тези ограничения с такива стойности, които да позволяват да се получи решение на формулираните оптимизационни задачи. Подобни модификации могат да бъдат подпомогнати чрез използване на следоптимизационен анализ, предоставян от повечето софтуерни пакети за оптимизация. Накрая може да се отбележи, че изчислителната трудност на подобни задачи позволява получаване на решения за времена от порядъка на секунди. Численият пример потвърждава практическата приложимост на предложения модел за избор на алтернатива като едновременно с това се отчитат и предполагаемите стойностите на външните условия, които имат значение за този избор.

7.5. Избор на алтернатива чрез групово вземане на решения

Присъствието на субективен фактор във всяко човешко решение се обяснява с различната степен, в която различните индивиди притежават определени качества, имащи пряко отношение към решаването на проблема (аналитичност, импулсивност и т.н.). Затова много често няколко ръководители, вземащи решение за една и съща ситуация, породена от един и същ обект, достигат до различни резултати. Ако ликвидирането на субективния фактор е почти невъзможно, то неговото минимизиране трябва да е винаги стремеж на добрия мениджмънт. Това може да стане, когато се използват повече мнения, т.е. когато се прилага групов подход за вземане на решения. При вземане на важни решения, ръководителят трябва да се основава не само на фактите, но и на мненията на всички (или на повечето) засегнати лица, защото в много случаи един или друг факт може да заблуждава, ако не е взет в правилния контекст или когато липсват основни и съществени факти.

Вземането на решения е процес, резултатът от който е избор на една единствена алтернатива от множество налични алтернативи. В организационен контекст, крайното решение се взема от управителя или от група от мениджъри с опит, които се стремят да постигнат на целите на организацията. Известно е, че всяко лице, вземащо решение, има свои специфични начин за използване на наличната информация. Някои ръководители вземат решенията въз основа на количествен анализ, докато други предпочитат да доверят на интуицията, натрупана на база на опита (Curtis & Cobham, 2008). Вземайки предвид горните съображения е предложен алгоритъм, описващ процеса на груповото вземане на решения както е показано на Фиг. 7.7 (Mustakerov & Borissova, 2014).

Отличителна особеност на груповото вземане на решения е, че в повечето случаи съществува лице, т. нар. *супер ЛВР*, определено от ръководителя, на което са възложени специфични отговорности. Това лице е отговорно за организиране на процеса на груповото вземане на решения – определяне на основните фактори (критерии) за оценка и възможните алтернативи (Фиг. 7.7). Обикновено тази отговорност може да се поеме от мениджъра, който ще вземе крайното решение. Определянето броя на алтернативите и съответните критерии за тези алтернативи, се извършва с помощта на група от лица, имащи отношение към вземането на даденото решение (ЛВР).



Фиг. 7.7. Алгоритъм за групово вземане на решения

Характерна особеност на разглеждания процес на групово вземане на решения е, че всеки участник в групата определя свои тегла за оценяваните критерии, както и съответни оценки за всеки критерий и за всяка алтернатива. Това може да се покаже като обобщена структура на матрица на алтернативите при груповото вземане на решение, показана в Таблица 7.19.

Таблица 7.19 Матрица на алтернативите при групово вземане на решение

Критерий	Тегла			Алтернатива 1			Алтернатива 2			...	Алтернатива J		
	ЛВР ¹	...	ЛВР ^K	ЛВР ¹	...	ЛВР ^k	ЛВР ¹	...	ЛВР ^k		ЛВР ¹	...	ЛВР ^k
c_1	w_1^1	...	w_1^K	$a_{1,1}^1$	...	$a_{1,1}^K$	$a_{1,2}^1$	...	$a_{1,2}^K$	...	$a_{1,J}^1$	...	$a_{1,J}^K$
c_2	w_2^1	...	w_2^K	$a_{2,1}^1$	...	$a_{2,1}^K$	$a_{2,2}^1$	...	$a_{2,2}^K$	...	$a_{2,J}^1$	...	$a_{2,J}^K$
...	...	...	...	...	...	...	...	...	...	...	...	...	...
c_M	w_M^1	...	w_M^K	$a_{M,1}^1$	...	$a_{M,1}^K$	$a_{M,2}^1$	...	$a_{M,2}^K$	...	$a_{M,J}^1$	...	$a_{M,J}^K$

Критериите за всяка алтернатива са означени чрез c_i ($i = 1, 2, \dots, M$), групата от ЛВР са означени с $ЛВР^k$ ($k = 1, 2, \dots, K$), тегловите коефициенти (изразяващи относителната важност на критериите) на всеки ЛВР за съответните критерии са означени с w_i^k и с a_{ij}^k са изразени оценъчните точки за всяка алтернатива j ($j = 1, 2, \dots, J$)

и съответстващия ѝ критерий c_i . По-голяма стойност на оценъчните точки означава по-добро представяне и следователно най-добрата алтернатива ще бъде тази, която получи най-много точки.

7.5.1. Обобщен модел на групово вземане на решения

На база на описаната структура на матрицата на алтернативите за групово вземане на решения е формулиран комбинаторен оптимизационен модел (Mustakeroov & Borissova, 2014):

$$\text{maximize } \sum_{i=1}^M \sum_{k=1}^K w_i^k A_i^k \quad (7.43)$$

при ограничения

$$\forall i = 1, 2, \dots, M : (\forall k = 1, 2, \dots, K : A_i^k = \sum_{j=1}^J a_{i,j}^k x_j) \quad (7.44)$$

$$\sum_{j=1}^J x_j = 1, \quad x_j \in \{0, 1\} \quad (7.45)$$

където двоичните целочислени променливи x_j са използвани за избор на алтернатива.

Така дефинирания оптимизационен модел (7.43) – (7.45) позволява формулиране на еднокритериална задача за избор на най-добра алтернатива. Решението на тази задача ще определи най-добрата алтернатива, като се вземат предвид едновременно както съответните теглови коефициенти за критериите, определени от ЛВР, така и съответните им оценки.

7.5.2. Числено тестване

За да се илюстрира практическата приложимост на предложения модел за избор на алтернатива чрез групово вземане на решение е използван реален проблем, описан в (Kgarohla, 2013). Група от 6 ЛВР трябва да вземе решение относно избор на софтуер, за който са представени 3 алтернативи, като всяка алтернатива се оценява по 19 критерия. За всеки критерий ЛВР определя съответно тегло, изразяващо важността на критерия, както е показано в Таблица 7.20.

Таблица 7.20 Критерии и теглови коефициенти на всеки ЛВР за всеки критерий

Критерий	ЛВР-1	ЛВР-2	ЛВР-3	ЛВР-4	ЛВР-5	ЛВР-6
c_1	2	4	8	8	6	10
c_2	10	8	8	8	8	4
c_3	10	10	8	4	8	10
c_4	6	2	6	6	6	6
c_5	10	4	10	10	10	8
c_6	10	2	8	6	6	8
c_7	6	6	6	8	8	6
c_8	10	8	10	8	8	10
c_9	6	10	8	10	10	8
c_{10}	4	2	8	6	6	6
c_{11}	8	8	10	10	10	10
c_{12}	10	4	10	6	8	10
c_{13}	10	2	10	6	8	10
c_{14}	8	6	8	10	10	8
c_{15}	4	8	6	8	10	10
c_{16}	4	4	8	10	10	10
c_{17}	6	6	10	10	10	6
c_{18}	6	8	8	10	10	6
c_{19}	4	2	6	10	10	4

Освен това, всяка алтернатива се оценява от ЛВР спрямо всеки критерий по скалата от 0 до 10, като са допустими само цели числа. Ако алтернативата няма принос към даден критерий, нейната оценка е 0 и обратно, ако нейният принос е значителен за изпълнение на друг критерий, тя може да има оценка 10. Освен това, тъй като алтернативите се оценяват спрямо всеки от критериите, то е възможно оценките за дадена алтернатива спрямо някои критерии да получат еднакви точки – Таблица 7.21.

Таблица 7.21. Оценки на алтернативите от ЛВР

Кри- терий	Оценки за Алтернатива 1						Оценки за Алтернатива 2						Оценки за Алтернатива 3					
	ЛВР1	ЛВР2	ЛВР3	ЛВР4	ЛВР5	ЛВР6	ЛВР1	ЛВР2	ЛВР3	ЛВР4	ЛВР5	ЛВР6	ЛВР1	ЛВР2	ЛВР3	ЛВР4	ЛВР5	ЛВР6
c_1	2	4	6	1	1	1	10	5	8	9	9	9	8	4	6	8	9	2
c_2	9	8	7	7	6	7	9	8	7	1	2	10	3	1	5	1	2	10
c_3	9	2	8	7	6	7	10	4	7	10	9	10	6	5	7	8	8	10
c_4	10	6	4	4	4	3	10	2	8	7	8	8	8	8	8	6	7	2
c_5	3	1	9	1	2	1	5	10	7	3	2	7	3	7	2	8	9	9
c_6	4	5	3	1	2	1	10	5	7	8	7	7	5	9	2	7	7	3
c_7	8	9	4	2	2	8	9	6	10	7	5	10	7	4	6	7	8	8
c_8	6	4	6	2	2	2	8	7	8	7	5	10	5	10	6	7	8	3
c_9	5	6	8	2	2	8	6	10	6	2	1	5	6	5	6	5	7	9
c_{10}	2	10	8	1	1	1	4	8	8	7	6	9	10	10	6	9	9	9
c_{11}	6	5	6	1	1	1	8	9	8	8	9	7	7	4	2	7	8	2
c_{12}	2	8	2	1	1	1	10	1	7	6	6	8	6	6	5	6	6	5
c_{13}	1	1	2	1	1	1	8	2	8	7	6	9	5	10	5	7	6	5
c_{14}	3	6	2	1	1	1	9	5	8	7	8	10	4	4	2	6	6	7
c_{15}	5	4	2	1	1	1	8	10	6	8	8	10	4	8	2	8	7	8
c_{16}	7	8	1	1	1	1	7	9	9	8	7	9	7	10	1	7	8	7
c_{17}	10	9	10	7	5	10	3	8	4	5	4	5	5	9	2	8	8	7
c_{18}	4	4	1	1	1	1	2	6	1	1	1	1	1	10	1	1	2	1
c_{19}	4	2	1	1	2	1	8	8	6	7	6	8	3	8	2	6	5	6

Като се вземе предвид обобщената формулировка на модела (7.43) – (7.45) и информацията от Табл. 7.20 и Табл. 7.21, е формулирана следната оптимизационна задача (Mustakerov & Borissova, 2014):

$$\max \left(\begin{array}{l} w_1^1 A_1^1 + w_2^1 A_2^1 + \dots + w_{19}^1 A_{19}^1 + \\ + w_1^2 A_1^2 + w_2^2 A_2^2 + \dots + w_{19}^2 A_{19}^2 + \\ \dots \\ + w_1^6 A_1^6 + w_2^6 A_2^6 + \dots + w_{19}^6 A_{19}^6 \end{array} \right) \quad (7.46)$$

при ограничения

$$A_1^1 = (a_{1,1}^1 x_1 + a_{1,2}^1 x_2 + a_{1,3}^1 x_3) \quad (7.47)$$

$$A_2^1 = (a_{2,1}^1 x_1 + a_{2,2}^1 x_2 + a_{2,3}^1 x_3) \quad (7.48)$$

• • • • •

$$A_{19}^1 = (a_{19,1}^1 x_1 + a_{19,2}^1 x_2 + a_{19,3}^1 x_3) \quad (7.65)$$

$$A_1^2 = (a_{1,1}^2 x_1 + a_{1,2}^2 x_2 + a_{1,3}^2 x_3) \quad (7.66)$$

$$A_2^2 = (a_{2,1}^2 x_1 + a_{2,2}^2 x_2 + a_{2,3}^2 x_3) \quad (7.67)$$

$$\dots\dots\dots (\dots)$$

$$A_{19}^2 = (a_{19,1}^2 x_1 + a_{19,2}^2 x_2 + a_{19,3}^2 x_3) \quad (7.84)$$

$$\dots\dots\dots (\dots)$$

$$A_1^6 = (a_{1,1}^6 x_1 + a_{1,2}^6 x_2 + a_{1,3}^6 x_3) \quad (7.132)$$

$$A_2^6 = (a_{2,1}^6 x_1 + a_{2,2}^6 x_2 + a_{2,3}^6 x_3) \quad (7.133)$$

$$\dots \qquad \qquad \qquad (\dots)$$

$$A_{19}^6 = (a_{19,1}^6 x_1 + a_{19,2}^6 x_2 + a_{19,3}^6 x_3) \quad (7.150)$$

$$x_1 + x_2 + x_3 = 1, x_i \in \{0, 1\} \quad (7.151)$$

Така формулираната задача е решена чрез софтуерната система LINGO v.12 и полученото решение определя като най-добър избор, алтернатива 2. За разлика от подхода в (Krapohl, 2013), използването на модела (7.43) – (7.45) води до решаване на оптимизационна задача (7.46) – (7.151), с което се гарантира, че полученото решение е оптимално.

Друго предимство на предложения оптимизационен подход, е лесното му модифициране и разширяване, когато е необходимо да се вземат предвид и други аспекти на груповото вземане на решения. Например, когато е необходимо да бъдат

отчитани компетентностите на участниците в групата за вземане на решение. В този случай, описаният модел (7.43) – (7.45) се модифицира до следната формулировка:

$$\text{maximize } \sum_{i=1}^M \sum_{k=1}^K \alpha^k w_i^k A_i^k \quad (7.152)$$

при ограничения

$$\forall i = 1, 2, \dots, M : (\forall k = 1, 2, \dots, K : A_i^k = \sum_{j=1}^J a_{i,j}^k x_j) \quad (7.153)$$

$$\sum_{j=1}^J x_j = 1, \quad x_j \in \{0, 1\} \quad (7.154)$$

$$\alpha^k \in (0, 1) \quad (7.155)$$

където допълнително са въведени α^k като коефициенти, изразяващи компетентността на всеки ЛВР.

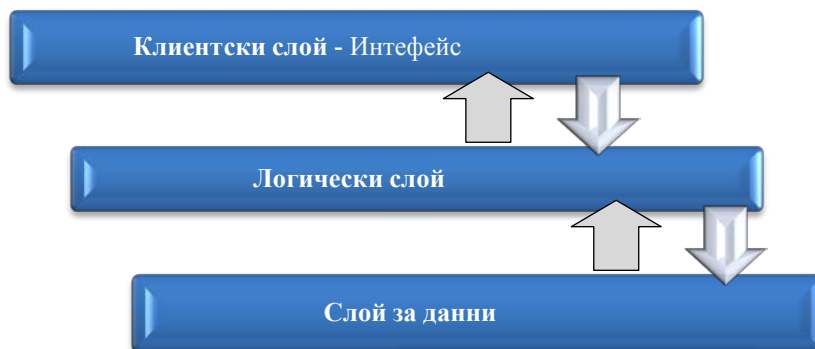
Глава 8:

Софтуерни инструменти за приложения в инженерни системи

В тази глава са описани софтуерни инструменти, реализиращи описаните в предните глави модели, методи и алгоритми. Част от тези инструменти са разработени като напълно функционални CAD системи, докато друга част са изследователски прототипи, позволяващи апробирането на предложените методи. Софтуерните инструменти са реализирани като уеб-базирани системи за подпомагане проектирането на модулни инженерни системи при проектирането на ПК, проектирането на УНВ, проектирането на ВЕП и при груповото вземане на решение. Друга част от разработените инструменти са специализирани софтуерни системи, използващи програмни библиотеки (API) на комерсиални оптимизационни системи (LINDO), както и оптимизационни модули, съвместими с електронни таблици за определяне на оптимални разписания, за оптимално проектиране на ВЕП и определяне на оптимално линейно разкрояване. Получените резултати са публикувани в статии D2, D3, D7, D8, D24, D28.

8.1. Уеб-базирани системи за подпомагане проектирането на модулни инженерни системи

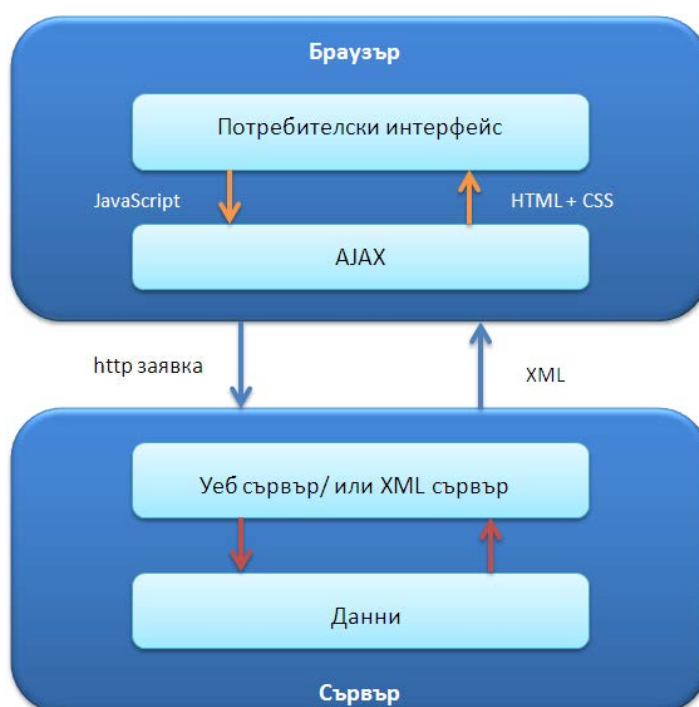
При разработването на уеб базирани системи е използвана широко използвана трислойна архитектура, показана на Фиг. 8.1.



Фиг. 8.1. Трислойна архитектура на уеб приложение

При тази архитектура се разграничат три ясно дефинирани слоя, всеки от които може да е разположен на отделна платформа: *Клиентският слой* отговаря за представянето на информацията пред потребителя и осигурява формирането на различните заявки към системата. Той се реализира като потребителския интерфейс, осигуряващ достъп до приложението в средата на браузер. Този слой осигурява взаимодействието с потребителя и изпращането на заявки към приложния слой. *Логическият слой* осигурява управлението на обмена на данни, реализирайки логиката на приложението, обслужващо клиента (функционална логика). Реализира се като набор от модули, всеки със специфично предназначение, под управлението на сървър на приложенията. *Слоят за данните* съхранява информацията, която се предава от и към логическия слой за обработка и след това се предава обратно към потребителя. Въведените данни за параметрите на модулите се използват за да се направят съответните изчисления, които да са визуализират. Като източник на данни може да се ползва един или няколко разпределени сървъри в мрежата.

При разработването на уеб приложенията е използвана AJAX технологията, (Asynchronous JavaScript and XML) основаваща се на два основни принципа: използване DHTML за динамично променяне на съдържанието на приложението и използване на XMLHttpRequest команда на JavaScript за достъп до сървъра "в движение", без да се рестартира цялото приложение (Фиг. 8.2).



Фиг. 8.2. AJAX технология за уеб приложения

AJAX технологията представлява нов начин за използване на съществуващите технологии. AJAX комбинира по ефективен начин разпространените уеб технологии: HTML, JavaScript, DOM и CSS (Борисова, 2014). На практика, AJAX технологията се реализира чрез използването на JavaScript функции, които се извикват при необходимост от актуализиране на информация от сървъра (Hertel, 2007). Благодарение на възможността за актуализиране само на отделни компоненти на уеб приложението, се създават нови модели за взаимодействие за реализиране на необходимата интерактивност с потребителя.

8.1.1. Система за подпомагане проектирането на персонални компютри

Опиисаният в Глава 2 оптимизационен подход към проектиране на модулна техническа система с ярко изразена зависимост между модулите, е илюстриран чрез уеб-приложение за конфигуриране на персонални компютри (ПК). Един от най-важните проблеми при изграждането на всяка софтуерна система е проектирането на нейния потребителски интерфейс. Целта е да се предложи опростен и интуитивен графичен интерфейс, който да подпомага потребителя при реализиране на приложените алгоритми. Основната идея при разработване на графичния интерфейс трябва да следва същата логика, използвана при разработването на алгоритмичните реализации, т.е. графичният интерфейс е в тясна връзка със съответните алгоритми.

Описаните в Глава 2 алгоритми (A1, A2, A3, A4, A5a, A5b и A6) са използвани при реализирането на уеб-приложението за подпомагане конфигурирането на ПК. Графичният потребителски интерфейс на това приложение е показан на Фиг. 8.3 (Mustakerov & Borissova, 2013).

