



**БЪЛГАРСКА АКАДЕМИЯ НА НАУКИТЕ
ИНСТИТУТ ПО ИНФОРМАЦИОННИ И КОМУНИКАЦИОННИ
ТЕХНОЛОГИИ**

Борис Атанасов Стайков

**МЕТОДИ, АЛГОРИТМИ И СОФТУЕРНИ СИСТЕМИ ЗА
ПОДПОМАГАНЕ ВЗЕМАНЕТО НА РЕШЕНИЯ**

ДИСЕРТАЦИЯ

за присъждане на образователна и научна степен „доктор“

по докторска програма „Информатика“

професионално направление „4.6. Информатика и компютърни науки“

Научен ръководител:

проф. д-р Васил Василев

Научен консултант:

доц. д-р Татяна Атанасова

София, 2020 г.

СЪДЪРЖАНИЕ

Списък на използвани съкращения и означения.....	5
Увод.....	6
1. Актуалност на темата.....	6
2. Задачи за многокритериален анализ.....	10
3. Задачи за многокритериална оптимизация.....	19
Цели и задачи на дисертацията.....	52
Глава I. Класификационно-ориентирани скаларизиращи задачи и алгоритми	53
1.1. Класификационно-ориентирани скаларизиращи задачи DAL	53
1.2. Класификационно-ориентирани скаларизиращи задачи DALDI.....	57
1.3. Обобщени скаларизиращи задачи.....	59
1.4. Обобщен интерактивен алгоритъм, базиран на метода gens-im за решаване на линейни и линейни целочислени задачи на многокритериалната оптимизация	73
Глава II. Програмна система МКО-2.1	79
2.1. Предназначение на системата МКО-2.1.....	79
2.2. Синтаксис за дефиниране на задачи за многокритериална оптимизация.	81
2.3. Основни модули в системата МКО-2.1.	84
2.4. Работа със системата МКО-2.1.	87

2.4.1.	Въвеждане на задача.....	88
2.4.2.	Решаване на задачата за многокритериална оптимизация	93
2.4.3.	Настройки.....	97
Глава III. Програмна система WebOptim.....		101
3.1.	Предназначение на системата WebOptim.....	101
3.2.	Структура на системата WebOptim.....	101
3.3.	Основни компоненти в WebOptim	103
3.4.	Работа със системата WebOptim	107
3.4.1.	Решаване на задачи за еднокритериална оптимизация.....	110
3.4.2.	Решаване на задачи за многокритериална оптимизация	113
3.5.	Интерфейс за връзка с външни системи.....	125
Глава IV. Експериментални изследвания – решаване на задача за оперативно планиране чрез многокритериална оптимизация		129
4.1.	Описание на задачата	129
4.2.	Построяване на математически модел на задачата	130
4.3.	Решаване на задачата	132
4.4.	Сравнителен анализ на резултатите.....	141
Заклучение		146

Списък на публикациите по дисертацията	148
Списък на забелязани цитирания на публикациите по дисертацията	150
Резюме на постигнатите резултати	152
Декларация за оригиналност на резултатите	153
Благодарности	154
Библиография	155

СПИСЪК НА ИЗПОЛЗВАНИ СЪКРАЩЕНИЯ И ОЗНАЧЕНИЯ

ЕО	Еднокритериална оптимизация
МО	Многокритериална оптимизация
МА	Многокритериален анализ
ЛВР	Лице вземащо решение
СПВР	Системи за подпомагане вземането на решения
МСПВР	Многокритериални системи за подпомагане вземането на решения
WWW	World Wide Web
OLAP	Online Analytical Processing
LAMP	Linux Apache MySql PHP

УВОД

1. Актуалност на темата

Управлението на бизнеса в съвременната икономика представлява процес, при който предварително определени цели се постигат чрез използване на различни ресурси – финанси, материали, хора, енергия и др. Ефективното управление и използване на тези ресурси зависи пряко от управленските решения, които се взимат от съответните ръководни лица. Дълги години умението за вземане на правилни управленски решения е бил разглеждан по-скоро като талант, придобиван с годините и опита, главно чрез метода на проби и грешки. Разчитало се е главно на персонални качества като талантливост, прозорливост и нестандартно мислене, отколкото на използването на математически количествени методи. С течение на времето и развитието на технологиите, реализирането на успешен бизнес става все по-трудно поради нарастването на броя на алтернативите и все по-сложното планиране на ресурсите и логистиката. Резултатите и цената от грешно взетите решения в много случаи се оказват пагубни [Singiresu, Rao , (2009)].

Поради тези причини в процесът на вземане на решения при съвременния бизнес, наред с персоналните качества на управленските кадри, се налага да бъде включено използването на техники, подходи и софтуерни системи в чиято основа стоят количествени анализи и математически прийоми. Тези анализи стават предмет на изследване от научната дисциплина „Изследване на операциите“.

Различните задачи за планиране, контрол и анализ в производството [Kirilov, Gulishaki и др. (2016)], транспорта, екологията [Jaszkiewicz, Slowinski, (1997)], образованието и други области могат да бъдат определени като многокритериални задачи за вземане на решения [White, (1990)]. В зависимост от тяхната формална постановка, тези задачи могат да бъдат

разделени на два отделни класа [Vincke, (1992)]. При първия клас задачи краен брой алтернативи са представени явно в таблична форма. Те се наричат многокритериални задачи за вземане на решение с дискретни алтернативи или задачи на многокритериалния анализ (МА) [Dyer, (2004)]. При втория клас задачи, краен брой от явно зададени ограничения във вид на функции определят безкраен брой допустими алтернативи [Sawaragi, Nakayama и др., (1985)]. Те се наричат задачи на многокритериалната оптимизация (МО). Именно този клас задачи е предмет на по-задълбочено разглеждане в настоящата дисертация.

В задачите за многокритериална оптимизация и многокритериален анализ няколко на брой критерии се оптимизират едновременно в дадено множеството от допустими алтернативи. В общия случай не съществува алтернатива, която е оптимална за всички критерии, но съществува множество от алтернативи, които притежават следното общо свойство: всяко подобрение в стойността на някой от критериите води до влошаване в стойността на поне един от останалите критерии. Това свойство е открито за първи път през 1896 г. от италианския математик-икономист Vilfredo Pareto и поради това, през 1951 г. множеството е наречено на негово име - множество на недоминираните (Парето-оптимални) алтернативи (решения) [Collette, Siarry, (2013)]. От чисто математическа гледна точка всяка алтернатива от множеството на Парето може да бъде решение на многокритериалната задача. В крайна сметка за да бъде избрана една алтернатива е необходима допълнителна информация, която се определя от т.нар. “лице, вземащо решение (ЛВР)”. Информацията, предоставена от ЛВР е отражение на неговите лични предпочитания с оглед на качествата на търсената най-предпочитана алтернатива.

Математически, решаването на такъв вид задачи е дълъг интерактивен процес, свързан със сложни и големи по обем изчисления. Това автоматично налага в практиката да бъдат използвани помощни софтуерни системи, разработвани специално за асистиране при решаването на такъв вид задачи [Sierksma, (2001)].

Системите, за подпомагане вземането на решения (СПВР) представляват интерактивни компютърно-базирани системи, предназначени за подпомагане и асистиране на лицето, вземащо решения при решаване на неформализирани или слабо формализирани задачи [Miettinen, (1994)]. Разработените многокритериални системи за подпомагане вземането на решения (МСПВР) могат да се разделят на три групи: комерсиални, изследователски или учебни и експериментални [Clemen (1996)]. Понякога е трудно да се направи ясно разделение между тези групи. Някои експериментални МСПВР могат успешно да се използват за изследователски и учебни цели. В [Wiestroffer, Narula, (1997)] е представено състоянието до 1996 година на многокритериалните системи за подпомагане вземането на решения. Описани са повече от 40 системи за решаване както на задачи с непрекъснати алтернативи, така и на задачи с дискретни алтернативи. В края на 90-те г. и началото на 21-ви век с главоломното развитие на технологиите и Интернет комуникациите, WWW технологията е действала насърчаващо за започване на по-широко използване на системите за подпомагане на вземането на решения [Berners-Lee, Cailliau и др., (1994)]. Тази технология е променила изцяло дизайна, разработването, прилагането и внедряването на системите за подпомагане вземането на решения [Bhargava, Power, (2001)]. Започва се активно разработване на нови експериментални системи [Vassilev, Djambov и др. (1995)], [Sun, Steuer, (1996)] [Vassilev, (1997)], [Tan, Lee и др. (2000)], [Fielding, (2000)], [Kikerl, [Marinchev, Kirilov, (2003)], [Bridges, (2005)] и адаптиране на версии на стари системи за работа в Интернет среда [Valente, (2007)]. Възникват нови, изцяло уеб-базирани СПВР [Miettinen, 2014], [Tsangaratos P, Kallioras, (2017)]. В наше време, когато Интернет неоспоримо играе ключова роля в развитието и управлението на бизнеса, разработката на този вид софтуерни системи се насочва основно към това, те да бъдат изцяло уеб-базирани, съответно по-леснодостъпни от все по-широк кръг потребители с различни умения и квалификация [Bhargava, Power и др., 2007].

Основното предизвикателство е, че СПВР исторически са предназначени за промишлени и организационни потребители, които обикновено са с професионален интерес към използването на тези системи. Уеб подпомага развитието на СПВР за по-широка гама от потребители, включително и непрофесионалисти в областта, които по някакъв начин трябва да бъдат информирани, че могат да се възползват от тези системи и те ще работят еднакво добре както за професионалистите така и за тях! Една от представените в дисертационния труд разработена уеб-базирана СПВР - WebOptim, има за цел да предложи решение именно на тези предизвикателства. Най-удобния начин да се предостави подобна платформа, която да бъде използвана главно от непрофесионалисти в областта, без да ги ангажира с локално инсталиране и изучаване на специфичен софтуер или с изисквания откъм изчислителна мощ е системата да се базира на вече широко използваната в ежедневието WWW технология [Kirilov, Guliashki и др. (2015)].

2. Задачи за многокритериален анализ

2.1. Формална постановка на задачите за многокритериален анализ

При задачите на многокритериалния анализ явно се задават краен брой алтернативи в таблична форма. Многокритериалният анализ се нарича още многоатрибутен анализ или многокритериално вземане на решение при зададени дискретни алтернативи.

Формално задачата може да бъде описана чрез матрицата на алтернативите A ($n \times k$), която може да бъде зададена по два начина:

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}

Или:

f_k a_i	$f_1(.)$	$f_2(.)$	...	$f_j(.)$	...	$f_k(.)$
a_1	$f_1(a_1)$	$f_2(a_1)$	...	$f_j(a_1)$	...	$f_k(a_1)$
a_2	$f_1(a_2)$	$f_2(a_2)$	...	$f_j(a_2)$	...	$f_k(a_2)$
...	...	...	...	...	...	...

a_i	$f_1(a_i)$	$f_2(a_i)$	$\dots$	$f_k(a_i)$	$\dots$	$f_k(a_i)$
$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$
a_n	$f_1(a_n)$	$f_2(a_n)$	$\dots$	$f_j(a_n)$	$\dots$	$f_k(a_n)$

Където:

a_i - означава алтернатива с индекс $i, i=1, \dots, n$;

$k_j(.)$ или $f_j(.)$ - означава критерий с индекс $j, j=1, \dots, k$.

I – представлява множеството от индексите на алтернативите

J - представлява индексно множество на критериите

Оценката на i -тата алтернатива по отношение на всички критерии се задава с вектора-ред $(a_{i1}, a_{i2}, \dots, a_{ik})$ или $(f_1(a_i), \dots, f_k(a_i))$.

Оценката на всички алтернативи по отношение на j -тия критерий се задава с вектора-стълб $(a_{1j}, a_{2j}, \dots, a_{nj})^T$ или $(f_j(a_1), \dots, f_j(a_n))^T$.

На базата на матрицата на алтернативите A могат да бъдат формирани три типа задачи:

Задача 1. Избор на най-добра алтернатива от ЛВР в съответствие с неговите предпочитания (задача за многокритериален избор).

Задача 2. Подреждане на всички алтернативи в низходящ или възходящ ред (задача за многокритериална наредба, рейтингиране, ранкиране и др.).

Задача 3. Разделяне на алтернативите на групи (задача за многокритериална класификация или многокритериално сортиране).

Критерий е мярка за оценка на ефективността и се проявява като атрибут или целева функция.

Атрибут е свойство на задачата, което може да бъде измерено или оценено. Параметри, компоненти, характеристики са синоними на атрибутите. Всяка алтернатива може да бъде описана с набор от атрибути. Атрибутите могат да бъдат количествени и качествени.

Целева функция е това, което трябва да бъде постигнато в колкото се може по-голяма степен. Целевата функция показва посоката на желаните промени.

Критерият е най-общ термин. Той се проявява като целева функция, когато е възможна пълна формализация и като атрибут, когато такава формализация не е възможна.

Допустимата алтернатива с индекс i' се нарича **недоминирана алтернатива**, ако не съществува друга алтернатива с индекс i , за която да е изпълнено условието:

$$a_{ij} \leq a_{i'j}, j = 1, \dots, k$$

и поне за един индекс $j=s$ да бъде изпълнено условието:

$$a_{is} < a_{i's}.$$

Допустимата алтернатива с индекс i се нарича **удовлетворителна алтернатива**, ако е изпълнено условието:

$$a_{ij} \leq \overline{a_j}, j = 1 \dots k,$$

където $\overline{a_j}$ е аспирационното ниво на критерия с индекс $j, j = 1, \dots, k$.

Алтернативата с индекс i^* се нарича **идеална алтернатива**, ако е изпълнено условието:

$$i^* = (a_1^*, a_2^*, \dots, a_n^*), \text{ където } a_j^* = \min_{1 \leq i \leq n} a_{ij}.$$

В общия случай идеалната алтернатива не съществува.

Недоминираната алтернатива се нарича **най-предпочитана алтернатива**, ако тя най-точно отразява предпочитанията на ЛВР.

2.2. Методи за решаване на задачи за многокритериален анализ

За решаване на задачите за многокритериалния анализ се използват два основни модела за използване на информацията от задачата: **компенсаторен** и **некомпенсаторен** [Clemen, (1996)].

При *некомпенсаторния модел* не се разрешава да се правят компромиси между критериите. Лошата стойност по един критерий в дадена алтернатива не може да бъде преодоляна за сметка на стойностите на тази алтернатива по другите критерии.

Компенсаторният модел от своя страна разрешава да се правят компромиси между критериите. Малки промени по един критерий могат да бъдат извършени за сметка на промени по други критерии. При компенсаторните модели на всяка многомерна характеристика на алтернативите, може да съответства едно число (числова характеристика). На основата на подхода за определяне на тази числова характеристика, компенсаторните модели се делят на три групи:

резултативен модел – определя алтернативата, която има най-голям резултат (най-голяма стойност на функцията за полезност). Проблемът при тези модели е как да се построи тази функция на полезност.

компромисен модел – определя алтернативата, която е най-близо до идеалната алтернатива. При този модел алтернативите се разглеждат като точки в критериалното пространство.

модел на съгласието – определя множество от отношения, които най-добре удовлетворяват съответна мярка за съгласие.

При моделиране и решаване на задачите на многокритериалния анализ възникват следните основни проблеми:

- как да изследваме даден количествен атрибут
- как да сравняваме количествени и качествени атрибути
- как да сравняваме повече от два количествени атрибута

Първият проблем е свързан със скалите на измерване. Съществуват три скали за измерване на количествени атрибути: *ординарна, интервална и пропорционална*.

При *ординарната скала* измерваните величини се поставят на определено място в една възходяща или низходяща наредба. При тази наредба не се говори за разстоянието между величините, а само за мястото по важност, което заемат.

При *интервалната скала* отделните стойности са на еднакво разстояние една от друга. Величината на тази стойност се измерва до произволно избрано начало.

При *пропорционалната скала* отделните стойности са също на еднакво разстояние една от друга, като величината на стойността показва разстоянието до естественото начало. За тази скала са валидни всички аритметични операции.

Вторият проблем се решава като качествените атрибути се превръщат в количествени чрез използване на интервална скала. Един от най-известните начини за такава трансформация е използването на т.нар. **биполярна скала**. Взема се, например, десет точкова скала. Десет точки се дават на максималната стойност на атрибута. Средният брой точки (пет) се дават за отделяне на добрите от лошите стойности на атрибута.

За решаване на третия проблем се извършва трансформация на количествените атрибути, така че те да станат безразмерни и стойностите им да се изменят в един и същ интервал.

Две от известните трансформации са:

Чрез използване на L2 - нормата за елементите a_{ij}

При тази трансформация елементът a_{ij} на матрицата на алтернативите A се преобразува в елемента r_{ij} , както следва:

$$r_{ij} = \frac{a_{ij}}{\sqrt{\sum_{l=1}^n a_{lj}^2}} \quad j = 1, \dots, n; \quad j = 1, \dots, k.$$

Предимството на тази трансформация е, че критериите стават безразмерни. А основният недостатък е, че отделните атрибути нямат еднакъв диапазон на изменение.

Чрез линейна трансформация

При тази трансформация елементът a_{ij} от вектор-стълба $(a_{1j}, a_{2j}, \dots, a_{nj})^T$ се разделя на максималния елемент в този стълб и така се получава новият елемент r_{ij} :

$$r_{ij} = \frac{a_{ij}}{\max_i a_{ij}}, \quad i = 1, \dots, n; \quad j = 1, \dots, k \quad - \text{за атрибути, които се максимизират}$$

и

$$r_{ij} = \frac{\frac{1}{a_{ij}}}{\max_i \left(\frac{1}{a_{ij}} \right)}, i = 1, \dots, n; j = 1, \dots, k - \text{за атрибути, които се минимизират.}$$

Предимството на тази трансформация е, че атрибутите стават безразмерни и стойностите за всеки критерий варират между 0 и 1.

2.3. Алгоритми за решаване на задачи за многокритериален анализ

Разработените алгоритми за решаването на задачи на многокритериалния анализ могат да бъдат групирани в три основни класа [Fishburn, (1970)]. Първият клас включва алгоритмите на теорията за многоатрибутната полезност (алгоритъма на компромиса [Farquhar, (1984)], [Keeney, Raiffa, (1993)]), алгоритъма UTA [Beuthe, Scannella, (2001)], алгоритъма MACBETH [Bana, 1999], директния тегловен алгоритъм [Vansnick, (1986)], [Koksalan, Karwan и др. (1984)] и тегловните алгоритми АНР [Saaty, 1980], [Saaty, 1994]. По начин на агрегиране на глобалните предпочитания на ЛВР, този клас алгоритми се дели на два подкласа [Jaszkiewicz, Slowinski (1997)], [Jaszkiewicz, Slowinski (1999)]. При първия подклас директно се синтезира обобщен функционален критерий, докато при втория - тегловните алгоритми, може да се каже, че такъв критерий се синтезира индиректно в адитивна форма [Vassilev, Kirilov (1995)], [Roy, 1996], [Kirilov, Vassilev (1996)]. Двата подкласа алгоритми са базирани на предположението, че не съществува ограничение във възможностите на ЛВР за сравнение между алтернативите. За изразяване на предпочитанията на ЛВР при сравняването на две алтернативи, е достатъчно използването само на бинарна релация на силно предпочитание Р (антирефлексивна, асиметрична и транзитивна) и бинарна релация на неразличимост І (рефлексивна, симетрична и транзитивна). Тук спадат алгоритмите на отправното направление

[Korhonen, (1988)], на аспирационните нива [Lotfi, Stewart и др. (1992)], партишън метода [Narula, Vassilev и др., (2003)] и др.

През 2009 година българска следа оставят изследванията на методите и алгоритмите за т. нар. групово вземане на решение. В тези случаи в задачите за многокритериален анализ се появява още един фактор – човешки, когато решението трябва да се вземе групово от повече от един човек [Andonov 2009].

2.4. Софтуерни системи, подпомагащи решаването на задачи на многокритериалния анализ

Софтуерните системи, подпомагащи решаването на задачи на многокритериалния анализ, могат да бъдат разделени на два класа – софтуерни системи с общо предназначение и проблемно-ориентирани софтуерни системи. Софтуерните системи с общо предназначение подпомагат решаването на различни задачи на многокритериалния анализ от различни ЛВР. Обикновено в тях са реализирани един или няколко алгоритъма от една и съща група за решаване на задачите. Това се дължи на следните причини:

- В алгоритмите от различните групи се използват различни по тип процедури за извличане на информация от ЛВР, което води до значителни трудности при реализирането на подходящ потребителски интерфейс в софтуерните системи.
- Разработчиците на софтуерните системи обикновено имат интерес към реализирането на собствения си алгоритъм (алгоритми) или имат определени предпочитания към алгоритми от една и съща група.

Някои от по-известните софтуерни системи с общо предназначение са Expert Choice [Saaty, 1980], PROMCALC и GAIA [Brans, 1994], ELECTRE III-IV [Roy, 1990], MACBETH

[Bana, 1990], VIP [Dias, 2000], Web-HIPRE [Mustajoki, 2000], IRIS [Dias, 2003], IDS [Ху, 2003], KnowCube [Hanne, 2003], MultiChoice [Vassilev, 2004]. Те реализират един или няколко алгоритма от една и съща група. В началото на 90-те години на миналия век се появяват и първите български изследвания в областта [Vassilev, Genova и др. (1991)], [Narula, Kirilov и др. (1992)], [Vassilev, Narula (1993)], [Kirilov (1993)], [Vassilev, Kirilov (1995)], [Vassilev, Genova и др. (1996)], [Kirilov, Vassilev (1997)], [Vassilev, Kirilov и др. (1997)], както и първата софтуерна имплементация на разработените алгоритми [Vassilev, Atanasov и др. (1993)]. Една от значимите за тези години българска софтуерна система е MOIP [Vassilev, Narula и др. (1997)].

От групата на проблемно-ориентирани софтуерни системи основни представители са FINCLAS – за решаване на финансови задачи [Zopounidis, 1998], Agland Decision system – за аграрни дейности [Parsons, 2002], системата DESYRE - за възстановяване на замърсени местности [Carlon, 2007], системата MultCSync – за планиране на опазването на околната среда [Moffett, 2005]. От наскоро разработените по-значителна е системата за избор на локация за построяване на изкуствени водоеми [Tsangaratos, 2017]. От значимите български разработки в областта, могат да бъдат споменати както изследвания и разработки съвместно със световно-известни учени [Narula, Vassilev и др. (2003)], [Kirilov, Vassilev и др. (2000)], така и тези, изцяло с участие само на български учени [Vassilev, Genova и др. (2004)], [Genova, Vassilev и др. (2004)]. От български софтуерни системи за решаване на задачи за многокритериален анализ, като най-значима трябва да бъде спомената системата Multichoice [Genova, Vassilev и др. (2003)].

В повечето случаи, този вид системи са включени в други информационно-управляващи системи и служат за подпомагане на решаването на един или няколко типа специфични задачи на многокритериалния анализ. В тях обикновено е реализиран проблемно-ориентиран потребителски интерфейс и са включени алгоритми от различни групи на многокритериалния анализ.

3. Задачи за многокритериална оптимизация

3.1. Формална постановка на задачите за многокритериална оптимизация

Основен етап при дефинирането и решаването на задачи за многокритериална оптимизация е построяването на математическия модел на съдържателната постановка на задачата. За успешното намиране на решението на модела от особено значение е подбирането на подходящ метод за решаване. За тази цел от основно значение е да се има широка и задълбочена представа за различните задачи, математически модели и методите за тяхното решаване, които предлага една от динамично развиващите се области в математиката - “Изследване на операциите”. Задачите за многокритериална оптимизация могат да бъдат както линейни, така и нелинейни. Съществуват подходи и методи за решаване и на двата вида задачи [Miettinen, Kirilov (2005)], но фокуса на настоящата дисертация е върху линейните задачи за многокритериална оптимизация.

В най-общия случай се използват два типа статични математически модели.

Първият модел има следният вид:

$$\max z = f(x)$$

(I) при ограниченията:

$$g_j(x) \leq b_j, j=1, \dots, m,$$

където:

– символът “max” означава, че трябва да се максимизират едновременно всички критерии (целеви функции);

– $f(x)$ е целевата функция (критерий);

– $x = (x_1, \dots, x_n)^T$ е вектор на променливите (на решенията);

– $g_j(x), j=1, \dots, m$, са функции на ограниченията на задачата;

Функциите $g_j(x), j=1, \dots, m$, и целевата функция $f(x)$ са реални функции: $g_j: \mathbb{R}^n \rightarrow \mathbb{R}, j=1, \dots, m$, и $f: \mathbb{R}^n \rightarrow \mathbb{R}$, където $\mathbb{R}$ е множеството на реалните числа, а $\mathbb{R}^n$ е n -мерното евклидово пространство.

Тези функции определят така нареченото множество на допустимите решения.

Означаваме това множество с $S, S \subset \mathbb{R}^n$.

Този модел може да се запише и така:

$$(I) \quad \left| \begin{array}{l} \max z = f(x) \\ x \in S \end{array} \right.,$$

където S е множеството, определено от функциите $\{g_j / j=1, \dots, m\}$.

В този математически модел $f(x)$ и $\{g_j(x) / j=1, K, m\}$ могат да имат различни икономически, физически, технически и др. интерпретации, като цена, печалба, загуба, разход, надеждност, маса, разстояние и др.

Другият тип математически модел, който често пъти е по-адекватен на съответния приложен проблем и който ще бъде обект на разглеждане, е следният:

$$“\max” \{z_1 = f_1(x), z_2 = f_2(x), \dots, z_k = f_k(x)\}$$

(II) при ограниченията:

$$g_j(x) \leq b_j, j=1, \dots, m,$$

където:

– “max” означава, че целевите функции се максимизират едновременно;

– $f_i(x)$, $i=1, \dots, k$, са целевите функции (критериите). Те са реални функции: $f_i: R^n \rightarrow R$, $i=1, \dots, k$. Особеното за тях е, че те обикновено са противоречиви и несъизмерими.

– $f(x) = (f_1(x), \dots, f_k(x))$ или $z = (z_1, \dots, z_k)$, е вектор на целевите функции;

– $k \geq 2$;

– $g_j(x)$, $j=1, \dots, m$, са функции на ограниченията на задачата и те са реални функции:

$g_j: R^n \rightarrow R$, $j=1, \dots, m$

Тези функции определят допустимото множество на решенията (променливите). Ще го означаваме със S и $S \subset R^n$.

– Означаваме с $f(S) = Z$ образа на множеството S в критериалното пространство R^k , $Z \subset R^k$. Z се нарича допустимо множество в критериалното пространство.

– Означаваме със $z = (z_1, \dots, z_k)^T$, където $z_i = f_i(x)$, $i=1, \dots, k$. z се нарича критериален вектор и $z \in Z$.

Този модел може да се запише и така:

$$(II) \quad \left| \begin{array}{l} \text{"max"} \{z_1 = f_1(x), \dots, z_k = f_k(x)\} \\ x \in S \end{array} \right.$$

Моделът (I) се нарича задача на еднокритериалната оптимизация (ЕО) или още обща задача на математическото оптимизиране.

Моделът (II) се нарича задача на векторната оптимизация, или още задача на многокритериалната оптимизация.

3.2. Методи за решаване на задачите за многокритериална оптимизация.

Съществуват два основни подхода при решаването на задачи на многокритериалната оптимизация: скаларизационен подход [Miettinen (1999)] и апроксимационен подход [Ehrgott и Wiecek (2004)]. Основните представители на скаларизационния подход са интерактивните алгоритми. В тези алгоритми задачите на многокритериалната оптимизация се разглеждат като задачи за вземане на решение, а ударението е поставено върху реалното участие на ЛВР в процеса на решаването на задачата.

Интерактивните алгоритми са най-усъвършенствани и широко разпространени. Те притежават следните основни характеристики:

- малка част от Парето-оптималните решения трябва да бъдат генерирани и оценени от ЛВР.
- в процеса на решаване на многокритериалната задача ЛВР може да се обучава в спецификата на задачата.
- ЛВР се чувства по-уверен в крайния резултат.

Обикновено всеки интерактивен алгоритъм се състои от две процедури – оптимизационна и оценяваща, които се повтарят циклично до достигане на условията за край на задачата. В оценяващата процедура ЛВР прави оценка на текущо намереното Парето-оптимално решение (решения) и определя предпочитанията си за търсенето на ново решение. На базата на тези предпочитания се формулира скаларизиращата задача, тя се решава в оптимизационната процедура и с нейна помощ се получава ново Парето-оптимално решение (решения), което се представя на ЛВР за оценка и избор. Основно свойство на всяка скаларизираща задача е, че всяко оптимално решение, което се получава е Парето-оптимално (слабо Парето-оптимално) решение на съответстващата задача на многокритериалната

оптимизация. Скаларизиращата задача е еднокритериална задача за оптимизация, което позволява използването на теорията и алгоритмите на еднокритериалната оптимизация. Всеки от интерактивните алгоритми за решаването на различни класове от задачи на многокритериалната оптимизация, разработени досега, има своите предимства и недостатъци, свързани най-вече с типа информация, предоставена от ЛВР и отразяваща глобалните и локалните му предпочитания, както и начина, по който тя се извлича от него; типа и начина на решаване на скаларизиращата задача. Интерактивните алгоритми са особено подходящи за решаването на линейни и изпъкнали нелинейни задачи на многокритериалната оптимизация, в които времето за решаване на скаларизиращата задача (времето за генериране на ново решение) не е от голямо значение. В NP- задачите (целочислени, дискретни, комбинаторни, неизпъкнали нелинейни задачи на многокритериалната оптимизация), това време може да нарасне значително и така да затрудни работата на ЛВР. При някои от тези задачи, например при многокритериалните линейни целочислени задачи [Narula, Vassilev, (1994)], [Vassilev, Genova (1998)], [Miettinen, Makela (2001)], са предложени различни стратегии за намаляване на времето за очакване на ново решение за оценка. Във фазата на обучение, на дадена итерация на ЛВР се предоставя не целочислено Парето-оптимално (слабо Парето-оптимално) решение, а апроксимирано Парето-оптимално (слабо Парето-оптимално) решение или Парето-оптимално решение на релаксираща (в случая на непрекъсната) линейна задача. Интерактивните алгоритми на отправната точка (направление) [Narula, Kirilov и др. (1992)], [Narula, Kirilov и др. (1994)], [Guliashki, Kirilov (2015)], [Karaivanova, Korhonen и др. (1995)], еволюционните (генетични) алгоритми [Guliashki, Kirilov, 2011], [Guliashki, Kirilov, 2013] и класификационно-ориентираните интерактивни алгоритми [Marcotte, Soland (1986)], [Kirilov, Vassilev, (1997)] са най-широко разпространените интерактивни алгоритми за решаване на задачи на многокритериалната оптимизация. Въпреки че интерактивните алгоритми на отправната точка все още са доминиращи, класификационно-ориентираните интерактивни алгоритми дават по-добро решение на някои основни проблеми в диалога с ЛВР, свързани с

определянето на неговите предпочитания и отнасящи се до времето, необходимо за генериране на нови недоминирани решения за оценяване и избор [Marler, Arora, (2004)], [Guliashki, Kirilov (2009).], [Tonchev, Kirilov (2007)].

Разработени са алгоритми от различен тип, които апроксимират оптимални решения в множеството на Парето [Kirilov, (1993)], [Athanasopoulos, Papalambros, (1996)], [Marler, Arora, (2010)], [Kirilov, Guliashki, (2014)] Повечето от тях са итеративни и пораждат точки от това множество. Някои от тези алгоритми са прецизно обосновани от гледна точка на теоретични доказателства и оптималност [Ehrgott, Wiecek (2004)], докато други са евристични и често нямат теоретична обосновка. За много 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 точки. Дори и от теоретична гледна точка да е възможно намирането на всичките тези точки, обикновено се оказва, че това е скъпо и изисква много изчислителни ресурси.

Основните представители на евристичните алгоритми са многокритериалните генетични (еволюционни) алгоритми [Goldberg, (1989)]. Тези алгоритми третираат задачата за многокритериална оптимизация по-скоро като задача за векторна оптимизация, отколкото като задача за вземане на решение и ударението е поставено върху намирането на подмножество от потенциални Парето-оптимални решения, които апроксимират достатъчно добре цялото множество на Парето. Това се постига чрез поддържането на популация от кандидати за апроксимиращото множество през целия процес на оптимизация. Тази популация се подобрява на всяка итерация с помощта на различни операции, моделиращи основните процеси на генетиката в биологията – селекция, рекомбинация и мутация. Различни модификации на тези операции се употребяват в генетичните алгоритми, а така също и различни популационни модели за подобряване на тяхната сходимост до Парето-оптимално

множество. Прилагат се и различни механизми за разпръскване на текущата популация, което гарантира добра апроксимация на цялото множество на Парето.

Генетичните алгоритми са алгоритми с вграден паралелизъм, позволяващ преодоляването на много трудности (наличие на множество от локални оптимуми, множество от решения и т.н.) при решаването на едно- или многокритериални комбинаторни и неизпъкнали нелинейни задачи. Основните недостатъци на тези алгоритми са свързани с непълното използване на спецификата на решаваните задачи, с необходимостта от относително големи изчислителни ресурси, с липсата на условия за оптималност. Развива се интензивна изследователска дейност за преодоляване на някои от тези трудности. Разработени са хибридни генетични алгоритми [Ishibuchi, Murata, (1998)], [Guliashki, Genova (2013)], които използват спецификите на решаваните задачи за локално подобряване на популациите, модифицирани са процедурите за избор на ново решение, включени са нови механизми за разпръскване на популацията.

Решенията, получени с помощта на генетичните алгоритми, в общия случай са приближени Парето-оптималните решения. По време на процеса на определяне на апроксимиращото множество ЛВР е изолиран, а накрая му се представя за оценяване и избор твърде голямо множество от решения (това е сравнително труден проблем на многокритериалния анализ). Голяма част от тези решения не са необходими на ЛВР, а най-предпочитаното решение може дори да не бъде намерено.

Намирането на крайно (окончателно) решение при този вид задачи е свързано с активното участие на лице или група от хора с еднакви разбирания (ЛВР). Пълното формализиране (автоматизиране) на решаването на задачата е невъзможно или нежелателно.

Има различни области на многокритериалната оптимизация (МО), свързани с качествата (степената на познаване на задачата, поведенчески, психологически и др.) на ЛВР,

предметната област на решаваните задачи и вида на формалните модели и методи, използвани от ЛВР, при решаването на конкретната задача.

Чрез сравнение със задачата на ЕО ще разгледаме особеностите, свързани със задачата на МО. Най-напред трябва да се отбележи, че в зависимост от вида на функциите $f_i(x)$, $i=1, \dots, k$ и $g_j(x)$, $j=1, \dots, m$, по отношение на променливите $x_1, \dots, x_n$, както и от вида на тези променливи, се различават линейни, нелинейни, непрекъснати и дискретни, детерминирани и стохастични, мрежови и немрежови задачи на многокритериалната (еднокритериалната) оптимизация.

Решението на задача (I) се нарича оптимално решение на задачата. В общия случай то е единствено решение. Нека го означим с x^* . За него е изпълнено условието:

$$f(x^*) \geq f(x), \text{ за } x \in S.$$

За x^* целевата функция $f(x)$ достига своя максимум.

Това обаче не е валидно при решаването на задача (II). Чисто математически при многокритериалната задача се оказва невъзможно да се намери едно решение, което би било оптимално за всички целеви функции едновременно. Това се дължи на факта, че целевите функции са противоречиви и несъизмерими. В този смисъл задача (II) принадлежи към лошо дефинираните математически задачи и затова тук се говори за оптималност по Парето.

Известно е, че в n -мерното Евклидово пространство не съществува пълна наредба, която да позволява сравняването на всеки два вектора от това пространство. Такава наредба съществува само в R^1 . В този смисъл векторите от R^n можем да разделим на сравними и несравними вектори.

Ще разгледаме две форми за сравняване на сравними вектори в критериалното пространство - наречени доминираност и строга доминираност [Steuer (1986), Vassileva(2004)]:

Дефиниция 1:

Нека $z^1 \in Z$ и $z^2 \in Z$ са два вектора на критерии. Казваме, че векторът z^1 доминира вектора z^2 , ако е изпълнено следното условие:

$$z_i^1 \geq z_i^2, \forall i=1, \dots, k$$

и за поне един индекс $j \in \{1, \dots, k\}$ е изпълнено условието:

$$z_j^1 > z_j^2,$$

т.е. за всички компоненти на двата вектора е изпълнено нестрого неравенство и поне за една компонента (един индекс) имаме строго неравенство.

Забележка 1: Разглеждаме понятието доминираност в смисъла на задачата за максимум.

Дефиниция 2:

Нека $z^1 \in Z$ и $z^2 \in Z$ са два вектора на критерии. Казваме, че векторът z^1 доминира строго вектора z^2 , ако е изпълнено следното условие:

$$z_i^1 > z_i^2, \forall i = 1, \dots, k,$$

т.е. всяка компонента на z^1 да е по-голяма от съответната компонента на z^2 .

Дефиниция 3:

Векторът на променливите $x^* \in S$ е Парето-оптимален, ако не съществува друг вектор на променливите $x \in S$, така че да е изпълнено условието:

$$f_i(x) \geq f_i(x^*), i=1, \dots, k$$

и поне за един индекс $j \in \{1, \dots, k\}$ да е изпълнено:

$$f_j(x) > f_j(x^*).$$

Дефиниция 4:

Критериалният вектор $z^* \in Z$ е оптимален по Парето, ако не съществува друг критериален вектор $z \in Z$, така че да е изпълнено условието:

$$z_i \geq z_i^*, i=1, \dots, k$$

и поне за един индекс $j \in \{1, \dots, k\}$ да е изпълнено:

$$z_j > z_j^*.$$

Забележка 2: Дефиниция 3 се отнася за оптималност по Парето в пространството на променливите $S \subset \mathbb{R}^n$, а Дефиниция 4 - в критериалното пространство $Z \subset \mathbb{R}^k$.

Ако обобщим двете дефиниции за оптималност по Парето, ще видим, че съответният вектор е оптимален по Парето, ако не съществува друг вектор (от съответния вид), който да го доминира (да е по-добър от него).

Забележка 3: Често елементите на множеството на Парето в пространството на променливите се наричат още ефективни вектори (точки, решения), а в пространството на критериите - недоминирани вектори (точки).

Още може да се каже, че $z^* \in Z$ е Парето-оптимален критериален вектор, ако съответният му вектор на променливите $x^* \in S$ е също Парето-оптимален (т.е. съществува съответствие).

Друга важна особеност е, че множеството на Парето, което се състои от Паретовски вектори, се намира винаги по границата на допустимото множество в критериалното пространство $Z \subset \mathbb{R}^k$.

Дефиниция 5:

Векторът на променливите $x^* \in S$ е слабо Парето-оптимален, ако не съществува друг допустим вектор $x \in S$, за който да е изпълнено условието:

$$f_j(x) > f_j(x^*), i=1, \dots, k.$$

(т.е векторът $x^* \in S$ е слабо Парето-оптимален, ако не съществува друг допустим вектор, който да го доминира строго).

Дефиниция 6:

Критериалният вектор $z^* \in Z$ е слабо Парето-оптимален, ако не съществува друг вектор $z \in Z$, за който да е изпълнено условието:

$$z_i > z_i^*, i=1, \dots, k.$$

Забележка 4: Дефиниция 5 се отнася за понятието слаба оптималност по Парето в пространството на променливите $S \subset \mathbb{R}^n$, а Дефиниция 6 - в критериалното пространство $Z \subset \mathbb{R}^k$.

Забележка 5: Често слабо Парето-оптималните вектори в пространството на променливите се наричат още слабо ефективни вектори, а в критериалното пространство - слабо недоминирани вектори.

От математическа гледна точка определянето на множеството на Парето може да се счита за решение на задача (II). Това множество обаче, се състои от безкраен брой елементи.

Всеки елемент от него е еднакво добър за да бъде решение на задачата (II). Генерирането на всички Парето-оптимални вектори практически е невъзможно, но не е и необходимо. Желателно е да се определи едно решение на задачата - една точка от множеството на Парето. Следователно е нужна допълнителна информация за да се направи този избор. От една страна, както знаем, Парето-оптималните вектори са еднакво добри един спрямо друг (т.е. те са несравними помежду си), а от друга страна стои следният проблем: в целевите функции не се съдържа информация, с помощта на която може да се извърши изборът на едно такова решение. Поради това, в сравнение с еднокритериалната оптимизация, тук е необходимо да се включи един нов елемент, за да се осъществи решаването на задачата на многокритериалната оптимизация. Това е още една от основните разлики между еднокритериалната и многокритериалната оптимизация. Този нов елемент се нарича лице, вземащо решение (ЛВР). ЛВР представлява човек (или група хора с еднакви разбирания), който е запознат със задачата и може да изрази своите предпочитания между отделните Парето-оптимални решения. ЛВР трябва да направи избора на крайното решение на задачата и е отговорно за това решение.

Дефиниция 7:

Под *крайно решение на задачата (II)* ще разбираме Парето-оптимално решение, което удовлетворява в най-голяма степен предпочитанията и изискванията на ЛВР. Това решение се нарича още и най-предпочитано решение на задачата. Решаването на задачата (II) се извършва чрез кооперация между ЛВР и така наречения РЕШАТЕЛ.

РЕШАТЕЛЯТ представлява обикновена компютърна програма, която отговаря за математическата страна на процеса на решаване на задачата. По-точно казано, РЕШАТЕЛЯТ генерира Парето-оптимални решения, а ЛВР извършва селекцията на едно решение - крайното, от определените вече от РЕШАТЕЛЯ Парето-оптимални решения. За генерирането на Парето-оптималните решения РЕШАТЕЛЯТ използва информацията за предпочитанията

на ЛВР, т.е. не е необходимо да се генерират изобщо всички Парето-оптимални решения, а само тези от тях, които удовлетворяват предпочитанията на ЛВР.

Обикновено се приема, че ЛВР взема своите решения на основата на една неявна (скрита) функция, наречена ценностна функция или функция на полезността.

Дефиниция 8:

Функцията $U: R^k \rightarrow R$ се нарича *ценностна функция*, ако отчита предпочитанията на ЛВР.

Разглеждаме допустимото критериално множество $Z \subset R^k$. Нека разгледаме $z^1, z^2 \in Z$ - два различни критериални вектора. Ако $U(z^1) > U(z^2)$, то ЛВР предпочита вектора z^1 . Ако $U(z^1) = U(z^2)$, то ЛВР приема, че двата вектора са еднакви за него - не може да ги различи.

Трябва да се подчертае, че ценностната функция е концепция, изцяло ориентирана към ЛВР. За една и съща задача различни ЛВР имат различни ценностни функции. Ценностната функция на ЛВР въвежда пълна наредба в допустимото критериално множество Z , в което като подмножество на R^k , такава по начало не съществува.

За съжаление, в практиката ценностната функция не може да бъде използвана лесно поради следните причини:

- Изключително трудно е да се определи явната форма (математическия вид) на ценностната функция за всеки ЛВР;
- Дори да се определи математическия вид на ценностната функция, той обикновено е много сложен.

Обикновено се приема, че ценностната функция на ЛВР е покомпонентно нарастваща.

Освен концепцията за ценностна функция в действията на ЛВР, която се нарича оптимизационна концепция, съществуват и други - т. нар. *удовлетворителна (задоволителна)* концепция в действията на ЛВР.

Удовлетворителното (задоволителното) вземане на решения означава, че ЛВР не оптимизира своя ценностна функция, а се старее да достигне някакви аспирационни нива на целевите функции. Решенията, които удовлетворяват всички аспирационни нива на критериите, се наричат удовлетворителни решения. В общия случай удовлетворителното решение може да не е Парето-оптимално решение.

Генерирането на Парето-оптималните решения е основна фаза в многокритериалната оптимизация и от математическа гледна точка многокритериалната задача се смята за решена, когато е определено множеството на Парето. Но обикновено се търси само едно решение и това означава, че трябва да се намери начин за подреждане на елементите на множеството на Парето. Затова е нужен ЛВР и неговата структура на предпочитанията, за да се извърши това подреждане. Така се вижда, че докато при еднокритериалната оптимизация субективност има на входа (при разработката на математическия модел на задачата), при многокритериалната оптимизация субективност има и на входа и на изхода.

Разработени са различни методи за решаване на задачата на многокритериалната оптимизация. Някои от тях работят в пространството на променливите, други (и то повечето) - в критериалното пространство. Това се дължи на две по-важни причини:

Критериалното пространство е с по-малка размерност;

ЛВР се ориентира и се интересува от стойностите на критериите (критериите са физични величини, докато променливите често са формални величини).

3.3. Скаларизиращ подход за трансформация на задачите за многокритериална в задачи за еднокритериална оптимизация

Обикновено задачите за многокритериална оптимизация се решават чрез скаларизация. Скаларизацията означава трансформация на многокритериалната оптимизационна задача в една или множество от еднокритериални оптимизационни задачи с реална целева функция, зависеща обикновено от един или няколко параметъра. Тази трансформация позволява да се използват теорията и резултатите на еднокритериалната оптимизация.

Основата за такава трансформация се състои във факта, че всяко Парето-оптимално решение на многокритериалната оптимизационна задача може да бъде получено като оптимално решение на еднокритериални оптимизационни задачи. Целевите функции на еднокритериалните задачи зависят от един или няколко параметъра.

Основният проблем при скаларизацията е определяне на параметрите така, че да има съответствие между даден набор от параметри и дадено Парето-оптимално решение.

Поради тези причини се поставят определени изисквания към скаларизиращите задачи. Обикновено те са следните две изисквания:

- Чрез промяна на параметрите на скаларизиращата задача да може да се намери всяко Парето-оптимално решение на многокритериалната задача.
- За всяко Парето-оптимално решение на многокритериалната задача да могат да се намерят подходящи (съответни) стойности на параметрите на еднокритериалната скаларизираща задача.

Трябва обаче да се има пред вид, че чрез този основен подход за решаване на многокритериалната задача, а именно скаларизацията, се пренасят слабостите на еднокритериалната оптимизация, свързани с:

- намирането само на екстремалните точки на линейните задачи;
- намиране на локалните оптимални решения на нелинейни задачи.

При решаване на линейни задачи на многокритериалната оптимизация като скаларизиращи задачи се използват задачите на линейното програмиране [Genova, Gulishaki и др. (2013)].

Някои от най-значимите видове скаларизиращи задачи и техните свойства са следните [Vassileva (2003)]:

Скаларизираща задача на отпавното направление [Narula, Kirilov и др. (1992)]

Представява задача на параметричното линейно програмиране и е разработена за интерактивно решаване на линейни многокритериални задачи.

Задачата има следния вид:

$$(S1) \quad \left| \begin{array}{l} \min \max_{1 \leq k \leq p} \left\{ \frac{\bar{z}_k - f_k(x)}{\omega_k} \right\} \\ \bar{z}_k = f_k + t(\bar{z}_k - f_k), k = \overline{1, p} \\ x \in X, \end{array} \right.$$

където: ω_k е k-тата компонента на тегловен (мащабиращ) вектор ω ;

f_k е стойността на k-тия критерий в текущото решение;

t е параметър: $t = 1, 2, \dots, \infty$.

Скаларизираща задача на претеглената сума на критериите [Maler, Arona (2010)]

Основната идея на тази скаларизираща задача е всеки критерий да получи собствен приоритет чрез задаване на тегловен фактор и да се максимизира сумата от всички критерии, умножени по съответния тегловен фактор.

Този скаларизираща задача има следния вид:

$$(S2) \quad \left| \begin{array}{l} \max \sum_{i=1}^k \omega_i f_i(x) \\ x \in S \end{array} \right.,$$

където параметрите трябва да изпълняват следните условия:

$$\left| \begin{array}{l} \omega_i \geq 0, i=1, K, k \\ \sum \omega_i = 1 \end{array} \right.$$

Задачата има следните свойства:

Свойство 1: Нека $x^* \in S$ е оптимално решение на задача (S2). Тогава x^* е слабо Парето-оптимално решение на задача (II).

Свойство 2: Нека $x^* \in S$ е оптимално решение на задача (S2) при положителни тегловни коефициенти $\omega_i > 0, i=1, \dots, k$. Тогава x^* е Парето-оптимално решение на задача (II).

Свойство 3: Нека $x^* \in S$ е единствено оптимално решение на задача (S2). Тогава x^* е Парето-оптимално решение на задача (II).

Свойство 4: Нека задача (II) е изпъкнала задача. Ако x^* е Парето-оптимално решение на задачата, то съществува тегловен вектор $\omega = (\omega_1, \dots, \omega_k)^T$, такъв че x^* е оптимално решение на задача (S1).

Трябва да се отбележи, че в случай, че задача (II) е неизпълнена задача, то с помощта на скаларизиращата задача (S2) не могат да се намерят всички Парето-оптимални решения.

Скаларизираща задача на ε -ограниченията [Yang, Cai и др. (2014)]

При тази задача трансформацията се състои в това, че един от критериите се избира за оптимизация, а всички останали се трансформират ограничения, като се задава долна граница за всеки един от тях. Използва се за генериране на недоминирани решения.

Задачата изглежда така:

$$(S3) \quad \begin{cases} \max f_l(x) \\ f_i(x) \leq \varepsilon_i, \quad i = 1, K, k, i \neq l. \\ x \in S \end{cases}$$

При тази скаларизираща задача една от функциите се максимизира, докато другите се използват във вид на ограничения.

Свойствата на тази задача са следните:

Свойство 1: Нека $x^* \in S$ е оптимално решение на задача (S3). Тогава x^* е слабо Парето-оптимално решение на задача (II).

Свойство 2: Векторът x^* е Парето-оптимално решение на изходната многокритериална задача (II) тогава и само тогава, когато е оптимално решение на задача (S3) за всяко $l=1, \dots, k$ при $\varepsilon_i = f_i(x^*)$, $i = 1, K, k$ и $i \neq l$.

Свойство 3: Ако $x^* \in S$ е единствено оптимално решение на задача (S3) за някое l при $\varepsilon_i = f_i(x^*)$, $i=1, K, k$, $i \neq l$, то $x^* \in S$ е Парето-оптимално решение на изходната многокритериална задача (II).

Свойство 4: За дадени горни граници $\varepsilon = (\varepsilon_1, \dots, \varepsilon_{l-1}, \varepsilon_{l+1}, \dots, \varepsilon_k)^T$, оптималното решение на задача (S3) е Парето-оптимално решение на изходната многокритериална задача (II), ако то е единствено решение.

Със задачата (S3) може да се намери всяко Парето-оптимално решение на задача (II). От изчислителна гледна точка обаче, тя е много трудоемка, тъй като трудно се доказва единственост на решението и трябва да се решават много еднокритериални задачи.

Скаларираща задача на глобалния критерий [Proos, Steven и др. (2001)]

Идеята при тази задача е да се минимизира претегленото разстояние между идеалния вектор и допустимото множество Z в критериалното пространство. В зависимост от избора на отправна точка и на метриката за измерване на претегленото разстояние между нея и допустимото критериално множество могат да се получат различни скалариращи задачи.

Тази задача има следния вид:

$$(S4) \quad \min_{x \in S} \left[\sum_{i=1}^k \omega_i (z_i^* - f_i(x))^p \right]^{1/p},$$

където $p=1, 2, \dots$, а z_i^* , $i=1, \dots, k$ са компонентите на така наречения идеален вектор.

i -тата компонента на идеалния вектор z_i^* е равна на оптималното решение на следната задача:

$$\max_{x \in S} f_i(x).$$

С други думи, за да се намерят компонентите на идеалния вектор, се извършва индивидуална оптимизация на целевите функции на многокритериалната задача и получените решения за отделните функции са точно компонентите на идеалния вектор.

По-важни свойства на задача (S4) са следните:

Свойство 1: Всяко оптимално решение на задача (S4) е Парето-оптимално решение на изходната многокритериална задача (II), ако решението е единствено и тегловните коефициенти са строго положителни.

Свойство 2: Ако изходната многокритериална задача (II) е изпъкнала, с помощта на задача (S4) може да се намери всяко нейно Парето-оптимално решение.

Свойство 3: Оптималното решение на задача (S4) е Парето-оптимално решение на изходната многокритериална задача, ако е изпълнено поне едно от трите условия:

- отправната точка е недопустима точка в критериалното пространство;
- решението е единствено;
- тегловните коефициенти са строго положителни.

Чебишевска скаларизираща задача [Gong (2011)]

Тази задача се явява частен случай на скаларизиращата задача на глобалния критерий, когато претегленото разстояние $r = \infty$. Специфичните свойства на тази задача и нейните

модификации я правят изключително добър инструмент за разработка на интерактивни методи за решаване на задачи за многокритериална оптимизация.

Задачата има следния вид:

$$(S5) \quad \min_{x \in S} \max_{1 \leq i \leq k} \{ \omega_i (\bar{z}_i - f_i(x)) \},$$

където $\bar{z}_i$, $i=1, \dots, k$, са компонентите на така наречения еталонен (отправен) аспирационен вектор. ($\bar{z}_i$ могат да бъдат и компонентите на идеалния вектор). В общия случай компонентите на еталонния вектор се определят от аспирационните (желаните) нива на целевите функции, които нива ЛВР желае да постигне.

Основните свойства на задача (S5) са:

Свойство 1: Всяко оптимално решение на задача (S5) при положителни $\omega_i > 0$, $i=1, \dots, k$ е слабо Парето-оптимално решение.

Свойство 2: Задача (S5) има поне едно Парето-оптимално решение.

Свойство 3: Всяко Парето-оптимално решение на изходната многокритериална задача (II) (независимо дали е изпъкнала или не) може да бъде намерено чрез решаване на задача (S5), при някои стойности на параметрите ω_i , $i=1, \dots, k$ и $\bar{z}_i$, $i=1, \dots, k$.

Трябва да се отбележи, че задача (S5) има една слаба страна, и тя е, че при нейното решаване могат да се получат и слабо Парето-оптимални решения. Но въпреки тази слабост, много силно свойство на задача (S5) е, че с нея могат да бъдат намерени всички Парето-оптимални решения на изходната многокритериална задача (II).

За премахването на слабостта на задача (S5), тя може да се модифицира малко. Така се получава т. нар. Разширена Чебишевска скаларизираща задача, която има следния вид:

$$(S6) \quad \begin{cases} \min \max_{1 \leq i \leq k} \{ \omega_i (\bar{z}_i - f_i(x)) \} + \beta \sum_{i=1}^k (\bar{z}_i - f_i(x)) \\ x \in S, \end{cases}$$

където β е малко положително число.