Основната концепция на този интерфейс е използването на т. нар. “падащи” менюта за избор на модули. При зареждане на уеб приложението, списъците в “падащите” менюта се инициализират с всички налични модули. Когато конкретен модул е избран от списъка с модули, “падащите” списъци за другите модули се актуализират така, че да показват само съвместимите модули с избрания вече модул. Тази функция на системата е отражение на идеята за съвместимост между модулите. След избор на конкретен модул, неговите параметри се показват в съответните текстови полета. При активирането на бутона “calculate” се определят параметрите на системата и общата ѝ цена. Друга характеристика на системата е възможността за редактиране на текстовите полета, съдържащи параметри за модулите.

pc.html - Avant Browser

File View Bookmarks AutoFill Tools Help

File:///pc.html

pc.html

PC design via modular configuration

MB CPU RAM RAM number

MB		CPU		RAM	
Socket	LGA775	Socket	LGA775	RAM Capacity	2
RAM type	DDR2	Clock Speed	3,06	RAM Frequency	800
RAM capacity	16	Core Number	2	Price	77,5
Price	178,5	Price	251		

RAM capacity Modules price

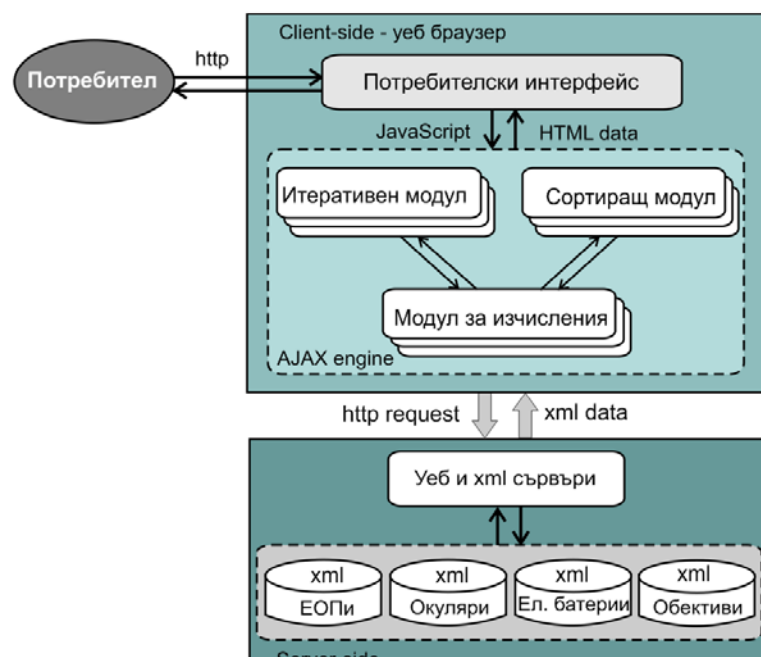
Фиг. 8.3. Интерфейс на уеб приложението за проектиране конфигурацията на ПК

Рационалният избор се реализира чрез предоставяне на възможност на потребителя да въведе желана стойност за някой параметър на проектираната конфигурация и след активиране на бутона “calculate” системата да намери такава комбинация от модули, които да удовлетворяват въведеното ограничение. Ако това не е възможно, се показва съобщение и потребителят трябва да вземе друго решение .

8.1.2. Система за подпомагане проектирането на уреди за нощно виждане (очила, прицели и бинокли)

Описаният в Глава 3 алгоритъм за подпомагане проектирането на различни типове УНВ (очила, прицели и бинокли), отчитащи и външните условия за наблюдение, е реализиран в уеб базирано приложение. Архитектурата на този софтуерен инструмент е показана на Фиг. 8.4 (Borissova & Mustakerov, 2013).

И в този случай използваната архитектура е базирана на технологията AJAX. Използването на AJAX води до динамично променяне на само тази част от уеб съдържанието, която трябва да се обнови и само когато това е необходимо, без да се налага презареждане на цялото съдържание.



Фиг. 8.4. Архитектура на веб приложението за подпомагане проектирането УНВ

Потребителският интерфейс на веб базираното приложение, подпомагащо проектирането на уреди за нощно виждане е показан на Фиг. 8.5 (Borissova и др, 2013).

IIT	Objective	Ocular	El. batteries	El. batteries number	Target type	Ambient light	Contrast	Atm. transmittance
IIT	Limiting resolution, [lp/mm]				Objective	Focal lenght, [mm]		
	Luminous sensitivity, [μA/lm]					F-number		
	Signal/noise ratio					Transmittance		
	Weight, [gr]					Field of view, [degree]		
	Price, [\$]					Focus range, [mm]		
Ocular	Diopter adjustment, [dpt]					Weight, [gr]		
	Eye relief, [mm]					Price, [\$]		
	Focal lenght, [mm]				Battery	Capacity, [mA/h]		
	Weight, [gr]					Number		
	Price, [\$]					Price, [\$]		
Reset					Calculation			
NVD detection range, [m]					El. battery power supply lifetime duration, [h]			
NVD recognition range, [m]					El. battery number			
NVD identification range, [m]					Weight of the NVD optoelectronic channel, [gr]			
NVD magnification, [x]					Price of the NVD optoelectronic channel, [\$]			

Фиг. 8.5. Потребителски интерфейс на приложението за проектиране на УНВ

Положени са усилия потребителския интерфейс да е интуитивен и дружелюбен при задаването на данни. Чрез “радио” бутони се избира типът на проектирания уред за нощно виждане – очила, бинокъл или прицел. Една част от менюта от типа «падащи списъци» са предназначени за избор на основните модули за УНВ (ЕОП, обектив със съответен окуляр, ел. батерии), а друга част – за избор на различни външни условия за наблюдение (наблюдаван обект, естествена нощна осветеност, контраст и пропускане на атмосферата). Изборът на конкретен модул автоматично зарежда неговите параметри в предвидените за целта текстови полета. При изчисляването на параметрите на УНВ – 3 разновидности на разстоянието на действие, увеличението, теглото, цената и времето за непрекъсната работа, се използват описаните в т. 3.2.1 и 3.2.2 модели на процеса на проектиране на УНВ.

За реализиране метода на рационалното проектиране на УНВ е предвидена възможност за въвеждане на изисквания относно параметрите на проектираното устройство. Основната идея е да се намери такава допустима комбинация от модули, която да отговаря на зададените потребителски изисквания. Алгоритмичната реализация на рационалния избор се основава на определяне на параметрите на УНВ за всички възможни комбинации от модули, и да избере такава комбинация, която отразява зададените предпочитания на потребителя. Ако такава комбинация не съществува, потребителят получава съобщение, че изискването не може да бъде изпълнено. В този случай потребителят трябва да зададе друго изискване и да повтори процедурата. Част от използвания програмен код е показан по-долу:

```
....
function calculateAll() {
if(document.getElementById("E").value==-1 ||
document.getElementById("Target").value==-1 ||
document.getElementById("K").value==-1 || document.getElementById("Ta").value==-1
|| document.getElementById("E").value==' ' ||
document.getElementById("Target").value==' ' ||
document.getElementById("K").value==' ' || document.getElementById("Ta").value==' ')
{
    alert("Missing data for external conditions!");
    return;
}
else if(chek!=0)
//else if(document.getElementById("RDa").value==' ' || chek!=0)
{
    results();
    chek=1;
}
else if(chek==0)
{ //alert('2: chek='+chek);
    initArray(); chek=1;
    for (i=0; i<12; i++)
```

```
{ dd = d[i];SS = S[i];MM = M[i];TTE= TE[i];CCE= CE[i];
for (j=0; j<3; j++)
{
  if (device==1)
  { ff=fg[j];TTo=Tog[j];kkn=kng[j];TTOb=TObg[j];CCOb=CObg[j];
    if (j==0)
    { Ind=j;
      for (k=0; k<2; k++)
      { TT0k=T0kg1[k]; CC0k=C0kg1[k];
        Cena = eval(CCE) + eval(CCOb)+ eval(CC0k); //
        Teglo = eval(TTE) + eval(TTOb)+ eval(TT0k);
cen[(i*6)+(j*2)+k]=Cena;
teg[(i*6)+(j*2)+k]=Teglo;  //alert((i*6)+(j*2)+k+' Teglo='+Teglo);
      }
    }
  }
  else if (j==1)
  { Ind=j;
    for (k=0; k<2; k++)
    { TT0k=T0kg2[k]; CC0k=C0kg2[k];
      Cena = eval(CCE) + eval(CCOb)+ eval(CC0k);
      Teglo = eval(TTE) + eval(TTOb)+ eval(TT0k);
cen[(i*6)+(j*2)+k]=Cena;
teg[(i*6)+(j*2)+k]=Teglo;
    }
  }
  if (j==2)
  { Ind=j;
    for (k=0; k<2; k++)
    { TT0k=T0kg3[k]; CC0k=C0kg3[k];
      Cena = eval(CCE) + eval(CCOb)+ eval(CC0k); //alert(Cena);
      Teglo = eval(TTE) + eval(TTOb)+ eval(TT0k);
cen[(i*6)+(j*2)+k]=Cena;
teg[(i*6)+(j*2)+k]=Teglo;
    }
  }
}
else if (device==2)
{ ff=fb[j]; TTo=Tob[j];kkn=knb[j]; TTOb=TObb[j]; CCOb=CObb[j];
  if (j==0)
  { Ind=j;
    for (k=0; k<2; k++)
    { TT0k=T0kb1[k]; CC0k=C0kb1[k];
      Cena = eval(CCE) + eval(CCOb)+ eval(CC0k); //
      Teglo = eval(TTE) + eval(TTOb)+ eval(TT0k);
cen[(i*6)+(j*2)+k]=Cena;
teg[(i*6)+(j*2)+k]=Teglo;
    }
  }
  else if (j==1)
  { Ind=j;
    for (k=0; k<2; k++)
    { TT0k=T0kb2[k]; CC0k=C0kb2[k];
      Cena = eval(CCE) + eval(CCOb)+ eval(CC0k);
      Teglo = eval(TTE) + eval(TTOb)+ eval(TT0k);
cen[(i*6)+(j*2)+k]=Cena;
teg[(i*6)+(j*2)+k]=Teglo;
    }
  }
}
```

```

    }
    else if (j==2)
    { Ind=j;
      for (k=0; k<2; k++)
      { TT0k=T0kb3[k]; CC0k=C0kb3[k];
        Cena = eval(CCE) + eval(CC0b)+ eval(CC0k);
        Teglo = eval(TTE) + eval(TT0b)+ eval(TT0k);
        cen[(i*6)+(j*2)+k]=Cena;
        teg[(i*6)+(j*2)+k]=Teglo;
      }
    }
  }
else if (device==3)
{ ff=fs[j]; TTo=Tos[j]; kkn=kns[j]; TT0b=T0bs[j]; CC0b=C0bs[j];
  if (j==0)
  { Ind=j;
    for (k=0; k<2; k++)
    { TT0k=T0ks1[k]; CC0k=C0ks1[k];
      Cena = eval(CCE) + eval(CC0b)+ eval(CC0k); //
      Teglo = eval(TTE) + eval(TT0b)+ eval(TT0k);
      cen[(i*6)+(j*2)+k]=Cena;
      teg[(i*6)+(j*2)+k]=Teglo;
    }
  }
else if (j==1)
{ Ind=j;
  for (k=0; k<2; k++)
  { TT0k=T0ks2[k]; CC0k=C0ks2[k];
    Cena = eval(CCE) + eval(CC0b)+ eval(CC0k);
    Teglo = eval(TTE) + eval(TT0b)+ eval(TT0k);
    cen[(i*6)+(j*2)+k]=Cena;
    teg[(i*6)+(j*2)+k]=Teglo;
  }
}
else if (j==2)
{ Ind=j;
  for (k=0; k<2; k++)
  { TT0k=T0ks3[k]; CC0k=C0ks3[k];
    Cena = eval(CCE) + eval(CC0b)+ eval(CC0k);
    Teglo = eval(TTE) + eval(TT0b)+ eval(TT0k);
    cen[(i*6)+(j*2)+k]=Cena;
    teg[(i*6)+(j*2)+k]=Teglo;
  }
}
}
Ap=0.001*(ff/kkn);
RD=Math.sqrt((0.07*Ap*ff*TTo*SS*0.000001*dd*Ta*E*K*Aobd)/(0.4*0.000000000001*MM));
RD1=Math.round(RD); //RDold=RD1;
rdet[(i*3)+j]=RD1;
RR=Math.sqrt((0.07*Ap*ff*TTo*SS*0.000001*dd*Ta*E*K*Aobr)/(0.4*0.000000000001*MM));
RR1=Math.round(RR); //RRold=RR1;
rrec[(i*3)+j]=RR1;
RI=Math.sqrt((0.07*Ap*ff*TTo*SS*0.000001*dd*Ta*E*K*Aobi)/(0.4*0.000000000001*MM));
RI1=Math.round(RI); //RRold=RI1;
ride[(i*3)+j]=RI1;
}
}

```

```
magarray();
var size=12*3;
sort(rdet,rdindex,size);
sort(rrec,rrindex,size);
sort(ride,riindex,size);
sort(magn,mindex,3);
sort(cen,cindex,size*2);
sort(teg,tindex,size*2);
if(document.getElementById("RDa").value!=RDold)
{ if (!RDchange())
  return; }
if(document.getElementById("RRa").value!=RRold)
{ if (!RRchange())
  return;}
if(document.getElementById("RIa").value!=RIold)
{ if (!RIchange())
  return; }
if(document.getElementById("magnification").value!=MAGold)
{ if (!MAGchange())
  return; }
if(document.getElementById("Tegloo").value!=Told)
{ if (!Tchange())
  return; }
if(document.getElementById("Cenaa").value!=Cold)
{ if (!Cchange())
  return; }
}
}
.....
```

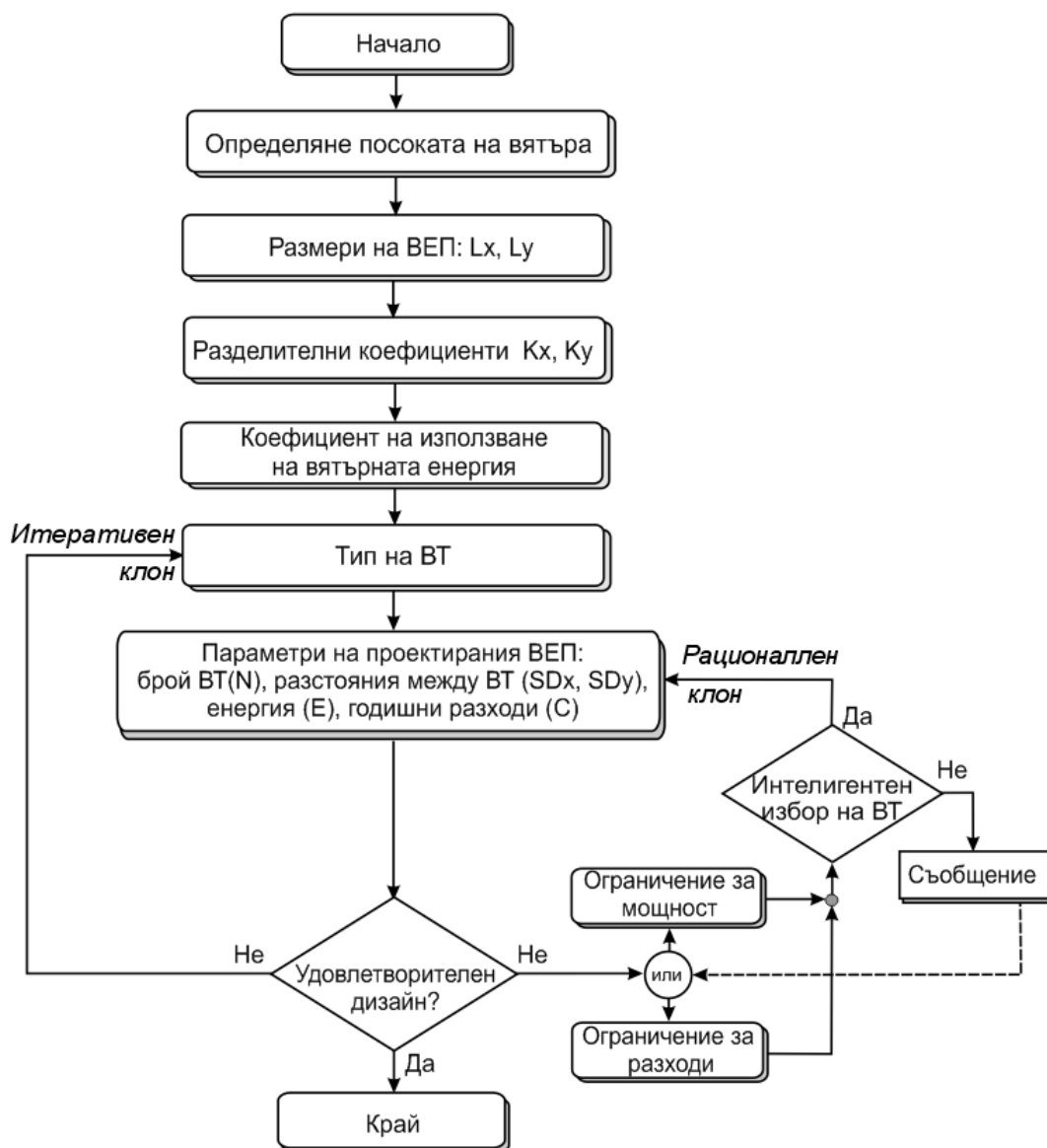
8.1.3. Система за подпомагане проектирането на ветроенергийни паркове

Като се имат предвид големите инвестиционни разходи и проблемите при изграждане на ВЕП, то важността на етапа на проектиране е от съществено значение. В тази връзка, възниква необходимост от предварителна оценка на проекта на ВЕП по отношение удовлетворяване на поставените критерии и цели. За целта е разработено уеб базирано приложение, реализиращо описания в Глава 4 алгоритъм за предварителна оценка на проект за ВЕП.

Предложеният алгоритъм за подпомагане проектирането на ВЕП е съставен от два основни клона, реализиращи съответно 1) итеративни процедури за оценка на ВЕП (итеративно проектиране) и 2) процедури за рационално проектиране (Borissova & Mustakerov, 2014), както е показано на Фиг. 8.6.

Изпълнението на алгоритъма започва с определяне на посоката на вятъра – преобладаваща или от всички посоки. На втория етап се определят размерите на ветровото поле. След това трябва да бъдат определени коефициентите за разстояния между турбините k_x и k_y и стойността на коефициента на използване на вятърната

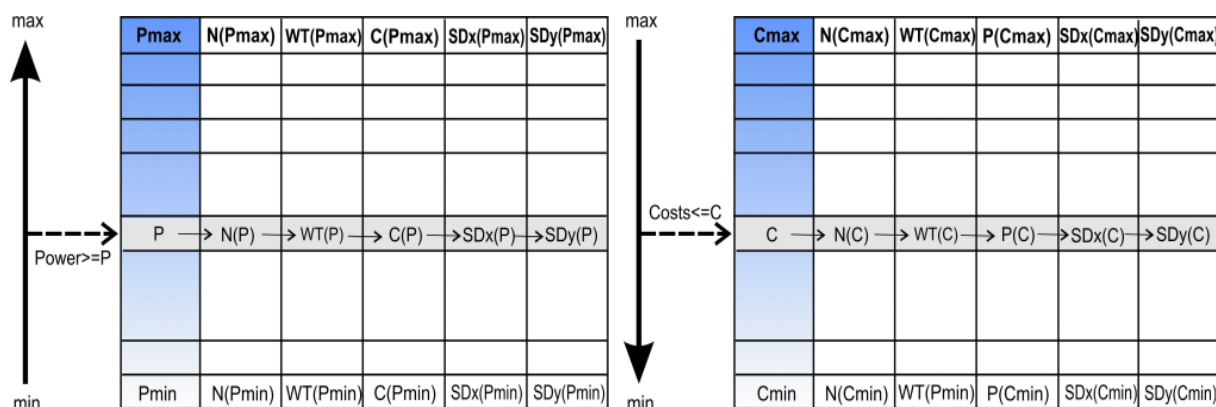
енергия η . След това е възможен избор на един от двата клона на алгоритъма – итеративен и рационален. Чрез итеративния клон, лицето, вземащо решение (ЛВР) може да направи избор на конкретен вид вятърна турбина (ВТ) и да получи съответните коефициенти SD_x и SD_y определящи броя и разположението на ВТ, както и да получи стойности за инвестиционните разходи и очакваната мощност на ВЕП. Ако получените стойности не са удовлетворителни, може да избере друга ВТ и да продължи този итеративен процес докато не бъде удовлетворен от постигнатите резултати. Друга възможност е използването на рационалния клон. При него могат да се зададат граници за разходите и/или мощността. Основната идея тук е да се намери подходящ тип ВТ, която за удовлетвори зададените от потребителя ограничения.