Скаларизиращите задачи (S5) и (S6) имат една особеност. Даже и $f_i(x)$, $i=1, \dots, k$ да са диференцируеми, задачите са недиференцируеми. Тези задачи обаче, могат да се трансформират в диференцируема форма, съответно (S5') и (S6'):

$$(S5') \quad \begin{cases} \min \alpha \\ \alpha \geq \omega_i (\bar{z}_i - f_i(x)), i = 1, K, k \\ x \in S \end{cases}$$

и

$$(S6') \quad \begin{cases} \min (\alpha + \beta \sum_{i=1}^k \bar{z}_i) \\ \alpha \geq \omega_i (\bar{z}_i - f_i(x)), i = 1, K, k \\ \bar{z}_i - f_i(x) = z_i, i = 1, K, k \\ x \in S \end{cases}$$

Съществуват и други модификации на чебишевските скаларизиращи задачи (S5) и (S6), ориентирани за решаване на различни типове многокритериални задачи. Придобилите най-голяма известност модификации могат да бъдат отделени в две групи: скаларизиращи задачи на отправната точка и класификационно-ориентирани скаларизиращи задачи.

Скаларираща задача RD [Narula, Kirilov и др. (1992)], [Kirilov (1999)]

Тази задача се използва в метода RD, представляващ интерактивен метод на отправно направление, предназначен за решаване на нелинейни задачи на многокритериалната оптимизация. Идеята тук е, че ЛВР разделя неявно множеството на индексите на критериите (означено с K) на три групи, означени съответно, с $K^{\geq}, K^{\leq}$ и $K^=$.

Задачата има следния вид::

$$(S7) \quad \begin{cases} \min \max_{k \in K^{\geq}} \left(\frac{f_k - f_k(x)}{\bar{z}_k - f_k} \right) \\ f_k(x) \geq \bar{z}_k + \alpha(f_k - \bar{z}_k), k \in K^{\leq}, \\ f_k(x) \geq f_k, k \in K^=, \\ x \in X, \end{cases}$$

където: $0 \leq \alpha < 1$;

$$\bar{z}_k = \begin{cases} f_k + \Delta_k, \text{ ако } k \in K^{\geq}, \\ f_k - \delta_k, \text{ ако } k \in K^{\leq}, \\ f_k, \text{ ако } k \in K^=; \end{cases}$$

$f_k, k \in K$ е стойността на критерия с индекс k в текущото намерено решение.

Скаларираща задача на Wierzbicki [Wierzbicki (1980)]

При скалариращата задача на Wierzbicki се използва отправна точка - допустима или недопустима в критериалното пространство. Нейните компоненти се формират посредством аспирационни нива на критериите, които ЛВР определя. Основната идея тук е да се минимизира претегленото отклонение на допустимото множество на критериите Z от отправната точка, задавана от ЛВР.

Задачата има следния вид:

$$(S9) \quad \left| \begin{array}{l} \min \max_{1 \leq k \leq p} \{w_k (\bar{z}_k - f_k(x))\} \\ x \in X, \end{array} \right.$$

където:

$\bar{z}_k$ е k-тата компонента на отправната точка $\bar{z}$;

w_k е k-тата компонента на фиксиран положителен тегловен вектор ω .

В общия случай, решенията на задача (S9) са слабо Парето-оптимални. Когато се търсят само Парето-оптимални решения, видът на задачата е следният:

$$(S10) \quad \left| \begin{array}{l} \min \left[\max_{1 \leq k \leq p} \{w_k (\bar{z}_k - f_k(x))\} + \rho \sum_{k=1}^p w_k (\bar{z}_k - f_k(x)) \right] \\ x \in X, \end{array} \right.$$

където ρ е малко положително число.

Скалариизираща задача STOM [Sawaragi, Nakayama и др. (1985)]

При тази задача се използва в метода на удовлетворителния компромис.

Задачата има следния вид:

$$(S11) \quad \left| \begin{array}{l} \min \max_{1 \leq k \leq p} \left\{ \frac{z_k^{00} - f_k(x)}{z_k^{00} - \bar{z}_k} \right\} \\ x \in X, \end{array} \right.$$

където:

$\bar{z}_k$ е k-тата компонента на отправната точка $\bar{z}$;

z_k^{00} е k-тата компонента на утопия-точката z^{00} .

Основната цел тук е да се минимизира претегленото отклонение на допустимото множество на критериите Z от утопия-точката, като тегловният коефициент ω_k е равен на

$$1 / (z_k^{00} - \bar{z}_k).$$

Скаларизираща задача STEP [Benayoun, Montgolfier и др. (1971)]

Методът STEP може да се каже, че е първият интерактивен метод за решаване на линейни задачи на многокритериалната оптимизация ориентиран към обучение. Идеята е, че ЛВР трябва да раздели множеството от индексите на критериите на два класа. Първият клас съдържа индексите на тези критерии, чиито стойности ЛВР желае да подобри в сравнение със стойностите им в текущото решение. Вторият клас се състои от индексите на тези критерии, чиито стойности в текущото решение ЛВР е съгласен да бъдат влошени с максимални допустими величини $\delta_k > 0$, $k \in \{1, 2, \dots, p\}$.

Задача има следния вид:

$$(S11) \quad \left| \begin{array}{l} \min \max_{k \in K^>} \left(\frac{e_k}{\sum_{j \in K^>} e_j} (z_k^0 - f_k(x)) \right) \\ f_k(x) \geq f_k - \delta_k, k \in K^{\leq}, \\ f_k(x) \geq f_k, k \in K^>, \\ x \in X, \end{array} \right.$$

където:

$$e_k = \frac{1}{z_k^{nad}} \frac{z_k^0 - z_k^{nad}}{z_k^0},$$

Скаларизираща задача GUESS [Buchanan (1997)]

В основата на тази задача е т. нар. “Наивен” метод. В задачата се изисква информация за надир-точката z^{nad} . Тази точка представлява долната граница на множеството на Парето. Определянето на компонентите на тази точка е трудна задача. Техни приближения могат да бъдат изчислени от т. нар. рау-off таблица.

Задачата има следния вид:

$$(S12) \quad \left| \begin{array}{l} \min \max_{1 \leq k \leq p} \left\{ \frac{z_k^{nad} - f_k(x)}{\bar{z}_k - z_k^{nad}} \right\} \\ x \in X, \end{array} \right.$$

където z_k^{nad} , $k = \overline{1, p}$ са компонентите на надир-точката z^{nad} .

Основната идея на тази скаларизираща задача е да се максимизира претегленото отклонение на допустимото множество на критериите Z от надир-точката. Тегловният

коэффициент w_k е равен на $\frac{1}{\bar{z}_k - z_k^{nad}}$. Аспирационните нива $\bar{z}_k$, $k = \overline{1, p}$ трябва да бъдат по-големи от стойностите на съответните компоненти на надир-точката. В общия случай, решенията на задача (S12) са слабо Парето-оптимални.

3.4. Софтуерни системи подпомагащи решаването на задачи за многокритериална оптимизация

Както при системите за решаване на задачи на многокритериалния анализ, софтуерните системи разработени за подпомагане на решаването на задачи на многокритериалната оптимизация също се делят на две групи: софтуерни системи с общо предназначение и проблемно-ориентирани такива. Първата група софтуерни системи служи за подпомагане на решаването на различни задачи на многокритериалната оптимизация от различни ЛВР. Обикновено в тях е реализиран един алгоритъм за решаване на многокритериални задачи. Това се дължи на следните причини:

- [1]. Различните алгоритми са предназначени за решаването на различни типове задачи на многокритериалната оптимизация (линейни, нелинейни, дискретни, непрекъснати, мрежови и т.н.).
- [2]. В различните алгоритми се използват различни по вид процедури за извличане и установяване на информация от ЛВР, което причинява значителни трудности при реализирането на модулите за дружелюбен потребителски интерфейс.
- [3]. В различните алгоритми се използват различни стратегии за обучение на ЛВР и различни подходи за намаляване на времето за решаване на скаларизиращата задача.

[4].Разработчиците на софтуерните системи обикновено имат интерес към реализирането на собствения си алгоритъм.

От историческа гледна точка, може да се каже, че разработването на софтуерни системи за подпомагане вземането на решения започва около средата на 60-те години на миналия век [Davis (1974)]. Дотогава използването на компютърни ресурси е било силно ограничено поради скъпата и неразвитите все още технология. По това време се появява изчислителни машини като IBM Systems 360, които правят тези ресурсите по-лесно достъпни и не толкова скъпи. Тогава поставя началото на разработването на информационни системи за управление за големите предприятия. Целта тогава е била да се предоставят на мениджмънта периодични структурирани аналитични отчети от счетоводни системи. Към края на 60-те години в практиката започват да се използват нов вид моделно-ориентирани системи за подпомагане вземането на решения. Една от първите значими СПВР, наречена Brandaïd се появява около 1975 година и служи за подпомагане на управленския състав при вземане на решения, касаещи ценообразуване и маркетинг и реклама на продукти [Power 2003].

Еволюцията на СПВР във вида, в който ги познаваме днес, започва в края на 80-те години, когато вече има изградена солидна теоретична основа, математически модели и методи. Някои добре известни софтуерни системи с общо предназначение, решаващи задачи на многокритериалната оптимизация са системите VIG [Korhonen, (1987)], DIDAS [Lewandowski, Kreglewski и др. (1989)]. В началото на 90-те години започват да се появяват и първите български изследвания в областта, създават се методи и алгоритми [Narula, Kirilov (1992)], [Sgurev, Vassilev и др. 1990], [Vassilev, Kirilov (1992)], [Kirilov, 1993], [Narula, Kirilov и др. (1994)], [Vassilev, Gouljashki и др. (1996)], [Kirilov, Vassilev (1996)], [Gouljashki, Kirilov и др. (1997)] заедно с тяхната първа софтуерна имплементация: MONP-16 [Sgurev, Vassilev и др. (1990)], MOLP-16 [Sgurev, Vassilev и др. (1990)], [Vassilev, Atanasov и др., (1993)]. От чуждестранните изследвания в областта и като софтуерна имплементация, през този период

по-широка популярност добиват системите LBS [Jaskiewicz, Slowinski, (1997)], SOMMIX [Climaco, Antunes и др., (1997)]. В някои от тези системи са реализиран повече от един алгоритъм за решаване на задачи на многокритериалната оптимизация. По-интересните системи от този тип са: системата ADELAIS [Zopounidis, Despotis и др., (1998)] – за избор на ценни книжа и Multicriteria DSS – за планиране на качеството на речната вода [Lotov, Bushenkov и др. (1997)]. В класа от софтуерни системи за многокритериална оптимизация трябва да бъдат включени и софтуерни системи, имплементиращи различни многокритериални еволюционни алгоритми. Въпреки, че те не гарантират получаването на точни решения, могат успешно да открият апроксимации на множеството на Парето-оптималните решения при дискретни, комбинаторни, неизпъкнали и нелинейни многокритериални задачи. Разработени са много такива софтуерни системи, но в този период по-значимите от тях са системата NSGM [Srinivas, Deb, (1994)], системата MOSES [Coello, Christiansen, (1999)]. Също така през тези години има и активно българско участие в разработката на системи за нелинейна многокритериална оптимизация [Vassilev, Genova (1994)], [Narula, Kirilov и др. (1994)], [Vassilev, Djambov и др. (1995)].

В началото на ново хилядолетие изпъкват системи като NIMBUS [Miettinen, (2000)], MOMILP [Alves, Climaco, (2004)]. Проблемно-ориентираните софтуерни системи за многокритериална оптимизация, които се появяват през този период най-често са включени в други софтуерни системи за управление и служат за решаване на един или няколко типа задачи на многокритериалната оптимизация [Kenney, Garson и др. (2002)], [Kirilov, Koynov и др. (2002)] [Boyko, Rudnichenko и др. (2017)]. Обикновено в тях е имплементиран проблемно-ориентиран потребителски интерфейс. От българските постижения като значими могат да бъдат отбелязани разработките на софтуерни системи за многокритериална оптимизация за общо ползване [Genova, Vassileva и др. (2003)], както и някои нови проблемно-ориентирани алгоритми и системи [Kirilov, Dokev (2002)], [Kirilov, Vassilev и др. (2003)], [Metev, Vassilev

(2003)], [Marinchev, Kirilov (2005)], както и разработката на нелинейни алгоритми за многокритериална оптимизация [Vassilev, Gouliashki и др. (1996)], [Kirilov (2000)]. Също така, значими са и съвместните разработки заедно със световно-признати учени в областта. Такива са например [Kirilov, Miettinen (2003)], [Kirilov, Miettinen (2005)]. Разработени и тествани са и няколко проблемно-ориентирани алгоритми за решаване на различни индустриални практически задачи [Kirilov, Tontchev и др. (2005)], [Tontchev, Kirilov (2005)], [Tontchev, Kirilov (2008)]. Софтуерните системи, имплементиращи еволюционни алгоритми за този период, които могат да се отбележат като значими са PAES [Knowles, Corne, (2000)], системата MOEA toolbox за MATLAB [<http://yarpiz.com/95/ypeal24-moead>]. Появяват се и първите български разработки на еволюционни алгоритми за многокритериална оптимизация [Kirilov, Guliashki (2011)], [Guliashki, Kirilov (2011)], [Guliashki, Kirilov и др. (2012)], [Guliashki, Genova и др. (2013)].

Значителна стъпка в развитието на СПБР е преминаването от стационарни системи към клиент-сървър базирани такива. Това започва също в началото на 90-те години. Тогава няколко фирми представят нов начин за подпомагане вземането на решения чрез използването на онлайн инструмент за анализ на информацията (online analytical processing - OLAP). През 1992-1993 година е завършена първата онлайн система за обработка на информацията. Скоро след това с огромното нарастване използването на Интернет от частния сектор, много фирми започват изграждане и подобряване на мрежовите си инфраструктури. Според едно изследване на Power [Power, (1998)], разработчиците на информационни системи разбират, че подпомагането вземането на решения е вече на съвсем различно ниво, и започват да добавят мрежови функционалности към базите данни и информационните системи. До към средата на 1990 г., изследователите и разработчиците разучавали възможностите за създаване на следващо поколение СПБР [Holsapple, Whinston, (1995)], [Bhargava, Krishnan и др., (1997)], [Bui, (1997)]. Те стигат до заключението, че WWW и Интернет технологиите създават нови

възможности за създаване и разпространение на СПВР. През 1995 на международната конференция „International Society for Decision Support Systems (ISDSS)” Bhargava и Krishnan, заедно с други съавтори представят серия от публикации на тема „DecisionNet, an electronic marketplace of Web-based decision support systems” [Bhargava, King , (1995)], [Kalakota, Whinston, (1995)]. Също така още две статии [Goul, Philippakis и др. (1995)], [Jeusfeld, Bui, (1995)], представени на същата конференция анализират използването на Уеб и Интернет за подпомагане вземането на решения. Тези и други подобни статии [Bhargava, Krishnan, (1998)] разглеждащи разработката на уеб-базирани СПВР са публикувани през 2002 г. в специално издание на списание за системи за подпомагане вземането на решения [Shim, Warkentin и др. (2002)]. Целта било да се създаде началото на достъпа до уеб-базирани материали касаещи дизайна и разработката на СПВР. Започва се използване на новите за тези години технологии като Web Services, SOAP, XML, API, Microsoft SQL server и др.

След 2002 г. софтуерни и хардуерни разработчици започват да обсъждат идеите около Grid [Power, (2002)], [Chang, Dasari и др., (2004)] и Utility [Naik, Mohindra и др., (2004)] изчисленията. Влиянието на тези технологии върху уеб-базираните СПВР остава неизяснено. Изглежда нереално, че с развитието на тези технологии, СПВР ще могат да работят с все по-големи и сложни модели и огромни обеми от информация на драстично по-ниска цена. Интегрирани среди за уеб-базирани разработки като LAMP (LINUX, Apache уеб сървър, MySQL бази от данни, PHP език за програмиране) стават все по-популярни за разработка на СПВР.

През 2004 г. WWW вече е утвърдена платформа за изграждане на такива системи и индустриални портали за информация, касаеща подпомагането вземането на решения [Bhargava, Power и др. (2007)]. За пример „DSSResources.com”, „OLAP Report”, и „DataWarehousing Online” са индустриални портали, предоставящи информация за софтуерни продукти, разработчици, методология и публикации в областта на технологиите за

подпомагане вземането на решения. Най-важното предимство на World Wide Web е, че тя е платформено-независима с възможност за отдалечени и разпределени изчисления, и обмен на сложна мултимедийна информация. От гледна точка на крайния потребител предимството е, че не е зависим от операционна система или изчислителна мощ тъй като за достъп до тези системи се изисква единствено Интернет свързаност и уеб-браузър.

В периода до 2010 г. много фирми използват Интернет като средство за предаване информация за СПВР или просто за използването на такъв вид софтуер. Въпреки това, употребата на уеб-базирана комуникация за демонстриране, разпространение или предоставяне на отдалечен достъп до СПВР все още е рядкост [Power, 2008]. След 2010 г. благодарение на все по-силната и утвърждаваща се интеграция на Интернет в ежедневието и практически във всяка една област, се наблюдават все по-големи усилия за разработване и прилагане на уеб-базираните СПВР в различни сфери, като медицина и здравеопазване [Sharma, Virmani, (2017)], хранително-вкусовата промишленост [Dellino, Laudadio, 2017], автоматизация на производството [Filip, Zamfirescu и др., (2017)], частния бизнес [Arrais-Castro, Leonilde и др., (2016)], различни задачи на правителствата и образованието [Guliashki, Genova и др. (2013)], [Khalili-Damghani, Abtahi и др. (2014)] и др. През 2011 година започва моделирането и създаването на първата българска уеб базирана система за решаване на задачи за многокритериална оптимизация, която е описана подробно в настоящата дисертация [Staykov, Andonov и др. (2011)], [Genova, Kirilov и др. (2011)].

В днешни дни научните изследвания и разработките в областта на СПВР са по-активни от всякога, служат за решаване на все по-сложни проблеми и благодарение на глобализацията и Интернет стават все по-достъпни както до специалистите в тази област, така и до по-широката публика от обучаващи се студенти, мениджъри или хора, проявяващи непрофесионален интерес в тази област [Hosack, Hall и др., (2012)].

Въпреки че уеб-базираните системи за подпомагане вземането на решения са значително обещаващи, има и някои важни предизвикателства, които трябва да бъдат преодоляни. Уеб архитектурния модел като замисъл е проектиран за случайни скокове в киберпространството и от това следва, че мрежата не осигурява постоянна връзка между клиента и сървъра. Това е основната разлика между настолните и уеб-базираните СПВР. Тази разлика води след себе си нужда от по-различен дизайн и многократен обмен на големи обеми от данни между клиента и сървъра [Hosack, Hall и др. (2012)], [Staykov, Kirilov и др. (2012)].

ЦЕЛИ И ЗАДАЧИ НА ДИСЕРТАЦИЯТА

Целта на настоящата дисертация е проектиране и реализиране на СПВР, базиращи се на избрани методи за решаване на различни класове задачи за многокритериална оптимизация.

Задачи:

1. Избор на методи за решаване на задачи за многокритериална оптимизация, които ще стоят в основата на изгражданите системи.
2. Алгоритмично и програмно осигуряване на избраните методи, което ще представлява ядрото на разработваните системи.
3. Разработване на синтаксис за дефиниране на задачи за многокритериална оптимизация и програмното му осигуряване за целите на разработваните системи.
4. Реализиране на СПВР за използване в работна среда под операционна система WINDOWS.
5. Реализиране на уеб базирана СПВР.
6. Разработка на комуникационни модули, позволяващи обмен на данни с външни системи.

ГЛАВА I. КЛАСИФИКАЦИОННО-ОРИЕНТИРАНИ СКАЛАРИЗИРАЩИ ЗАДАЧИ И АЛГОРИТМИ

В главата се описват скаларизиращи задачи и алгоритми за решаване на задачи за многокритериална оптимизация, разработени съвместно с екип от учени в БАН. В резултат е разработен и програмно реализиран интерактивен алгоритъм, базиран на метода GENS-IM, който служи за основа на изчислителните модули в разработваните приложни системи за многокритериална оптимизация МКО-2.1. и WebOptim. Част от резултатите са представени в публикация номер 5.

1.1. Класификационно-ориентирани скаларизиращи задачи DAL

Името на скаларизиращите задачи *DAL* [Vassileva (2004)] (още наречени задачи на желаните и приемливите нива), произлиза от първите букви на английските думи: *Desired* (Желан), *Acceptable* (Приемлив) и *Level* (Ниво). При тях идеята е на всяка итерация ЛВР да оценява текущото слабо недоминирано решение или недоминирано решение на линейната или линейната целочислена задача на многокритериалната оптимизация. Ако ЛВР желае да търси по – добро решение, той декларира своите предпочитания за желани и приемливи промени на стойностите на някои или всички критерии. В зависимост от тези предпочитания, множеството на критериите може да бъде разделено индиректно от ЛВР на четири или по-малко от четири класа, означени, съответно, с: $K^{\geq}$, $K^=$, $K^{\leq}$ и K^0 .

Всеки критерий $f_k(x)$, $k \in K$ може да попадне в един от тези класове, както следва:

$k \in K^{\geq}$, ако ЛВР желае критерият $f_k(x)$ да бъде подобрен с желаната стойност Δ_k , $\Delta_k > 0$;

$k \in K^{\leq}$, ако ЛВР е съгласен критерият $f_k(x)$ да бъде влошен най-много с някаква приемлива стойност $\delta_k, \delta_k > 0$;

$k \in K^=$, ако ЛВР желае да не се влошава текущата стойност на критерия $f_k(x)$;

$k \in K^0$, ако ЛВР е съгласен критерият $f_k(x)$ да се променя свободно.

За получаване на по-добро от текущото слабо недоминирано решение на линейната целочислена задача на многокритериалната оптимизация, на основата на направената от ЛВР неявна класификация на критериите, може да се използва следната целочислена скаларизираща задача *DAL-1* [Vassileva (2004)]:

Да се минимизира:

$$S(x) = \max \left[\max_{k \in K^{\geq}} (\bar{f}_k - f_k(x)) / |f'_k|, \max_{k \in K^{\leq}} (f_k - f_k(x)) / |f'_k| \right]$$

при ограниченията:

$$f_k(x) \geq f_k, \quad k \in K^=,$$

$$f_k(x) \geq f_k - \delta_k, \quad k \in K^{\leq},$$

$$x \in X_1,$$

където:

$f_k, k \in K$ е стойността на критерия $f_k(x)$ в текущото предпочитано решение;

$\bar{f}_k = f_k + \Delta_k, k \in K^{\geq}$ е желаното ниво на критерия $f_k(x)$;

X_1 е допустимото множество на изменение на целочислените променливи;

f'_k , $k \in K$ е мащабиращ коефициент:

$$f'_k = \begin{cases} \varepsilon, & \text{ако } |f'_k| \leq \varepsilon \\ f_k, & \text{ако } |f'_k| > \varepsilon \end{cases},$$

където ε е малко положително число.

Скаларизиращата задача *DAL-1* има три свойства, които позволяват да се преодолеят до голяма степен изчислителните трудности, свързани с нейното решаване като задача на целочисленото програмиране, а също така, и да се намали натоварването на ЛВР при сравняването на новите решения. Първото свойство е свързано с това, че текущото предпочитано целочислено решение (намерено в предходната итерация) на многокритериалната задача е допустимо целочислено решение на задача *DAL-1*. Това улеснява както точните, така и приближените алгоритми за решаване на задачата *DAL-1*, защото стартират с известно начално допустимо целочислено решение. Второто свойство е свързано с това, че допустимата област на задача *DAL-1* е част от допустимата област на целочислената многокритериалната задача, за разлика от допустимите области на скаларизиращите задачи на отправните точки, чиито допустими области съвпадат с допустимата област на изходната задача. В зависимост от значенията на параметрите $\Delta_k/k \in K^{\geq}$, $\delta_k/k \in K^{\leq}$ допустимата област на задача *DAL-1* може да бъде сравнително тясна и допустимите решения в критериалното пространство, намерени с точни или приближени алгоритми на целочисленото програмиране, могат да се намират съвсем близо до недоминираната повърхност на изходната целочислена многокритериална задача. Получаването и използването на такива приближени слабо недоминирани решения може значително да намали времето, през което ЛВР чака за оценка на нови решения. С незначително намаляване качеството на получаваните решения в критериалното пространство може значително да се подобри диалогът с ЛВР. Третото свойство е свързано с възможността ЛВР да реализира стратегия на търсене от типа “не големи печалби - малки загуби”. Това се дължи на факта, че със задача *DAL-1* се търси оптимално

решение, което минимизира Чебишевското разстояние от допустимото критериално множество до текущата еталонна точка, чиито компоненти са равни на желаните от ЛВР стойности на критериите, които се подобряват, и на текущите стойности на критериите, които се влошават. Получаваното слабо недоминирано решение и текущото решение са сравнително близки и ЛВР може по-лесно да направи своя избор. Това е в сила също, когато скаларизиращата задача се решава приближено и се получат повече допустими решения, които са сравнително близки, както с текущото решение, така и помежду си. С други думи, намалява се влиянието на т. нар. “ограничена сравняемост” на слабо недоминираните решения.

За получаване на по-добро от текущото слабо недоминирано решение на линейната задача на многокритериалната оптимизация може да се използва Чебишевската непрекъсната скаларизираща задача *DAL-2*, която се получава от целочислената скаларизираща задача *DAL-1*, замествайки ограничението (4) със следното ограничение:

$$x \in X_2,$$

където X_2 е допустимото множество на изменение на непрекъснатите променливи.

В групата на задачите *DAL*, освен *DAL-1* и *DAL-2*, има още скаларизиращи задачи-*DAL-3*, *DAL-4*, *DAL-5*, *DAL-5p* и *DAL-6*. Докато скаларизиращите задачи *DAL-1*, *DAL-2* и *DAL-3* са предназначени за намиране на слабо недоминирани решения, то скаларизиращи задачи *DAL-4*, *DAL-5*, *DAL-5p* и *DAL-6* са предназначени за намиране на недоминирани решения. За всички тези скаларизиращи задачи съществуват еквивалентни задачи с диференцируеми целеви функции.

1.2. Класификационно-ориентирани скаларизиращи задачи DALDI

Името на скаларизиращата задача *DALDI-1* [Vassilev, Genova и др.(2004)] произлиза от първите букви на английските думи: *Desired* (Желан), *Acceptable* (Приемлив), *Level* (Ниво), *Direction* (Посока) и *Interval* (Интервал).

Скаларизиращата задача *DALDI-1* [Vassilev, Genova и др.(2004)] има подобни характеристики като скаларизиращата задача *DAL-1*. Но между тях има и две различия. Първото различие се състои в това, че скаларизиращата задача *DALDI-1* дава по-голяма свобода на ЛВР да изразява своите локални предпочитания при търсенето на по-добро слабо недоминирано решение. Освен желани и приемливи стойности на част или всички критерии, ЛВР има възможност да задава и желани и приемливи посоки и интервали на изменение на стойностите на някои от критериите. По този начин ЛВР може да описва своите локални предпочитания с по-голяма гъвкавост, точност и сигурност. Второто различие между скаларизиращите задачи *DAL-1* и *DALDI-1* се отнася до възможността да се променят допустимите им множества (да се направят “по-тесни”), така че допустимите им решения да се намират близо до недоминираните (ефективните) решения на многокритериалната задача. Колкото повече са критериите, които ЛВР желае да бъдат свободно подобрявани или за които е съгласен да бъдат свободно влошавани, толкова тази възможност е по-малка. Тесните допустими области на скаларизиращите задачи *DAL-1* и *DALDI-1* позволяват успешно да се използват приближени еднокритериални алгоритми, което е особено важно, когато тези задачи са целочислени. Както лесно може да се види в това отношение, скаларизиращата задача *DAL-1* е по-добра от скаларизиращата задача *DALDI-1*.

Задачата *DALDI-1* има следния вид:

Да се минимизира:

$$S(x) = \max \left[\max_{k \in K^{\geq}} (\bar{f}_k - f_k(x)) / |f'_k|, \max_{k \in K^{<} \cup K^{\leq}} (f_k - f_k(x)) / |f'_k| \right] + \\ + \max_{k \in K^{>}} (f_k - f_k(x)) / |f'_k|,$$

при ограниченията:

$$f_k(x) \geq f_k, \quad k \in K^{>} \cup K^{=},$$

$$f_k(x) \geq f_k - \delta_k, \quad k \in K^{\leq},$$

$$f_k(x) \geq f_k - t_k^-, \quad k \in K^{> <},$$

$$f_k(x) \leq f_k + t_k^+, \quad k \in K^{> <},$$

$$x \in X_1.$$

За получаване на по – добро от текущото слабо недоминирано решение на линейната задача на многокритериалната оптимизация може да се използва Чебишевската непрекъсната скаларизираща задача *DALDI-2*, която се получава от целочислената скаларизираща задача *DALDI-1* [Vassilev, Genova и др. (2004)], замествайки ограничението

$$x \in X_1.$$

със следното ограничение:

$$x \in X_2$$

В групата на задачите *DALDI*, освен *DALDI-1* и *DALDI-2*, има още скаларизиращи задачи-*DALDI-3*, *DALDI-4*, *DALDI-5*, *DALDI-5p* и *DALDI-6*. Докато скаларизиращите задачи *DALDI-1*, *DALDI-2* и *DALDI-3* са предназначени за намиране на слабо недоминирани решения, то скаларизиращи задачи *DALDI-4*, *DALDI-5*, *DALDI-5p* и *DALDI-6* са предназначени за

намиране на недоминирани решения. За всички тези скаларизиращи задачи съществуват еквивалентни задачи с диференцируеми целеви функции.

1.3. Обобщени скаларизиращи задачи

Описаните в тази точка обобщени скаларизиращи задачи представляват обобщение на скаларизиращите задачи на претеглената сума, на задачите на ε -ограниченията, на задачите на отправната точка, на задачите на отправното направление и на класификационно – ориентираните задачи. Обобщените скаларизиращи задачи GENWS [Vassileva, Miettinen и др. (2005)] до голяма степен обединяват отделните скаларизиращи задачи и по този начин до голяма степен обединяват и техните качества. На тяхна основа може да се разработи и обобщен интерактивен алгоритъм на многокритериалната оптимизация с променлива параметризация и променлива скаларизация, който също обединява до голяма степен качества на отделните интерактивни методи, използващи съответните скаларизиращи задачи, и разширява възможностите на ЛВР за задаване на неговите предпочитания.

За получаване на слабо Парето-оптимално решение, стартирайки директно или индиректно от текущото слабо Парето-оптимално решение, може да бъде използвана следната обобщена скаларизираща задача GENWS:

Да се минимизира

$$2. \quad S(x) = \max_{k \in K^{\geq}} \left(\max_{k_1 \in K^{\leq}} (F_k^1 - f_k(x))^1 \max_{k_2 \in K^{\leq}} (F_k^2 - f_k(x))^2 \right. \\ \left. \max_{k \in K^{\leq}} (F_k^3 - f_k(x)) G_k^3 R_3 \max_{k \in K^{\geq}} (F_k^4 - f_k(x)) G_k^4 + \sum_{k \in K^0} (F_k^5 - f_k(x)) G_k^5 \right),$$

при ограниченията

$$3. \quad f_k(x) \geq f_k, k \in K^{\geq} \cup K^=,$$

4. $f_k(x) \geq f_k - D_k, k \in K^{\leq},$
5. $f_k(x) \geq f_k - t_k^-, k \in K^{><},$
6. $f_k(x) \leq f_k + t_k^+, k \in K^{><},$
7. $x \in X,$

където:

- K е множеството от всички критерии;
- $G_k^1, G_k^2, G_k^3, G_k^4, G_k^5$ са мащабиращи коефициенти;
- $F_k^1, F_k^2, F_k^3, F_k^4, F_k^5$ са параметри, свързани с аспирационни, текущи и други нива на стойностите на критериите;
- R_1, R_2, R_3 са равни на аритметичната операция “+” или на разделител “,”;
- D_k е величина, с която ЛВР е съгласен да се влоши критерия с индекс $k \in K$ ($0 < D_k < \infty$);
- t_k^- и t_k^+ са долна и горна граница на приемливия за ЛВР интервал на промяна на стойността на критерия с индекс $k \in K^{><};$
- f_k - стойност на критерия с индекс $k \in K$ в текущото получено решение;
- $K^{\geq}$ е множеството от критерии, за които ЛВР желае да бъдат подобрени техните текущи стойности до зададени от него желани нива F_k^1 ;
- $K^{>}$ е множеството от критерии, за които ЛВР желае да бъдат подобрени техните текущи стойности;

- $K^{\leq}$ е множеството от критерии, за които ЛВР е съгласен да бъдат влошени техните текущи стойности до зададени от него приемливи нива F_k^2 , но не повече от определени стойности $f_k - D_k$ ($F_k^2 \geq f_k - D_k$);

- $K^{<}$ е множеството от критерии, за които ЛВР е съгласен да бъдат влошени техните текущи стойности;

- $K^{=}$ е множеството от критерии, за които ЛВР желае да не бъдат влошени техните текущи стойности;

- $K^{><}$ е множеството от критерии, за които ЛВР е съгласен техните стойности да се изменят в определени интервали;

- K^0 е множеството от критерии, за които ЛВР не задава явно предпочитания за промяната на техните стойности.

Скаларизиращата задача GENWS има смисъл, когато $K^{\geq} \neq \emptyset$ и/или $K^{>} \neq \emptyset$ или $K^0 = K$. Затова приемаме, че $K^{\geq}$ и/или $K^{>} \neq \emptyset$ или $K^0 = K$. В общия случай, ограниченията (2)-(5) определят подмножество $X^1 \in X$, което съдържа слабо Парето-оптимални решения.

В общия случай, скаларизиращата задача GENWS е оптимизационна задача с недиференцируема целева функция. Всяка конкретна скаларизираща задача GENWS (определени значения на R_1, R_2, R_3) може да бъде сведена за сметка на допълнителни променливи и ограничения към еквивалентна оптимизационна задача с диференцируема целева функция. Еквивалентността на всяка двойка оптимизационни задачи е по отношение на получените стойности на критериите (целевите функции) и на основните променливи. При различни значения на R_1, R_2, R_3 могат да се получат различни типове еквивалентни задачи. Всяка еквивалентна задача може да се представи като

$$(7) \quad \min (\mu + \sum_{k \in K^0} y_k),$$

при изпълнението на две групи от ограничения. Първата група от ограничения е еднаква за всички типове еквивалентни задачи и има следния вид:

$$(8) \quad \alpha \geq (F_k^1 - f_k(x)) G_k^1, \quad k \in K^{\geq},$$

$$(9) \quad \beta \geq (F_k^2 - f_k(x)) G_k^2, \quad k \in K^{\leq},$$

$$(10) \quad \gamma \geq (F_k^3 - f_k(x)) G_k^3, \quad k \in K^{<},$$

$$(11) \quad \Omega \geq (F_k^4 - f_k(x)) G_k^4, \quad k \in K^{>},$$

$$(12) \quad (F_k^5 - f_k(x)) G_k^5 = y_k, \quad k \in K^0,$$

$$(13) \quad f_k(x) \geq f_k, \quad k \in K^{>} \cup K^{=},$$

$$(14) \quad f_k(x) \geq f_k - D_k, \quad k \in K^{\leq},$$

$$(15) \quad f_k(x) \geq f_k - t_k^-, \quad k \in K^{> <},$$

$$(16) \quad f_k(x) \leq f_k + t_k^+, \quad k \in K^{> <},$$

$$(17) \quad x \in X,$$

$\alpha, \beta, \gamma, \Omega$ – произволни.

Втората група от ограничения има различен вид и брой ограничения в зависимост от значенията на R_1, R_2, R_3 . Ще разгледаме тези ограничения за три различни еквивалентни задачи, които ще означим съответно с GENWS 1, GENWS 2 и GENWS 3.

Еквивалентната задача GENWS 1 се получава, когато R_1 и R_3 са равни на аритметичната операция “+”, а R_2 е равно на разделителя “,”. Ограниченията от втората група в този случай са две и имат следния вид:

$$(18) \quad \mu \geq \alpha + \beta,$$

$$(19) \quad \mu \geq \gamma + \Omega,$$

μ - произволно.

Еквивалентната задача GENWS 2 се получава, когато R_1, R_2, R_3 са равни на разделителя “,” и при нея ограниченията от втората група имат следния вид:

$$(20) \quad \mu \geq \alpha,$$

$$(21) \quad \mu \geq \beta,$$

$$(22) \quad \mu \geq \gamma,$$

$$(23) \quad \mu \geq \Omega,$$

μ - произволно.

Еквивалентната задача GENWS 3 се получава, ако R_1, R_2, R_3 са равни на аритметичната операция “+” и при нея втората група от ограничения се състои от едно ограничение:

$$(24) \quad \mu \geq \alpha + \beta + \gamma + \Omega,$$

μ - произволно.

Ограниченията от втората група при останалите еквивалентни задачи могат да се формулират по подобен начин.

В общия случай скаларизиращата задача GENS, подобно на скаларизиращата задача GENWS, е оптимизационна задача с недиференцируема целева функция. Всяка конкретна скаларизираща задача GENS (определени значения на R_1, R_2, R_3) може да бъде сведена, подобно на задача GENWS, към еквивалентна оптимизационна задача с диференцируема целева функция за сметка на допълнителни променливи и ограничения. Еквивалентността на всяка двойка оптимизационни задачи е по отношение на получените стойности на критериите (целевите функции) и на основните променливи. При различни значения на R_1, R_2, R_3 могат да се получат различни типове еквивалентни задачи. Всяка еквивалентна задача може да се представи, както следва:

$$(25) \quad \min(\mu + \sum_{k \in K^0} y_k + \rho \sum_{k \in K \setminus K^0} y_k),$$

при изпълнението на две групи от ограничения.

Първата група от ограничения е еднаква за всички типове еквивалентни задачи и има следния вид:

$$(26) \quad \alpha \geq (F_k^1 - f_k(x)) G_k^1, \quad k \in K^{\geq},$$

$$(27) \quad \beta \geq (F_k^2 - f_k(x)) G_k^2, \quad k \in K^{\leq},$$

$$(28) \quad \gamma \geq (F_k^3 - f_k(x)) G_k^3, \quad k \in K^{<},$$

$$(29) \quad \Omega \geq (F_k^4 - f_k(x)) G_k^4, \quad k \in K^{>},$$

$$(30) \quad (F_k^1 - f_k(x)) G_k^1 = y_k, \quad k \in K^{\geq},$$

$$(31) \quad (F_k^2 - f_k(x)) G_k^2 = y_k, \quad k \in K^{\leq},$$

$$(32) \quad (F_k^3 - f_k(x)) G_k^3 = y_k, \quad k \in K^<,$$

$$(33) \quad (F_k^4 - f_k(x)) G_k^4 = y_k, \quad k \in K^>,$$

$$(34) \quad (F_k^5 - f_k(x)) G_k^5 = y_k, \quad k \in K^0,$$

$$(35) \quad -f_k(x) G_k^6 = y_k, \quad k \in K^= \cup K^{><},$$

$$(36) \quad f_k(x) \geq f_k, \quad k \in K^> \cup K^=,$$

$$(37) \quad f_k(x) \geq f_k - D_k, \quad k \in K^{\leq},$$

$$(38) \quad f_k(x) \geq f_k - t_k^-, \quad k \in K^{><},$$

$$(39) \quad f_k(x) \leq f_k + t_k^+, \quad k \in K^{><},$$

$$(40) \quad x \in X,$$

$\alpha, \beta, \gamma, \Omega, y_k/k \in K$ - произволни.

Втората група от ограничения има различен вид и брой ограничения в зависимост от значенията на R_1, R_2 и R_3 . Тези ограничения са еднакви с ограниченията при еквивалентните задачи на скаларизиращата задача GENWS.

Ще разгледаме една еквивалентна задача на скаларизиращата задача GENS, която ще означим с GENS 1 и която се получава, когато R_1 е равно на разделителя “,” , а R_2 и R_3 са равни на аритметичната операция “+”. Ограниченията от втората група в този случай имат следния вид:

$$(41) \quad \mu \geq \alpha,$$

$$(42) \quad \mu \geq \beta + \gamma + \Omega,$$

μ - произволно.

Ограниченията от втората група при останалите еквивалентни задачи могат да се формулират по подобен начин.

Чрез промяна на параметрите на обобщената скаларизираща задача GENWS могат да се получат голяма част от придобилите известност скаларизиращи задачи, а могат, също така, да се генерират и нови скаларизиращи задачи. Качествата на всяка скаларизираща задача се определят от възможностите, които се предоставят на ЛВР за задаване на неговите предпочитания, от възможностите за преодоляване на изчислителните трудности при решаването на някои типове оптимизационни задачи, от качеството на получаваните решения.

Скаларизиращите задачи GENWS са обобщение на известни скаларизиращи задачи, като скаларизиращата задача на претеглената сума, скаларизиращата задача на ε -ограниченията, Чебишевската скаларизираща задача, скаларизиращите задачи на отправната точка, скаларизиращите задачи на отправното направление и класификационно-ориентираните скаларизиращи задачи.

При първите пет типа скаларизиращи задачи, стойностите на критериите в текущото получено решение не се включват директно в параметрите на скаларизиращата задача, макар че на основата на тези стойности ЛВР задава стойностите на параметрите на скаларизиращата задача. При класификационно-ориентираните скаларизиращи задачи, стойностите на критериите в текущото получено решение са параметри на скаларизиращата задача. При скаларизиращите задачи на претеглената сума ЛВР задава своите предпочитания чрез стойностите на теглата на критериите, а при скаларизиращите задачи на ε -ограниченията – чрез избор на една функция за максимизация и определяне на долни граници на изменение на останалите критерии. Отправната точка се определя от аспирационните нива на критериите, които нива ЛВР желае или е съгласен да се получат в новото решение. Тези аспирационни

нива на критериите са параметри на скаларизиращите задачи на отправната точка. Параметри на скаларизиращата задача могат да бъдат и стойностите на критериите в текущото получено решение. Задавайки желаните или приемливите промени на стойностите на критериите в текущото получено решение, ЛВР индиректно класифицира отделните критерии в различни групи.

От скаларизиращата задача GENWS могат да се получат някои от придобилите известност скаларизиращи задачи по следния начин:

Скаларизираща задача на претеглената сума

Скаларизиращата задача на претеглената сума може да бъде получена от скаларизиращата задача GENWS чрез следните замествания:

$$1) K^{\geq} = K^{\leq} = K^{<} = K^{>} = K^{=} = K^{> <} = \emptyset ,$$

$$2) K^0 = K ,$$

$$3) F_k^5 = 0 ,$$

$$4) G_k^5 = \omega_k \text{ и } \sum_{k \in K} G_k^5 = 1 .$$

Скаларизираща задача на ε – ограниченията

Скаларизиращата задача на ε – ограниченията може да бъде получена от скаларизиращата задача GENWS, както следва:

$$1) K^{\geq} = K^{<} = K^{=} = K^0 = K^{> <} = \emptyset ,$$

$$2) K^> = \{l\},$$

$$3) K^{\leq} = K \setminus K^>,$$

$$4) G_k^2 = 0,$$

$$5) F_k^4 = 0,$$

$$6) G_k^4 = 1,$$

$$7) D_k = f_k - \varepsilon_k.$$

Чебишевска скаларизираща задача

Чебишевската скаларизираща задача може да бъде получена от скаларизиращата задача GENWS, както следва:

$$1) K^0 = K^{\leq} = K^{<} = K^{>} = K^{=} = K^{>=} = \emptyset,$$

$$2) K^{\geq} = K,$$

$$3) G_k^1 = \omega_k,$$

$$4) F_k^1 = z_k^{**}.$$

Скаларизираща задача STEM

Скаларизиращата задача STEM може да се получи от скаларизиращата задача GENWS чрез следните замествания:

$$1) K^{<} = K^{\geq} = K^{=} = K^{>=} = K^0 = \emptyset,$$

$$2) G_k^2 = 0, k \in K^{\leq},$$

$$3) D_k = \Delta_k,$$

$$4) F_k^4 = z_k^*,$$

$$5) G_k^4 = \frac{e_k}{\sum_{j \in K^>} e_j}.$$

Скаларизираща задача STOM

Скаларизиращата задача STOM може да се получи от скаларизиращата задача GENWS чрез следните замествания:

$$1) K^> = K^< = K^= = K^{><} = K^0 = \emptyset,$$

$$2) R_1 = ",",$$

$$3) F_k^1 = F_k^2 = z_k^{**},$$

$$4) G_k^1 = G_k^2 = \frac{1}{z_k^{**} - \bar{z}_k},$$

$$5) D_k = \infty.$$

Скаларизираща задача на отправната точка

Скаларизиращата задача на отправната точка може да се получи от скаларизиращата задача GENWS чрез следните замествания:

$$1) K^> = K^< = K^= = K^{><} = K^0 = \emptyset,$$

$$2) R_1 = “, ” ,$$

$$3) F_k^1 = F_k^2 = \bar{z}_k,$$

$$4) G_k^1 = G_k^2 = \frac{1}{|\bar{z}_k|},$$

$$5) D_k = \infty.$$

Скаларизираща задача GUESS

Скаларизиращата задача GUESS може да се получи от скаларизиращата задача GENWS чрез следните замествания:

$$1) K^> = K^< = K^= = K^{><} = K^0 = \emptyset,$$

$$2) R_1 = ” , ” ,$$

$$3) F_k^1 = F_k^2 = z_k^{nad},$$

$$4) G_k^1 = G_k^2 = \frac{1}{\bar{z}_k - z_k^{nad}},$$

$$5) D_k = \infty.$$

Скаларизираща задача на модифицираната отправна точка MRP

Скаларизиращата задача на модифицираната отправна точка MRP може да бъде получена от скаларизиращата задача GENWS чрез следните замествания:

$$1) K^> = K^< = K^{><} = K^0 = \emptyset,$$

$$2) R_1 = “, ”,$$

$$3) F_k^1 = \bar{z}_k,$$

$$4) F_k^2 = f_k,$$

$$5) G_k^1 = \frac{1}{\bar{z}_k - f_k},$$

$$6) G_k^2 = \frac{1}{f_k - \bar{z}_k},$$

$$7) D_k = \infty.$$

Скаларизираща задача NIMBUS

Скаларизиращата задача NIMBUS може да бъде получена от скаларизиращата задача GENWS чрез следните замествания:

$$1) K^< = K^{> <} = \emptyset,$$

$$2) R_3 = “, ”,$$

$$3) F_k^1 = \bar{z}_k,$$

$$4) F_k^4 = z_k^*,$$

$$5) D_k = \Delta_k,$$

$$6) G_k^2 = G_k^3 = 0,$$

$$7) G_k^1 = G_k^4 = \frac{1}{|z_k^*|}.$$

Класификационно–ориентирана скаларизираща задача

Класификационно–ориентираната скаларизираща задача може да се получи от скаларизиращата задача GENWS чрез следните замествания:

$$1) R_1 = R_2 = “,” ,$$

$$2) R_3 = “+” ,$$

$$3) F_k^1 = \bar{f}_k ,$$

$$4) F_k^2 = F_k^3 = F_k^4 = f_k ,$$

$$5) G_k^1 = G_k^2 = G_k^3 = G_k^4 = \frac{1}{|f_k|} ,$$

$$6) D_k = \Delta_k .$$

Разгледаните десет скаларизиращи задачи могат да се получат чрез различни трансформации на обобщената скаларизираща задача GENWS. Чрез първите две от тях могат да се получат слабо Парето-оптимални и Парето-оптимални решения, докато останалите осигуряват в общия случай слабо Парето-оптимални решения. Всяка от тези скаларизиращи задачи обаче има разширение, с помощта на които се осигуряват Парето-оптимални решения.

Тези разширени скаларизиращи задачи могат да се получат чрез подобни разширения на обобщените скаларизиращи задачи GENS. Чрез промяна на параметрите на скаларизиращата задача GENWS могат да се получат и много нови скаларизиращи задачи с различни характеристики.

1.4. Обобщен интерактивен алгоритъм, базиран на метода GENS-IM за решаване на линейни и линейни целочислени задачи на многокритериалната оптимизация

Разработените досега интерактивни алгоритми за решаване на задачи на многокритериалната оптимизация се отличават със следните особености:

- всеки интерактивен метод е разработван за решаване на конкретен тип задачи – линейни, нелинейни, целочислени и т.н.;
- всеки интерактивен метод е построен на основата на определен тип скаларизираща задача и носи специфичните особености на тази задача, свързани както с възможностите за задаване на предпочитанията на ЛВР, така и с типа и времето за получаване на текущото предпочитано Парето-оптимално решение;
- всеки интерактивен метод е ориентиран към работа с един ЛВР (ако са повече, се предполага, че са с еднакви разбираня);
- по-голямата част от интерактивните методи са методи, ориентирани към обучение, и при тях ЛВР управлява процеса на решаване на многокритериалната задача;
- всеки интерактивен метод е в основата на разработена програмна система с общо предназначение за решаване на съответния клас многокритериални задачи и определя до голяма степен характеристиките на тази система.

Както беше показано, чрез промяна на съответни параметри на обобщените скаларизиращи задачи GENWS могат да се получат по-голяма част от известните

скаларизиращи задачи. Нещо повече – могат да се генерират и нови скаларизиращи задачи с определени свойства.

В тази точка ще бъде описан разработения на основата на скаларизиращите задачи GENWS и GENS обобщен интерактивен алгоритъм с променлива скаларизация и параметризация, който се отличава със следните характеристики:

- ЛБР може да задава своите предпочитания с помощта на тегла на критериите, на ε – ограничения, на желани и приемливи нива на промяна на стойностите на критериите, на желани и приемливи направления на промяна на стойностите на критериите, на желани и приемливи нива, посоки и интервали на промяна на стойностите на критериите и т.н.;
- В процеса на решаване на многокритериалните задачи ЛБР може да променя начина на задаване на своите предпочитания;

На дадена итерация, стартирайки от едно и също текущо Парето-оптимално решение и използвайки различни скаларизиращи задачи (чрез съответна промяна на параметрите на GENWS и GENS), ЛБР може да получи различни нови Парето-оптимални решения, като тази възможност е особено полезна при обучение и сравняване на различните скаларизиращи задачи.

В обобщения интерактивен алгоритъм, наречен GENS-IM, стъпките за задаване на предпочитанията на ЛБР и за оценката на получената информация са усложнени в зависимост от това какъв тип задача се решава, дали на една итерация се търси едно или повече решения, дали се решават реални задачи или ЛБР се обучава в спецификата на различни скаларизиращи задачи, като се търсят решения при едни и същи предпочитания на ЛБР, зададени по различен начин. Стъпката за решаване на еднокритериалната задача също се усложнява в зависимост от

вида на решаваната многокритериална задача (линейна, нелинейна, целочислена), в зависимост от броя и типа на получаваните решения (точни или приближени).

Основните стъпки на обобщения интерактивен алгоритъм са следните:

Стъпка 1.

Избира се по какъв начин ще бъдат задавани предпочитанията за намиране на следващи Парето-оптимални решения. На базата на този избор, се избира подходящ метод за решаване:

1.1. Избор на задаване на предпочитания чрез тегла

1.1.1. Използване на обобщен критерий под формата на претеглена сума на критериите. Преминаване към Стъпка 2.

1.1.2. Използване на обобщен критерий под формата на минимизация на разстоянието до идеалната точка. Преминаване към Стъпка 2.

1.2. Избор за оптимизация на един критерий, за сметка на останалите.
Преминаване към Стъпка 2.

1.3. Избор чрез задаване на отправна точка в критериалното пространство

1.3.1. Задаване на аспирационни нива. Преминаване към Стъпка 2.

1.3.2. Задаване на горни и долни граници. Преминаване към Стъпка 2.

1.3.3. Търсене на решение, максимално отдалечено от надир точката. Преминаване към Стъпка 2.

1.3.4. Задаване на отправни точки, нямащи отношение към текущото решение. Преминаване към Стъпка 2.

1.3.5. Задаване на оправни точки, имащи отношение към текущото решение. Преминаване към Стъпка 2.

1.4. Задаване на оправно направление.

1.4.1. Задаване на оправно направление, нямащо отношение към текущото решение. Преминаване към Стъпка 2.

1.4.2. Разделяне на критериите в 3 класа ($K^{\geq}$, $K^=$, $K^{\leq}$), взимайки предвид текущото решение. Преминаване към Стъпка 2.

1.5. Задаване на предпочитания чрез класифициране на критериите в един от няколко предварително дефинирани класа

1.5.1. Класифициране в 5 възможни класа:

- свободно подобряване
- подобряване до определено аспирационно ниво
- запазване или подобряване на текущата стойност
- влошаване до определено аспирационно ниво
- свободна промяна (без значение)

Преминаване към Стъпка 2.

1.5.2. Класифициране в 7 възможни класа:

- свободно подобряване
- подобряване до определена стойност или аспирационно ниво
- запазване или подобряване на текущата стойност
- влошаване без ограничение
- влошаване до определена стойност или аспирационно ниво
- промяна в определени интервали
- свободна промяна (без значение)

Преминаване към Стъпка 2.

Стъпка 2.

Задаване на предпочитания според избраните начини и метод, определени в стъпка 1.

Стъпка 3.

Намиране на Парето-оптимално решение.

Стъпка 4.

Оценка на текущото решение. Ако намереното решение удовлетворява ЛВР, тогава то се приема за финално и се преминава към Стъпка 5. В противен случай се преминава към Стъпка 2.

Стъпка 5.

Край.

В Стъпка 1 ЛВР определя какъв е типът на решаваната задача – линейна или нелинейна, непрекъсната или целочислена и т.н. За всяка една от участващите променливи се задават вид (целочислена/непрекъсната) и горна и долна граница. След Стъпка 1, за съответната задача, независимо от ЛВР, се намира начално Парето-оптимално решение, след което процесът продължава от Стъпка 4. Оценката се извършва в зависимост от целта на ЛВР – дали реално решава многокритериалната задача, дали се обучава или просто изследва различните скаларизиращи задачи.

ГЛАВА II. ПРОГРАМНА СИСТЕМА МКО-2.1

В главата са описани разработките, направени за целите на приложната СПВР МКО-2.1. Това включва: синтаксис за дефиниране на задачи за многокритериална оптимизация; управляващ модул; оптимизационен модул. Направено е и подробно описание на работата със системата. Резултатите са представени в публикации номер 7 и 8.

2.1. Предназначение на системата МКО-2.1.

Програмната система МКО-2.1 е предназначена за подпомагане решаването на линейни и линейни целочислени задачи на многокритериалната оптимизация. В системата на основата на обобщена скаларизираща задача се генерират 12 известни скаларизиращи задачи от различен тип и съответстващите им 12 интерактивни алгоритми. В зависимост от типа на информацията за предпочитанията на ЛВР при търсенето на нови решения тези алгоритми могат да бъдат отделени в пет групи (табл. 1). Системата се отличава с удобен потребителски интерфейс, свързан с въвеждането на входните данни, задаването в началото или на произволна итерация на различен тип информация за предпочитанията на ЛВР, избор на различни интерактивни алгоритми за решаване, извеждането в числов и графичен вид на получените изходни резултати, както и тяхното архивиране, възможностите за прекъсване на процеса на решаване на многокритериалната задача и стартирането му отново след време в друга посока и др.