Фиг. 8.6. Алгоритъм за предварителна оценка на проект за ВЕП

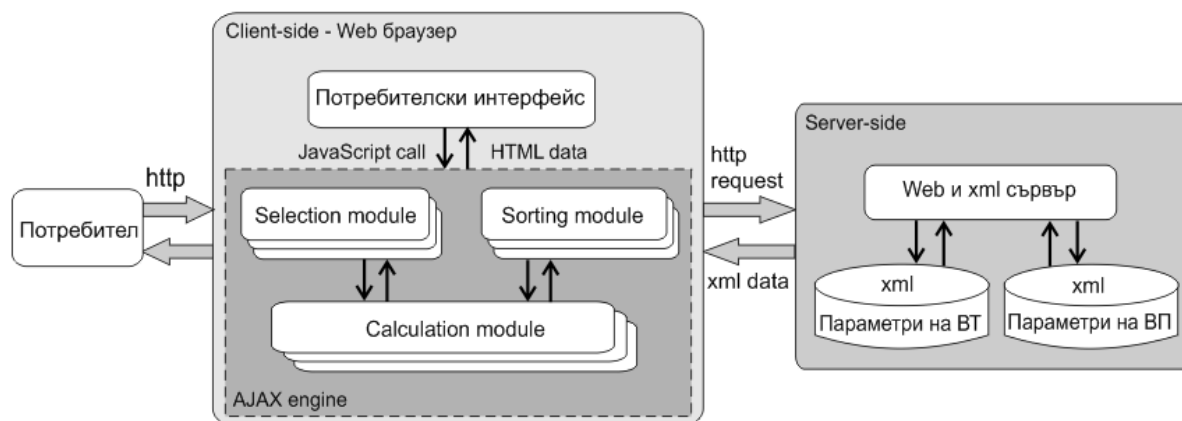
Алгоритмичната реализация на рационалния избор на вида на турбината се основава на изчисляване на мощността и разходите за всеки тип турбина, отчитайки размерите на ветровото поле и посока на вятъра. Получените масиви от данни за мощността и разходите се сортират във възходящ ред, като всеки елемент от масива елемент съответства на определен параметър на ВЕП (брой и тип турбини, разстояния между ВТ, разходи, енергия).

При наличие на зададено ограничение търсенето на решение, удовлетворяващо зададена граница се реализира, както е показано на Фиг. 8.7 (Borissova & Mustakerov, 2014). Ако зададеното ограничение не може да бъде изпълнено, съответно съобщение се показва и ЛВР има възможност да коригира зададеното ограничение.



Фиг. 8.7. Реализация на рационалното проектиране

Описаният в т. 4.5.1. алгоритъм, е реализиран в разработен прототип на уеб приложение с архитектура, използваща AJAX технологията – Фиг. 4.23.



Фиг. 8.8. Архитектура на уеб приложението

Графичният потребителски интерфейс на това уеб приложение е показан на Фиг. 8.9 (Borissova & Mustakerov, 2014).

Избор на преобладаващата посока на вятъра за конкретен ВЕП се осъществява чрез радио бутони. С помощта на съответни падащи списъци ЛВР може да избере тип ВТ и стойност на коефициента на използване на ветровата енергия. Размерите на ВП, както и коефициентите k_x и k_y се въвеждат от ЛВР в предвидени за целта текстови полета. При всеки избор на турбина нейните параметри се визуализират в съответните полета. С активирането на бутона “Calculation” се изчисляват и показват параметрите на ВЕП. По този начин, се реализира предварителната оценка на проектирания ВЕП чрез итеративния клон на алгоритъма. За активиране на рационалния клон е необходимо ЛВР за въведе граници за мощността или разходите. На база на получените сортирани масиви се определят вида и броя на ВТ, и се параметрите на ВЕП се визуализират в съответните полета.

The screenshot shows the 'Windpro - Avant Browser' window displaying a web application titled 'A preliminary assessment of wind power park project'. The interface includes a navigation bar with links for 'Main menu', 'Print', 'Help', and 'About'. Below the title, there are radio buttons for 'predominant' (selected) and 'uniform' wind patterns. The main input section contains a table with columns for 'Wind turbine type', 'Wind site dimension, Lx (km)', 'Wind site dimension, Ly (km)', 'Separation coefficients, Kx', 'Separation coefficients, Ky', and 'Capacity factor'. Below this is a section for 'Selected turbine parameters' with fields for 'Rated power, [KW]', 'Wind turbine rotor diameter, [m]', and 'Wind turbine rated wind speed [m/s]'. At the bottom, there are 'Reset' and 'Calculation' buttons, followed by a table for results including 'Number of installed turbines, [m]', 'Separation distances between turbines, Sx, [m]', 'Separation distances between turbines, Sy, [m]', 'Installed power, [MW]', 'Expected energy output, [MWh]', and 'Costs/year'.

Фиг. 8.9. Графичен потребителски интерфейс на уеб приложението

Предложеният алгоритъм за предварителна оценка на проектирания ВЕП, позволява да бъде оценена ефективността на проекта на ВЕП чрез задаване на вида на ВТ. Друг подход за проектиране на ВЕП е чрез използването на формулирани модели и съответни оптимизационни задачи. Този подход е описан в т. 8.2.2 и неговото използване води до определяне оптимален брой и разположения на ВТ спрямо зададените критерии/и и ограничения.

8.1.4. Система за подпомагане на групово вземане на решения

Предложеният в Глава 7, т. 7.5, подход за групово вземане на решение е апробиран в разработен прототип на софтуерен инструмент за групово вземане на решение, чийто основен екран е показан на Фиг. 8.10 (Mustakerov & Borissova, 2014). Целта е да се вземат предвид различните оценки на различните ЛВР (с различни компетентности) по отношение на оценяваните алтернативи.

Number of Decision Makers: 6 Number of Criteria: 19 Number of Alternatives: 3

Generate the matrix

Criteria	Weightings						Alternative 1						Alternative 2						Alternative 1					
	DM1	DM2	DM3	DM4	DM5	DM6	DM1 eval	DM2 eval	DM3 eval	DM4 eval	DM5 eval	DM6 eval	DM1 eval	DM2 eval	DM3 eval	DM4 eval	DM5 eval	DM6 eval	DM1 eval	DM2 eval	DM3 eval	DM4 eval	DM5 eval	DM6 eval
C1	2	4	8	8	6	10	2	4	6	1	1	1	10	5	8	9	9	9	8	4	6	8	9	2
C2	10	8	8	8	8	4	9	8	7	7	6	7	9	8	7	1	2	10	3	1	5	1	2	10
C3	10	10	8	4	8	10	9	2	8	7	6	7	10	4	7	10	9	10	6	5	7	8	8	10
C4	6	2	6	6	6	6	10	6	4	4	4	3	10	2	8	7	8	8	8	8	6	6	7	2
C5	10	4	10	10	10	8	3	1	9	1	2	1	5	10	7	3	2	7	3	7	2	8	9	9
C6		2	8	6	6	8	4	5	3	1	2	1	10	5	7	8	7	7	5	9	2	7	7	3
C7	6	6	6	8	8	6	8	9	4	2	2	8	9	6	10	7	5	10	7	4	6	7	8	8
C8	10	8	10	8	8	10	6	4	6	2	2	2	8	7	8	7	5	10	5	10	6	7	8	3
C9	6	10	8	10	10	8	5	6	8	2	2	8	6	10	6	2	1	5	6	5	6	5	7	9
C10	4	2	8	6	6	6	2	10	8	1	1	1	4	8	8	7	6	9	10	10	6	9	9	9
C11	8	8	10	10	10	10	6	5	6	1	1	1	8	9	8	8	9	7	7	4	2	7	8	2
C12	10	4	10	6	8	10	2	8	2	1	1	1	10	1	7	6	6	8	6	6	5	6	6	5
C13	10	2	10	6	8	10	1	1	2	1	1	1	8	2	8	7	6	9	5	10	5	7	6	5
C14	8	6	8	10	10	8	3	6	2	1	1	1	9	5	8	7	8	10	4	4	2	6	6	7
C15	4	8	6	8	10	10	5	4	2	1	1	1	8	10	6	8	8	10	4	8	2	8	7	8
C16	4	4	8	10	10	10	7	8	1	1	1	1	7	9	9	8	7	9	7	10	1	7	8	7
C17	6	6	10	10	10	6	10	9	10	7	5	10	3	8	4	5	4	5	5	9	2	8	8	7
C18	6	8	8	10	10	6	4	4	1	1	1	1	2	6	1	1	1	1	1	10	1	1	2	1
C19	4	2	6	10	10	4	4	2	1	1	2	1	8	8	6	7	6	8	3	8	2	6	5	6

Optimal solution: alternative A2

Фиг. 8.10. Основен екран на прототип на система за групово вземане на решение

Този софтуерен инструмент позволява да се генерира съответна таблица, описваща матрицата за алтернативите, критериите и оценките на отделните ЛВР, участващи при груповото вземане на решение. Например, след въвеждането на броя ЛВР, могат да се добавят или премахват динамично критериите за определените алтернативи, като се използва следния код:

```
.....
function addRow() {
    var Cr = document.getElementById("cr");
    var DM1 = document.getElementById("dm1-w");
    var DM1_Eaval1 = document.getElementById("dm1-eval1");
    var DM1_Eaval2 = document.getElementById("dm1-eval2");
    var DM1_Eaval3 = document.getElementById("dm1-eval3");
    var DM2 = document.getElementById("dm2-w");
```

```
var DM2_Eaval1 = document.getElementById("dm2-eval1");
var DM2_Eaval2 = document.getElementById("dm2-eval2");
var DM2_Eaval3 = document.getElementById("dm2-eval3");
var DM3 = document.getElementById("dm3-w");
var DM3_Eaval1 = document.getElementById("dm3-eval1");
var DM3_Eaval2 = document.getElementById("dm3-eval2");
var DM3_Eaval3 = document.getElementById("dm3-eval3");
var DM4 = document.getElementById("dm4-w");
var DM4_Eaval1 = document.getElementById("dm4-eval1");
var DM4_Eaval2 = document.getElementById("dm4-eval2");
var DM4_Eaval3 = document.getElementById("dm4-eval3");
var DM5 = document.getElementById("dm5-w");
var DM5_Eaval1 = document.getElementById("dm5-eval1");
var DM5_Eaval2 = document.getElementById("dm5-eval2");
var DM5_Eaval3 = document.getElementById("dm5-eval3");
var DM6 = document.getElementById("dm6-w");
var DM6_Eaval1 = document.getElementById("dm6-eval1");
var DM6_Eaval2 = document.getElementById("dm6-eval2");
var DM6_Eaval3 = document.getElementById("dm6-eval3");
var table = document.getElementById("myTableData");
var rowCount = table.rows.length;
var row = table.insertRow(rowCount);
row.insertCell(0).innerHTML= 'Crerion ' + Cr.value+ ' <input type="button" value =
"Delete" onClick="Javascript:deleteRow(this)">';
row.insertCell(1).innerHTML= DM1.value;
row.insertCell(2).innerHTML= DM1_Eaval1.value;
row.insertCell(3).innerHTML= DM1_Eaval2.value;
row.insertCell(4).innerHTML= DM1_Eaval3.value;
row.insertCell(5).innerHTML= DM2.value;
row.insertCell(6).innerHTML= DM2_Eaval1.value;
row.insertCell(7).innerHTML= DM2_Eaval2.value;
row.insertCell(8).innerHTML= DM2_Eaval3.value;
row.insertCell(9).innerHTML= DM3.value;
row.insertCell(10).innerHTML= DM3_Eaval1.value;
row.insertCell(11).innerHTML= DM3_Eaval2.value;
row.insertCell(12).innerHTML= DM3_Eaval3.value;
row.insertCell(13).innerHTML= DM4.value;
row.insertCell(14).innerHTML= DM4_Eaval1.value;
row.insertCell(15).innerHTML= DM4_Eaval2.value;
row.insertCell(16).innerHTML= DM4_Eaval3.value;
row.insertCell(17).innerHTML= DM5.value;
row.insertCell(18).innerHTML= DM5_Eaval1.value;
row.insertCell(19).innerHTML= DM5_Eaval2.value;
row.insertCell(20).innerHTML= DM5_Eaval3.value;
row.insertCell(21).innerHTML= DM6.value;
row.insertCell(22).innerHTML= DM6_Eaval1.value;
row.insertCell(23).innerHTML= DM6_Eaval2.value;
row.insertCell(24).innerHTML= DM6_Eaval3.value;
}
```

```
function deleteRow(obj) {  
    var index = obj.parentNode.parentNode.rowIndex;  
    var table = document.getElementById("myTableData");  
    table.deleteRow(index);  
}  
.....
```

След въвеждане от ЛВР на тегловите коефициенти за отделните критерии и на оценките за всеки критерий, се определя най-добрата алтернатива.

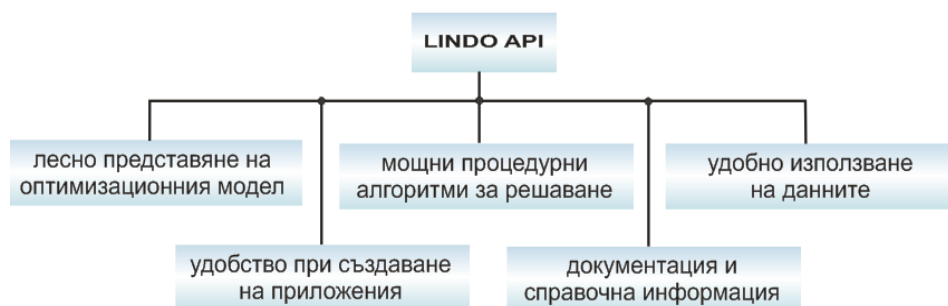
8.2. Специализирани софтуерни системи

При разработването на специализирани софтуерни системи за оптимизация е необходимо да се вземат предвид следните аспекти: 1) какви оптимизационни проблеми ще се решават и какво оптимизационно моделиране ще се прилага; 2) как ще се представят оптимизационните проблеми; с какви методи ще се решават оптимизационните проблеми; как ще се визуализират и анализират резултатите; каква помощ се предвижда за диагностика на грешки и каква помощ е предвидена за управление на модела.

Един от възможните подходи към създаването на програмни системи, съобразени с конкретната специфика, е разработването и използването на собствени оптимизационни модули, реализиращи специфични за целта алгоритми. Основен недостатък на този подход е значително увеличаване на времето за разработване, свързано с необходимостта от тестване на тези алгоритми и доказване на практическата им приложимост. Друг възможен подход, избягващ тези недостатъци, е използването на практически доказани готови оптимизационни модули, включени в разработваните системи. В този случай, основните усилия се насочват към разработването на входно-изходните интерфейсни модули, където възможните алгоритмични и програмни грешки могат да бъдат сравнително лесно открити и коригирани. Този подход се базира на използването на комерсиални програмни библиотеки от оптимизационни модули чрез т.нар. API (*Application Programming Interface*) което намалява съществено времето за разработка и гарантира в по-голяма степен практическата приложимост на разработваната система.

За реализация на оптимизационното ядро на описваната програмна система е избрана библиотеката LINDO API (LINDO API 7.0, 2012), на утвърдената в средата на специалистите по оптимизация, програмна система LINGO. Тя осигурява необходимите средства за решаване на широк спектър от задачи за оптимизация от линейно

програмиране, смесено-целочислено програмиране, квадратично програмиране, както и нелинейни и неизпъкнали задачи (Фиг. 8.11). Една от най-съществените особености при използването на LINDO API е въвеждането на входните данни на решаваната оптимизационна задача. В много реални практически проблеми формулираните оптимизационни задачи се характеризират с голям брой ограничения и променливи, но с малък брой ненулеви коефициенти на матрицата на ограниченията – т.е. притежават *разредена матрица*.



Фиг. 8.11 LINDO API характеристики

В такива случаи е целесъобразно да се използва икономичен начин за въвеждане на входните данни, при които се въвеждат само ненулевите елементи на матрицата на ограниченията. За да се възстанови оригиналната матрица е необходимо да се въведе допълнителна информация за разположението на тези ненулеви елементи в матрицата, а останалите елементи в матрицата по подразбиране се приемат за нули. За целта се изисква: 1) описание на матрицата на коефициентите на ограниченията, 2) редуциране на матрицата на ограниченията до описание само на ненулевите ѝ членове, чрез въвеждане на векторите (в терминологията на LINDO API) *Value*, *Column-start*, *Row-index*, 3) въвеждане на стойностите на десните страни на ограниченията – *Right-hand side values*, 4) определяне на коефициентите на целевата функция – *Objective coefficients*, и 5) определяне на типа на ограниченията – *Constraint senses*.

Един добре познат софтуерен инструмент е системата MS Office. Част от тази система е софтуерният инструмент за електронни таблици MS Excel, която днес е един от най-популярните и гъвкави инструменти за бизнес моделиране. В средата на MS Excel могат да се създават и решават оптимизационни задачи, като се използват специализирани оптимизационни модули, предназначени за работа в тази среда. Чрез комбиниране на графичния потребителски интерфейс с алгебричен език за моделиране и оптимизиране, MS Excel Solver може да се разглежда като един добър избор за създаване и решаване на различни оптимизационни задачи (Mason & Dunning, 2010). Съществуват оптимизационни модули, които могат да бъдат използвани в средата на

За рачунањето на формулирането в Глава 5 оптимизационно е зададено (5.9) – (5.16) с

Фиг. 8.12. Матрица на ограниченията на формулираната задача

```
Value = [-1., -1., 1., -1., -1., 1., -1., -1., -1., 1.,
        -1., 1., -1., 1., -1., 1., -1., 1., -1., -1., 1.,
        -1., 1., -1., -1., 1., -1., 1., -1., 1., 1., -1.,
        -1., -1., 1., -1., 1., -1., 1., -1., 1., -1., 1.,
        -1., -1., 1., -1., 1., -1., 1., 1., -1., 1., -1.,
        1., -1., -1., 1., 1., -1., 1., -1., -1., 1., 1.,
        -1., 1., 1., 1., 1., 1., 1., 1., 1.]
```

Вторият вектор, *Column-start* описва кои елементи от *Value*-вектора представляват първия ненулев елемент от колона в оригиналната матрица, т.е. *n*-ия запис в този вектор показва къде във *Value* вектора се намира началото на *n*-тата колона. Важно е да се отбележи, че LINDO API използва индексване от нула, така че *Column-start* вектора за разглеждания проблем ще има вида:

```
Column-start = [0, 2, 5, 8, 11, 15, 19, 22, 25, 29, 32, 36,
                40, 44, 47, 50, 54, 56, 58, 62, 64, 67, 68,
                69, 70, 71, 72, 73]
```

Също така е важно да се отбележи, че *Column-start* вектора има един запис в повече, отколкото колоните в разглежданата матрица. Този запис показва къде завършва последната колона и винаги е равен на дължината на *Value* вектора. От *Column-start* вектора може да се получи информация, коя колона с кой запис се асоциира във *Value* вектора. Допълнителната информация, която е необходима, за да се възстановят адекватните позиции на ненулевите елементи в матрицата, е номера на реда на съответния елемент, т.е. т. нар. *Row-index* вектор. Този вектор има същата дължина като *Value* вектора и показва коя стойност от *Value* вектора на кой ред от матрицата на ограниченията принадлежи. За конкретния разглеждан пример вектора *Row-index* има вида:

```
Row-index = [0, 21, 0, 1, 26, 1, 2, 34, 3, 21, 22, 3, 4, 26, 27, 4, 5, 34,
              35, 6, 22, 23, 6, 7, 30, 7, 8, 27, 28, 8, 35, 36, 9, 10, 23,
              24, 10, 11, 28, 29, 11, 12, 30, 31, 13, 24, 25, 13, 14, 29,
              14, 15, 31, 32, 15, 16, 17, 25, 17, 18, 32, 33, 19, 33, 19,
              20, 36, 2, 5, 9, 12, 16, 18, 20].
```

Други данни, необходими за пълното описание на оптимизационната задача, са десните страни на ограниченията *Right-hand side values*, вида на ограниченията *Constraint senses* и коефициентите на целевата функция *Objective coefficients*, които за разглеждания пример са както следва:

```
Right-hand side values = [8., 6., 6., 8., 10., 6., 8., 8., 8., 4., 4., 1.,
                          2., 4., 12., 4., 8., 6., 8., 6., 8., 8., 8., 8.,
                          4., 4., 6., 10., 8., 1., 8., 2., 4., 8., 6.,
                          6., 4.].
```

```
Constraint senses = [G G G G G G G G G G G G G G G G G G G G G
                     G G G G G G G G G G G G G G G G G G G G G].
```

```
Objective coefficients = [0., 0.,0., 0., 0., 0., 0., 0., 0., 0.,0., 0., 0., 0.,  
                        0., 0., 0., 0., 0.,0., 0., 0., 1., 1., 1., 1.,  
                        1., 1., 1.].
```

Вида на ограниченията в терминологията на LINDO API се изразява по следния начин: L – за по-малко, E – за равенство, G – за по-голямо.