<i>Тип на задаваната от ЛВР информация за неговите предпочитания</i>	<i>Интерактивен алгоритъм</i>
--	-------------------------------

Тегловни фактори (приоритети) на всеки критерий	1) Чебишевски алгоритъм 2) Тегловен алгоритъм.
Избор на един критерий за оптимизация и трансформация на останалите критерии в ограничения, задавайки долна граница на изменение за всеки от тях	1) Алгоритъм на ϵ – ограниченията
Аспирационни нива на всеки критерий	1) Алгоритъм STEM 2) Алгоритъм STOM 3) Алгоритъм на отправната точка RP 4) Алгоритъм GUESS 5) Алгоритъм на модифицираната отправна точка MRP
Аспирационни нива на всеки критерий и параметър за определяне на отправно направление	1) Алгоритъм на отправно направление VIG 2) Алгоритъм на отправно направление RD2
Желани и приемливи нива, посоки и интервали на изменение на част или всички критерии	1) Класификационно-ориентиран алгоритъм NIMBUS 2) Класификационно-ориентиран алгоритъм DALDI

Таблица 1. МКО-2.1 - Алгоритми за търсене на нови Парето-оптимални решения
спрямо начина на задаване на предпочитания.

2.2. Синтаксис за дефиниране на задачи за многокритериална оптимизация.

За представяне на данните в електронен вид на изчислителните модули в системата МКО-2.1., е необходимо те да бъдат транслирани от подходящ, четим за хора синтаксис за описание на такъв вид задачи. За целта е разработена граматика със съответния парсер, който приема описанието на задачата в текстов вид. Граматиката се състои от набор от компоненти и правила за тяхната подредба.

Компоненти на граматиката, дефиниращи граматическата азбука:

1. Литерал – за дефиниране на имена на критерий или променливи.
2. Оператор за вид на критерия – min; max. Определят дали критерия ще се минимизира или максимизира.
3. Оператори за сравнение: =; <; >; <=; >=.
4. Математически оператори: +; -; *.
5. Двойка коефициент – променлива.
6. Цели и реални числа.

Правила на граматиката:

Правила, касаещи литералите за означаване на имена на критерии и променливи:

1. Литералите започват винаги със символ, различен от цифра.
2. Литералите не могат да съдържат празни символи.
3. Няма ограничение в използваната азбука за литерал.
4. Няма ограничение в дължината на литерала.

Двойките коефициент-променлива се подчиняват на следните правила:

1. Коефициентът представлява положително или отрицателно число пред променливата, играещо ролята на математически множител. Положителните числа се обозначават като се префиксират със знак „+“ (незадължителен), а отрицателните със знак „-“.
2. Положителния знак „+“ не е задължителен и може да се пропусне.
3. Когато знакът „+“ е пропуснат, между литерала от предхождащата двойка и коефициента, трябва да има поне един празен символ.
4. Когато коефициентът е равен на 1, може да се пропусне.
5. Между коефициент и променлива стои оператор за математическа операция умножение. Той може да се пропусне изцяло или да съществува между коефициента и променливата, с или без празни символи между тях.
6. Не може да съществува свободен коефициент без променлива. Винаги трябва да се дефинира двойка коефициент-литерал.

Правила на граматиката за дефиниране на критерий:

1. Всеки критерий се дефинира на отделен ред.
2. Критерий се описва в следния ред:
3. Литерал за име на критерия.
4. Оператор „=“
5. Оператор за определяне вида на критерия (min/max).
6. Празен символ
7. Низ от двойки коефициент-променлива.

Следват примери за валидно дефиниран критерий. Всичките примери имат едно и също математическо значение:

$$F1 = \max 1 \ x1 \ 2 \ x2 \ -1 \ x3$$

$$F1 = \max x1 \ 2x2 \ -x3$$

$$F1 = \max x1 + 2*x2 -1*x3$$

$$F1 = \max x1 \ 2 * x2 -x3$$

$$F1 = \max x1+2x2-x3$$

Правила на граматиката за дефиниране на ограничения:

1. Всяко ограничение се дефинират на отделен ред.
2. Ограниченията се дефинират в следния ред:
3. Низ от двойки коефициент – променлива
4. Оператор за сравнение: =; <; >; <=; >=
5. Число

Следват примери за валидно дефиниране на ограничение. Всичките примери имат едно и също математическо значение:

$$2x1 -3x2 \ 1 \ x4 \ <= \ 5$$

$$2*x1 -3*x2 +x4 \ <= \ 5$$

$$2 \ x1 \ -3 \ x2 \ x4<=5$$

Парсерът, разработен за транслиране на тази граматика представлява линеен детерминиран автомат с фиксирани състояния.

Автоматът дефинира следните състояния и подсъстояния в точна последователност:

2. Четене на критерий.

- Четене на литерал име на критерий.
- Четене на оператор за вида на критерия. Четене на двойка коефициент-променлива.
- Четене на знак за положително/отрицателно число.
- Четене на число.
- Четене на литерал – име на променлива.

3. Четене на ограничение.

- Четене на знак за положително/отрицателно число.
- Четене на число.
- Четене на литерал – име на променлива.
- Четене на оператор за сравнение: =; <; >; <=; >=
- Четене на число

Така проектирания синтаксис, заедно с реализирания парсер, представляват основна част от изчислителния модул на системата МКО-2.1.

2.3. Основни модули в системата МКО-2.1.

Системата МКО-2.1 се състои от три основни групи модули: управляващ модул, оптимизационни модули и интерфейсни модули.

Управляващият модул е интегрирана софтуерна среда за създаване, обработка и съхранение на асоциирани със системата файлове (с разширение “*.mlp”), както и за свързване и изпълнение на различни типове програмни модули.

Основните функционални възможности на управляващия модул могат да бъдат отделени в три групи. Първата група включва възможностите за използване на стандартните

за MS Windows приложения, менюта и системни функции - Файл, Настройки, Редакция, Изглед, Прозорци, Помощ и др., в собствената среда на системата.

Втората група от функционални възможности включва управлението на взаимодействието между модулите, реализиращи:

- създаване, модификация и съхранение на асоциираните със системата файлове “*.mlp”, съдържащи входни данни и данни за междинните и крайните резултати от решаването на линейни и линейни целочислени задачи на многокритериалната оптимизация
- интерактивно решаване на въведените задачи на многокритериалната оптимизация
- локализация и идентификация на грешките, възникнали в процеса на работа със системата МКО-2.1

Третата група от функционални възможности на управляващия модул включва възможностите за визуализация на съществена информация както за ЛВР, така и за работа на системата МКО-2.1 като цяло.

Управляващият модул е разработен в стил Multy Document Interface. Има една главна форма (прозорец), съдържаща главното меню, а всички други форми (прозорци) са нейни “дъщерни” форми. По едно и също време могат да се визуализират няколко “дъщерни” форми едновременно.

За осъществяване на комуникация с библиотеките за динамично свързване на различните еднокритериални оптимизационни модули се използва специално разработен програмен Решател, който е процесор за обработка на данни. Той от своя страна, съдържа модули за подаване на входни данни, за получаване на изходни данни и за извършване на съответните настройки за типа на променливите и типа на решението, което се изисква. Решателят работи на принципа на черна кутия – на входа, като параметър, се подава

оптимизационната задача в определен формат и синтаксис, извършва се синтактичен анализ, определят се имената и типът на променливите (непрекъснати/целочислени), определя се видът на решението (непрекъснато/целочислено) и оптимизационната задача се трансформира във формат, изискван от съответния еднокритериален оптимизационен модул. Информацията за получените решения се записва в изходните точки на класа, откъдето тя може да бъде получена.

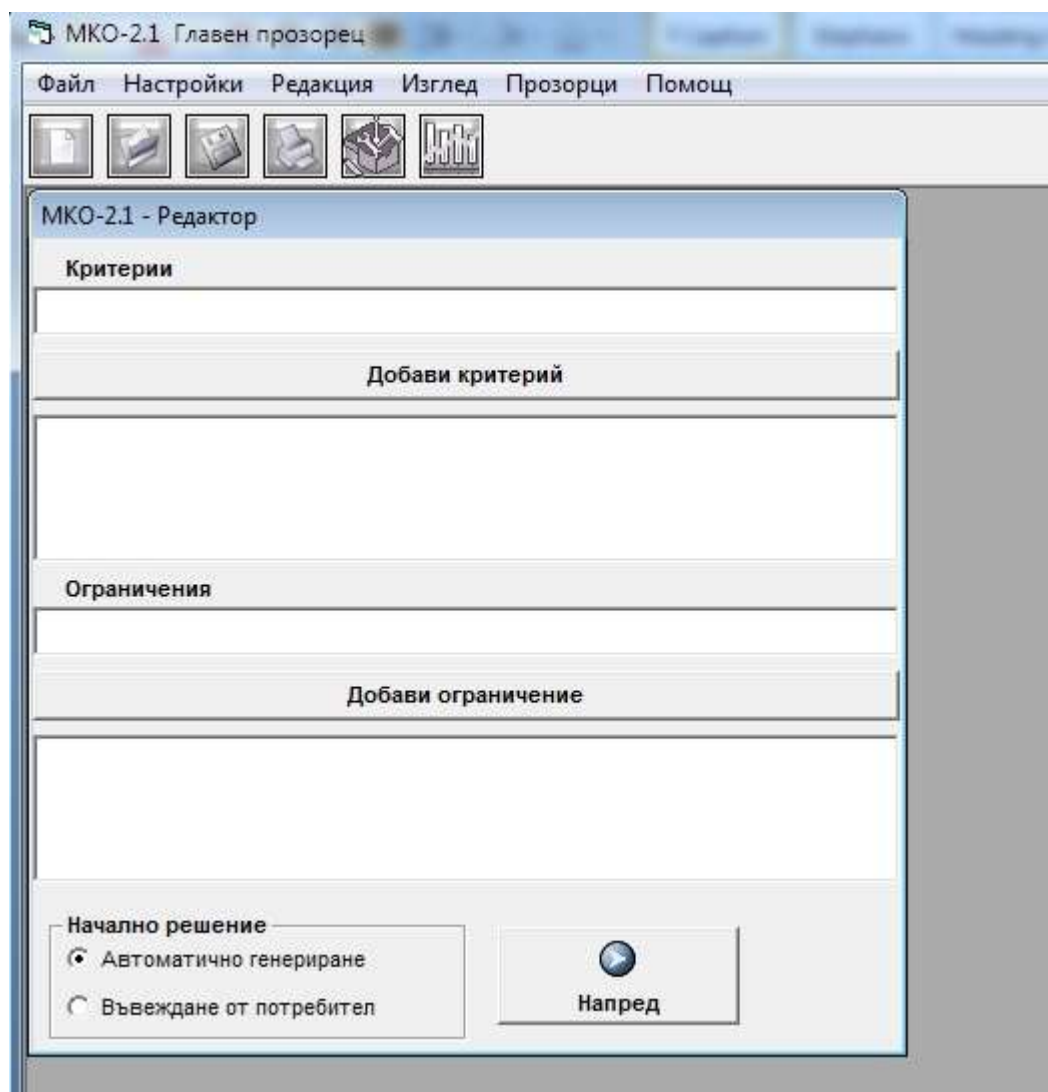
Другият основен компонент, който се използва от управляващия модул, е подмодулът за управление на историческите данни. Неговата цел е да се създаде структура, в която да се съхраняват получените резултати на всяка итерация от интерактивния алгоритъм – стойностите на критерии и променливи, зададените от ЛВР предпочитания и използваната на тази итерация скаларизираща задача. Това позволява интерактивният процес на решаване на всяка задача на многокритериалната оптимизация да бъде прекъсван и стартиран отново от мястото на прекъсване. ЛВР има възможност, също така, да се връща назад и да търси решения в друга посока и с други предпочитания.

Интерфейсните модули осигуряват диалога между ЛВР и системата при въвеждането и корекцията на входните данни на решаваната задача, по време на интерактивния процес на решаването ѝ и за динамична числова и графична визуализация на основните параметри на този процес. Чрез редактиращ модул се въвеждат, променят и запазват описанията на критериите, ограниченията, както и типът и границите на изменение на променливите. Чрез друг интерфейсен модул се реализират два вида графично представяне на информация за стойностите на критериите на различни стъпки и възможностите за сравнение. Осигурена е и динамична информация за предназначението и начина на използване на полетата и радио бутоните.

Оптимизационните модули реализират 12 интерактивни алгоритъма на многокритериалната оптимизация, както и два алгоритъма на линейната и линейната целочислена еднокритериална оптимизация.

2.4. Работа със системата МКО-2.1.

Системата МКО-2.1 работи под операционна система MS Windows. При стартирането се отваря прозорец “МКО-2.1 Главен прозорец” (фиг. 1), съдържащ лента с шест главни менюта – Файл, Настройки, Редакция, Изглед, Прозорци и Помощ и втора лента с икони за бърз достъп – Нов, Отваряне, Съхраняване, Печат, Настройки и Графика.



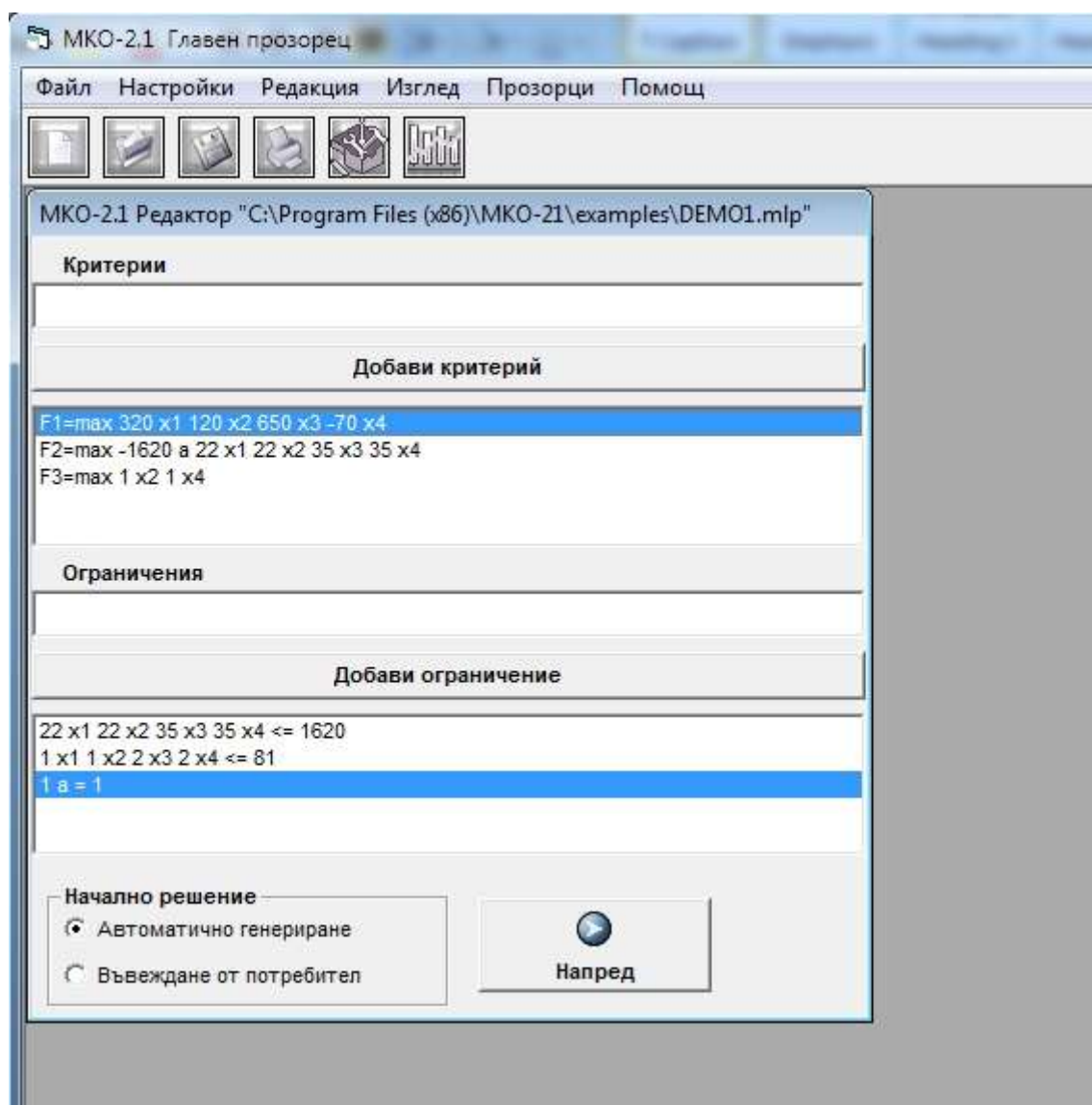
Фигура 1. МКО-2.1 – Основен прозорец.

С помощта на команда “Нов” се отваря прозорец за въвеждане на нова задача, данните за която се записват от системата във файл с разширение “*.mlp”. Записът се осъществява с

командите “Съхраняване” или “Съхраняване като”. Ако данните за задачата, която този файл съдържа, не са изцяло въведени или не е стартиран процесът за решаване на тази задача, с помощта на команда “Отваряне” се отваря прозорецът “МКО-2.1 Редактор”. В противен случай се отваря прозорец “МКО-2.1 Решаване”.

2.4.1. Въвеждане на задача

Въвеждането и корекцията на критериите и ограниченията на задачата се извършва в две отделни полета на прозореца “МКО-2.1 Редактор” (фиг. 2).

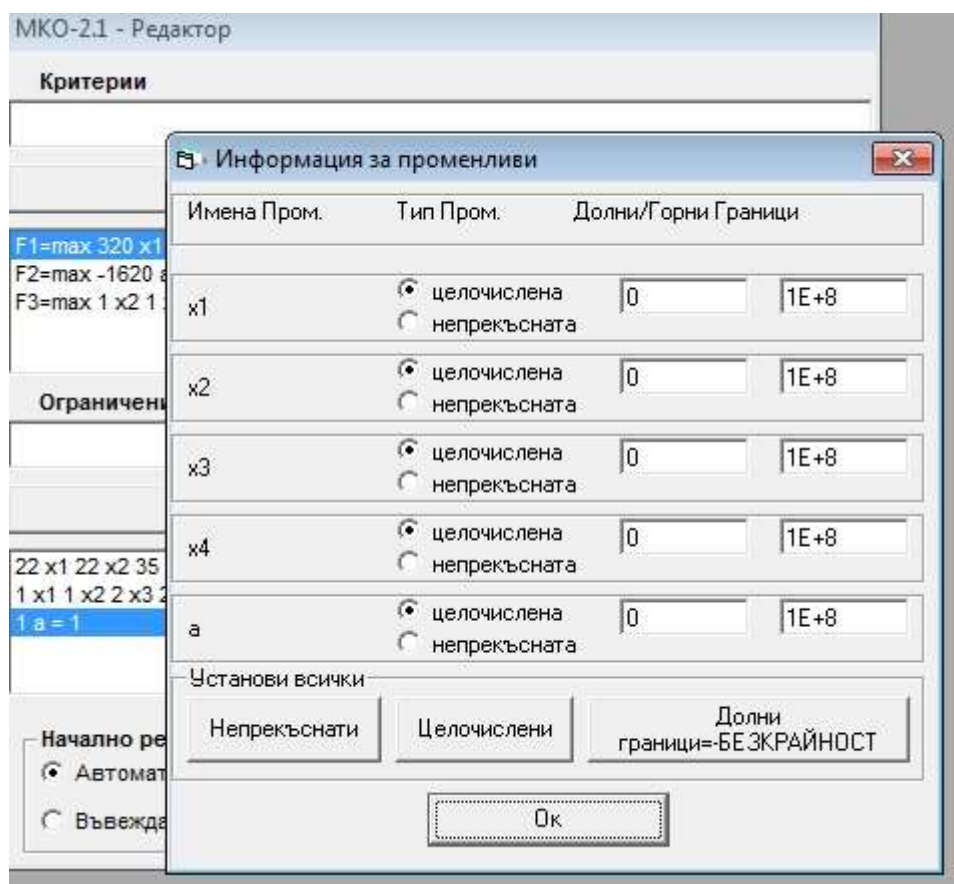


Фигура 2. MKO-2.1. - Редактор.

Всеки критерий и всяко ограничение се въвеждат поотделно в съответните текстови полета за въвеждане и редактиране. При добавянето им към вече въведените критерии или ограничения се извършва синтактична проверка. Приетият синтаксис е подобен на математическия запис на този клас оптимизационни задачи. За всеки критерий първо се записва името на критерия, последвано от знак за равенство. Ако името на критерия не се зададе, за име на критерия автоматично се присвоява името F_i , където индексът i е число, показващо последователността на въвеждане на критериите. След това се записва типът на търсения оптимум – “min” или “max”. По-нататък се въвежда числовият коефициент със знака си, следван от името на променливата, за която се отнася. Имената на променливите могат да бъдат произволен набор от букви и цифри. Всеки един от тези елементи е отделен с интервал. Ограниченията имат подобен синтаксис – въвеждат се последователно числови коефициенти и имена на променливи. Типът на ограниченията се задава с някой от знаците “<”, “<=” или “=”. С двойно кликане върху вече въведено ограничение или критерий, те отново се прехвърлят в полето за редактиране, ако се налагат следващи корекции.

За генериране на начално допустимо решение на ЛВР се предоставят две възможности. От групата радио бутони „Начално Решение” се активира „Автоматично генериране” или „Въвеждане от потребител”. Следващата стъпка от работата със системата е да се укаже как ще бъдат задавани предпочитанията на ЛВР в процеса на решаване на задачата.

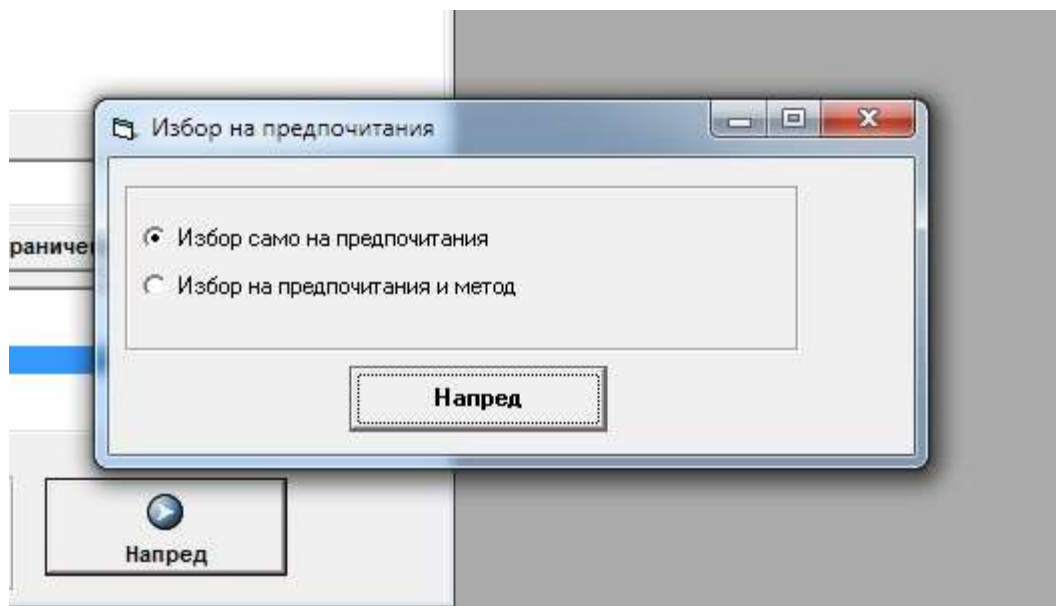
Чрез бутона „Напред“ се отваря прозорец “Информация за променливи” (фиг. 3), в който се задава информация за типа и границите на изменение на променливите.



Фигура 3. МКО-2.1. - Задаване интервали на валидни стойности за променливите.

По подразбиране всички променливи са от тип Целочислени, с Долни граници = 0 и Горни граници = $1E+24$, което е прието за безкрайност. С двата бутона Непрекъснати и Долни граници = - БЕЗКРАЙНОСТ може автоматично да се измени информацията за всички променливи. Затварянето на прозореца “Информация за променливи” и съхранението на направените корекции се извършва след натискане на бутон **Ок**, след което се преминава към прозореца “МКО-2.1 Решаване”.

За генериране на начално допустимо решение на ЛВР се предоставят две възможности: „Автоматично генериране” или „Въвеждане от потребител” (фиг. 2). От групата радио бутони „Начално Решение” се активира интерфейс за указване как ще бъдат задавани предпочитанията на ЛВР в процеса на решаване на задачата (фиг.4).

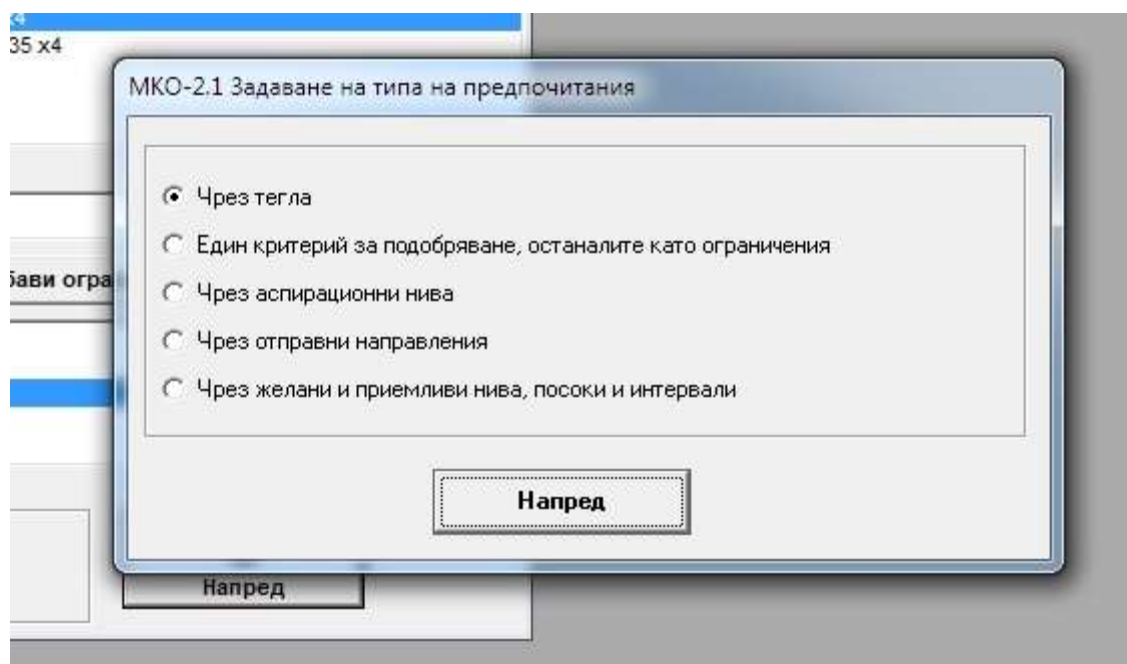


Фигура 4. МКО-2.1. - Избор на начин на задаване на предпочитания.