Оптимизационните модули на LINDO API са вградени в разработен прототип на изследователски прототип “J-Opti”, илюстриран със следните извадки от програмния код (Mustakerov & Borissova, 2009):

```
..... *  
public static void main(String[] args) {  
SwingUtilities.invokeLater(new Runnable() {  
    public void run() {  
        try{UIManager.setLookAndFeel(UIManager.  
                                getSystemLookAndFeelClassName());}  
        catch (Exception exception) { exception.printStackTrace();}  
        System.loadLibrary("lindojni");  
        new Schedule1Class(); } }  
.....  
public class schedule extends Lindo {  
public schedule( ) { try{jbInit(); }  
catch(Exception ex) { ex.printStackTrace(); } }  
.....  
void jbInit() throws Exception {  
    nM = 37; /* Number of constraints */  
    nN = 28; /* Number of variables */  
    /* declare an instance of the LINDO environment object */  
    int pEnv[] = new int[1];  
    /* declare an instance of the LINDO model object */  
    int pModel[] = new int[1];  
    int nSolStatus[] = new int[1];  
    /* Стъпка 1 Зареждане на LINDO среда */  
    nErrorCode[0] = LSloadLicenseString("lndapi50.lic", cLicenseKey);  
    APIErrorCheck(pEnv);  
    pEnv[0]=LScreateEnv(nErrorCode,cLicenseKey.toString());  
    APIErrorCheck(pEnv);  
    /* Стъпка 2 Създаване на модел в средата */  
    pModel[0]=schedule.LScreateModel(pEnv[0],nErrorCode);  
    APIErrorCheck(pEnv);  
    /* Стъпка 3 Описание на модела: int nDir = schedule.LS_MIN; double dObjConst  
= 0., adC[] = ..., adB[] = ..., adA[] = ..., pdLower[] = ..., pdUpper[] = ..., int nNZ = 74, int  
anBegCol[] = ..., anRowX[] = ..., pnLenCol[] = ..., varnames= new String[]{...},  
connames= new String[] {...} */  
    /* Стъпка 4 Изпълнение на оптимизацията */  
    nErrorCode[0] = schedule.LSoptimize(pModel[0],  
        schedule.LS_METHOD_PSIMPLEX,nSolStatus);  
    APIErrorCheck(pEnv);  
    /* Стъпка 5 Получаване на решението */  
    int i; adX = new double[nN]; adR = new double[nN];  
    adS = new double[nM]; adY = new double[nM];  
    dObj = new double[1];  
    APIErrorCheck(pEnv);  
    System.out.print ("Стойност на целевата функция="+ dObj[0]+"\\n"+"Решение на  
правата задача");
```



```

    for (i= 0; i < nN; i++) System.out.println("    " + varnames[i] + "\t" +
adX[i] + "\t" + adR[i]);.....
/* Стъпка 6 Изтриване на LINDO средата */
nErrorCode[0] = schedule.LSdeleteModel(pModel);
nErrorCode[0] = schedule.LSdeleteEnv(pEnv);
.....

```

Интерфейсът на разработената система е илюстриран чрез *Главно меню* от вида, показан на Фиг. 8.13 (Mustakerov & Borissova, 2009):

The screenshot shows a window titled "Java Optimization" with a menu bar (File, Help) and buttons for LOAD and RUN. The interface is split into two panes. The left pane displays the optimization problem, and the right pane displays the solution results.

ОПТИМИЗАЦИОННА ЗАДАЧА:
 $\min (X1E \cdot X2E \cdot X3E \cdot X4E \cdot X5E \cdot X6E \cdot X7E)$,
при ограничения за последователност на обработките:

$X12 - X11 \geq 8$
 $X14 - X12 \geq 8$
 $X1E - X14 \geq 8$
 $X22 - X21 \geq 8$
 $X24 - X22 \geq 10$
 $X2E - X24 \geq 8$
 $X33 - X31 \geq 8$
 $X32 - X33 \geq 8$
 $X34 - X32 \geq 8$
 $X3E - X34 \geq 4$
 $X42 - X41 \geq 4$
 $X43 - X42 \geq 1$
 $X4E - X43 \geq 2$
 $X52 - X51 \geq 4$
 $X53 - X52 \geq 12$
 $X55 - X53 \geq 4$
 $X5E - X55 \geq 8$
 $X63 - X61 \geq 6$
 $X6E - X63 \geq 8$
 $X74 - X73 \geq 6$
 $X7E - X74 \geq 8$

и при ограничения за заетост на машините:

$X21 - X11 \geq 8$
 $X31 - X21 \geq 8$
 $X41 - X31 \geq 8$
 $X51 - X41 \geq 4$
 $X61 - X51 \geq 4$
 $X22 - X12 \geq 6$
 $X32 - X22 \geq 10$
 $X42 - X32 \geq 8$
 $X52 - X42 \geq 1$
 $X43 - X33 \geq 8$
 $X53 - X43 \geq 2$
 $X63 - X53 \geq 4$
 $X73 - X63 \geq 8$
 $X24 - X14 \geq 6$
 $X34 - X24 \geq 6$
 $X74 - X34 \geq 4$

РЕШЕНИЕ НА ОПТИМИЗАЦИОННАТА ЗАДАЧА:
Стойност на целевата функция = 332.0

Решение на правата задача

Променлива	Стойност	Отн. оценка
x1	0.0	7.0
x2	8.0	0.0
x3	14.0	0.0
x4	8.0	0.0
x5	16.0	0.0
x6	26.0	0.0
x7	16.0	0.0
x8	24.0	0.0
x9	32.0	0.0
x10	40.0	0.0
x11	24.0	0.0
x12	40.0	0.0
x13	41.0	0.0
x14	28.0	0.0
x15	41.0	0.0
x16	53.0	0.0
x17	57.0	0.0
x18	32.0	0.0
x19	57.0	0.0
x20	65.0	0.0
x21	71.0	0.0
x22	20.0	0.0
x23	32.0	0.0
x24	28.0	0.0
x25	43.0	0.0
x26	65.0	0.0
x27	65.0	0.0
x28	79.0	0.0

Решение на двойствената задача

Огр.	Допълн. пром.	Двойств. оценка
C1	1.0	0.0
C2	1.0	0.0
C3	1.0	0.0
C4	1.0	0.0
C5	1.0	0.0
C6	1.0	0.0
C7	4.0	0.0
C8	4.0	0.0
C9	0.0	0.0
C10	1.0	0.0
C11	0.0	-12.0
C12	1.0	0.0
C13	1.0	0.0
C14	0.0	-9.0
C15	3.0	0.0
C16	1.0	0.0
C17	1.0	0.0
C18	0.0	-19.0
C19	1.0	0.0
C20	1.0	0.0
C21	1.0	0.0
C22	6.0	0.0
C23	5.0	0.0
C24	1.0	0.0
C25	0.0	0.0
C26	0.0	0.0
C27	-0.0	-2.0
C28	0.0	-6.0
C29	4.0	0.0
C30	3.0	0.0
C31	0.0	-9.0
C32	-0.0	-10.0
C33	2.0	0.0
C34	1.0	0.0
C35	0.0	-6.0
C36	0.0	-8.0
C37	0.0	-27.0

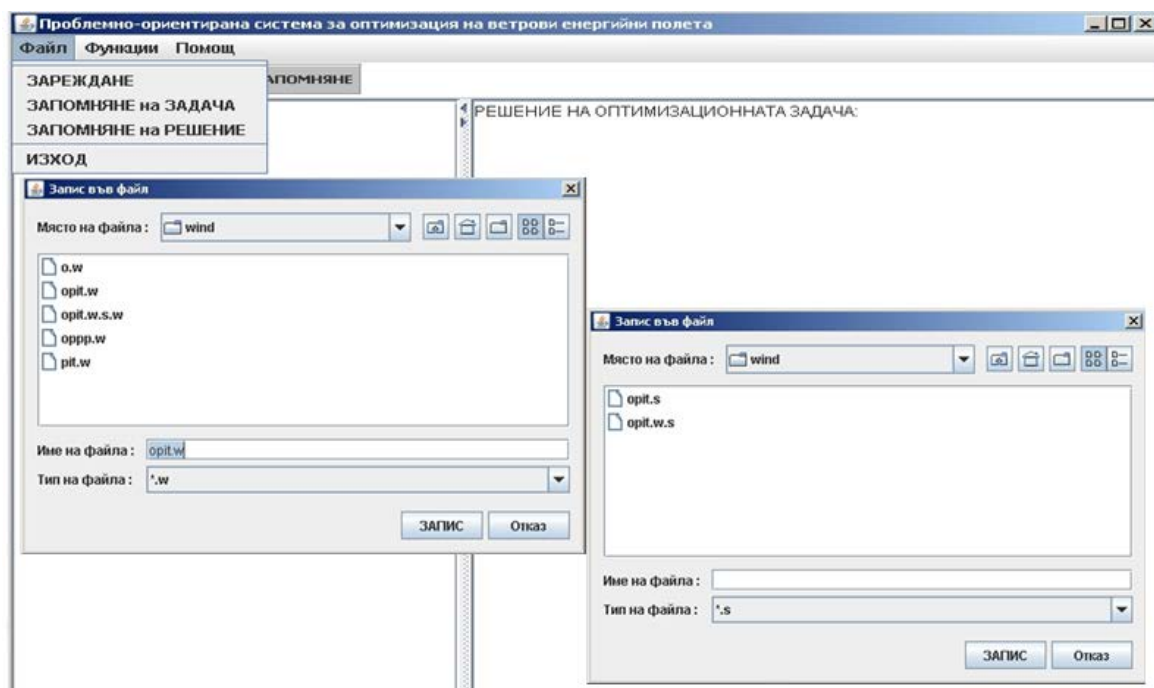
Фиг. 8.13. Главно меню на системата

В прозореца на *главното меню* освен стандартно приетите панели за менюта и за бутони са използвани два разделени панела – за входните и изходните данни съответно. Панелът с входните данни се запълва с информация за решаваната задача след зареждането ѝ с помощта на бутона LOAD, а панелът с резултатите – след активиране на бутона RUN.

За целите на решаване на формулираните оптимизационни задачи за оптимално проектиране на ВЕП е разработена проблемно-ориентирана изследователска система за оптимално проектиране на ветроенергийни паркове (Фиг. 8.14). Системата също се базира на използването на оптимизационните модули на LINDO API. Основните функции на системата позволяват: въвеждане и запис на оптимизационна задача, решаване на задачата и запис на полученото решение. За разлика от описаната по-горе

система “J-Opti”, в системата *Wind* формулираните задачи се запомнят в различен формат. Това позволява на по-късен етап те да бъдат редактирани или модифицирани.

Тази система е използвана за решаване на формулираните оптимизационни задачи, описани в Глава 4.



Фиг. 8.14. Проблемно-ориентирана система за оптимално проектиране на ВЕП

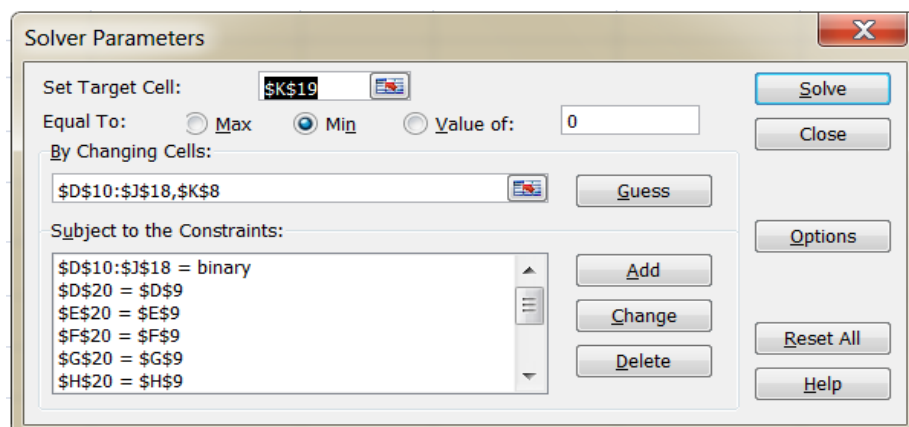
8.2.2. Специализирани софтуерни системи, използващи оптимизационен модул, работещ в средата на електронни таблици

Формулираната оптимизационната задача (5.9) – (5.46), на база на описаният в Глава 5 обобщен модел за оптимални разписания, е представена в средата на MS Excel следвайки следните стъпки (Borissova, 2008):

1. Въвеждане на текстова информация за решаваната задача, в т.ч. и легенда за използваните съкращения в клетките A1 до AD7 за (Фиг. 8.15).
2. Клетка E8 ще се използва за стойността на общата продължителност на процеса на обработка на детайлите.
3. Клетки A8 до D8 съдържат обяснителен текст за съдържанието на клетка E8.
4. В клетки C11 до C31 се въвежда информацията за последователната обработка на детайлите на отделните машини,
5. В клетки D11 до D31 се въвежда продължителността на обработките в часове. Използваните машини се кодират като M1, M2, M3 и т.н.
6. В клетки B11 до B31 се въвеждат съответните детайли, за които са в сила обработките на машините в клетките C11 до C31.

9. Клетки F9 – AG9 се използват за стойностите на управляващите променливи, а клетки F10 до AG10 съдържат означения на управляващите променливи x_{11} , x_{12} , x_{14} , ..., до x_{6e} , x_{7e} .
10. В клетки F11 до AG31 се въвежда матрицата на коефициентите пред променливите в ограниченията.
11. Продължителността на обработките в колона D11 до D31 т.е. стойностите на десните страни на ограниченията определят редовете на ограниченията в попълваната матрица. В клетките, пресечна точка на редовете на ограниченията и на колоните на участващите в ограниченията променливи, се въвеждат съответните коефициенти +1 или -1, а в останалите клетки се въвежда 0.
12. В клетки B32 до D47 се въвежда информацията за изчакването между детайлите, които трябва да се обработват на една и съща машина, като в колона D32-D47 се въвеждат десните страни на съответното ограничение, а в клетки F32-AG47 се въвеждат коефициентите пред променливите в тези ограничения.
13. В клетки F49 до AG49 се въвеждат коефициентите пред променливите в целевата функция, съобразени с колоните за променливите, въведени в F10 до AG10.
14. В клетка F50 ще се запише изчислената стойност на целевата функция, използвайки формулата =SUMPRODUCT(F49:AG49;F9:AG9) .
15. В клетки A49 до E49 и от A50 до E50 се въвеждат пояснителни текстове: “коефициенти на целевата функция” и “стойност на целевата функция”.
16. В клетка E11 е формулата =SUMPRODUCT(F11:AG11;\$F\$9:\$AG\$9), която се копира последователно в клетки E12 до E47, като при копирането формулата се модифицира автоматично да съответства на клетката, например в клетка E47 тя ще има вида: =SUMPRODUCT(F47:AG47;\$F\$9:\$AG\$9) .
17. В клетка E8 се въвежда формулата за определяне на крайната продължителност на процеса на обработка на корпусните детайли =MAX(F9:AG9), която избира най-голямата стойност от реда F9 до AG9. Стойностите в реда F9 до AG9 се попълват след решаване на така въведената оптимизационна задача с помощта на модула Solver в електронната таблица Excel (Фиг.8.15).

Формулираният в Глава 5 оптимизационен модел за линейно разкрояване също е представен в средата на MS Excel. За въвеждане на оптимизационната задача (5.86) – (5.91) се създава електронна таблица, включваща определяне на клетката, която ще съдържа стойността целевата функция; определяне на клетките за променливите и определяне на клетките, съдържащи ограниченията. Въвеждането на тези параметрите се реализира посредством диалогов прозорец, показан на Фиг. 8.16.



Фиг. 8.16. Въвеждане на модела

Представянето на модела за линейно разкрояване и решението в средата на MS Excel има вида, показан на Фиг. 8.17.

task-one-dimensional-cutting-stock-model - Microsoft Excel										
	A	B	C	D	E	F	G	H	I	J
1		<u>Combinatorial optimization model for one-dimensional cutting stock problem</u>								
2		$l_1, l_2, \dots, l_7$: length of the elements that are to be cut								
3		ΔL_i : waste for each blank = optimal length of blank - utilized length of each blank								
4		Overall waste = sum of waste of all blanks								
5		L_{min}, L_{max} : lower and upper limits of optimal length of blank								
7		$L_{min} =$	5000	l_1	l_2	l_3	l_4	l_5	l_6	l_7
8		$L_{max} =$	7000	650	730	1400	1700	2000	2100	2200
9		Demand of elements	4	8	4	4	4	4	4	8
10		Cutting patterns for each blank	0	1	1	0	1	0	1	
11			0	1	1	0	1	0	1	
12			0	1	1	1	0	0	1	
13			0	1	1	1	0	0	1	
14			1	1	0	0	0	1	1	
15			1	1	0	0	0	1	1	
16			1	1	0	0	0	1	1	
17			0	1	0	1	1	1	0	
18			1	0	0	1	1	0	1	
19		Minimal waste:								
20		Elements # in solution:	4	8	4	4	4	4	4	8

Фиг. 8.17. Модел за оптимално линейно разкрояване

И двете приложения показват възможността за създаване на различни потребителски интерфейси, което е предпоставка за осигуряване на подходящи интерфейси за конкретен тип потребители.

Нарупаният опит в разработването на описаните в тази глава софтуерни инструменти ще бъде използван при създаването на други подобни системи, реализиращи предложените в дисертационния труд модели, методи и алгоритми.

Заклучение

Глобализационните процеси, наблюдавани в съвременния свят поставят все по-големи изисквания за понижаване на производствените разходи, за конкурентоспособност в световен мащаб и за използване на научно-обосновани методи при вземането на решения. Развитието на компютърните технологии и постоянно растящата им изчислителна мощност позволяват ефективно използване на оптимизационни модели, водещи до задачи с голяма размерност. Наличието на модерни компютърни средства и ефективни изчислителни методи, са добра основа за развитие на научно-обосновани подходи при решаване на реални научно-приложни проблеми от различни области. За построяването на адекватен математически модел от особена важност е определянето на зависимостите, описващи количествените връзки между изходните данни и параметрите на решението, които се използват като индикатор за ефективността на това решение. За всеки конкретен проблем е необходимо да бъде формулиран съответен модел, който най-добре да отразява спецификата и изискванията към моделирания обект или процес. В дисертационния труд са предложени обобщени модели, приложими за определени класове инженерни проблеми. Те са използвани за формулиране на едно- или многокритериални оптимизационни задачи, които са тествани числено на базата на реални примери от практиката на инженерните системи. Част от предложените методи, модели и алгоритми са реализирани в подходящи софтуерни инструменти, подпомагащи оптимално проектиране, планиране и управление на инженерни системи.

Като бъдещо развитие на изследванията в дисертационния труд се предвижда използване на други методи за решаване на формулираните задачи, както и модифициране и разширяване на предложените модели, методи и алгоритми за други области на приложение.

Получените резултати, описани в дисертационния труд, са отразени в общо 33 научни публикации, представени на 5 международни научни форума, в 13 специализирани международни списания и в списания с импакт фактор, и 1 глава от книга. До момента са забелязани общо 89 цитирания на публикациите по настоящия дисертационен труд, 28 от които са в списания с импакт фактор.

Авторска справка

Научни и научно-приложни приноси

Приносите с научен и научно-приложен характер в настоящия дисертационен труд са свързани с обработката на информация чрез формализация на проблеми, свързани с инженерни системи. Чрез тази формализация са създадени адекватни математически модели, позволяващи използване на оптимизационни методи и алгоритми. Част от предложените методи и алгоритми са реализирани в софтуерни инструменти, подпомагащи вземането на решения. Получените основни резултати в дисертационния труд са групирани както следва:

1. Предложени са обобщени модели и методи за оптимално проектиране на модулни инженерни системи от следните класове:
 - системи, вземащи предвид съществуващите взаимовръзки между модулите,
 - системи, вземащи предвид съществуващите взаимовръзки между модулите и външните експлоатационни условия,
 - системи, вземащи предвид както съществуващите взаимовръзки и външните експлоатационни условия, така и икономическата ефективност.
2. Предложени са обобщени модели и алгоритми за оптимално планиране на ресурси при:
 - зависима и смесена обработка на детайли,
 - линейно разкрояване, определящо едновременно както оптималните модели на разкрояване за всяка заготовка, осигуряваща минимални отпадъци, така и оптималната дължина на заготовките,
 - определяне на оптимален брой на персонала и на съответстващия му производствен график, осигуряващ минимален престой на персонала.
3. Предложени са модели и алгоритми за оптимално управление на предсказващо поддържане чрез:
 - определяне на оптимална стратегия за поддръжка, обосноваваща ремонт или смяна на машината като цяло или на отделни нейни компоненти,
 - определяне на оптимално разположение на зададен брой сензори и при едновременно определяне на броя и разположението на сензорите.
4. Предложени са обобщени методи и алгоритми за обоснован избор на алтернативи за проблеми от следните класове:
 - класиране на алтернативи,
 - определяне на k най-добри алтернативи,
 - избор на алтернатива с отчитане влиянието на околната среда,
 - групово вземане на решения.
5. Разработени са софтуерни инструменти за приложения в инженерни системи:
 - уеб-базирани системи за проектиране на: персонални компютри, уреди за нощно виждане, ветроенергийни паркове и за групово вземане на решение,
 - специализирани софтуерни системи, използващи оптимизационни модули за оптимални разписания, оптимално проектиране на ветроенергийни паркове и оптимално линейно разкрояване.

Апробация на резултатите

Резултатите, описани в дисертацията, са представени на 5 международни конференции, и са публикувани в 13 специализирани международни списания и 1 глава от книга, както следва:

1. *Applied Mathematical Modelling*
2. *Comptes rendus de l'Academie bulgare des Sciences*
3. *Cybernetics and Information Technologies*
4. *Engineering Science*
5. *IEEE International Symposium on Innovations in Intelligent Systems and Applications (INISTA)*
6. *International Conference "Automatics and Informatics'09"*
7. *International Conference on Circuits, Systems, Communications and Computers (CSCC 2014)*
8. *International Conference on Information Systems and Technologies (ICIST 2014)*
9. *International Conference on Theory and Applications in Mathematics and Informatics, (ICTAMI 2007)*
10. *International Journal of Advanced Modeling and Optimization*
11. *International Journal of Computer and Information Engineering*
12. *International Journal of Computer Science and Engineering*
13. *International Journal of Electronics and Telecommunications*
14. *International Journal of Information Science and Engineering*
15. *International Journal of Information Technologies & Knowledge*
16. *International Journal of Management Science and Engineering*
17. *Renewable Energy*
18. *Wind Turbines (книга)*
19. *World Academy of Science, Engineering and Technology*

Декларация

Дисертационният труд се базира на оригинални научноизследователски резултати, получени от автора или с основно участие на автора, което декларирам с подписа си.