Тук системата предоставя изключително разнообразни възможности относно начините на задаване на предпочитания. Това прави системата лесна и подходяща за използване както за обучение, така и за решаване на задачи от експерти с различни нива на познания в областта, а също така и за такива потребители със съвсем бегли и общи познания на методиката.

Начините на задаване на предпочитания са разделени на две основни групи – „Избор само на предпочитания” и „Избор на предпочитания и метод”.

Опцията „Избор само на предпочитания” предполага използване на системата от потребители с не толкова задълбочени познания на методиката, като им се дава възможност само да избират по какъв начин ще задават своите предпочитания (Фиг. 5).

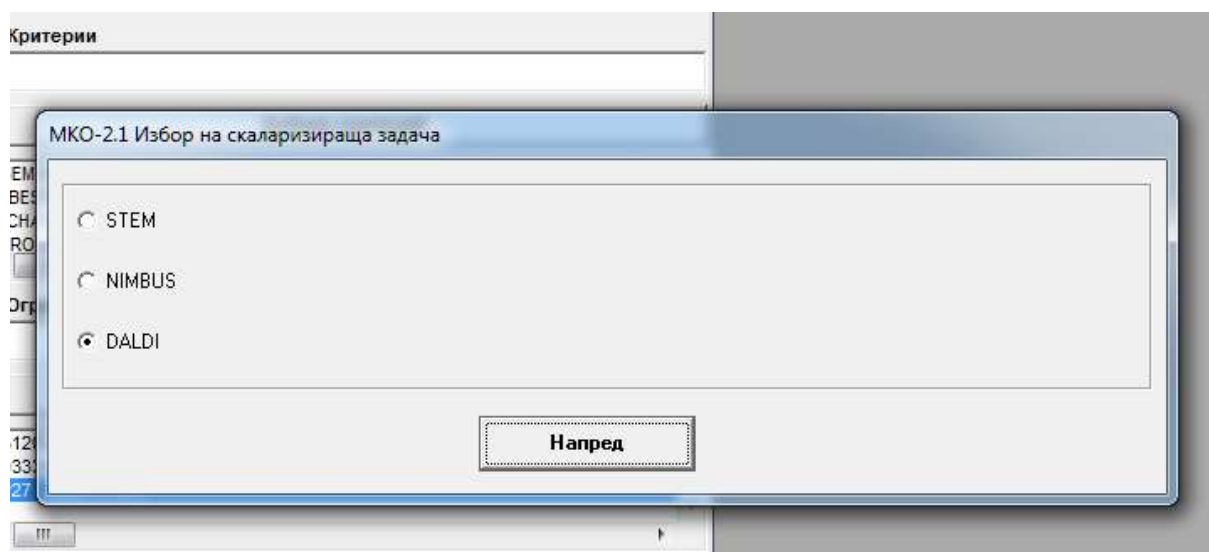


Фигура 5. МКО-2.1. - Избор на интерактивен начин за задаване на предпочитания за намиране на следващо Парето-оптимално решение.

Възможностите са пет:

- чрез задаване на тегла на критериите, като по този начин се указва тяхната относителна важност
- чрез задаване на един критерий за подобряване, като останалите се трансформират в ограничения
- чрез аспирационни нива (задават се горни и долни граници, в които стойностите на критериите могат да се променят)
- чрез отправни направления – указва се за всеки критерий как да се промени неговата стойност, като възможностите за избор са подобряване; влошаване; запазване на текуща стойност
- чрез задаване на желани и приемливи нива, посоки и интервали

При избиране на опцията „Избор на предпочитания и метод”, освен начина на задаване на предпочитания, вече може да се избира и конкретен метод, което прави опцията подходяща за използване от потребители на вече по-високо експертно ниво (фиг. 6).

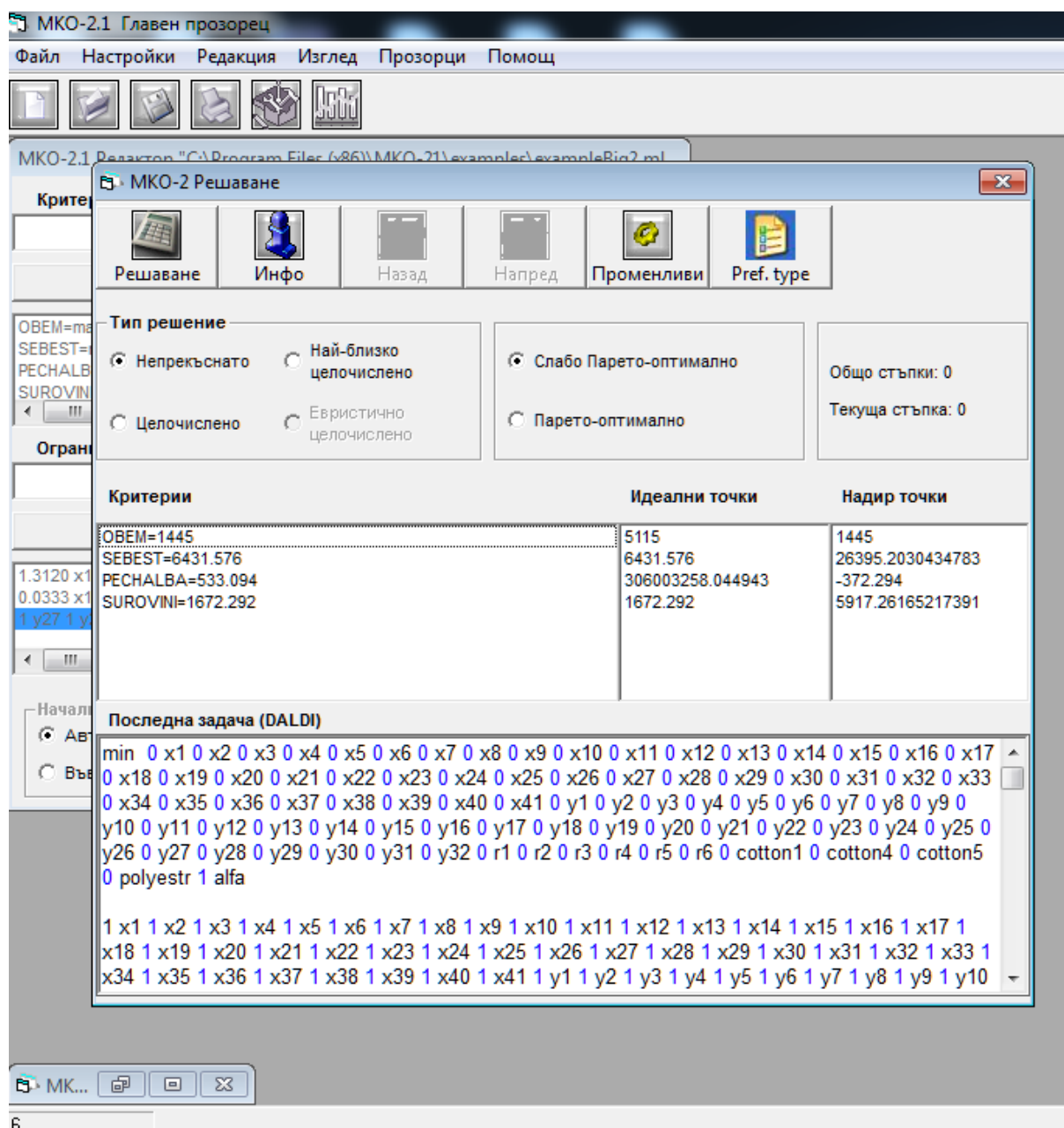


Фигура 6. МКО-2.1. – Задаване на скаларизираща задача според избрания начин на задаване на предпочитания.

Ако е активиран бутонът за автоматично генериране с бутон „Напред” се отваря основния прозорец “МКО-2.1 Решаване” и се представя генерираното непрекъснато начално решение. Ако е активиран радио бутонът Въвеждане от потребител с натискане на Напред се отваря прозорец за задаване на началните стойности на критериите и с бутон Напред тези стойности се съхраняват, след което се преминава в прозореца “МКО-2 Решаване”.

2.4.2. Решаване на задачата за многокритериална оптимизация

Решаването на линейни и линейни целочислени многокритериални задачи се осъществява чрез 12 помощни прозореца в интерфейсия модул “МКО-2.1 Решаване” (фиг. 7). Всеки от тях е ориентиран за работа с един от 12-те интерактивни алгоритъма.

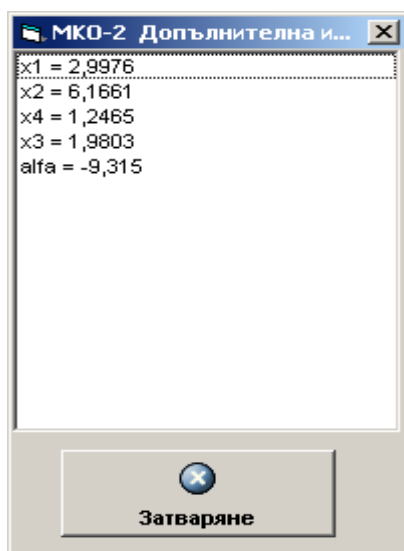


Фигура 7. MKO-2.1. – Решаване на задачата.

В горната част на основния прозорец за решаване е разположена лентата с бутоните, с които се реализират основните функции на процеса на интерактивно решаване на линейни и линейни целочислени задачи на многокритериалната оптимизация. Това са бутоните:

7. *Решаване* - за стартиране на оптимизационни модули на системата MKO-2.1 за намиране на ново текущо решение, чрез решаване на генерираната на тази итерация скаларизираща задача

8. *Инфо* - за визуализация в отделен прозорец на стойностите на променливите за текущото решение (фиг. 8);
9. *Назад / Напред* - бутони за навигация; те предоставят на ЛВР възможност да се връща назад към предишни итерации или да отива напред към следващи итерации и да разглежда намерените текущи решения. Ако ЛВР прецени, той може да промени зададените от него предпочитания за промяна на критериите на някоя от итерациите и оттам да стартира процесът на търсене на по-добро решение;
10. *Променливи* - за отваряне на прозорец, в който е зададена информация за типа и границите на изменения на променливите (фиг. 8).



Фигура 8. МКО-2.1. – Стойности на променливите за текущото решение.

11. *Тип на предпочитания* - за определяне на типа на предпочитания на ЛВР при търсенето на нови решения (фиг. 4) и за избор на алгоритъм, ориентиран към такъв тип предпочитания (фиг. 5). Ако ЛВР среща затруднения с избора на алгоритъм, автоматично се избира алгоритъм, представителен за дадена група от алгоритми. Изборът от ЛВР на типа на неговите предпочитания и на съответния алгоритъм се извършва на първата итерация. Изборът може да се променя на всяка итерация или през произволен брой итерации.

Под лентата с бутоните на прозореца “МКО-2.1. Решаване” са разположени радио бутони за избор на типа на търсеното решение: непрекъснато, целочислено, най-близко целочислено, както и слабо Парето-оптимално (слабо недоминирано) или Парето-оптимално (недоминирано) решение. Под тях се предоставя информация за номера на текущо разглежданата итерация и за общия брой на изпълнените итерации.

Следват две текстови полета. В първото са изведени последователно стойностите на критериите, получени на текущата итерация. То е и работното поле за задаване на предпочитанията на ЛВР за търсене на следващо решение. При маркиране на всеки един от критериите, с десния бутон на мишката се отваря контекстно меню, от което ЛВР избира желаната промяна на стойността на този критерий за следващата итерация. Ако изборът е свързан с необходимостта от въвеждане на конкретна стойност, то системата МКО-2.1 отваря допълнителен диалогов прозорец и изчаква въвеждането на съответната числова информация от ЛВР.

Във второто текстово поле се визуализира скаларизиращата задача, при решаването на която е намерено съответното решение на текущата итерация. Тази информация би била полезна за изучаващите и работещите в областта на многокритериалната оптимизация.

За останалите 11 алгоритъма тези две текстови полета имат подобен вид.

При решаването на задачи на многокритериалната оптимизация е важно да се предостави не само информация за последното намерено решение, но и такава за намерените решения в предишните итерации. За ЛВР е важно да може да “удостовери” как е стигнал до крайното решение. Ето защо, информацията за интерактивния процес на решаване на конкретна задача на многокритериалната оптимизация, която се състои от входните данни на задачата, получените на всяка итерация решения, зададените от ЛВР предпочитания за ново търсене и конструирания скаларизиращи задачи (записани в асоциираните със системата МКО-2.1. файлове (“*.mlp”)) служат не само за рестартиране на процеса на решаване (в случай, че е прекъснат) на задачата, но и за документиране на този процес.

2.4.3. Настройки

Системните настройки на системата МКО-2.1. могат да бъдат променени с помощта на меню “Настройки”, съдържащо три команди – , “Език”, “Глобални променливи” и “Файлови асоциации”.

Език: Интерфейсът на системата може да се използва на два езика - български и английски. Избраният език се указва чрез съответния радио-бутон.

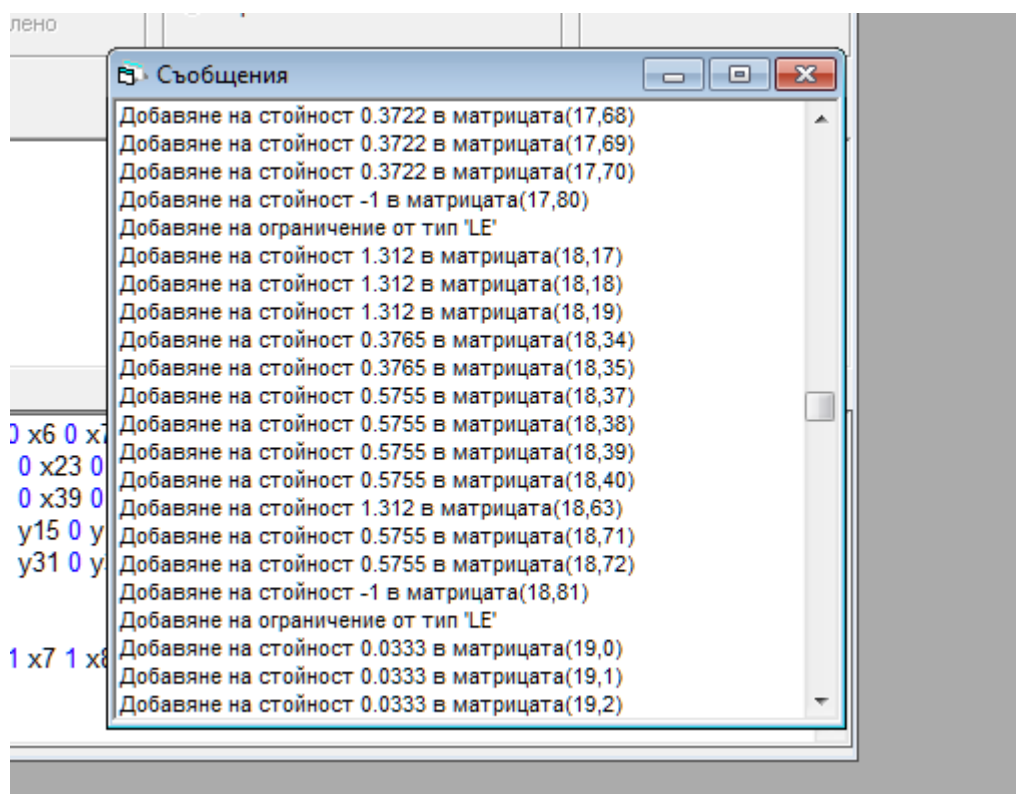
Глобални променливи: служебни променливи, използвани от системата. Тяхната промяна изисква много добро познаване на използваните алгоритми.

Файлови асоциации: с натискане на бутон Асоцииране на работния файл се присвоява разширение “.mlp”.

Изглед: с помощта на меню “Изглед” може да се визуализира допълнителна информация, свързана с решаваната задача.

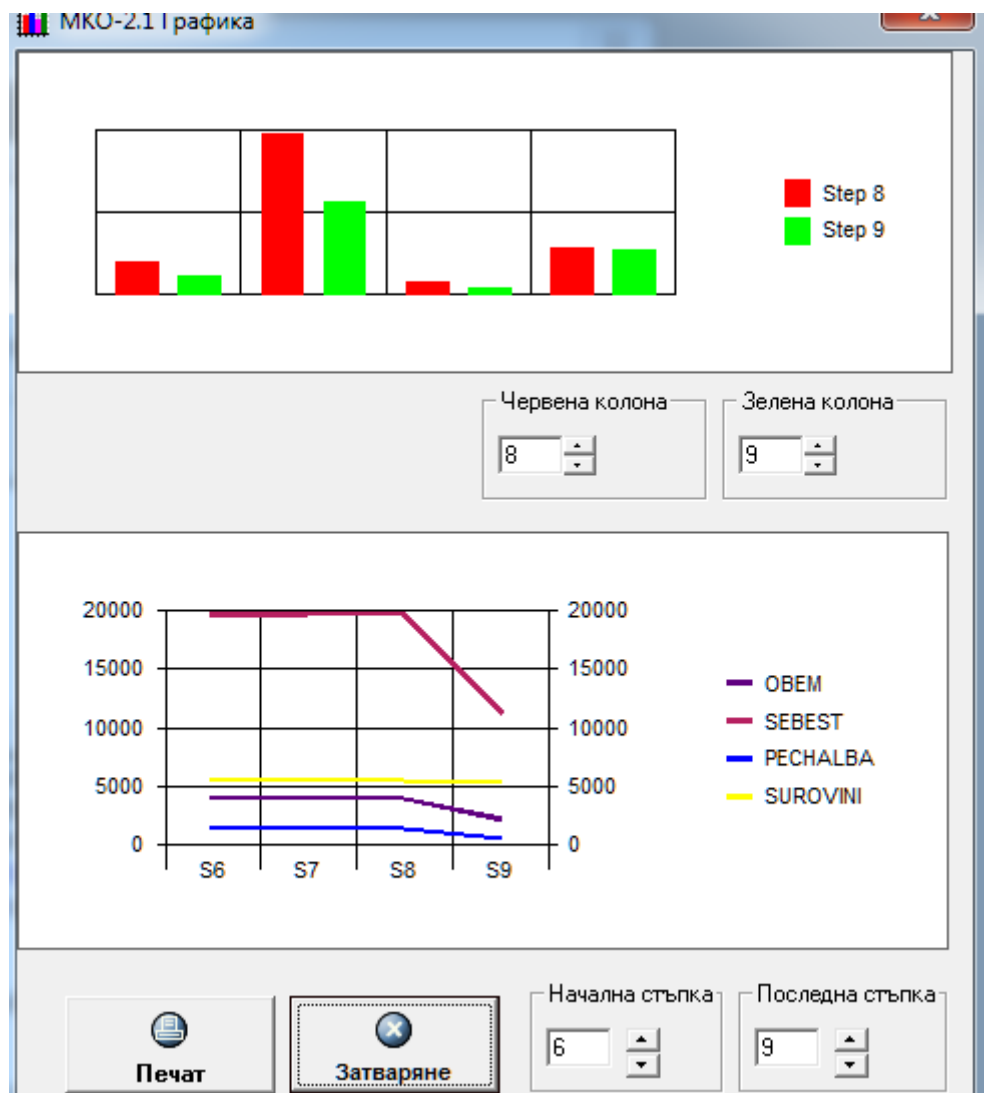
Съветник: с команда “Съветник” се предоставя помощна информация за визуалните компоненти на системата МКО-2.1. След избирането на тази опция се отваря прозорец “Прозорец на съветника” и при преместване на курсора върху някой визуален елемент на системата в този прозорец се появява пояснителен текст.

Съобщения: след избирането на команда “Съобщения” се отваря прозорец “Съобщения” (фиг. 9). Всички модули на системата МКО-2.1. извеждат служебна информация за работата си в този прозорец. Чрез контекстно меню съдържанието на прозореца може да бъде изчистено или съхранено във файл.



Фигура 9. МКО-2.1. – Съобщения относно процеса на решаване.

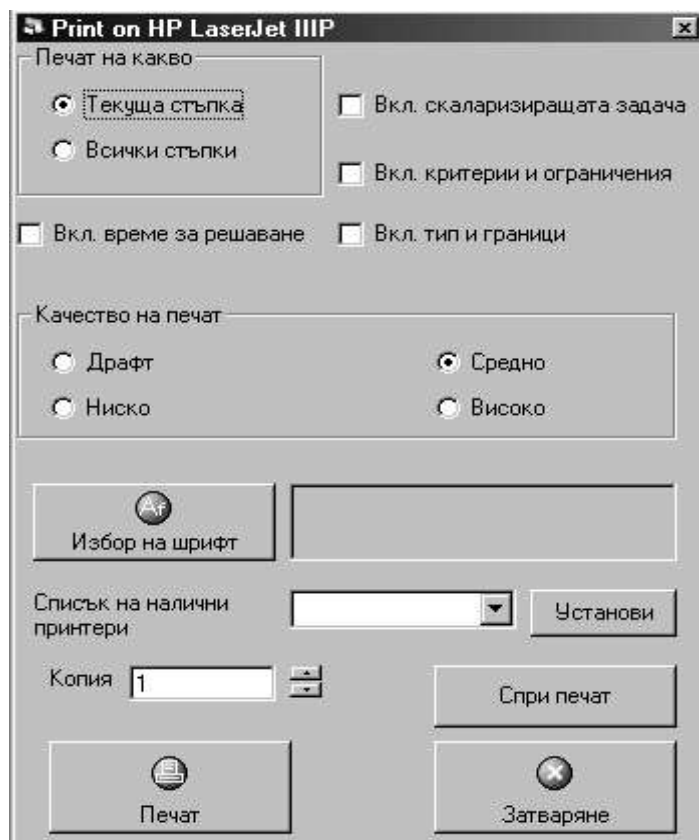
Графика: избирането на команда “Графика” позволява да се визуализират два типа графична информация за процеса на решаването на задачата. За целта се отваря прозорец с два типа графики (фиг. 10). Чрез горната бар – графика може да се направи визуално сравнение на решенията, намерени на две итерации, избрани чрез полетата под нея за избор на итерации. Чрез долната графика може визуално да се проследи промяната на стойностите на отделните критерии на различни итерации от интерактивния процес на търсене на по-добро решение. Чрез бутоните за избор се определя начална и крайна итерация на интервала от итерации, в който се проследяват стойностите на всички критерии.



Фигура 10. MKO-2.1. – Графика за визуализация на процеса на решаване.

Помощ: след избирането на меню “Помощ” може да се отвори прозорец на Интернет браузър, в който се показва “Ръководството на потребителя” или може да се получи информация за разработчиците на системата.

Печат: отпечатване на кратка или подробна информация за текущо решаваната задача се извършва с помощта на команда “Печат”. Отваря се форма за избор на информация за отпечатване (фиг. 11).



Фигура 11. МКО-2.1. – Печат.

Може да се избира отпечатване на информация за текущата стъпка или за всички междинни стъпки на процеса на решаване на многокритериалната задача; може да се отпечатва и информация за отделните скаларизиращи задачи, за критериите, за ограниченията, за типа и вида на променливите, както и за крайното време за решаване на задачата. Могат да се задават и изисквания за качеството на отпечатване, за броя на копията и за вида на шрифта.

ГЛАВА III. ПРОГРАМНА СИСТЕМА WEBOPTIM

В главата са описани разработките направени за целите на уеб базираната СПВР WebOptim. Това включва: цялостната софтуерна архитектура на системата; архитектура на базата данни; интерфейсни модули; управляващ модул; модул сигурност и управление на потребителите; модул за управление и поддръжка на изчислителните подмодули (решатели); междинна система за вътрешномодулна комуникация; публичен API модул за връзка и обмен на данни с външни системи. Направено е и подробно описание на работата със системата. Резултатите са представени в публикации с номера 2, 4 и 6.

3.1. Предназначение на системата WebOptim

Програмната система WebOptim е естествен наследник на системата МКО-2.1 и е създадена с идеята да бъде съвременна, изцяло уеб базирана и по този начин да предостави лесен и свободен достъп до възможно най-голям брой потребители. Другите две основни цели при проектирането на системата са да може лесно да се добавят нови изчислителни модули, имплементиращи различни алгоритми и осигуряване на връзка за обмен на данни с други независими външни системи. Това стана възможно чрез използването на т.нар. уеб услуги (web services) и технологии като SOAP, XML и WCF.

3.2. Структура на системата WebOptim

Системата WebOptim е уеб-базирана интерактивна софтуерна система за решаване на задачи за еднокритериална и многокритериална оптимизация. Изградена е на модулен принцип и позволява както надграждане на самата нея с други различни модули за решаване (решатели), така и комуникация с външни софтуерни системи чрез универсален публичен модул за комуникация – API. Както беше споменато в увода - основния недостатък на уеб-

базираните архитектури е невъзможността за установяване на устойчива комуникационна връзка между клиента и сървърното приложение. Това от своя страна налага и съответни ограничения върху обема на трансферираната информация и времето за отговор на дадена заявка между клиента и сървъра. Времето, необходимо за решаване на една задача за многокритериална оптимизация е невъзможно да бъде предварително определено, тъй като в зависимост от конкретната задача, това е свързано с огромни по обем математически изчисления и многобройни итерации. Тези два факта налагат в системата WebOptim да бъде използвана друг вид клиент-сървър комуникация, базирана на протокол за т. нар. уеб услуги (web services). За осигуряване на тази комуникация е разработен специален модул, представляващ междинна система за вътрешно-модулна комуникация.

Системата WebOptim е изградена на модулен принцип и съдържа следните основни модули:

- База данни
- Интерфейсни модули
- Модул сигурност и управление на потребителите
- Изчислителни подмодули (решатели)
- Модул за управление и поддръжка на изчислителни подмодули (решатели)
- Междинна система за вътрешно-модулна комуникация.
- Публичен API модул за външна връзка с други независими софтуерни системи.

Системата е разработена чрез използване изцяло на Майкрософт технологии:

- MS SQL Server
- MS.NET framework
- MS Visual Studio version 10

За съхранението на цялата информация относно потребителите, решаваните задачи, метаданните касаещи различните методи и решатели се използва релационна база данни, базирана на MS SQL Server.

3.3. Основни компоненти в WebOptim

Модул управление на потребители и сигурност

WebOptim е изградена така, че се изисква предварителна регистрация на потребителя за да се позволи по-нататъшна работа. По този начин се поддържа персонален профил, с който се асоциират въведените за решаване задачи, решения и допълнителна информация, касаеща възникнали грешки в процеса на решаване.

Регистрацията става чрез стандартен уеб интерфейс, който е добре познат на хората използващи активно Интернет.

След регистрация и вход в персоналния профил, на потребителите са предоставен два отделни интерфейса за решаване на еднокритериални или многокритериални задачи за оптимизация.

Модул за управление и поддръжка на решаващи подмодули (решатели)

Имайки предвид, че съществуват много алгоритми и софтуерни продукти с отворен код или комерсиални такива за решаване на задачи за оптимизация, WebOptim е така проектирана, че да позволява лесно интегриране на решаващи модули (решатели), написани на различни програмни езици и чрез използването на различни технологии за програмиране. Решателите могат както да бъдат инсталирани локално на същата машина, където се намира системата, също така и на физически съвсем различна отдалечена машина. Това става възможно чрез използването на добре познатите в наши стандарти за уеб услуги и дистанционен обмен на данни като XML, WSDL, SOAP.

За да бъде интегриран със системата WebOptim, всеки един изчислителен модул (решател), трябва да имплементира и поддържа два вида комуникации базирани на протокол за уеб услуги (web services) – входяща и изходяща.