Настоящият дисертационен труд не е представян за придобиване на научна степен в друго висше училище, университет или научен институт.

.....
Даниела Борисова

Списък на научноизследователските и приложни проекти по темата на дисертацията с участието на автора

1. “Приложение на операционните изследвания за вземане на решения в инженерни системи”. Научноизследователски проект, (2012-2013), изпълнител.
2. “Предсказващо поддържане на технологични съоръжения въз основа на диагностика и анализ на риска”. Договор с Ф"НИ" – МОН, (2010-2013), изпълнител.
3. “Разработване на оптимизационни модели за проектиране на ветроенергийни паркове”. Научноизследователски проект, (2011), изпълнител.
4. “Разработване и изследване на модели и методи за оптимизация в инженерни системи”. Научноизследователски проект, (2010), изпълнител.
5. “Изграждане на висококвалифицирани млади изследователи по информационни технологии за обработка и управление на знания”. Договор BG051PO001/07/3.3-02/7, Проектът се осъществява с финансовата подкрепа на ЕСФ 2007-2013 и РБългария – МОН по ОП "Развитие на човешките ресурси" в направление “Подкрепа за развитието на докторанти, постдокторанти, специализанти и млади учени”, (2008-2010), изпълнител.
6. “Иновационни информационни технологии при проектиране на енергийни полета на витлови генератори”. Договор с ИАНМСП № ИФ-00-04-163-1/28.12.2007 г., съвместно с “Фриймейзанри” ЕООД, (2007-2009), изпълнител.
7. “Проучване възможностите за проектирането на ношен псевдобинокулярен уред с възможност за цифрова обработка на изображението”, Договор с “ЕлкоФ” ООД, (2008 – 2009), ръководител.
8. “Системна актуализация на база данни и поддръжка на информационна система”. Договор с “АНАНИЕВ-АГА и с-ие” СД, 2008, изпълнител.
9. “Проучване и идеен проект за ношен монокуляр с възможност за цифрова обработка на изображението”. Договор с “ЕлкоФ” ООД, 2007–2008, ръководител.
10. “Оптимизационни методи с използване на Java-технология”. Научноизследователски проект, (2007-2009), изпълнител.
11. “Изследване възможностите за проектиране на уреди за нощно виждане чрез методи на операционните изследвания”. Научноизследователски проект, (2007-2009), ръководител.
12. “Оптимално планиране на производството на детайли и възли за металорежещи машини с ЦПУ”. Договор No МСП-201 с ФНИ, Повишаване на иновациите в малки и средни предприятия, (2006-2007), изпълнител.
13. “Системна поддръжка на информационна система и обработка на база данни”. Договор с “АНАНИЕВ-АГА и с-ие” СД, 2006, изпълнител.
14. “Разработване на информационна система за изследване на наноструктури на базата на сканиращ микроскоп модел “Philips” SEM”. Договор с ЦИАФОП-БАН, 2005, изпълнител.
15. “Изследване и разработване на компютърни програмни системи за обработка на информация”. Научноизследователски проект, (2004-2006), изпълнител.
16. “Теоретично изследване и моделиране на оптоелектронни уреди”. Научноизследователски проект, (2004-2006), ръководител.
17. “Проучване на съвременното състояние и перспективи за развитие на оптичните системи, използвани в мерно-наблюдателните нощни уреди за стрелково оръжие”. Договор No 10.8, с Министерство на Отбраната, 2000 г., ръководител.
18. “Очила за нощно виждане”. Договор с фирма “ЕлкоЕ”, (1998-2000), ръководител.

Списък на публикациите, в които са представени резултати от дисертационния труд

- D1. Borissova, D., I. Mustakerov. A parallel algorithm for optimal job shop scheduling of semi-constrained details processing on multiple machines. *In Proc. Advanced Information Science and Applications – Vol. I, Int. Conf. on Circuits, Systems, Communications and Computers (CSCC 2014)*, July 17-21, 2014, Santorini Island, Greece, ISBN: 978-1-61804-236-1, pp. 145-150.
- D2. Mustakerov, I., D. Borissova. A Web application for group decision-making based on combinatorial optimization. *In Proc. 4th Int. Conf. on Information Systems and Technologies (ICIST 2014)*, ISBN: 978-0-9561122-5-5, March 22-24, 2014, Valencia, Spain, pp. 46-56.
- D3. Borissova, D., I. Mustakerov. Web-based tool for preliminary assessment of wind power plant design. *In Proc. 4th Int. Confe. on Information Systems and Technologies (ICIST 2014)*, ISBN: 978-0-9561122-5-5, March 22-24, 2014, Valencia, Spain, pp. 139-149.
- D4. Mustakerov, I., D. Borissova. Multi-criteria model for optimal number and placement of sensors for structural health monitoring: Lexicographic method implementation. *Int. J. Advanced Modeling and Optimization*, ISSN: 1841-4311, Vol. 16(1), 2014, pp. 103-112.
- D5. Borissova, D., I. Mustakerov, E. Bantutov. Methodology for determining of surveillance conditions in relation to night vision devices performance. *Int. J. Advanced Modeling and Optimization*, ISSN: 1841-4311, Vol. 16(1), 2014, pp. 51-59.
- D6. Mustakerov, I., D. Borissova. One-dimensional cutting stock model for joinery manufacturing. *In Proc. Advanced Information Science and Applications – Vol. I, Int. Conf. on Circuits, Systems, Communications and Computers (CSCC 2014)*, July 17-21, 2014, Santorini Island, Greece, ISBN: 978-1-61804-236-1, pp. 51-55.
- D7. Borissova, D., I. Mustakerov, E. Bantutov. Web-based architecture of a system for design assessment of night vision devices. *Int. J. of Information Science and Engineering*, pISSN 2010-376X, Vol. 7(7), 2013, pp. 62-67.
- D8. Mustakerov, I., D. Borissova. Data structures and algorithms of intelligent Web-based system for modular design. *Int. J. of Computer Science and Engineering*, pISSN 2010-376X, eISSN 2010-3778, Vol. 7(7), 2013, pp. 87-92.
- D9. Borissova, D., I. Mustakerov. An algorithm for an optimal staffing problem in open shop environment. *World Academy of Science, Engineering and Technology*, pISSN 2010-376X, Vol. 76, 2013, pp. 46-50.
- D10. Mustakerov, I., D. Borissova. Modular systems design via multi-objective optimization. *Int. J. Advanced Modeling and Optimization*, ISSN: 1841-4311, Vol. 15(2), 2013, pp. 421-430.
- D11. Borissova, D., I. Mustakerov. K-best night vision devices by multi-criteria mixed-integer optimization modeling. *Int. J. of Information Science and Engineering*, pISSN 2010-376X, Vol. 7(10), 2013, pp. 205-210.
- D12. Mustakerov, I., D. Borissova. An intelligent approach for optimum maintenance strategy defining. *In Proc. Innovations in Intelligent Systems and Applications (INISTA), 2013 IEEE International Symposium on*. 19-21 June 2013, Print ISBN: 978-1-4799-0659-8, DOI: 10.1109/INISTA.2013.6577666.
- D13. Borissova, D., I. Mustakerov. A concept of intelligent e-maintenance decision making system. *In Proc. Innovations in Intelligent Systems and Applications (INISTA), 2013 IEEE International Symposium on*. 19-21 June 2013, Print ISBN: 978-1-4799-0659-8, DOI: 10.1109/INISTA.2013.6577668.
- D14. Mustakerov, I., D. Borissova. A combinatorial optimization ranking algorithm for reasonable decision making. *Comptes rendus de l'Academie bulgare des Sciences*, ISSN 0366-8681 0861-1459, Tom 66(1) 2013, pp. 101-110. (IF= 0.210)
- D15. Mustakerov, I., D. Borissova. Investments attractiveness via combinatorial optimization ranking. *Int. J. of Management Science and Engineering*, pISSN 2010-376X, Vol. 7(10), 2013, pp. 230-235.

- D16. Mustakerov, I., D. Borissova. A discrete choice modeling approach to modular systems design. *World Academy of Science, Engineering and Technology*, pISSN 2010-376X, Vol. 76, 2013, pp. 133-139.
- D17. Borissova, D., I. Mustakerov. An integrated framework of designing a decision support system for engineering predictive maintenance. *Int. Journal of Information Technologies & Knowledge*, ISSN 1310-0513, Vol. 6(4), 2012, pp. 366-376.
- D18. Borissova, D., I. Mustakerov, L. Doukovska. Predictive maintenance sensors placement by combinatorial optimization. *Int. J. of Electronics and Telecommunications*. ISSN 0867-6747, Vol. 58(2), 2012, pp. 153-158. (SJR = 0.24)
- D19. Mustakerov, I., D. Borissova. Wind park layout design using combinatorial optimization. *Wind Turbines*, Edited by: Ibrahim Al-Bahadly, ISBN 978-953-307-221-0, Hard cover, 652 pages, Publisher: InTech, April 2011, pp. 403-424.
- D20. Borissova, D., I. Mustakerov. A generalized combinatorial optimization approach to wind power plant design. *Cybernetics and Information Technologies*, ISSN: 1311-9702, Vol. 10(4), 2010, pp. 62-74.
- D21. Mustakerov, I., D. Borissova. Wind turbines type and number choice using combinatorial optimization. *Renewable Energy*. ISSN: 0960-1481, Vol. 35(9), 2010, pp. 1887-1894. (IF: 2.584).
- D22. Borissova, D., I. Mustakerov. A multicriteria approach to exploring combinations of external surveillance conditions defining a given NVD working range value. *Cybernetics and Information Technologies*, ISSN: 1311-9702, Vol. 9(4), 2009, pp. 102-109.
- D23. Borissova, D., I. Mustakerov. A generalized optimization method for night vision devices design considering stochastic external surveillance conditions. *Applied Mathematical Modelling*, ISSN: 0307-904X, Vol. 33, 2009, pp. 4078-4085. (IF: 1.375).
- D24. Мустакеров, И., Д. Борисова. Използване на LINDO API при разработване на приложен софтуер за оптимизация. *Int. Conference "Automatics and Informatics'09"*, ISSN: 1313-1850, 29 Септ – 4 Окт 2009, София, България, pp. I-113 - I-116.
- D25. Borissova D. Night Vision Devices Choice Taking into Account the External Surveillance Conditions. *Int. J. Advanced Modeling and Optimization*, ISSN: 1841-4311, Vol. 10(2), 2008, pp. 213-220.
- D26. Borissova D., I. Mustakerov. Multicriteria choice of night vision devices considering the impact of their performance parameters. *Int. J. Advanced Modeling and Optimization*, ISSN: 1841-4311, Vol. 10(1), 2008, pp. 81-93.
- D27. Mustakerov, I., D. Borissova. Optimal manufacturing scheduling for dependent details processing. *Int. J. of Computer and Information Engineering*. pISSN: 2010-3921, Vol. 2(8) 2008, pp. 493-497.
- D28. Borissova, D. Optimal scheduling for dependent details processing using MS Excel Solver. *Cybernetics and Information Technologies*, ISSN: 1311-9702, Vol. 8(2), 2008, pp. 102-111.
- D29. Borissova, D., I. Mustakerov. Night vision devices design process modeling. In *Proc. of Int. Conf. on Theory and Applications in Mathematics and Informatics, ICTAMI 2007*, Alba Iulia, Romania (Eds. D. Breaz, N. Breaz, D. Wainberg), ISBN 978-973-1890-01-2, 30 August - 02 September 2007, Alba Iulia, Romania, pp. 55-66.
- D30. Mustakerov, I., D. Borissova. Software system for night vision devices design by reasonable combinatorial choice. In *Proc. of Int. Conf. on Theory and Applications in Mathematics and Informatics, ICTAMI 2007*, Alba Iulia, Romania (Eds. D. Breaz, N. Breaz, D. Wainberg), ISBN 978-973-1890-01-2, 30 August - 02 September 2007, Alba Iulia, Romania, pp. 43-53.
- D31. Borissova, D. Using Weighted Sum Method for the Choice of the Night Vision Goggles Battery Power Supply. *Engineering Science*, ISSN 1312-5702, No 1, 2007, pp. 16-26.
- D32. Mustakerov, I., D. Borissova. Technical systems design by combinatorial optimization choice of elements on the example of night vision devices design. *Comptes rendus de l'Academie bulgare des Sciences*, ISSN 0366-8681 0861-1459, Tom 60(4), 2007, pp. 373-380. (IF= 0.106)
- D33. Borissova D., I. Mustakerov. A working distance formula for night vision devices quality preliminary information. *Cybernetics and Information Technologies*, ISSN: 1311-9702, Vol. 6(3), 2006, pp. 85-92.

Списък на цитиранията на публикации от дисертационния труд

- **Borissova D., I. Mustakerov. A working distance formula for night vision devices quality preliminary information. *Cybernetics and Information Technologies*, ISSN: 1311-9702, Vol. 6(3), 2006, pp. 85-92.**
 1. Bantutov E. *Night vision devices? It is simple!* Lambert Academic Publishing, ISBN: 978-3-659-63536-6, 2015.
 2. Bantutov E. An approach for estimation of night vision devices performance in respect of temperature variation. *Int. J. Advanced Modeling and Optimization*, Vol. 16(2), 2014, pp. 261-268.
 3. Бантутов, Е. Моделиране влиянието на температурата върху параметрите на уреди за нощно виждане. *Автореферати на дисертации на ИИКТ при БАН*, No 7, 2014, 43 стр.
 4. Quasay, A.H.Al-Naser. Development of NVD's Using XR5tm techniques and III-V photocathode under night sky conditions. *IRAQI Journal of Applied Physics*, Vol. 7(4), 2011, pp. 33-36. IF=1.1142
 5. Yadav, S., A. Chakraborty, S.P.S. Jalal, A. K. Musla. Design and development of passive night vision binocular with replaceable objective lenses. *ICOP 2009-International Conference on Optics and Photonics*, Chandigarh, India, 30 Oct.-1 Nov. 2009.
- **Mustakerov I., D. Borissova. Technical systems design by combinatorial optimization choice of elements on the example of night vision devices design. *Comptes rendus de l'Academie bulgare des Sciences*, ISSN 0366-8681 0861-1459, Tom 60(4), 2007, pp. 373-380**
 6. Peneva, V, I. Popchev. Fuzzy criteria importance with weighting functions. *Compt. rend. Acad. bulg. Sci.*, 61(3), 2008, pp. 293-300. IF = 0.152
 7. Doukovska, L. Moving Target Hough Detector in Randomly Arriving Impulse Interference. *Cybernetics and Information Technologies*, Vol. 7(3), 2007, pp. 55-72.
 8. Doukovska, L., I. Garvanov, Vl. Kyovtorov. Comparative analysis between HOUGH Detector and averaging detector in conditions of strong pulse jamming. *Problems of Engineering Cybernetics and Robotics*, Vol. 58, 2007, pp. 41-52.
- **Borissova D. Night vision devices choice taking into account the external surveillance conditions. *Int. J. Advanced Modeling and Optimization*, ISSN: 1841-4311, Vol. 10, No 2, 2008, pp.213-220.**
 9. Quasay, A.H.Al-Naser. Development of NVD's Using XR5tm techniques and III-V photocathode under night sky conditions. *IRAQI Journal of Applied Physics*, Vol. 7(4), 2011, pp. 33-36. IF=1.1142
- **Mustakerov I., D. Borissova. Optimal Manufacturing Scheduling for Dependent Details Processing. *International Journal of Computer and Information Engineering*. pISSN: 2010-3921, eISSN: 2010-393X, Vol. 2(8) 2008, pp. 493-497**
 10. University at Albany, State University of New York. Ph.D. Program in Information Science - General Comprehensive Examination, Summer 2012, http://www.albany.edu/informatics/files/2012_INFcomps.pdf
- **Borissova D., I. Mustakerov. Multicriteria choice of night vision devices considering the impact of their performance parameters. *Int. J. Advanced Modeling and Optimization*, ISSN: 1841-4311, Vol. 10, No 1, 2008, pp.81-93.**
 11. Islam, H., M. Jollands, S. Setunge, M. A. Bhuiyan. Optimization approach of balancing life cycle cost and environmental impacts on residential building design. *Energy and Buildings*, Vol. 87, 2015, pp 282-292. IF=3.076
 12. Bantutov E. *Night vision devices? It is simple!* Lambert Academic Publishing, ISBN: 978-3-659-63536-6, 2015.
 13. Islam, H. Use of Material in Residential House Design: An Optimisation Approach Balancing Life Cycle Cost & Life Cycle Environmental Impact. *Doctor of Philosophy, RMIT University, Melbourne, Australia*, 2012
 14. Quasay, A.H.Al-Naser. Development of NVD's Using XR5tm techniques and III-V photocathode under night sky conditions. *IRAQI Journal of Applied Physics*, Vol. 7(4), 2011, pp. 33-36. IF=1.1142

- **Borissova D., I. Mustakerov. A generalized optimization method for night vision devices design considering stochastic external surveillance conditions. *Applied Mathematical Modelling*, ISSN: 0307-904X, 33, 2009, pp. 4078-4085..**
 - 15. Parush, A., M. S. Gauthier, L. Arseneau, D. Tang. The Human Factors of Night Vision Goggles Perceptual, Cognitive, and Physical Factors. *Reviews of human factors and ergonomics*, Vol. 7, 2011, pp. 238-279, doi: 10.1177/1557234X11410392.
 - 16. Bantutov E. *Night vision devices? It is simple!* Lambert Academic Publishing, ISBN: 978-3-659-63536-6, 2015.
- **Borissova D., I. Mustakerov. A generalized combinatorial optimization approach to wind power plant design. *Cybernetics and Information Technologies*, ISSN: 1311-9702, 10(4), 2010, 62-74.**
 - 17. Korsemov, Ch., H. Toshev. Models of Wind Potential Prediction. *Problems of Engineering Cybernetics and Robotics*, Vol. 65, 2012, pp. 49-61.
 - 18. Nikolov, Z., C. Korsemov, H. Toshev. Reliability of Wind Turbine Generators and Exploitation of Wind Farms. *Problems of Engineering Cybernetics and Robotics*, Vol. 63, 2011, pp. 5-13.
 - 19. Nikolov, Z., C. Korsemov, H. Toshev. Reactive Power in Wind Generator Farms and Introduction of Flicker in a Power Line. *Problems of Engineering Cybernetics and Robotics*, Vol. 63, 2011, pp. 22-34.
- **Mustakerov, I., D. Borissova. Wind turbines type and number choice using combinatorial optimization. *Renewable Energy*. ISSN: 0960-1481, Vol. 35(9), 2010, pp. 1887-1894.**
 - 20. Chowdhury, S., W. Tong, A. Mehmani, A. Messac. A Visually-Informed Decision-Making Platform for Wind Farm Layout Optimization. *In Proc. of 11th World Congress on Structural and Multidisciplinary Optimization*, 7 - 12 June 2015, Sydney Australia, pp. 1-6.
 - 21. Rehman, S., S.S.A. Ali. Wind farm layout design using modified particle swarm optimization algorithm. *Journal of Clean Energy Technologies*, Vol. 4(1), 2016, DOI: 10.7763/JOCET.2016.V4.256, pp. 72-77.
 - 22. Dehghani, H., G. B. Gharepetian, B. Vahidi. Study of Effect of Wind DGs Place and Uncertainty on Reliability Indexes in Distribution Network. *Conference: 9th Power Systems Protection and Control (PSPC 2015)*, At Tehran.
 - 23. Jerez, S., F. Thais, I. Tobin, M. Wild, A. Colette, P. Yiou, R. Vautard. The CLIMIX model: A tool to create and evaluate spatially-resolved scenarios of photovoltaic and wind power development. *Renewable and Sustainable Energy Reviews*, Vol. 42, 2015, pp. 1-15. IF=6.796
 - 24. Herbert-Acero, J. F., O. Probst, P.E. Réthoré, G. C. Larsen and K. K. Castillo-Villar. A review of methodological approaches for the design and optimization of wind farms. *Energies*, Vol. 7(11), 2014, pp. 6930-7016. IF = 1.602
 - 25. Ismail, I., S. Kamal, P. Purnomo, S. Sarjiya, P. Prajitno. Optimized design of wind farm configuration: Case study. *Asian Journal of Applied Sciences*, Vol. 2(6), 2014, pp. 936-945.
 - 26. Iqbal, M., M. Azam, M. Naeem, A. S. Khwaja, A. Anpalagan. Optimization classification, algorithms and tools for renewable energy: A review. *Renewable and Sustainable Energy Reviews*, Vol. 39, 2014, pp. 640-654. IF=6.796
 - 27. Montoya, F. G., F. Manzano-Agugliaro, S. López-Márquez, Q. Hernández-Escobedo, C. Gil. Wind turbine selection for wind farm layout using multi-objective evolutionary algorithms. *Expert Systems with Applications*, Vol. 41(15), 2014, pp. 6585-6595. IF=2.254
 - 28. Dobrić, G., Ž. Đurišić. Double-stage genetic algorithm for wind farm layout optimization on complex terrains. *Journal of Renewable Sustainable Energy*, Vol. 6(3), 2014, <http://dx.doi.org/10.1063/1.4881684>. IF=1.176
 - 29. Petković, D., S. H. Ab Hamid, Ž. Čojbašić, N. T. Pavlović. Adapting project management method and ANFIS strategy for variables selection and analyzing wind turbine wake effect. *Natural Hazards*, 2014, DOI 10.1007/s11069-014-1189-1. IF=1.958
 - 30. Billionnet, A., M.-C. Costa, P.-L. Poirion. Robust optimal sizing of an hybrid energy stand-alone system. http://www.optimization-online.org/DB_FILE/2014/03/4291.pdf 2014

31. Chowdhury, S., J. Zhang, W.Tong and A. Messac. Modeling the influence of land-shape on the energy production potential of a wind farm site. *Journal of Energy Resources Technology*, Vol. 136(1), 011203 (2014), doi:10.1115/1.4026201. IF=1.472
32. Lu, S., H. M. Kim. Wind farm layout design optimization through multi-scenario decomposition with complementarity constraints. *Engineering Optimization*, 2014, DOI: 10.1080/0305215X.2013.861457. IF=1.230
33. Mulinazzi, T. E., Z. C. Zheng. Wind farm turbulence impacts on general aviation airports in Kansas. *Report No. K-TRAN: KU-13-6, FINAL REPORT*, January 2014, http://www.copanational.org/files/windfarms_kansas.pdf
34. Turner, S.D.O., D.A. Romero, P.Y. Zhang, C.H. Amon, T.C.Y. Chan. A new mathematical programming approach to optimize wind farm layouts. *Renewable Energy*, Vol. 63, 2014, 674-680. IF= 3.842
35. González, S. J., B. Payán, M., Santos, J.M.R., González-Longatt, F. A review and recent developments in the optimal wind-turbine micro-siting problem. *Renewable & Sustainable Energy Reviews*, Vol. 30, 2014, pp.133-144. IF= 6.796
36. Kurian, S., T. K. Sindhu, E. P. Cherian. Review on developments in wind energy generation and its integration to utility grid. *International Review on Modelling and Simulations (IREMOS)*, Vol 6, No 5, 2013, pp. 1523-1532.
37. Dobric, G., M. Zarkovic, Z. Djuriscic. Fuzzy based computational efficiency for optimal wind farm layout design. *Int. Conf. on Renewable Energy Research and Applications (ICRERA)*, DOI: 10.1109/ICRERA.2013.6749765, 2013, pp. 274 - 279
38. Patent: Methods and systems for warning a wind turbine generator in a wind park of an extreme wind event. Inventors: Martin Ansbjerg KJÆR, Per Brath, Jesper Sandberg Thomsen, Søren DALSGAARD, Publication number: WO2013083131 A1, Publication date: 13 Jun 2013, <http://www.google.st/patents/WO2013083131A1?cl=en>
39. DuPont, B. L., J. Cagan. Multi-stage optimization of wind farms with limiting factors. *ASME 2013 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, Vol. 3A: 39th Design Automation Conference, Portland, Oregon, USA, August 4-7, 2013, ISBN: 978-0-7918-5588-1, Paper No. DETC2013-12503, pp. V03AT03A017; doi:10.1115/DETC2013-12503.
40. Billionnet, A., M.-C.Costa, P.-L. Poirion. Optimisation robuste d'un parc autonome de production d'électricité. *14 Conference ROADEF*, 13-15 Feb, 2013, http://uma.ensta-paristech.fr/files/diam/docro/roadef_2013/papiers/36.pdf
41. Zhang, P. Y. Topics in wind farm layout optimization: Analytical wake models, noise Propagation, and energy production. *Thesis*. 2013, http://individual.utoronto.ca/yun_zhang/Zhang_Yun_201306_MASc_thesis.pdf
42. Gu, H., J. Wang. Irregular-shape wind farm micro-siting optimization. *Energy*, Vol. 57, 2013, pp. 535-544. IF=4.465
43. Lumbreras, S., A. Ramos. Offshore wind farm electrical design: A review. *Wind Energy*, Vol. 16(3), 2013, pp. 459-473. IF=2.556
44. Lumbreras, S., Ramos, A. Optimal design of the electrical layout of an offshore wind farm applying decomposition strategies. *IEEE Transactions on Power Systems*, ISSN: 0885-8950, 28(2), 2013, pp. 1434-1441.
45. Salcedo-Sanz, S., D. Gallo-Marazuela, A. Pastor-Sánchez, L. Carro-Calvo, A. Portilla-Figueras, L. Prieto. Evolutionary computation approaches for real offshore wind farm layout: A case study in northern Europe. *Expert Systems with Applications*, Vol. 40(16), 2013, pp. 6292-6297. IF=2.254
46. Khan, S. A., S. Rehman. Iterative non-deterministic algorithms in on-shore wind farm design: A brief survey. *Renewable and Sustainable Energy Reviews*, Vol. 19, 2013, pp. 370-384. IF=6.796
47. Ibrochim, M. Analisis tekno-ekonomi desain konfigurasi pusat listrik tenaga angin (wind farm) dan perhitungan feed in tariff di Indonesia. *Universitas Indonesia, Fakultas teknik, Teknik elektro depok, Juni, Thesis*, 2012.

48. Yilmaz, E. Benchmarking of optimization modules for two wind farm design software tools. *Gotland University, School of Culture, Energy and Environment, Thesis*, 2012, <http://uu.diva-portal.org/smash/get/diva2:631016/FULLTEXT01>,
49. Du Pont, B. L., J. Cagan. An extended pattern search approach to wind farm layout optimization. *Journal of Mechanical Design*. Vol. 134, 2012, pp.081002-1 – 081002-18. IF=1.394
50. Dobrić, G., Ž. Đurišić. Multi-criteria optimization of wind farm layout for WAsP application. *In Proc. of European Wind Energy Association (EWEA) April*, Copenhagen, Denmark, 2012, pp. 16-19.
51. Dobrić, G., Ž. Đurišić. Optimal wind farm layout design in a complex terrain using two-stage genetic algorithm. 2012, http://www.els.mycpanel.rs/me/images/radovi/31cigre/C6_02.pdf
52. Korsemov, C., H. Toshev. Models of wind potential prediction. *Problems of Engineering Cybernetics and Robotics*, Vol. 65, 2012, pp. 49-61.
53. Rajakumar, S., D. Ravindran. Optimization of wind turbine power coefficient parameters using hybrid technique. *Journal of the Institution of Engineers: Series C*, Vol. 93(2), 2012, pp 141-149.
54. Ekonomou, L., S. Lazarou, G.E. Chatzarakis, V. Vita. Estimation of wind turbines optimal number and produced power in a wind farm using an artificial neural network model. *Simulation Modelling Practice and Theory*, Vol. 21(1), 2012, pp. 21-25. IF= 1.169
55. Karampelas, P., L. Ekonomou, G. P. Fotis, V. Vita. Evaluation of the optimal number of wind turbines in a wind farm using the Downhill Simplex optimization method. *Int. Journal on Power System Optimization*, Vol. 3(1), 2011, pp.11-14.
56. Lazarou, S., V. Vita, L. Ekonomou. Application of Powell's optimisation method for the optimal number of wind turbines in a wind farm. *Science, Measurement & Technology, IET*, Vol. 5(3), 2011, pp.77-80. IF=0.592
57. Banos, R., F. Manzano-Agugliaro, F. G. Montoya, C. Gil, A. Alcayde, J. Gomez. Optimization methods applied to renewable and sustainable energy: A review. *Renewable and Sustainable Energy Reviews*, Vol. 15 (4), 2011, pp. 1753-1766. IF=6.796
58. Saavedra-Moreno B., S. Salcedo-Sanz, A. Paniagua-Tineo, L. Prieto and A. Portilla-Figueras. Seeding evolutionary algorithms with heuristics for optimal wind turbines positioning in wind farms. *Renewable Energy*, Vol. 36(11), 2011, pp. 2838-2844. IF=3.842
59. Billionnet, A., M.-C. Costa, P.-L. Poirion. Optimizing an hybrid energy system. *Conference on Optimization and Practices in Industry (COP11)*, November, 2011, pp. 20-22, http://uma.ensta-paristech.fr/work/labo_work/files/publis/2011/2011-conf-uma1250-COP12011.pdf
60. Billionnet, A., M.-C. Costa, P.-L. Poirion. Recherche opérationnelle et énergie renouvelable: une présentation de quelques problèmes, *ROADEF 2011, 12ème congrès annuel de la Société française de Recherche Opérationnelle et d'Aide à la Décision*, March 2011, pp. 145-146, Saint-Etienne, France, (http://uma.ensta-paristech.fr/files/diam/docro/roadef_2011/VERSION-ELECTRONIQUE/roadef2011_submission_285.pdf)
61. Dykes, K., R. Meadows. Applications of systems engineering to the research, design, and development of wind energy systems. *Technical Report NREL/TP-5000-52616*, December 2011, 92 pages.
62. Nikolov, Z., C. Korsemov, H. Toshev. Reliability of wind turbine generators and exploitation of wind farms. *Problems of Engineering Cybernetics and Robotics*, Vol. 63, 2011, pp. 5-13.
63. Nikolov, Z., C. Korsemov, H. Toshev. Reactive power in wind generator farms and introduction of flicker in a power line. *Problems of Engineering Cybernetics and Robotics*, Vol. 63, 2011, pp. 22-34.
64. Réthoré, P.-E. State of the art in wind farm layout optimization. *Wind Energy Research*. 2010. <http://windenergyresearch.org/2010/10/state-of-the-art-in-wind-farm-layout-optimization/>.

65. Khan, S. A., S. Rehman. Computational intelligence techniques for placement of wind turbines: A brief plan of research in Saudi Arabian perspective. *2010 IEEE International Energy Conference and Exhibition, EnergyCon 2010*, art. no. 5771736, pp. 519-523.
66. Monti, S. Tesi di laurea: Studio delle modifiche indotte al moto ondoso dalla presenza di un impianto eolico in mare. Univesita' Di Pisa, Facoltà di Scienze Matematiche, Fisiche e Naturali Corso di laurea in Scienze e Tecnologie per l'Ambiente ed il Territorio, 2009/2010, http://www.unifi.it/labima/upload/sub/Tesi_Presentazioni/Tesi_S.Monti.pdf
67. Henriksen, L. C. Wind energy literature survey. no. 18. *Wind Energy*, Vol. 13(7), 2010, pp. 685-688. doi: 10.1002/we.432. IF=2.556
- **Mustakerov I., D. Borissova. Wind park layout design using combinatorial optimization. *Wind Turbines*, Edited by: Ibrahim Al-Bahadly, ISBN 978-953-307-221-0, Hard cover, 652 pages, Publisher: InTech, Publication date: April 2011, pp. 403-424.**
 68. Iqbal, M., M. Azam, M. Naeem, A.S. Khwaja, A. Anpalagan. Optimization classification, algorithms and tools for renewable energy: A review. *Renewable and Sustainable Energy Reviews*, Vol. 39, 2014, pp. 640-654. IF=6.796
 69. José F. Herbert-Acero, Oliver Probst, Pierre-Elouan Réthoré, Gunner Chr. Larsen and Krystel K. Castillo-Villar. A Review of Methodological Approaches for the Design and Optimization of Wind Farms. *Energies*, Vol. 7(11), 2014, pp. 6930-7016; doi:10.3390/en7116930. IF = 1.602
 70. Wagner, M. Theory and applications of bio-inspired algorithms. *Thesis, 2013, The University of Adelaide, Adelaide, South Australia.*
 71. Wagner, M., J. Day, F. Neumann. A fast and effective local search algorithm for optimizing the placement of wind turbines. *Renewable Energy*, Vol. 51, 2013, pp. 64-70. IF= 3.842
 72. Dragoi, I. Comparison of optimization for non linear and linear wind resource grids, *Thesis, 2013, Department of Wind Energy, Visby, Sweden*, <http://www.diva-portal.org/smash/get/diva2:693225/FULLTEXT01.pdf>
 73. Yilmaz, E. Benchmarking of optimization modules for two wind farm design software tools. *Gotland University, School of Culture, Energy and Environment, Thesis, 2012*, <http://uu.diva-portal.org/smash/get/diva2:631016/FULLTEXT01>
 74. Korsemov, C., H. Toshev. Models of wind potential prediction. *Problems of Engineering Cybernetics and Robotics*, Vol. 65, 2012, pp. 49-61.
 75. Nikolov, Z., C. Korsemov, H. Toshev. Reliability of wind turbine generators and exploitation of wind farms. *Problems of Engineering Cybernetics and Robotics*, Vol. 63, 2011, pp. 5-13.
 76. Nikolov, Z., C. Korsemov, H. Toshev. Reactive power in wind generator farms and introduction of flicker in a power line. *Problems of Engineering Cybernetics and Robotics*, Vol. 63, 2011, pp. 22-34.
 77. Réthoré, P.-E. State of the art in wind farm layout optimization. *Wind Energy Research*. 2010. <http://windenergyresearch.org/2010/10/state-of-the-art-in-wind-farm-layout-optimization/>.
- **Borissova, D., I. Mustakerov. An integrated framework of designing a decision support system for engineering predictive maintenance. *Int. Journal of Information Technologies & Knowledge*, ISSN 1310-0513 (printed), ISSN 1313-0463 (online), Vol. 6, No 4, 2012, pp. 366-376**
 78. Bao-luo, L. Framework and intervention type of sharable clinical decision support system and its knowledge representation. *China Digital Medicine*. Vol.9, No 6, 2014 pp. 42 to 45.
 79. Гарванов, И. *Методи и алгоритми за откриване на цели*. Изд. „Авангард Прима”, ISBN 978-619-160-317-6, 2014, 277 стр.
- **Borissova D., I. Mustakerov, L. Doukovska. Predictive maintenance sensors placement by combinatorial optimization. *Int. J. of Electronics and Telecommunications*. ISSN 0867-6747, 58(2), 2012, pp. 153-158.**
 80. Vincenzi, L., L. Simonini. Influence of correlation length in optimal sensor placement. *In Proc. of 11th Int. Conference on Vibration Problems*, Z. Dimitrovova et.al. (eds.), Lisbon, Portugal, 9–12 September 2013

- **Borissova, D., I. Mustakerov. An algorithm for an optimal staffing problem in open shop environment. *World Academy of Science, Engineering and Technology*, pISSN 2010-376X, eISSN 2010-3778, Vol. 76, 2013, pp. 46-50.**
 - 81. Гарванов, И. *Методи и алгоритми за откриване на цели*. Изд. „Авангард Прима”, ISBN 978-619-160-317-6, 2014, 277 стр.
- **Mustakerov I., D. Borissova. An intelligent approach for optimum maintenance strategy defining. *IEEE International Symposium on Innovations in Intelligent Systems and Applications (INISTA)*, 2013, Print ISBN: 978-1-4799-0659-8, DOI: 10.1109/INISTA.2013.6577666.**
 - 82. Tauterat, T. Development of a Method for the Economic Evaluation of Predictive Maintenance. *Lecture Notes in Business Information Processing*, ISBN: 978-3-319-19592-6, Vol. 210, 2015, pp 179-185.
 - 83. Segura, R.P., P. Ramírez, C. Vásquez. Criterios de evaluación de caducidad y obsolescencia de procesos, aplicado a la Caja Costarricense de Seguro Social. *Ingeniería*, 2014, Vol. 24 (2), pp. 93-104.
 - 84. Гарванов, И. *Методи и алгоритми за откриване на цели*. Изд. „Авангард Прима”, ISBN 978-619-160-317-6, 2014, 277 стр.
- **Borissova D., I. Mustakerov. A concept of intelligent e-maintenance decision making system. *IEEE International Symposium on Innovations in Intelligent Systems and Applications (INISTA)*, 2013, 19-21 June 2013, Print ISBN: 978-1-4799-0659-8, DOI: 10.1109/INISTA.2013.6577668.**
 - 85. Houessou, B. Conception d’une trousse d’analyse d’opportunités pour l’implantation d’une solution enrichissant les systèmes de GMAO existants, *Thesis, 2015, 219 pages, Ecole de Technologie Supérieure Université du Québec, Montreal, Canada*, <http://espace.etsmtl.ca/1438/1/HOUESSOUBruno.pdf>
 - 86. Paukkonen, H., T. Vartiainen. E-maintenance. *Lappeenranta teknillinen yliopisto, tuotantotalous, Finland, 2015, 47 pages*, <http://www.doria.fi/bitstream/handle/10024/105185/Kandidaatinty%C3%B6%20Paukkonen%20Vartiainen.pdf?sequence=2>
- **Mustakerov I., D. Borissova. A discrete choice modeling approach to modular systems design. *World Academy of Science, Engineering and Technology*, pISSN 2010-376X, eISSN 2010-3778, Vol. 76, 2013, pp. 133-139.**
 - 87. Tulu, N. G. Monitor for displaying the status of Real-Time simulation. Thesis, 2014, http://brage.bibsys.no/xmlui/bitstream/id/209056/Thesis_NigussieGirmaTulu.pdf
 - 88. Гарванов, И. *Методи и алгоритми за откриване на цели*. Изд. „Авангард Прима”, ISBN 978-619-160-317-6, 2014, 277 стр.
- **Mustakerov I., D. Borissova. A combinatorial optimization ranking algorithm for reasonable decision making. *Comptes rendus de l’Academie bulgare des Sciences*, ISSN 0366-8681 0861-1459, 66(1) 2013, pp. 101-110.**
 - 89. Гарванов, И. *Методи и алгоритми за откриване на цели*. Изд. „Авангард Прима”, ISBN 978-619-160-317-6, 2014, 277 стр.

Библиография

1. Aarts, E., J. Korst. *Simulated Annealing and Boltzmann Machines*, Wiley, Chichester, UK, 1989.
2. Andro-Vasko, J. *Shop problems in scheduling*. UNLV Theses/Dissertations/Professional Papers/Capstones. 2011, paper 945.
3. Akhshabi, M. A new fuzzy multi criteria model for maintenance policy. *Middle-East Journal of Scientific Research*, Vol. 10, 2011, pp. 33-38.
4. AlGeddawy, T., H. ElMaraghy. Reactive design methodology for product family platforms, modularity and parts integration. *CIRP Journal of Manufacturing Science and Technology*, Vol. 6, 2013, pp. 34-43.
5. Andersson, J. A survey of multiobjective optimization in engineering design. *Technical Report: LiTH-IKP-R-1097*, 2000, pp. 34.
6. Antonio, K.W. L., R. C. M. Yam, E. Tang. The impacts of product modularity on competitive capabilities and performance: An empirical study. *Int. Journal Production Economics*, Vol. 105, 2007, pp.1-20.
7. Akturk, M. S., J. B. Ghosh and E. D. Gunes. Scheduling with tool changes to minimize total completion time: A study of heuristics and their performance. *Naval Research Logistics*, Vol. 50(1), 2003, pp. 15-30.
8. Aktin T. and R. G. Ozdemir. An integrated approach to the one-dimensional cutting stock problem in coronary stent manufacturing. *European Journal of Operational Research*, Vol. 196, 2009, pp. 737-743
9. Atamturk A., M. W. P. Savelsbergh. Integer-Programming Software Systems. *Annals of Operations Research*, Vol. 140, 2005, pp. 67-124.
10. Atanassova, V., L. Doukovska, D. Mavrov, Kr. Atanasov. InterCriteria Decision Making Approach to EU Member States Competitiveness Analysis: Temporal and Threshold Analysis. *Intelligent Systems'2014, Advances in Intelligent Systems and Computing*, Vol. 322, 2015, pp. 95-106.
11. Athan, T.W., P.Y. Papalambros. A note on weighted criteria methods for compromise solutions in multi-objective optimization. *Optimization and Engineering*, Vol. 27, 1996, pp. 155-176.
12. Atkinson, J.B. A greedy look-ahead heuristic for combinatorial optimization: an application to vehicle scheduling with time windows. *Journal of Operational Research Society*, Vol. 45, 1994, pp. 673-684.
13. Avak, B. Variant management of modular product families in the market phase. VDI Verlag (2007), <http://dx.doi.org/10.3929/ethz-a-005320674>.
14. Bai, D., L. Tang. Open shop scheduling problem to minimize makespan with release dates. *Applied Mathematical Modelling*, Vol. 37(4), 2013, pp. 2008-2015.
15. Baker, D., D. Bridges, R. Hunter, G. Johnson, J. Krupa, J. Murphy, K. Sorenson, *Guidebook to decision-making methods*, WSRC-IM-2002-00002, Department of Energy, USA, 2002.
16. Baker, K. R., B. Keller. Solving the single-machine sequencing problem using integer programming. *Computers & Industrial Engineering*, Vol. 59(4), 2010, pp. 730-735.
17. Balas, E., S. Ceria, G. Cornuejols, N. Natraj, Gomory cuts revisited. *Operations Research Letters*, Vol. 19, 1996, pp. 1-9.
18. Balev, S., N. Yanev, A. Fréville, R. Andonov. A dynamic programming based reduction procedure for the multidimensional 0-1 knapsack problem. *EJOR*, Vol. 186(1), 2008, pp. 63-76.
19. Battarra, M., A. A. Pessoa, A. Subramanian, E. Uchoa. Exact algorithms for the traveling salesman problem with draft limits. *EJOR*, Vol. 235, 2014, pp. 115-128.
20. Behzadian M., S. K. Otaghsara, M. Yazdani, J. Ignatius. A state-of the-art survey of TOPSIS applications. *Expert Systems with Applications*, Vol. 39, 2012, pp. 13051-13069.
21. Behzadian, M., R. B. Kazemzadeh, A. Albadvi, M. Aghdasi. PROMETHEE: A comprehensive literature review on methodologies and applications. *EJOR*, Vol. 200, 2010, pp. 198-215.
22. Bjorndal, M.H., A. Caprara, P.I. Cowling, F. Della Croce, H. Lourenço, F. Malucelli, A.J. Orman, D. Pisinger, C. Rego, J.J. Salazar. Some thoughts on combinatorial optimisation. *EJOR*, Vol. 83(2), 1995, pp. 253-270.
23. Błażewicza, J., E. Peschb, M. Sternaa, F. Werner. Open shop scheduling problems with late work criteria. *Discrete Applied Mathematics*, Vol. 134(1-3), 2004, pp. 1-24.
24. Blochlinger, I. Modeling staff scheduling problems. A Tutorial. *EJOR*, Vol. 158, 2004, pp. 533-542.
25. Borissova D., I. Mustakerov. A working distance formula for night vision devices quality preliminary information. *Cybernetics and Information Technologies*, Vol. 6(3), 2006, pp. 85-92.
26. Borissova D., I. Mustakerov. Method of rational choice by sorting in the software system "NVGpro". *Cybernetics and Information Technologies*, Vol. 6(1), 2006, pp. 69-75.
27. Borissova D. Using weighted sum method for the choice of the night vision goggles battery power supply. *Engineering Science*, 2007, No 1, pp. 16-26.
28. Borissova D., I. Mustakerov. Night vision devices design process modeling. In *Proc. of Int. Conf. on Theory and Applications in Mathematics and Informatics, ICTAMI 2007*, Alba Iulia, Romania (Eds. D. Breaz, N. Breaz, D. Wainberg), 30 August - 02 September 2007, Alba Iulia, Romania, pp. 55-66.