Точката за входяща комуникация представлява уеб адрес, на който работи уеб услугата и където модулът за управление подава към решателя задачите за решаване.

Изходящата комуникация от страна на решателя, представлява изпращане на готовото решение обратно към модула за управление, отново чрез уеб услуга на предварително зададен уеб адрес.

Всеки изчислителен модул (решател) се характеризира със следните специфични параметри:

- Входна точка на уеб услугата (Web Service end point). Представлява уеб адрес, на който се подават задачите за решаване чрез SOAP протокол за комуникация
- Версия на модула
- Кратко текстово описание на решателя
- Списък с поддържаните методи за решаване
- Списък с поддържаните видове синтаксиси за описание на задачи (LP, AMPL, MPS, LINDO и др.)
- Други специфични параметри, необходими за работата на конкретния решател.

Съществуват много разновидности на синтаксиси на езици за описание на задачи за оптимизация, адаптирани за машинна обработка. Задачите се подават на съответните решатели, описани на някои от тези синтаксиси във вид на текстови файлове. Основният решаващ модул в системата е базиран на програмната библиотека с отворен код LPSOLVE, която поддържа следните синтаксиси:

- MPS – Синтаксис, поддържан от повечето решатели за линейна оптимизация, което го прави доста универсален. Този формат обаче е доста стар и труден за разчитане от хора.
- LP - Този синтаксис е разработен специално за библиотеката LPSOLVE. Интуитивен и удобен за разчитане от хора и е много близък до математическата формулировка на задачата.
- CPLEX – Също удобен за разчитане от хора и доста близък до математическата формулировка на задачата.
- LINDO – Синтаксис, специално разработен от създателите на добре познатите софтуерни продукти за линейна оптимизация Lindo Systems Inc за целите на техните продукти.
- GNU MathProg – Синтаксис за описание на модели за линейна оптимизация, лесно четим от хора и близък до математическата формулировка на задачите.
- LPFML/OSIL XML – Представява формат, описващ задачите във вид на XML файл. Един от най-новите формати за описание на такива модели задачи, но не е много удобен за четене и създаване от хора

Междинна система за вътрешно-модулна комуникация

WebOptim е замислена така, че да бъде максимално лесно разширяема и допълвана във всеки един аспект и по всяко едно време. Нови решатели, методи и алгоритми във вид на модули могат да бъдат добавяни изключително лесно. Както беше споменато, това се постига чрез изключителното използване на добре утвърдили се стандарт описание на данни - XML и т. нар. веб услуги.

Основния проблем при реализацията на системата е, че затворения цикъл на комуникацията клиент->сървър->решател->сървър->клиент трябва да се извършва

асинхронно, главно поради натурата на протоколите HTTP и SOAP, които имат много ограничена времева рамка за процеса на фактическия обмен на данни между сървъра и клиентското приложение. Това прави невъзможно този цикъл да бъде затворен и да се извърши наведнъж без прекъсване.

Този проблем в WebOptim е решен, като е добавен междинна система, която се грижи за осигуряване на комуникацията между потребителския интерфейс, базата данни и решателите.

Системата за вътрешно-модулна има две основни задачи – да изпраща заявки за решаване на задачи към решателите и да приема от тях готовите решения.

Изпращането на задача става чрез унифицирана уеб услуга чрез XML SOAP протокол, съдържащ следните полета:

- ID номер на проблема – уникално число, чрез което еднозначно да бъде разпозната конкретната задача в системата.
- Синтаксис, който е използван за описание на задачата
- Описание на задачата
- Допълнителни незадължителни параметри (специфични за всеки отделен решател)

При успешно приемане на задачата, решателят връща утвърдителен отговор и задачата бива маркирана със статус *„Изчакване на решение”*.

На следващия етап, междинната система очаква отговор от решателя относно решението на задачата, пак на уеб адрес, където е имплементирана уеб услуга за получаване на отговори от страна на междинната система.

Параметрите, които се съдържат в този отговор са следните:

- ID номер на проблема – уникално число, чрез което еднозначно да

- Решение (в случай на успешно намерено такова)
- Съобщение за грешка (в случай на грешка)
- Допълнителна информация

Когато се получи такъв отговор, модулът претърсва базата данни за задачата със съответния ID номер, записва решението (или грешката) и сменя статуса на задачата на „намерено решение“ или „грешка от решателя“ (в случай, че е възникнала синтактична или друга грешка в процеса на решаване).

Публичен API модул за външна връзка с други независими софтуерни системи

Целта на този модул е да предостави публичен програмен интерфейс за комуникация с други външни независими системи. На практика този модул позволява дистанционно да се извършват всички операции, които са предоставени от потребителския интерфейс в системата:

- изпращане на задачи за решаване към съответен решател
- изпращане на съобщение към външната система след получаване на решение или съобщение за грешка от даден решател на дадена задача.
- достъпът до този публичен интерфейс е реализиран като стандартна уеб услуга (web service). Това позволява лесна интеграция със системи написани на различни езици за програмиране, базирани на различни хардуерни платформи и операционни системи.

3.4. Работа със системата WebOptim

Тъй като системата е изцяло уеб-базирана, от потребителска гледна точка не се изисква нищо друго, освен свързаност с Интернет и уеб браузър (фиг. 12).



Фигура 12. WebOptim - начален екран.

Първата стъпка е да бъде регистриран персонален профил на потребителя. Това става чрез стандартна уеб форма и въвеждане на идентификационни данни за потребителя – адрес за електронна поща, потребителско име и парола (фиг. 13). На първо време това е напълно достатъчно за да бъде създаден потребителски профил и да се започне работа със системата.

Фигура 13. WebOptim – създаване на профил.

След като тази стъпка е изпълнена, вече може да се пристъпи към работа със системата. Извършва се автентикация чрез създадените потребителско име и парола (фиг. 14).

LOG IN

Please enter your username and password. [Register](#) if you don't have an .

Account Information

Username:

bobs

Password:

••••••••

☐ Keep me logged in

Фигура 14. WebOptim – форма за автентикация.

След успешен вход в системата, потребителят попада в интерфейс, съдържащ списък със задачи, чийто собственици са избрали те да бъдат публично достъпни (фиг.15).

Home

Members area

Browse public problems

Solvers

Syntaxes

Public API

About

My Problems

My MC Problems

LIST OF PUBLIC PROBLEMS.

	Title	Created on	Status	Sent to solver on	Syntax	Use solver
Preview	LP test big	6/6/2011 4:42:06 PM	Solution obtained	3/23/2012 8:26:23 PM	lp	LP Solve
Preview	MPS Test	6/9/2011 9:41:24 AM	Solution obtained	3/23/2012 8:26:21 PM	mps	LP Solve
Preview	CPLEX test	6/11/2011 1:21:11 AM	Solution obtained	3/23/2012 8:26:19 PM	lpt	LP Solve
Preview	Планиране производството на играчки	8/24/2011 2:50:53 PM	Solution obtained	8/26/2011 1:22:26 PM	lp	LP Solve
Preview	Задача за работен график	8/25/2011 4:07:37 PM	Solution obtained	8/26/2011 1:38:30 PM	lp	LP Solve
Preview	Задача за работен график	8/25/2011 4:12:00 PM	New		lp	LP Solve
Preview	Краткосрочно финансово планиране	8/26/2011 2:18:14 PM	Solution obtained	4/3/2012 2:41:36 PM	lp	LP Solve
Preview	Задача за смеси	8/30/2011 11:18:32 AM	Solution obtained	8/31/2011 11:51:47 AM	lp	LP Solve
Preview	Задача за тримесечно планиране	8/31/2011 2:46:02 PM	Solution obtained	5/7/2012 11:51:19 AM	lp	LP Solve
Preview	Задача за инвестиционна стратегия	9/2/2011 4:00:36 PM	Solution obtained	9/2/2011 4:26:59 PM	lp	LP Solve

Фигура 15. WebOptim – Списък с общодостъпни задачи.

Тук се дава възможност да бъдат прегледани и изследвани тези задачи, което се оказва от огромна помощ за новите потребители на системата и такива, които все още нямат достатъчно опит в работата със системата или решаването на оптимизационни задачи.

Потребителската част на системата се състои от два основни интерфейса – за решаване съответно на задачи за еднокритериална или многокритериална оптимизация.

3.4.1. Решаване на задачи за еднокритериална оптимизация

Решаването на задачи за еднокритериална оптимизация в системата WebOptim се извършва в секцията „My problems“. Основния екран, който се зарежда първоначално представлява списък със дефинираните задачи на потребителя и някои от по-важните им атрибути като дата на създаване, статус на задачата (нова, в процес на решаване, решена, грешка) и т.н.(фиг. 16).

WEBOPTIM									
Welcome bobs! [Log Out]									
Home Members area Browse public problems Solvers Syntaxes Public API About									
My Problems My MC Problems									
MY LP PROBLEMS									
	ID	Title	Created on	Status	Sent to solver on	Syntax	Use solver		
Select	Send to solver	113	Test MO 3 Predachen cech	4/13/2016 1:22:15 PM	Solution obtained	4/13/2016 1:22:29 PM	lp	GENS-IM	Delete
Select	Send to solver	92	Test MO 3 Predachen cech	7/13/2015 7:11:27 PM	Solution obtained	7/13/2015 7:13:12 PM	lp	GENS-IM	Delete
Select	Send to solver	41	cplex	9/12/2011 5:32:19 PM	Solution obtained	11/22/2012 3:49:11 PM	lpt	GENS-IM	Delete
Select	Send to solver	19	CPLEX test	6/12/2011 2:29:02 PM	Solution obtained	3/23/2012 8:26:15 PM	lpt	GENS-IM	Delete
Select	Send to solver	18	CPLEX test	6/12/2011 2:23:58 PM	Solution obtained	3/23/2012 8:26:17 PM	lpt	GENS-IM	Delete
Select	Send to solver	16	CPLEX test	6/11/2011 1:21:11 AM	Solution obtained	3/23/2012 8:26:19 PM	lpt	GENS-IM	Delete

Фигура 16. WebOptim – Списък със задачи за еднокритериална оптимизация

От този интерфейс на потребителя се дава възможност да изпрати дадена задача за решаване към решателя, да редактира задача или да създаде нова.

Създаването на нова задача или редактиране на съществуваща такава се извършва чрез интерфейс за дефиниране на задачи за еднокритериална оптимизация (фиг. 17).

problem_id	19
Title	CPLEX test
created_on	6/12/2011 2:29:02 PM
Status	Solution obtained
Sent to solver on	3/23/2012 8:26:15 PM
Description	CPLEX test problem
Public problem	☐
Syntax	LP - CPLEX
Solver	GENS-IM
Definition	<pre> Maximize obj: x1 + 2 x2 + 3 x3 + x4 Subject To c1: - x1 + x2 + x3 + 10 x4 <= 20 c2: x1 - 3 x2 + x3 <= 30 c3: x2 - 3.5 x4 = 0 Bounds 0 <= x1 <= 40 2 <= x4 <= 3 General x4 </pre>

Фигура 17. WebOptim – Създаване и редакция на задачи за еднокритериална оптимизация.

Тъй като при задачите за еднокритериална оптимизация няма процес на многостъпкова интеракция с потребителя, интерфейсът съдържа само 2 полета за въвеждане на задачата – текстово описание и самата задача, описана в някой от съвместимите стандартни синтаксиси поддържани от системата.

Информация за тези синтаксиси може да бъде намерена в секцията “Syntaxes” на главното меню (фиг. 18).

Home	Members area	Browse public problems	Solvers	Syntaxes	Public API	About	Publications
------	--------------	------------------------	---------	----------	------------	-------	--------------

LIST OF LP PROBLEM DEFINITION SYNTAXES CURRENTLY INSTALLED ON THE SYSTEM.

Syntax name	File extention	Description
LP - LPSolve	lp	This is the LPSolve lib native syntax. More information can be found here: http://lpsolve.sourceforge.net/5.5/lp-format.htm
LP - CPLEX	lpt	CPLEX LP syntax. More information can be found here: http://lpsolve.sourceforge.net/5.5/CPLEX-format.htm
LP- Xpress	lpx	Xpress LP syntax. More information can be found here: http://lpsolve.sourceforge.net/5.5/Xpress-format.htm
LP - LINDO	ltx	LINDO LP syntax. More information can be found here: http://lpsolve.sourceforge.net/5.5/LINDO-format.htm
MPS	mps	MPS syntax. Information about this syntax can be found here: http://lpsolve.sourceforge.net/5.0/mps-format.htm and here: http://lpsolve.sourceforge.net/5.5/bas-format.htm
ZIMPL	zpl	ZIMPL syntax. More information can be found here: http://lpsolve.sourceforge.net/5.5/Zimpl.htm
GNU - MathProg	mod	GNU MathProg syntax. More information can be found here: http://lpsolve.sourceforge.net/5.5/MathProg.htm
DIMACS	net	DIMACS syntax. More information can be found here: http://lpsolve.sourceforge.net/5.5/DIMACS_asn.htm and here: http://lpsolve.sourceforge.net/5.5/DIMACS_maxf.htm and here: http://lpsolve.sourceforge.net/5.5/DIMACS_mcf.htm
LPFML	xml	LPFML syntax. More information can be found here: http://lpsolve.sourceforge.net/5.5/LPFML.htm

The file extention parameter is also used by the solvers webservice for syntax description.

Фигура 18. WebOptim – Информация за поддържаните синтаксиси за дефиниране на задачи.

След като задачата е дефинирана и записана, следва изпращане към модула за решаване. На следваща стъпка се получава решението. Информацията за него е представена в две части – основна и разширена. Основната част съдържа стойностите на променливите и целевата функция, а разширената съдържа подробна информация, отнасяща се до самия процес на решаване – използвания метод, брой итерации, междинни решения и др. (фиг. 19).

Solution	<pre> Objective value:122.5 c1: 20 c2: 28 c3: 0 c1: 40 x1: 40 x2: 10.5 x3: 19.5 x4: 3 </pre>
Debug info	<pre> OPTIMAL Model name: '' - run #1 Objective: Maximize(obj) SUBMITTED Model size: 3 constraints, 4 variables, 9 non-zeros. Sets: 0 GUB, 0 SOS. CONSTRAINT CLASSES General REAL 1 General MIP 2 Using DUAL simplex for phase 1 and PRIMAL simplex for phase 2. The primal and dual simplex pricing strategy set to 'Devex'. Found feasibility by dual simplex after 3 iter. Relaxed solution 125.208333333 after 6 iter is B&B base. Feasible solution 122.5 after 7 iter, 1 nodes (gap 2.1%) get_ptr_sensitivity_objex: Sensitivity unknown Primal objective: Column name Value Objective Min Max </pre>

Фигура 19. WebOptim – Получаване на решение на задача за еднокритериална оптимизация.

3.4.2. Решаване на задачи за многокритериална оптимизация

Решаването на задачи за многокритериална оптимизация в системата WebOptim се извършва в секцията „My MCP problems“. Основния екран, който се зарежда е много подобен на този за решаване на задачи за еднокритериална оптимизация и съдържа същите атрибути. (фиг. 20).

My MCLP PROBLEMS

	ID	Title	Created on	Status	Sent to solver on	Syntax	Use solver	
Select	24	Batteries production problem	9/20/2017 4:21:20 PM	New		lp	GENS-IM	Delete
Select	19	Производство на фотоапарати	7/15/2016 4:11:04 PM	New		lp	GENS-IM	Delete
Select	18	test evil 4	4/15/2016 3:04:44 PM	Solver error	4/15/2016 3:04:49 PM	lp	FIEM	Delete
Select	11	Test evolution 2	12/10/2014 12:39:31 PM	Solution obtained	4/15/2016 3:06:58 PM	lp	FIEM	Delete
Select	10	Test evol	12/8/2014 1:57:01 PM	Solution obtained	12/8/2014 2:06:32 PM	mps	FIEM	Delete
Select	3	test 2	9/20/2013 5:58:13 PM	New		lp	GENS-IM	Delete
Select	1	test 1	6/19/2013 5:18:16 PM	New		lp	GENS-IM	Delete
Create new problem								

Фигура 20. WebOptim – Списък със задачи за многокритериална оптимизация

Интерфейсът за описание и дефиниране на задачата също е много подобен на този за еднокритериалната оптимизация (фиг. 21).

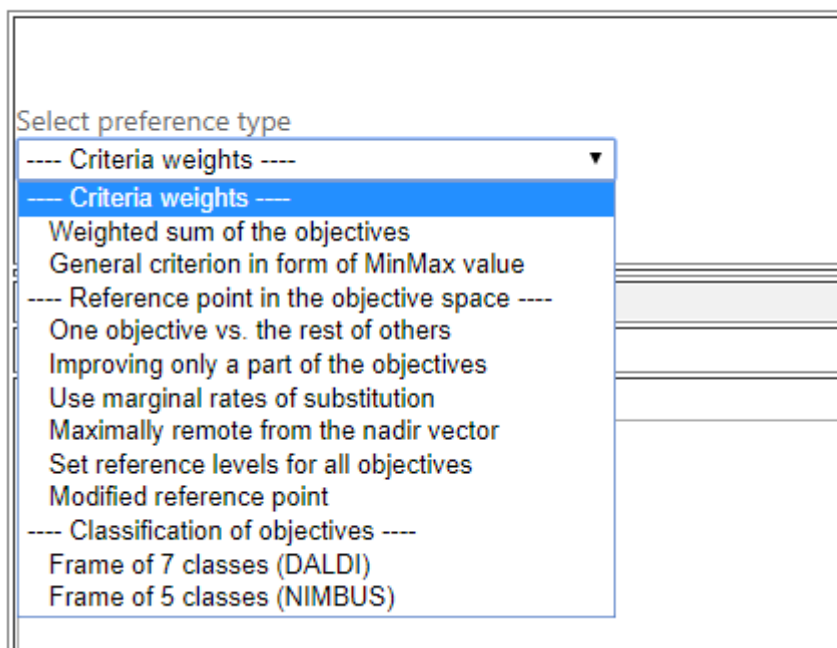
Select 1 test 1		6/19/2013 5:18:16 PM		New		lp		GENS-IM		Delete	
Update		Cancel									
problem_id	19										
Title	Производство на фотоапарати										
created_on	7/15/2016 4:11:04 PM										
Status	New										
Sent to solver on											
Description	Фабрика за фотоапарати произвежда 2 модела фотоапарати - стандартен и луксозен. Продукцията им е предназначена както за вътрешния, така и за международния пазар. Производствения процес определя съответни зависимости между човешката работната сила и машинното време за даден период. Целта е да се оптимизира производствения процес, имайки предвид, че марката стартира промоционална компания с цел да наложи името си на международния пазар.										
Problem Definition	<pre> max Z1 = 320 x1 120 x2 650 x3 70 x4 min Z2 = 1620 a 22 x1 22 x2 35 x3 + 35 x4 max Z3 = x2 x4 22 x1 + 22 x2 + 35 x3 + 35 x4 <= 1620 x1 + x2 + 2 x3 + 2 x4 <= 81 a = 1 x1 >= 1 x2 >= 1 x3 >= 1 x4 >= 1 </pre>										
Public problem	☒										
Syntax	LP - LPSolve										
Solver	GENS-IM										

Фигура 21. WebOptim – Създаване и редакция на задачи за многокритериална оптимизация

Подобно на системата МКО-2.1., в WebOptim са имплементирани общо 10 „aposteriori“ методи за решаване на задачи за многокритериална оптимизация. Това означава, че на потребителя се дава едно Парето-оптимално решение след което от него се изисква да въведе своите предпочитания спрямо всеки критерий (желание за подобряване, влошаване, запазване) и се генерира ново Парето-оптимално решение на базата на тези предпочитания. Процесът продължава докато ЛВР приеме едно от всичките решения за крайно.

След дефиниране на задачата следва избиране на начина на задаване на предпочитания при последващите итерации за намиране на нови Парето-оптимални решения.

Изборът става чрез падащо меню (фиг. 22), което предоставя 10 възможности, разпределени в 3 групи:



Фигура 22. WebOptim – Избор на начин за задаване на предпочитания за генериране на последващо Парето-оптимално решение.

1). Criteria weights (Задаване на предпочитания чрез задаване на тегла на критериите).

- Weighted sum of the objectives

Използва се скаларизираща задача на претеглената сума на критериите. ЛВР задава тегла на всеки един от критериите. По-голяма стойност на теглото означава по-голяма важност на критерия. Задаването става чрез помощен прозорец, който се отваря при клик на съответния критерий (фиг. 23). Стойностите на критериите за текущото решение са дадени в табличен вид, като в последната колона се задават предпочитанията.

Preference: Weight ▼

Value: 5

Set

Min/Max	x1	x2	x3	x4	a	Ideal	Nadir	Step1	Preference
Z1 = Max:	320	120	650	70	0	25535	1160	25535	Weight 5
Z2 = Min:	22	22	35	35	1620	114	1619.999	1426.5	Weight 6
Z3 = Max:	0	1	0	1	0	70.4545	2	2	Weight 2
a = Max:	0	0	0	0	0	0	0	0	Weight 10

Фигура 23. WebOptim – Задаване на тегловен коефициент на всеки един критерий.

- General criterion in form of MinMax value.

Използва се Чебишевска скаларизираща задача. Отново се задават тегловни коефициенти чрез същия помощен прозорец.

2). Reference point in the objective space (отправна точка в критериалното пространство).

- One objective vs. the rest of others.

Използва се скаларизираща задача на епсилон ограниченията. Един критерий се подобрява за сметка на всички останали. Тук изборът се прави като се избира един критерий за подобряване, а на всички останали се задава как да бъдат влошени. Влошаването може да бъде зададено или като гранична стойност, или като стойност с която да бъде влошена текущата стойност. Интерфейсът представлява помощен прозорец, подобен на този за задаване на тегла, но тук вече има включени 3 възможности за избор, чрез падащо меню (фиг.24):

- *Improve:* Подобряване
- *Worse with:* Влошаване със зададената стойност.
- *Worse to:* Влошаване, но не повече от зададената стойност.

Preference: Improve ▼
 Value: Undefined

Min/Max	x1	x2	x3	x4	a	Ideal	Nadir	Step1	Preference
Z1 = Max:	320	120	650	70	0	25535	1160	25535	Improve
Z2 = Min:	22	22	35	35	1620	114	1619.999	1426.5	Worse with 34.0000
Z3 = Max:	0	1	0	1	0	70.4545	2	2	Worse to 65.0000
a = Max:	0	0	0	0	0	0	0	0	Worse with 45.0000

Фигура 24. WebOptim – Задаване на предпочитания чрез отправна точка.

- Improving only a part of the objectives.

Използва се скаларизираща задача STEP. Избора е за подобряване на част от критериите за сметка на останалите. За задаване на предпочитания се използва същия прозорец, но този път могат да бъдат задавани повече от един критерий за подобряване.

- Use marginal rates of substitution.

Използва се скаларизираща задача STOM. Целта е установяване на отправна точка в критериалното пространство, като нейните координати са желаните приемливи стойности на критериите. Тези координати обаче, не трябва да надвишават стойностите на идеалния вектор! Предпочитанията се задават чрез помощен прозорец (фиг. 25), който предоставя следните възможности за задаване на стойност на всеки критерий:

Preference: Improve with ▼
 Value: Undefined

Min/Max	x1	x2	x3	x4	a	Ideal	Nadir	Step1	Preference	Step2	Preference
Z1 = Max:	320	120	650	70	0	25535	1160	25535	Improve	25530	Improve with 2.0000
Z2 = Min:	22	22	35	35	1620	114	1619.999	1426.5	Worse with 5.0000	1431	Improve to 44.0000
Z3 = Max:	0	1	0	1	0	70.4545	2	2	Worse to 1.0000	2	Worse with 3.0000
a = Max:	0	0	0	0	0	0	0	0	Worse to 1.0000	0	Worse to 55.0000

Фигура 25. WebOptim – Задаване на предпочитания чрез определяне на приемливи стойности на критериите.

- *Improve with*: Подобряване със зададената стойност.
 - *Improve to*: Подобряване до определена стойност.
 - *Worse with*: Влошаване със зададената стойност.
 - *Worse to*: Влошаване, но не повече от зададената стойност.
- Maximally remote from the nadir vector

Използва се скаларизираща задача GUESS. И тук предпочитанието е отправна точка със същите характеристики както при предишния случай, но вече имаме повече възможности за задаване на предпочитания (фиг. 26):

Preference: Improve with ▾ Value:											
Min/Max	x1	x2	x3	x4	a	Ideal	Nadir	Step1	Preference		Preference
Z1 = Max:	320	120	650	70	0	25535	1160	25535	Improve	25530	Improve with 4.0000
Z2 = Min:	22	22	35	35	1620	114	1619.999	1426.5	Worse with 5.0000	1431	Worse with 4.0000
Z3 = Max:	0	1	0	1	0	70.4545	2	2	Worse to 1.0000	2	Worse to 5.0000
a = Max:	0	0	0	0	0	0	0	0	Worse to 1.0000	0	Worse to 2.0000

Фигура 26. WebOptim – Задаване на отправна точка чрез определяне на приемливи стойности на критериите.

- *Improve with*. Подобряване със зададената стойност.
- *Improve to*. Подобряване до определена стойност.
- *Worse with*. Влошаване със зададената стойност.
- *Worse to*. Влошаване, но не повече от зададената стойност.

- Asp. level. Задаване на аспирационно ниво. В зависимост от зададеното ниво се определя дали стойността на критерия трябва да се подобрява или влошава.

- Set reference levels for all objectives.

Използва се скаларизираща задача RD. Чрез задаване на аспирационни нива се определят аспирационни направления за промяна стойностите на всеки един критерий (фиг. 27).

Preference: Asp. level ▾ Value: 2 Set											
Min/Max	x1	x2	x3	x4	a	Ideal	Nadir	Step1	Preference	Step2	Preference
Z1 = Max:	320	120	650	70	0	25535	1160	25535	Improve	25530	Asp. level 4.0000
Z2 = Min:	22	22	35	35	1620	114	1619.999	1426.5	Worse with 5.0000	1431	Asp. level 6.0000
Z3 = Max:	0	1	0	1	0	70.4545	2	2	Worse to 1.0000	2	Asp. level 7.0000
a = Max:	0	0	0	0	0	0	0	0	Worse to 1.0000	0	Asp. level 2.0000

Фигура 27. WebOptim – Задаване на аспирационни нива за всеки един от критериите.

- Modified reference point.

Използва се скаларизираща задача на Wierzbicki. Задаването на предпочитания става чрез класифициране на всеки критерий в един от 3 основни класа - за подобряване; за влошаване; за запазване на текуща стойност.

Изборът става от помощен прозорец, който предоставя 5 възможности за въвеждане (фиг. 28):

										Preference:	Save current value ▾
										Value:	Undefined
											Improve with
											Improve to
											Worse with
											Worse to
											Save current value
Min/Max	x1	x2	x3	x4	a	Ideal	Nadir	Step1	Prefe		reference
Z1 = Max:	320	120	650	70	0	25535	1160	25535	Improve		Improve with 4.0000
Z2 = Min:	22	22	35	35	1620	114	1619.999	1426.5	Worse with 5.0000	1431	Save current value
Z3 = Max:	0	1	0	1	0	70.4545	2	2	Worse to 1.0000	2	Worse to 4.0000
a = Max:	0	0	0	0	0	0	0	0	Worse to 1.0000	0	Save current value

Фигура 28. WebOptim – Класифициране на критериите в 3 основни класа - подобряване, влошаване, запазване на текуща стойност.