29. Borissova D., I. Mustakerov. A generalized optimization method for night vision devices design considering stochastic external surveillance conditions. *Applied Mathematical Modelling*, Vol. 33(11), 2009, pp. 4078-4085.
30. Borissova D. Optimal scheduling for dependent details processing using MS Excel solver. *Cybernetics and Information Technologies*, Vol. 8(2), 2008, pp. 102-111.
31. Borissova, D. and I. Mustakerov. Multicriteria choice of night vision devices considering the impact of their performance parameters. *Advanced Modeling and Optimization*, Vol. 10(1), 2008, pp. 81-93.
32. Borissova D., I. Mustakerov. A multicriteria approach to exploring combinations of external surveillance conditions defining a given NVD working range value. *Cybernetics and Information Technologies*, Vol. 9(4), 2009, pp. 102-109.
33. Borissova D., I. Mustakerov. A generalized combinatorial optimization approach to wind power plant design. *Cybernetics and Information Technologies*, Vol. 10(4), 2010, pp. 62-74.
34. Borissova D., I. Mustakerov, L. Doukovska. Predictive maintenance sensors placement by combinatorial optimization. *Int. Journal of Electronics and Telecommunications*, Vol. 58(2), 2012, pp. 153-158.
35. Borissova, D., I. Mustakerov. An integrated framework of designing a decision support system for engineering predictive maintenance. *Int. Journal of Information Technologies & Knowledge*, Vol. 6(4), 2012, pp. 366-376.
36. Borissova D., I. Mustakerov, E. Bantutov. Web-based architecture of a system for design assessment of night vision devices. *Int. Journal of Information Science and Engineering*, Vol. 7(7), 2013, pp. 62-67.
37. Borissova, D., I. Mustakerov. An algorithm for an optimal staffing problem in open shop environment. *World Academy of Science, Engineering and Technology*, Issue 76, 2013, pp. 46-50.
38. Borissova D., I. Mustakerov. A concept of intelligent e-maintenance decision making system. *In Proc. of Innovations in Intelligent Systems and Applications (INISTA)*, 2013 IEEE International Symposium on. 19-21 June 2013, DOI: 10.1109/INISTA.2013.6577668.
39. Borissova D., I. Mustakerov, E. Bantutov. Methodology for determining of surveillance conditions in relation to night vision devices performance. *Advanced Modeling and Optimization*, Vol. 16(1), 2014, pp. 51-59.
40. Borissova D., I. Mustakerov. Web-based tool for preliminary assessment of wind power plant design. *In Proc. of Int. Conf. on Information Systems and Technologies (ICIST 2014)*, March 22-24, 2014, Valencia, Spain, pp. 139-149.
41. Borissova D., I. Mustakerov. A parallel algorithm for optimal job shop scheduling of semi-constrained details processing on multiple machines. *In Proc. Advanced Information Science and Applications – Vol. I, In Proc. Int. Conf. on Circuits, Systems, Communications and Computers (CSCC 2014)*, July 17-21, 2014, Santorini Island, Greece, pp. 145-150.
42. Bouvard, K., S. Artus, C. Bérenguer and V. Cocquempot. Condition-based dynamic maintenance operations planning & grouping. Application to commercial heavy vehicles. *Reliability Engineering and System Safety*, Vol. 96, 2011, pp. 601-610.
43. Campbell H. F. and R. P.C. Brown. A multiple account framework for cost-benefit analysis. *Evaluation and Program Planning*, Vol. 28, 2005, pp. 23-32.
44. Cavalcante, C.A.V., R.J.P. Ferreira and A. Teixeira de Almeida. A preventive maintenance decision model based on multicriteria method PROMETHEE II integrated with Bayesian approach. *IMA Journal of Management Mathematics*, Vol. 21, 2010, pp. 333-348.
45. Charnes, A., W. W.Cooper. *Management models and industrial applications of linear programing*. New York, John Wiley & Sons, 1961.
46. Chen,W., M.M.Wiecek, J. Zhang. Quality utility – a compromise programming approach to robust design. *Journal of Mechanical Design*, Vol. 121, 1999, pp. 179-187.
47. Chen, W., A. Sahai, A. Messac, G. Sundararaj. Exploration of the effectiveness of physical programming in robust design. *Journal of Mechanical Design*. Vol. 122, 2000, pp. 155-163.
48. Cherri, C., M. N. Arenales and H. H. Yanasse. The one-dimensional cutting stock problem with usable leftover – A heuristic approach. *EJOR*, Vol. 196, 2009, pp. 897-908.
49. Chowdhury, S., A. Akram. E-maintenance: Opportunities and challenges. *Proc. IRIS*, 2011, pp. 68-81.
50. Clausen, J. Branch and Bound Algorithms – Principles and Examples. *Technical report*. 1999, University of Copenhagen.
51. Coello, C.A.C., D.A.V. Veldhuizen, G.B. Lamont. *Evolutionary algorithms for solving Multi-Objective Problems*. Kluwer, New York, 2002.
52. Cui, Y. and Y. Lu. Heuristic algorithm for a cutting stock problem in the steel bridge construction. *Computers & Operations Research*, Vol. 36, 2009, pp. 612-622.
53. Curtis, Gr. and D. Cobham. *Business Information Systems – Analysis, Design and Practice*. 2008. Pearson Education Ltd.
54. Deb, K. *Multi-Objective Optimization using Evolutionary Algorithms*. Wiley: New York, 2001

55. Dantzig, G. Origins of the Simplex Method. *Technical Report Sol 87-5*. May 1987, <http://www.dtic.mil/dtic/tr/fulltext/u2/a182708.pdf>
56. Desrosiers, J., M.E. Lubbecke. Branch-Price-and-Cut Algorithms. *Wiley Encyclopedia of Operations Research and Management Science*, 2010.
57. Dikili, A. C., A. C. Takinaci and N. A. Pek. A new heuristic approach to one-dimensional stock-cutting problems. *Ocean Engineering*, Vol. 35(7), 2008, pp. 637-645.
58. Dikili, A. C., E. Sarioz and N. A. Pek. A successive elimination method for one-dimensional stock cutting problems in ship production. *Ocean Engineering*, Vol. 34, 2007, pp. 1841-1849.
59. Dimov, I. T., R. Georgieva. Multidimensional Sensitivity Analysis of Large-scale Mathematical Models. O.P. Iliev et al. (eds.), *Numerical Solution of Partial Differential Equations: Theory, Algorithms, and Their Applications, Springer Proceedings in Mathematics & Statistics*, Vol. 45, 2013, pp. 137-156.
60. Doebling, S.W., C.R.Farrar, M. B.Prime, D.W. Shevitz. Damage identification and health monitoring of structural and mechanical systems from changes in their vibrational characteristics: a literature review. Los Alamos National Laboratory, 1996, LA-13070-MS.
61. Donovan, S. Wind farm optimization. In: *40th Annual Conference of the Operations Research Society*, Wellington: New Zealand; 2005; 196–205
62. Dorigo, M., T. Stutzle. *Ant Colony Optimization*. MIT Press, Cambridge, MA 2004.
63. Dorsey, J. T., T. J. Collins, W. R. Doggett, R. V. Moe. Framework for defining and assessing benefits of a modular assembly design approach for exploration systems. In *Proc. Space Technology and Applications International Forum – STAIF 2006*, Vol. 813, pp. 969-981.
64. Dowsland, K. Simulated Annealing, In: *Modern Heuristic Techniques for Combinatorial Problems*, (Reeves C.R. Ed.). Blackwell, 1993, pp. 20-69.
65. Dubois, D., H. Fargier, H. Prade. Fuzzy constraint in job shop scheduling. *Journal of Intelligent Manufacturing*, Vol. 6, 1995, pp. 215-234.
66. Ehrgott, M., M. Wiecek. *Multiobjective Programming*. In: *Multiple Criteria Decision Analysis. State of the Art Surveys*. (Figueire J., Greco S., Ehrgott M. Eds.). Springer. 2005.
67. Eiben, A., J.Smith. *Introduction to Evolutionary Computing*. Springer, Berlin Heidelberg, 2003.
68. Emami, A., P. Noghreh. New approach on optimization in placement of wind turbines within wind farm by genetic algorithms. *Renewable Energy*, Vol.35, 2010, pp.1559-1564.
69. Ettoumi, F. Y., A.El H. Adane, M.L. Benzaoui, N. Bouzergui. Comparative simulation of wind park design and siting in Algeria. *Renewable Energy*, Vol.33, 10, 2008, pp. 2333-2338.
70. Farrar, Ch. R., K. Worden. An introduction to structural health monitoring. *Transactions of the Royal Society A*, 2007, 365, doi: 10.1098/rsta.2006.1928.
71. Ferreira, R.J.P., A. Teixeira de Almeida and C.A.V. Cavalcante. A multi-criteria decision model to determine inspection intervals of condition monitoring based on delay time analysis. *Reliability Engineering and System Safety*, Vol. 94, 2009, pp. 905-912.
72. Fujita, K., H. Sakaguchi, S. Akagi. Product variety deployment and its optimization under modular architecture and module communalization. In *Proc. of the ASME Design Engineering Technical Conferences*, 1999, Las Vegas, Nevada, DETC99/DFM-8923.
73. Fujita, K. Product Variety Optimization under modular architecture. *Computer-Aided Design*, Vol. 4(12), 2002, pp. 953-965.
74. Gandhare, B. and M. Akarte. Maintenance strategy selection. In *Proc. of Ninth AIMS Int. Conf. on Management*, pp. 1330-1336, 2012.
75. Garey, M. R., D. S. Johnson, and R. Sethi. The complexity of flowshop and job shop scheduling. *Mathematics of Operations Research*, Vol. 1(2), 1976, pp. 117-129.
76. Garcia-Cascales, M. S., M. T. Lamata. Selection of a cleaning system for engine maintenance based on the analytic hierarchy process. *Computers & Industrial Engineering*, Vol. 56, 2009, pp. 1442-1451.
77. Garrett, J. Design and Technology. 1991, Cambridge. University Press, England
78. Gavish, B. Topological design of computer communication networks -The overall design problem. *EJOR*, Vol. 58, 1992, pp. 149-172.
79. Gershenson, J. K., G. J. Prasad. Modularity in product design for manufacturability. *Int. J. of Agile Manufacturing*, Vol. 1, no 1, pp. 1-11, 1997.
80. Gilmore, P. and R. Gomory. A linear programming approach to the cutting stock problem. *Operations Research*, Vol. 9, no. 6, 1961, pp. 848-859.
81. Gilmore, P. and R. Gomory. A linear programming approach to the cutting stock problem, part II. *Operations Research*, Vol. 11, 1963, pp. 863-888.
82. Ginevicius, R. A new determining method for the criteria weights in multicriteria evaluation. *Int. Journal of Information Technology & Decision Making*, Vol. 10, 2011, pp. 1067-1095.
83. Glover, F. Future paths for integer programming and artificial intelligence. *Computers and Operations Research*, Vol. 13(5), 1986, pp. 533-549.

84. Glover, F., M. Laguna. Tabu Search. In: *Handbook of Applied Optimization*, (Pardalos P. M. and Resende M.G.C., Eds.), Oxford University Press, 2002, pp. 194-208.
85. Glover, F. Tabu Search, Part I. *ORSA Journal on Computing*, Vol. 1(3), 1989, pp. 190-206.
86. Goldberg, D.E. *Genetic Algorithms in Search. Optimization and Machine Learning*. Addison-Wesley, Reading, MA, 1969.
87. Golden, B., L. Bodin, T. Doyle, W. Stewart Jr. Approximate traveling salesman algorithm. *Operations Research*, Vol. 28, 1980, pp. 694-711.
88. Golenko-Ginzburg, D., A. Gonik. Optimal job-shop scheduling with random operations and cost objectives. *Int. Journal of Production Economics*, Vol. 76(2), 2002, pp. 147-157.
89. Gomez, A. Oakes, William. Leone, Les. *Engineering Your Future*. Great Lake Press, 2004, pp.352-363.
90. Gonzálezp J.S., A.G.G. Rodriguez, J.C. Mora, J.R.Santos, M.B. Payan. Optimization of wind farm turbines layout using an evolutive algorithm. *Renew Energy* 2010; 35(8):1671–81
91. Grady, S.A., M.Y. Hussaini, M. M. Abdullah. Placement of wind turbines using genetic algorithms. *Renewable Energy*, Vol. 30, 2005, pp. 259-270.
92. Griffin, R. W. *Principles of Management*. Houghton Mifflin Company, 2007, 459 pages.
93. Grigorios, M., St. Lazarou, El. Pyrgioti. Optimal placement of wind turbines in a wind park using Monte Carlo simulation. *Renewable Energy*, Vol. 33, 2008, pp.1455-1460.
94. Groba, C., S. Cech, F. Rosenthal, A. Gossling. Architecture of a predictive maintenance framework. In *Proc. Int. Conf. Computer Information Systems and Industrial Management Applications*, 2007, pp. 59-64.
95. Gromicho, J. A. S., J. J. van Hoorn, F. Saldanha-da-Gama and G. T. Timmer. Solving the job-shop scheduling problem optimally by dynamic programming. *Computers & Operations Research*, Vol. 39, 2012, pp. 2968-2975.
96. Guratzsch, R. F., S. Mahadevan. Sensor placement design for SHM under uncertainty. *Third European Workshop on Structural Health Monitoring*, July 5-7, Granada, Spain, 2006.
97. Haessler, R. W. and P. E. Sweeney. Cutting stock problems and solution procedures. *EJOR*, Vol. 54, 1991, pp. 141-150.
98. Hameed, Z., Y. S.Hong, Y. M. Cho, S. H. Ahn, C. K. Song. Condition monitoring and fault detection of wind turbines and related algorithms: A review. *Renewable and Sustainable Energy Reviews*, Vol. 13, 2009, pp. 1-39.
99. Hasiija, S., E. Pinker, R. A. Shumsky. Capacity estimation and optimal staffing for an email contact center. 2008, http://mba.tuck.dartmouth.edu/pages/faculty/robert.shumsky/contact_center_estimation_july_2008.pdf.
100. Harrell, Frank E., Jr. *Regression modelling strategies*. Springer-Verlag, 2001.
101. Hertel, M., *Aspects of AJAX*, 2007, <http://www.mathertel.de/AJAX/AJAXeBook.aspx>.
102. Higginbotham, K.D. Effect of using high signal-to-noise image intensifier tubes on night vision goggle (NVG) aided visual acuity, USA, Master's Thesis, 2006.
103. Ignizio, J.P. *Goal programming and extensions*. Lexington Books, Lexington, KY1976.
104. Indiso, S.B. The transmittance of the atmosphere for solar radiation on individual clear days. *Journal of Applied Meteorology*, Vol. 9, 1970, pp. 239-241.
105. Jae-Kyu, L., S. Sung-Hoon, K. Suhn-Baum. Configuration of personal computer by constraint and rule satisfaction problem approach. In *Proc. First Asian Pacific DSI Conference*, Hongkong, 1996, pp. 1-22.
106. Jotsov V., V.Sgurev. Human-centered methods for applications of fuzzy probabilities in intelligent systems, *Proc. 2nd Int. IEEE Conference 'Intelligent Systems'* (R. Yager, V.Sgurev, Eds.), Varna, Bulgaria, June 22-24, V. I, 2004, pp. 122-129.
107. Kalouptsidis N., K. Koutroumbas, V. Psaraki. Classification methods for random utility models with i.i.d. disturbances under the most probable alternative rule. *EJOR*, Vol. 176, pp. 1778-1794, 2007.
108. Kantorovich, L. V. Mathematical methods of organizing and planning production. *Management Science*, Vol. 6, 1960, pp. 366-422.
109. Kamrani, A.K., E.A. Nasr. *Collaborative Engineering*. Springer, 2008, ch. 10.
110. Karmarkar, N. A. New Polynomial-time algorithm for linear programming. *Combinatorica*, Vol. 4(4), 1984, pp. 373-395.
111. Kaveh, A., S.M. Javadi, M.Maniat. Damage assessment via modal data with a mixed particle swarm strategy, Ray Optimizer, and Harmony Search. *Asian Journal of Civil Engineering (BHRC)*, Vol. 15(1), 2014, pp. 95-106.
112. Kidd, P. T. *Agile Manufacturing: Forging New Frontiers*, Addison Wesley, New York, 1994.
113. Kong, X., L.Gao, H. Ouyang, S. Li. A simplified binary harmony search algorithm for large scale 0–1 4 knapsack problems. *Expert Systems with Applications*, Vol. 42(12), 2015, pp. 5337-5355.
114. Korsemov, Ch., Hr. Toshev, I. Mustakerov, D. Borissova and V. Grigorova. An optimal approach to design of joinery for renovation of panel buildings. *Int. Journal of Science and Engineering Investigations*, Vol. 2(18), 2013, pp. 123-128.

115. Koski, J., R. Silvennoinen. Norm methods and partial weighting in multicriterion optimization of structures. *Int. Journal for Numerical Methods in Engineering*, Vol. 24, 1987, pp. 1101-1121.
116. Koski, J. Multicriterion optimization in structural design. In: Atrek, E.; Gallagher, R.H.; Ragsdell, K.M.; Zienkiewicz, O. C. (eds.) *New Directions in Optimum Structural Design*, 1984, pp. 483-503.
117. Koulamas, Ch., G. J. Kyparisis. The three-machine proportionate open shop and mixed shop minimum makespan problems. *EJOR*, Vol. 243, 2015, pp. 70-74.
118. Krapohla, D. Structured Methodology for Group Decision Making, 2013, <http://www.augmentedintel.com/wordpress/index.php/a-structured-methodology-for-group-decision-making/>
119. Kreng, V. B., Tseng-Pin Lee., Modular product design with grouping genetic algorithm – a case study. *Computers & Industrial Engineering*, Vol. 46(3), 2004, pp. 443-460.
120. Kulak O., S. Cebi, C. Kahraman. Applications of axiomatic design principles: A literature review. *Expert Systems with Applications*, Vol. 37, 2010, pp. 6705-6717.
121. Kwon, O., K. Yoo, E. Suh. UbiDSS: A proactive intelligent decision support system as an expert system deploying ubiquitous computing technologies. *Expert System with Applications*, Vol. 28(1), 2005, pp. 149-161.
122. Laporte, G. The traveling salesman problem – an overview of exact and approximate algorithms. *EJOR*, Vol. 59, 1992, pp. 231-247.
123. Lawler, E.L., Lenstra, J.K, Rinnooy Kan, A.H.G., D.B. Shmoys. *The traveling salesman problem*, Wiley, New York, 1985.
124. Lawrence, S. R., E.C. Sewell. Heuristic optimal static and dynamic schedules when processing times are uncertain. *Journal of Operations Management*, Vol. 15(1), 1997, pp. 71-82
125. Lee, J., J. Ni, D. Djurdjanovic, H. Qiu, H. Liao. Intelligent prognostics tools and e-maintenance. *Computers in Industry*, Vol. 57, 2006, pp. 476-489.
126. Lee, S.M. *Goal Programming for decision analysis*. Auerbach, Philadelphia, 1972.
127. Levin, M. S. Combinatorial optimization in system configuration design. *Automation and Remote Control*, Vol. 70(3), 2009, pp. 519-561.
128. Li, B., J.Ou, X.Zhao, D. Li. Optimal sensor placement in health monitoring system. *Int. Workshop on Advanced Smart Materials and Smart Structures Technology*, July 25-26, 2011, Dalian, China.
129. Yi, Ting-Hua, Li, Hong-Nan. Gu, Ming. Optimal sensor placement for structural health monitoring based on multiple optimization strategies. *The Structural Design of Tall and Special Buildings*, Vol. 20, 2011, pp. 881-900.
130. Li,Y., L. Chun, A. Nee and Y. Ching. An agent-based platform for web enabled equipment predictive maintenance. In: *Proc. of IAT'05 IEEE/WIC/ACM international conference on intelligent agent technology*, Compiègne, France, 2005.
131. Li, Z. N., J. Tang, Q. S. Li. Optimal sensor locations for structural vibration measurements. *Applied Acoustics*, Vol. 65, 2004, pp. 807-818.
132. *LINDO API 7.0 User Manual*, LINDO Systems, Inc., 2012
133. Liu, W., Gao, W., Sun, Y. and Xu, M., (2008) Optimal sensor placement for spatial lattice structure based on genetic algorithms. *Journal of Sound and Vibration*, Vol. 317, pp. 175-189.
134. Malapert, A., H. Cambazard, N. Jussien, A. Langevin, L.-M. Rousseau. An optimal constraint programming approach to the open-shop problem. *NFORMS Journal on Computing*, Vol. 24(2), pp. 2012, 228-244.
135. Mahadevan, M.L., S.P. Kumar and R.Vinodh. Preventive maintenance optimization of critical equipments in process plant using heuristic algorithms. In *Proc. of Conf. on Industrial Engineering and Operations Management*, Dhaka, Bangladesh, 2010.
136. Mahi, M., Ö. K. Baykan, H. Kodaz. A new hybrid method based on Particle Swarm Optimization, Ant Colony Optimization and 3-Opt algorithms for Traveling Salesman Problem. *Applied Soft Computing*, Vol. 30, 2015, pp. 484-490.
137. Malaguti, E., R. M. Durán, P. Toth. Approaches to real world two-dimensional cutting problems. *Omega*, Vol. 47, 2014, pp. 99-115.
138. Manwell, J. F., J. G. McGowan, A. L. Rogers. *Wind Energy Explained*. West Sussex, England, John Wiley & Sons Ltd. 2002.
139. Marasco, P.L. and Task, H.L., (2003) The impact of target luminance and radiance on night vision device visual performance testing. In *Proc. Helmet- and Head-Mounted Displays VIII: Technologies and Applications*, DOI: 10.1117/12.487869.
140. Marler, R.T., J.S. Arora. Survey of multi-objective optimization methods for engineering. *Structural and Multidisciplinary Optimization*, Vol. 26, 2004, pp. 369-395.
141. Marler, R.T., J.S. Arora. The weighted sum method for multi-objective optimization: new insights. *Structural and Multidisciplinary Optimization*, Vol. 41, 2010, pp. 853-862.
142. Marler, R.T., J.S. Arora. Function-transformation methods for multi-objective optimization. *Engineering Optimization*, Vol. 37(6), 2005, pp. 551-570.
143. Marmidis, G.; Lazarou, S. & Pyrgioti, E. Optimal placement of wind turbines in a wind park using Monte Carlo simulation. *Renewable Energy*, Vol. 33(7), 2008, pp. 1455-1460.

144. Mason A. J. and I. Dunning. OpenSolver: Open source optimisation for Excel. *In Proc. of the 45th Annual Conference of the Operations Research Society of New Zealand*, November 2010, pp. 181-190.
145. Mathew, A.D., S. Zhang, L. Ma, T. Earle, D.J. Hargreaves. Reducing maintenance cost through effective prediction analysis and process integration. *Advances in Vibration Engineering*, Vol. 5(2), 2006, pp. 87-96.
146. Mavrotasp G. Effective implementation of the e-constraint method in multi-fbjective mathematical programming problems. *Applied Mathematics and Computation*, Vol. 213, 2009, pp. 455-465.
147. McDermott, J. R1: A rule-based configurer of computer systems. *Artificial Intelligence*, Vol. 19(1), 1982, pp. 39-88.
148. Michalewicz Z., D.B. Fogel. *How to solve it: modern heuristics*. Springer-Verlag, New York, Inc 2000.
149. Miettinen, K., M.M. Makela. On scalarizing functions in multiobjective optimization. *OR Spectrum*, Vol. 24(2), 2002, pp. 193-213.
150. Miettinen, K. *Nonlinear Multiobjective Optimization*. Kluwer Academic Publishers, Boston 1999.
151. Minoux, M. Network synthesis and optimum network design problems: Models, solution methods and applications. *Networks*, Vol. 19, 1989, pp. 313-360.
152. Mitchell, J.E. *Branch-and-Cut Algorithms for Combinatorial Optimization Problems*. Handbook of Applied Optimization, 2002, pp. 65-77.
153. Miwa, K., S. Takakuwa. Optimization and analysis of staffing problems at a retail store. *In Proc. 2010 Winter Simulation Conf.*, 2010, pp. 1911-1923.
154. Mobasher, A. and A. Ekici. Solution approaches for the cutting stock problem with setup cost. *Computers & Operations Research*, Vol. 40, 2013, pp. 225-235.
155. Mobley K.R. *An Introduction to Predictive Maintenance*. Elsevier Science, 2010.
156. Morton, T.E., D.W. Pentico. *Heuristic Scheduling Systems*, Wiley, New York, 1993.
157. Mosetti G, Poloni C, Diviacco B. Optimization of wind turbine positioning in large wind farms by means of a genetic algorithm. *J. of Wind Engineering and Industrial Aerodynamics*, Vol. 51(1), 1994, pp. 105-16.
158. Mukhacheva E. A. and A. S. Mukhacheva. L. V. Kantorovich and cutting-packing problems: New approaches to combinatorial problems of linear cutting and rectangular packing. *Journal of Mathematical Sciences*, Vol. 133(4), 2006, pp. 1504-1512.
159. Muller, A., A.C. Marquez and B. Iung. On the concept of e-maintenance: review and current research. *Reliability Engineering & System Safety*, Vol. 93, 2008, pp. 1165-1187.
160. Mustakerov I., D. Borissova. Technical systems design by combinatorial optimization choice of elements on the example of night vision devices design. *Comptes rendus de l'Academie bulgare des Sciences*, Tom 60(4), 2007, pp. 373-380.
161. Mustakerov I., D. Borissova. Software system for night vision devices design by reasonable combinatorial choice. *In Proc. of Int. Conf. on Theory and Applications in Mathematics and Informatics, ICTAMI 2007*, Alba Iulia, Romania (Eds. D. Breaz, N. Breaz, D. Wainberg), 30 August - 02 Sept. 2007, Alba Iulia, Romania, pp. 43-53.
162. Mustakerov I., D. Borissova. Optimal manufacturing scheduling for dependent details processing. *International Journal of Computer and Information Engineering*. Vol. 2(8) 2008, pp. 493-497.
163. Mustakerov I, Borissova D. Wind turbines type and number choice using combinatorial optimization. *Renew Energy*, Vol. 35(9), 2010, pp. 1887-1894.
164. Mustakerov I, Borissova D. Wind park layout design using combinatorial optimization. In: *Wind Turbines*, Ed. I. Al-Bahadly, InTech; 2011, pp. 403-24.
165. Mustakerov I., D. Borissova. Modular systems design via multi-objective optimization. *Advanced Modeling and Optimization*, Vol. 15(2), 2013, pp. 421-430.
166. Mustakerov I., D. Borissova. A discrete choice modeling approach to modular systems design. *World Academy of Science, Engineering and Technology*, Issue 76, 2013, pp. 133-139.
167. Mustakerov I., D. Borissova. Data structures and algorithms of intelligent Web-based system for modular design. *International Journal of Computer Science and Engineering*, Vol. 7(7), 2013, 2013, pp. 87-92.
168. Mustakerov I., D. Borissova. An intelligent approach for optimum maintenance strategy defining. *In Proc. of Innovations in Intelligent Systems and Applications (INISTA)*, 2013 IEEE International Symposium on. 19-21 June 2013, DOI: 10.1109/INISTA.2013.6577666.
169. Mustakerov I., D. Borissova. A Web Application for Group Decision-Making based on Combinatorial Optimization. *In Proc. of Int. Conf. on Information Systems and Technologies (ICIST 2014)*, March 22-24, 2014, Valencia, Spain, pp. 46-56
170. Mustakerov I., D. Borissova. One-dimensional cutting stock model for joinery manufacturing. *In Proc. Advanced Information Science and Applications – Vol. I, 18th Int. Conf. on Circuits, Systems, Communications and Computers (CSCC 2014)*, July 17-21, 2014, Santorini Island, Greece, pp. 51-55.
171. Mustakerov I., D. Borissova. Multi-criteria model for optimal number and placement of sensors for structural health monitoring: Lexicographic method implementation. *Advanced Modeling and Optimization*, Vol. 16(1), 2014, pp. 103-112.

172. Naderi, B., A. Azab. Modeling and heuristics for scheduling of distributed job shops. *Expert Systems with Applications*, Vol. 41, 2014, pp. 7754-7763.
173. Naderi, B., M. Zandieh. Modeling and scheduling no-wait open shop problems. *Int. Journal of Production Economics* Vol. 158, 2014, pp. 256-266.
174. Nelson, C. A. A scoring model for flexible manufacturing system project selection. *EJOR*, Vol. 24, 1986, pp. 346-359.
175. Nielsen, U.D., Z. Lajic, J.J. Jensen. Towards fault-tolerant decision support systems for ship operator guidance. *Reliability Engineering & System Safety*, Vol. 104, 2012, pp. 1-14.
176. Ohkawara, N. Long-term variations of atmospheric transmittance from pyrliometer measurements. In *Proc. 12th BSRN Science and Review and Workshop*, Potsdam, Germany, 2012.
177. Osyczka, A. An approach to multicriterion optimization problems for engineering design. *Computer Methods in Applied Mechanics and Engineering*. Vol. 15(3), 1978, pp. 309-333.
178. Pareto, V. 1906. *Manuale di Economica Politica*, Societa Editrice Libreria. Milan; translated into English by A.S. Schwier as *Manual of Political Economy*, Eds. A.S. Schwier & A.N., New York, page, 1971.
179. Pawitan, Yudi. In *All Likelihood: Statistical Modelling and Inference Using Likelihood*, Oxford University, 2001.
180. Peneva, V., I. Popchev. Models for fuzzy multicriteria decision making based on fuzzy relations. *Comptes rendus de l'Acad'emie bulgare des Sciences*, Vol. 62(5), 2009, pp. 551-558.
181. Peneva, V., I. Popchev. Fuzzy multi-criteria decision making algorithms. *Comptes rendus de l'Acad'emie bulgare des Sciences*, Vol. 63(7), 2010, pp. 979-992.
182. Peneva, V., I. Popchev. Models for decision making by fuzzy relations and fuzzy numbers for criteria evaluations. *Comptes rendus de l'Acad'emie bulgare des Sciences*, Vol. 62(10), 2009, pp. 1217-1222.
183. Peng, Y., M. Dong and M.J. Zuo. Current status of machine prognostics in condition-based maintenance: a review. *Int. Journal of Advanced Manufacturing Technology*, Vol. 50, 2010, pp. 297-313.
184. Pinedo, M. Stochastic scheduling with release dates and due dates. *Operations Research*, Vol. 31(3), 1983, pp. 559-572.
185. Power, D. *Decision Support Systems: Frequently Asked Questions*. iUniverse, 2005.
186. Proos, K.A.; G.P. Steven, O.M. Querin, Y.M. Xie. Multicriterion evolutionary structural optimization using the weighted and the global criterion methods. *AIAA Journal*. Vol. 39(10), 2001, pp. 2006-2012.
187. Rao, S.S., T.I. Freiheit. A modified game theory approach to multiobjective optimization. *Journal of Mechanical Design*, Vol. 113, 1991, pp. 286-291.
188. Rigopoulos, G., D. TH. Askounis and K. Metaxiotis. NeXCLass: A decision support system for non-ordered multicriteria classification. *Int. Journal of Information Technology & Decision Making*, Vol. 9(53), 2010, pp. 53-79.
189. Russell, L., J. Lombardo. Target acquisition: It's not just for military imaging. *Photonics Spectra*, July, 1998, pp. 123-126.
190. Ryer A. *The Light Measurement Handbook*, International Light, Newburyport, MA, 1997, pp. 64.
191. Riegler, J.T., J.D. Whiteley, H.L. Task, J. Schueren. The effects of signal-to-noise ratio on visual acuity through night vision goggles, *AL-TR-1991-001*, 1991.
192. Saaty T. L. Making and validating complex decisions with the AHP/ANP. *Journal of Systems Science and Systems Engineering*, Vol. 14, pp. 1-36, 2005.
193. Salukvadze, M.E. *Vector-Valued Optimization Problems in Control Theory*. New York: Academic Press, 1979.
194. Sawaragi, Y., H. Nakayama, T. Tanino. *Theory of multiobjective optimization*. Academic Press, Orlando, 1985.
195. Sgurev V., M. Hadjiski. Intelligent control integration in thermal power plant rehabilitation, *Proc. IFAC Workshop on Automatic Systems for Building the Infrastructure in Developing Countries*, DICOM-TT'03, (T. Dinibutun, ed.), 2003, Istanbul, Turkey.
196. Sgurev, V., S. Kojnov. Modern Trends in the Intelligent Supervisory Control for Power Plants. In *Proc. Control of Thermal Power Plants and Systems*, 14-15 November 2008, Stara Zagora, pp. 13-19.
197. Silva, E., J. F. Oliveira, G. Wäscher. 2DCPackGen: A problem generator for two-dimensional rectangular cutting and packing problems. *EJOR*, Vol. 237, 2014, pp. 846-856.
198. Silver, E.A. An overview of heuristic solution methods. *Journal of the Operational Research Society*, Vol. 55, 2004, pp. 936-956.
199. Soininen T., I. Niemela, J. Tiihonen, R. Sulonen. Unified configuration knowledge representation using weight constraint rules. In *Proc. ECAI-2000 Workshop on Configuration*, Berlin, 2000, pp. 79-84.
200. Sotskov, Y. N. and N. V. Shakhlevich. NP-hardness of shop-scheduling problems with three jobs. *Discrete Applied Mathematics*, Vol. 59, 1995, pp. 237-266.
201. Sospedra, F., V. Caselles, E. Valor, C. Di Bella, C. Coll, E. Rubio. Night-time cloud cover estimation. *Int. Journal of Remote Sensing*, Vol. 25, No 11, 2004, pp. 2193-2205.
202. Staszewski, W. J., K. Worden. An overview of optimal sensor location methods for damage detection. *Smart Structures and Materials, Proc.* 4326, 2001, pp. 179-187.

203. Steckel, R. H., J. M. Prince. Tallest in the world: Native Americans of the great plains in the nineteenth century. *American Economic Review*, Vol. 91(1), 2001, pp. 287-294.
204. Steuer R., *Multiple criteria optimization: theory, computation and application*. New York, John Wiley & Sons, Inc., 1986.
205. Stoilov, T, K. Stoilova. Functional Analysis of Enterprise Resource Planning Systems. *International Conference on Computer Systems and Technologies - CompSysTech'08*, 2008, pp. IIIB.8-1 - IIIB.8-6.
206. Stoilova, K, T. Stoilov. *Hierarchical Optimization for Fast Resource Allocation. Time Management*, (Ed. T. Stoilov), ISBN 978-953-51-0335-6, 114 pages, InTech, 2012, pp. 31-46.
207. Suliman, S. M. A. Pattern generating procedure for the cutting stock problem. *Int. Journal of Production Economics*, Vol. 74, 2001, pp. 293-301.
208. Taha, H.A. *Operations Research: An Introduction*. Prentice Hall, 2010.
209. Tam, V., K. T. Ma. Using heuristic-based optimizers to handle the personal computer configuration problems. *In Proc. 12th IEEE Int. Conf. on Tools with Artificial Intelligence*, 2000, pp. 108-111.
210. Tam, V., K. T. Ma. Optimizing personal computer configurations with heuristic-based search methods. *Artificial Intelligence Review*, Vol. 17(2), 2002, pp. 129-140.
211. Task, H.L. Night vision devices and characteristics, *ASC 91-2961*, 1992.
212. Tamiz M., D. Jones, C. Romero. Goal programming for decision making: An overview of the current state-of-the-art. *EJOR*, Vol. 111, 1998, pp. 569-581.
213. Tavakkoli-Moghaddam, R., F. Jolai, F. Vaziri, P. K. Ahmed, A. Azaron. A hybrid method for solving stochastic job shop scheduling problems. *Applied Mathematics and Computation*, Vol. 170, 2005, pp.185-206.
214. Tejada, A. A mode-shape-based fault detection methodology for cantilever beams. *NASA/CR-2009-215721*. 2009 http://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20090021633_2009020629.pdf.
215. Tseng, F. T., E. F. Stafford Jr., J. N. D Gupta. An empirical analysis of integer programming formulations for the permutation flowshop. *Omega*, Vol. 32(4), 2004, pp. 285-293
216. Tseng, T.L.B, C.C. Huang. Design support systems: A case study of modular design of the set-top box from design knowledge externalization perspective. *Decision Support Systems*, Vol. 44(4), 2008, pp. 909-924.
217. Vassilev, V., K. Genova, M. Vassileva. A brief survey of multicriteria decision making methods and software systems. *Cybernetics and Information Technologies*, Vol. 5(2), 2005, pp. 14-20.
218. Vassilev, V., K. Genova, F. Andonov, B. Staykov. Multicriteria Decision Support System MultiChoice *Cybernetics and Information Technologies*, Vol. 4(1), 2004, pp. 65-75.
219. Wang,X., E. Triantaphyllou. Ranking irregularities when evaluating alternatives by using some ELECTRE methods. *Omega*, Vol. 36, 2008, pp. 45-63.
220. Wang, Y.W., Q. Ma, W. Li. Structural damage detection by multi-objective intelligent algorithm. *In Proc. of 15th World Conference on Earthquake Engineering*, Lisbon, Portugal, 2012.
221. Weber, R.R., P. Varaiya, J. Walrand. Scheduling jobs with stochastically ordered processing times on parallel machines to minimize expected flowtime. *J. of Applied Probability*, Vol. 23(3), 1986, pp. 841-847.
222. Wolsey, L. *Integer Programming*. Wiley-Interscience. 1998.
223. Worden, K., A. P. Burrows, Optimal sensor placement for fault detection. *Engineering Structures*, Vol. 23, 2001, pp. 885-901.
224. Wu,W.W., Y.T. Lee. Selecting knowledge management strategies by using the analytic network process. *Expert Systems with Applications*, Vol. 32, 2007, pp. 841-847.
225. Yang, S-I., D. M. Frangopol, L.C. Neves. Service life prediction of structural systems using lifetime functions with emphasis on bridges. *Reliability Engineering & Systems Safety*, Vol. 86(1), 2004, pp. 39-51.
226. Yu, S., Q.Yang, J. Tao, X. Xu. Incorporating quality function deployment with modularity for the end-of-life of a product family. *J. of Cleaner Production*, Vol. 87, 2015, pp. 423-430.
227. Yu, P.L. *Multiple criteria decision making: concepts, techniques and extensions*. Plenum Press, New York 1985.
228. Zhang, M., X. Zhao, Y. Qi. The effects of organizational flatness, coordination, and product modularity on mass customization capability. *Int. J. Production Economics*, Vol. 158, 2014, pp. 145-155.
229. Вирт Н. *Алгоритмы + структуры данных = программы*, Москва, „МИР” 1986
230. Борисова, Д., *Основы на веб програмирането*. Изд. "За Буквите - О писменахъ", ISBN: 978-619-185-052-5, София 2014, 255 стр.
231. Волков В. Г., Приборы ночного видения новых поколений, *Специальная техника*, № 5, 2001
232. Гарванов, И. *Методи и алгоритми за откриване на цели*. Изд. „Авангард Прима”, 2014, 277 стр.
233. Дубров, А. М., Б. А. Лагоша Е. Ю. Хрусталеv, Моделирование рисковvх ситуаций в экономике и бизнесе, 2000, <http://bibliotekar.ru/riskovye-situacii-2/index.htm>.
234. Кошавцев Н. Ф., А. Н. Кошавцев, С. Ф. Федотова. Анализ перспектив развития приборов ночного видения. *Прикладная физика*, No 3, 1999, УДК 621.384.326.23.
235. Кнут Д. *Искусство программирования для ЭВМ*, т. 1, Основные алгоритмы, Москва, „МИР”, 1976
236. Леготин, Ф. Я. *Экономико-кибернетическая природа затрат*. Изд-во Урал. гос. экон. у-т, 2008.