- *Improve with*: Подобряване със зададената стойност.
- *Improve to*: Подобряване до определена стойност.
- *Worse with*: Влошаване със зададената стойност.
- *Worse to*: Влошаване, но не повече от зададената стойност.
- *Save current value*: Запазване на текущата стойност.

3). Classification objectives (класификация на целевите функции)

- Frame of 7 classes (daldi)

Използва се скаларизираща задача DALDI. Тук целта е всеки критерий да бъде поставен в рамка от 7 основни класа

- подобряване (най-доброто възможно)
- подобряване с дадена стойност или до дадено аспирационно ниво
- запазване на текущата стойност
- влошаване
- влошаване с дадена стойност или до дадено аспирационно ниво

- задаване на горна и долна граница за възможна промяна на текущата стойност.
- без предпочитания.

Класификацията става чрез помощен прозорец (фиг. 29), който дава тази възможност.

Min/Max	x1	x2	x3	x4	a	Ideal	Nadir	Step1	Preference
Z1 = Max:	320	120	650	70	0	25535	1160	25535	Improve
Z2 = Min:	22	22	35	35	1620	114	1619.999	1426.5	Worse with 5
Z3 = Max:	0	1	0	1	0	70.4545	2	2	Worse to 1.0
a = Max:	0	0	0	0	0	0	0	0	Worse to 1.0

Фигура 29. WebOptim – Класифициране на критериите в 7 основни класа.

В случаите в които има задаване на някакъв вид интервали, това става чрез определяне на горна и долна стойност на интервала (фиг. 30).

Min/Max	x1	x2	x3	x4	a	Ideal	Nadir	Step1	Preference	Step2	Preference
Z1 = Max:	320	120	650	70	0	25535	1160	25535	Improve	25530	Interval 2.0000,6.0000
Z2 = Min:	22	22	35	35	1620	114	1619.999	1426.5	Worse with 5.0000	1431	Worse
Z3 = Max:	0	1	0	1	0	70.4545	2	2	Worse to 1.0000	2	Improve
a = Max:	0	0	0	0	0	0	0	0	Worse to 1.0000	0	

Фигура 30. WebOptim – Задаване граници на интервал

- Frame of 5 classes (nimbus)

Използва се скаларизираща задача NIMBUS. Също се използва класифициране, но този път в 5 класа:

- подобряване (най-доброто възможно)
- подобряване с дадена стойност или до дадено аспирационно ниво
- запазване на текущата стойност
- влошаване
- влошаване с дадена стойност или до дадено аспирационно ниво

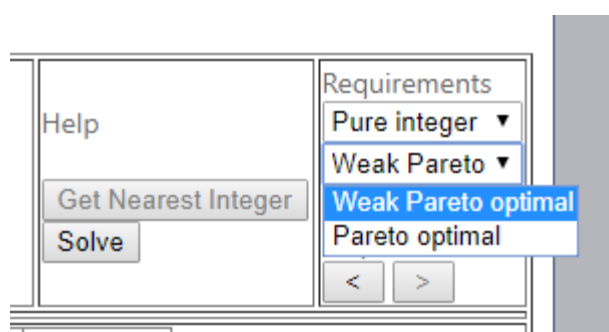
Изборът става чрез помощен прозорец (фиг. 17)

Preference: Undefined									
Undefined Undefined Improve Improve with Improve to Worse with Worse to Save current value Don't care									
Min/Max	x1	x2	x3	x4	a	Ideal	Nadir	Step1	Preference
Z1 = Max:	320	120	650	70	0	25535	1160	25535	Improve
Z2 = Min:	22	22	35	35	1620	114	1619.999	1426.5	Worse with 5
Z3 = Max:	0	1	0	1	0	70.4545	2	2	Worse to 1.0000
a = Max:	0	0	0	0	0	0	0	0	Worse to 1.0000

Фигура 31. WebOptim – Класифициране на критериите в 5 основни класа.

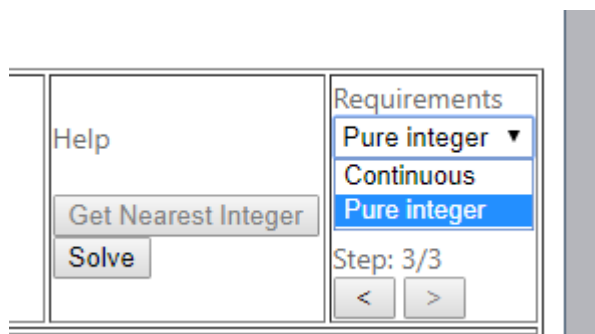
Важна подробност е, че избрания начин за задаване на предпочитания може да бъде променян на всяка стъпка преди генериране на следващо решение.

След като бъдат избрани предпочитанията, на ЛВР се дава избор за настройка на още 2 параметъра - генерираното решение да бъде слабо Парето-оптимално или Парето-оптимално (фиг. 32) и дали да се търси целочислено или непрекъснато решение (фиг. 33).



Фигура 32. WebOptim – Избор на вида на следващото генерирано решение - слабо/силно

Парето-оптимално.



Фигура 33. WebOptim – Избор на вида на следващото генерирано решение -

целочислено/непрекъснато.

По този начин и в тази част на системата се извършва целия интерактивен процес на търсене на решение стъпка след стъпка (фиг. 34).

Select preference type
Weighted sum of the objectives
WS - scalarising problem of weighted sum
Input parameters: The DM sets weights for each objective function
Output parameters: one supported Pareto optimal solution.

Help
Get Nearest Integer
Solve

Requirements
Pure integer
Weak Pareto
Step: 4/4
< >

Min/Max	x1	x2	x3	x4	a	Ideal	Nadir	Step1	Preference	Step2	Preference	Step3	Preference	Step4	Preference
Z1 = Max	320	120	650	70	0	25535	1160	25535	Improve	25530	Improve	25530	Worse	22930	undefined
Z2 = Min	22	22	35	35	1620	114	1619.999	1426.5	Worse with 5.0000	1431	Worse with 4.0000	1431	Interval -1.0000,1.300.0000	1291	undefined
Z3 = Max	0	1	0	1	0	70.4545	2	2	Worse to 1.0000	2	Worse with 3.0000	2		2	undefined
a = Max	0	0	0	0	0	0	0	0	Worse to 1.0000	0	Worse with 3.0000	0	Worse	0	undefined

Last Error:
Result
An optimal solution was obtained.
x1=2.0000 x2=1.0000 x3=34.0000 x4=1.0000 a=0.0000
Scalarized problem: (DALDI/Pure Integer)
Min: Alfa
320 x1 +120 x2 +650 x3 +70 x4 +25530 Alfa >= 25530
22 x1 +22 x2 +35 x3 +35 x4 +1620 a >= -1
22 x1 +22 x2 +35 x3 +35 x4 +1620 a <= 1300
0.01 Alfa >= 0
22 x1 +22 x2 +35 x3 +35 x4 <= 1620
x1 + x2 +2 x3 +2 x4 <= 83
x1 >= 1
x2 >= 1
x3 >= 1
x4 >= 1
a >= 0

Фигура 34. WebOptim – Основен интерфейс за търсене на Парето-оптимални решения.

В най-горната част се намира прозорецът за настройки. Тук се дава възможност във всеки един момент ЛВР да се върне на предишна стъпка и да продължи по нов начин ако не е

удовлетворен от последните намерени решения. Под него е таблицата с намерените до момента решения, разделени на стъпки. Последната колона на таблицата е интерактивна и в зависимост от избраните начини на задаване на предпочитания относно критериите, отваря допълнителните прозорци за въвеждане на стойности. Следва табличен ред за изписване на съобщения за грешки. Тук се извеждат потребителски съобщения за грешки свързани с процента на решаване - например несъвместим модел в следствие на грешно зададени предпочитания, грешки от математически операции като делене на 0 или други и т.н. Най-долу се намира прозорец с информация за текущото решение на задачата. Показани са стойностите на променливите, коя скаларизираща задача е използвана и самата генерирана задача.

3.5. Интерфейс за връзка с външни системи

WebOptim дава възможност за работа с външни системи чрез т.нар. уеб услуги.

По този начин процесът на работа вече не е обвързан с локалния потребителски интерфейс на системата. Чрез технологията на уеб услугите (web services) се предоставя машинен достъп до всички функционалности на системата. Всичката необходима информация за осъществяване на комуникацията със системата е дадена в раздела „Public API“ на главното меню на WebOptim (фиг. 35).

[Home](#)
[Members area](#)
[Browse public problems](#)
[Solvers](#)
[Syntaxes](#)
[Public API](#)
[About](#)
[Publications](#)

BETA VERSION OF THE PUBLIC API WEBSERVICE CONTRACTS FOR REMOTE USING.

WSDL schemas of solver services for posting LP problems. Can be invoked using SOAP protocol.

- [LP Solve Service](#)

Service parameters description	
problemKey	Unique identifier for the problem
syntax	Syntax name for the problem description. (See syntaxes section for more details)
definition	Definition of the problem.
verbose	Currently not used.
settings	Currently not used.
userKey	Currently not used.
returnURL	URL address of the webservice endpoint where the solution will be returned.

(For security reasons the IP address of the "returnURL" parameter (this is the address where the solution postback service is implemented) must match the sender IP address.)

WSDL schema of solution obtaining service of WebOptim. The same service must be implemented on the client side in order to receive solutions of the posted problems.

- [Solution Postback Service](#)

Webservice parameters description	
problemKey	Unique identifier for the problem
solution	Obtained solution.
verbose	Additional debug information.
errorCode	0 - solution obtained 1 - Solver does not support the given syntax 2 - Parse (syntax) error 3 - Internal solver error

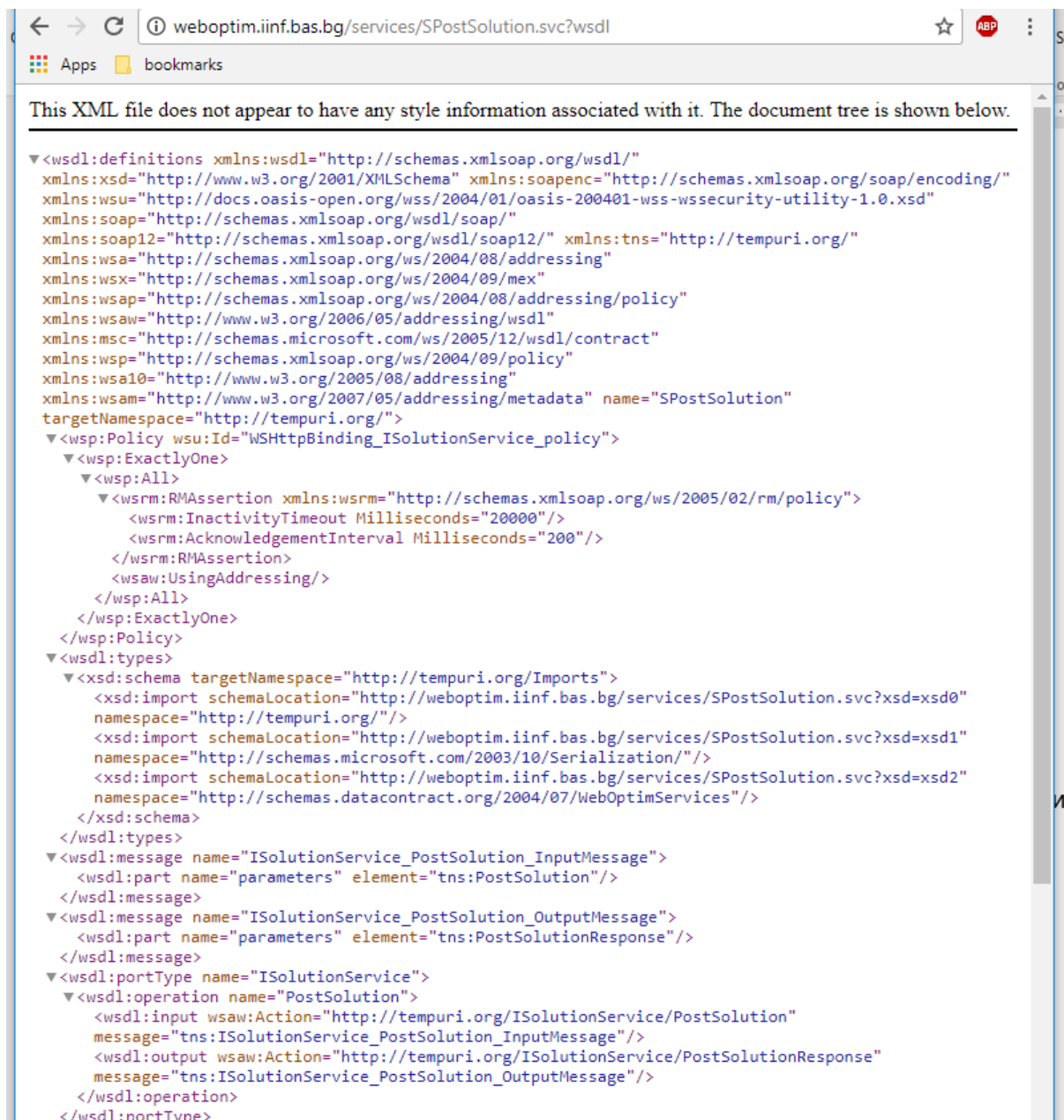
Фигура 35. WebOptim – Информация за осигуряване на комуникация с външни системи.

Там са посочени и връзките за намиране на WSDL схемите на двете уеб услуги, чрез които се осъществява комуникацията (фиг. 36 и фиг. 37).

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<?xml version="1.0" encoding="utf-8"?>
<wsdl:definitions xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/" xmlns:tns="http://tempuri.org/"
  xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
  xmlns:wsx="http://schemas.xmlsoap.org/ws/2004/09/mex"
  xmlns:wsap="http://schemas.xmlsoap.org/ws/2004/08/addressing/policy"
  xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"
  xmlns:msc="http://schemas.microsoft.com/ws/2005/12/wsdl/contract"
  xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy"
  xmlns:wsa10="http://www.w3.org/2005/08/addressing"
  xmlns:wsam="http://www.w3.org/2007/05/addressing/metadata" name="SLPSolve"
  targetNamespace="http://tempuri.org/">
  <wsp:Policy wsu:Id="WSHttpBinding_ISolverService_policy">
    <wsp:ExactlyOne>
      <wsp:All>
        <wsrm:RMAssertion xmlns:wsrm="http://schemas.xmlsoap.org/ws/2005/02/rm/policy">
          <wsrm:InactivityTimeout Milliseconds="20000"/>
          <wsrm:AcknowledgementInterval Milliseconds="200"/>
        </wsrm:RMAssertion>
        <wsaw:UsingAddressing/>
      </wsp:All>
    </wsp:ExactlyOne>
  </wsp:Policy>
  <wsdl:types>
    <xsd:schema targetNamespace="http://tempuri.org/Imports">
      <xsd:import schemaLocation="http://weboptim.iinf.bas.bg/services/SLPSolve.svc?xsd=xsd0"
        namespace="http://tempuri.org/" />
      <xsd:import schemaLocation="http://weboptim.iinf.bas.bg/services/SLPSolve.svc?xsd=xsd1"
        namespace="http://schemas.microsoft.com/2003/10/Serialization/" />
      <xsd:import schemaLocation="http://weboptim.iinf.bas.bg/services/SLPSolve.svc?xsd=xsd2"
        namespace="http://schemas.datacontract.org/2004/07/WebOptimServices/" />
    </xsd:schema>
  </wsdl:types>
  <wsdl:message name="ISolverService_PostProblem_InputMessage">
    <wsdl:part name="parameters" element="tns:PostProblem" />
  </wsdl:message>
  <wsdl:message name="ISolverService_PostProblem_OutputMessage">
    <wsdl:part name="parameters" element="tns:PostProblemResponse" />
  </wsdl:message>
  <wsdl:portType name="ISolverService">
    <wsdl:operation name="PostProblem">
      <wsdl:input wsaw:Action="http://tempuri.org/ISolverService/PostProblem"
        message="tns:ISolverService_PostProblem_InputMessage" />
      <wsdl:output wsaw:Action="http://tempuri.org/ISolverService/PostProblemResponse"
        message="tns:ISolverService_PostProblem_OutputMessage" />
    </wsdl:operation>
  </wsdl:portType>
</wsdl:definitions>
```

Фигура 36. WebOptim – WSDL схема на веб услугата за изпращане на задачи за решаване.



Фигура 37. WebOptim – WSDL схема на веб услугата за получаване на готовите решения.

ГЛАВА IV. ЕКСПЕРИМЕНТАЛНИ ИЗСЛЕДВАНИЯ – РЕШАВАНЕ НА ЗАДАЧА ЗА ОПЕРАТИВНО ПЛАНИРАНЕ ЧРЕЗ МНОГОКРИТЕРИАЛНА ОПТИМИЗАЦИЯ

В тази глава основната цел е да се тестват и валидират разработените системи и имплементираните в тях алгоритми за многокритериална оптимизация. Работоспособността на системите е доказана чрез решаване на реален пример за многокритериална оптимизация и съпоставяне на резултатите с тези, получени чрез решаване на същата задача с друга независима подобна система. Резултатите от експерименталната постановка са описани в публикация номер 3.

4.1. Описание на задачата

Фабрика за фотоапарати произвежда 2 модела фотоапарати - стандартен и луксозен. Продукцията им е предназначена както за вътрешния, така и за международния пазар.

Производствения процес определя съответни зависимости между човешката работната сила и машинното време за даден период. Тези зависимости са описани в таблица 2.

<i>Параметър</i>	<i>Часове човешки ресурс</i>	<i>Часове машинен ресурс</i>	<i>Продажна цена на вътрешен пазар</i>	<i>Продажна цена на международния пазар</i>
<i>Стандартен модел</i>	22	1	1000	800
<i>Луксозен модел</i>	35	0.5	1710	1130

<i>Ресурси на разположение</i>	1620	81		
<i>Цена на ресурса</i>	30	20		

Таблица 2. Параметри на задачата

Цел: Да се оптимизира производствения процес, имайки предвид, че марката стартира промоционална компания с цел да наложи името си на международния пазар.

Задачата е решена с помощта на софтуерната система МКО-2.1 и методът DALDI.

4.2. Построяване на математически модел на задачата

Параметри (променливи):

- x_1 : брой стандартни модели, произведени за вътрешния пазар
- x_2 : брой стандартни модели, произведени за износ
- x_3 : брой луксозни модели, произведени за вътрешния пазар
- x_4 : брой луксозни модели, произведени за износ

Целеви функции (критерии):

Да се максимизира печалбата (производствена цена - разходи):

$$\text{Max } Z_1 = 320 \cdot x_1 + 120 \cdot x_2 + 650 \cdot x_3 + 70 \cdot x_4$$

Където коефициентите пред всяко x , представляват разликата между продажната цена и производствените разходи и са получени чрез следната формула:

$$x_1: 1000 - (30 \cdot 22 + 20 \cdot 1) = 320$$

$$x_2: 800 - (30 \cdot 22 + 20 \cdot 1) = 120$$

$$x_3: 1710 - (30 \cdot 35 + 20 \cdot 0.5) = 650$$

$$x_4: 1130 - (30 \cdot 35 + 20 \cdot 0.5) = 70$$

Да се минимизират неоползотворените човекочасове или времето, през което персонала на фабриката ще стои незает. Представява минимизиране на разликата между наличния ресурс в човекочасове и сбора от необходимите човекочасове за производство:

$$\text{Min } Z_2 = 22 \cdot x_1 + 22 \cdot x_2 + 35 \cdot x_3 + 35 \cdot x_4 - 1620$$

Тъй като в тази функция съществува един свободен коефициент (1620), а синтаксисите за въвеждане на задача в МКО-2.1 не позволяват такива коефициенти, се налага добавянето на една допълнителна променлива, която ще бъде равна на 1: $a = 1$.

Максимизиране на бройката произведени фотоапарати за износ от двата модела:

$$\text{Max } Z_3 = x_2 + x_4$$

Ограничения:

Ограничение на ресурса от часовете за човешка работна сила:

$$22 \cdot x_1 + 22 \cdot x_2 + 35 \cdot x_3 + 35 \cdot x_4 \leq 1620$$

Ограничение на ресурса за машинно време:

$$x_1 + x_2 + 2 \cdot x_3 + 2 \cdot x_4 \leq 81$$

Изискване за смисъл и целочисленост на променливите:

$$x_1, x_2, x_3, x_4: \text{ цели числа, } \geq 1$$

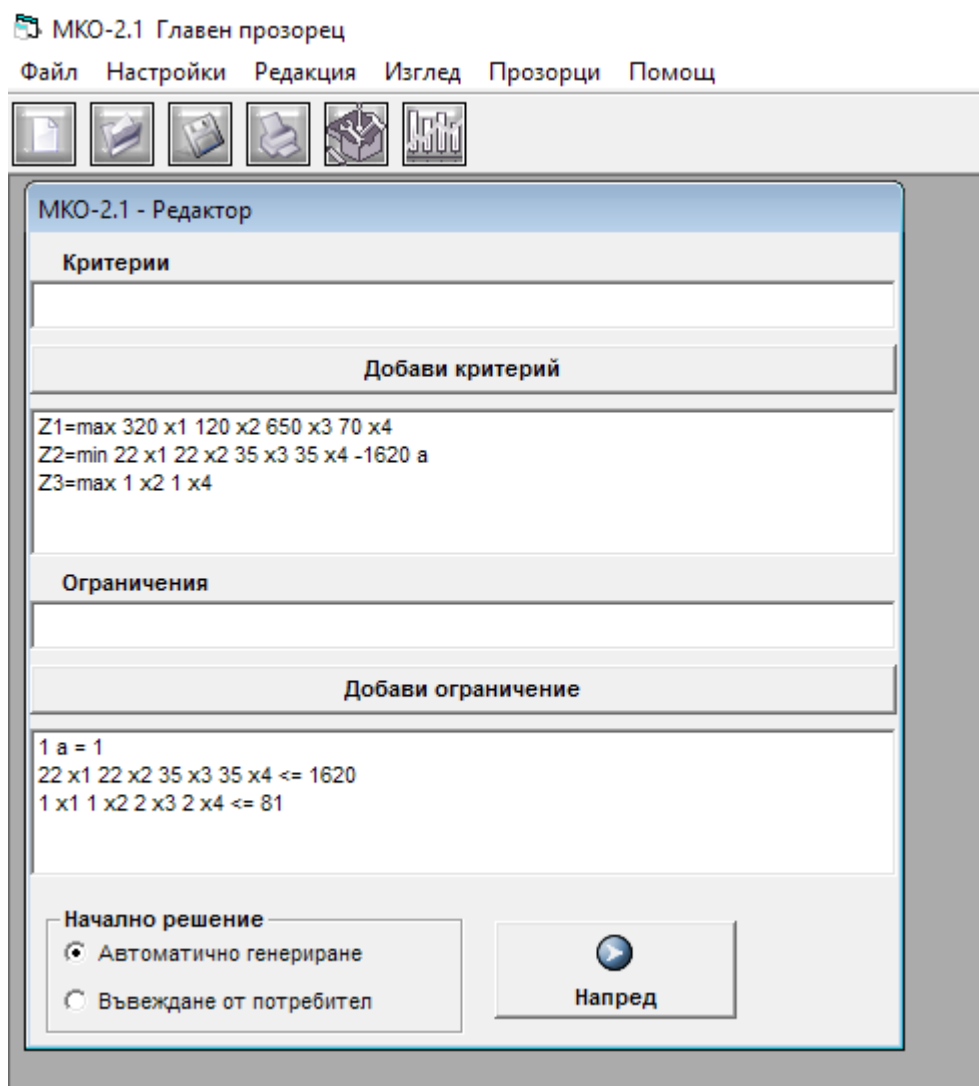
4.3. Решаване на задачата

Стартираме решаването чрез въвеждане на задачата в системата МКО-2.1 (фиг. 38).

При стартиране на процеса на решаване получаваме първоначално решение (фиг. 39) със следните данни за „идеален“ и „надир“ векторите (таблица. 3):

<i>Целева функция</i>	<i>Идеален вектор</i>	<i>Надир вектор</i>
<i>Z1 (max)</i>	25535	1160
<i>Z2 (min)</i>	-1506	-4.547
<i>Z3 (max)</i>	70.4545	2

Таблица 3. Стойности на идеалния и надир векторите



Фиг.38. МКО-2.1. – Въвеждане на задачата.

МКО-2 Решаване

Решаване Инфо Назад Напред Променливи Pref. type

Тип решение

☐ Непрекъснато
 ☒ Най-близко целочислено
 ☒ Слабо Парето-оптимално
 ☐ Парето-оптимално

☐ Целочислено
 ☐ Евристично целочислено

Общо стъпки: 1
Текуща стъпка: 1

Критерии	Идеални точки	Надир точки
Z1=8960	25535	1160
Z2=-76	-1506	-4.54747350886464E-13
Z3=67	70.4545454545454	2

Последна задача (DALDI)

min 0 x1 0 x2 0 x3 0 x4 0 a 1 alfa

1 a = 1

22 x1 22 x2 35 x3 35 x4 <= 1620

1 x1 1 x2 2 x3 2 x4 <= 81

320 x1 120 x2 650 x3 70 x4 9027.82608695644 alfa >= 9027.82608695644

Фиг.39. МКО-2.1. – Първоначално решение на задачата

Системата генерира автоматично първоначално недоминирано решение и след като изберем най-близкото му целочислено решение, получаваме следните стойности на критериите и променливите (табл. 4 и табл. 5):

Z1	Z2	Z3
8960	-76	67

Таблица 4. Стойности на критериите

x_1	x_2	x_3	x_4
1	66	1	1

Таблица 5. Стойности на променливите

Това първоначално решение е наша отправна точка, а интерпретацията му е, че трябва да се произведат 66 бройки стандартни фотоапарати за износ и по 1 бройка от всички останали. Чисто икономически, това не ни задоволява като решение, тъй като дава прекалено голям приоритет само стандартния модел апаратите за износ. Затова продължаваме да търсим ново Парето-оптимално решение, като решаваме да подобрим свободно критерия за печалба (Z_1) за сметка на критерия, който максимизира износа (Z_3). Вземайки предвид идеалната точка (70.4545) и надир точката (2) на критерия, който ще влошаваме, решаваме да позволим той да се влошава максимум до ниво 30. Критерият Z_2 , който се отнася до уплътняване на заетостта на работниците на тази стъпка ще го оставим да се променя свободно (фиг. 40)

МКО-2 Решаване

Решаване Инфо Назад Напред Променливи Pref. type

Тип решение

☐ Непрекъснато ☒ Най-близко целочислено
☐ Целочислено ☐ Евристично целочислено

☒ Слабо Парето-оптимално
☐ Парето-оптимално

Общо стъпки: 1
Текуща стъпка: 1

Критерии	Идеални точки	Надир точки
Z1=8960 (подобряване)	25535	1160
Z2=-76 (свободна промяна)	-1506	-4.54747350886464E-13
Z3=67 (влошаване -> аспирационно ниво: 30)	70.4545454545454	2

Последна задача (DALDI)

min 0 x1 0 x2 0 x3 0 x4 0 a 1 alfa

1 a = 1

22 x1 22 x2 35 x3 35 x4 <= 1620

1 x1 1 x2 2 x3 2 x4 <= 81

320 x1 120 x2 650 x3 70 x4 9027.82608695644 alfa >= 9027.82608695644

Фиг.39. МКО-2.1.– Задаване на предпочитания за промяна на критериите

Стартираме решаването на задачата с така зададените предпочитания и получаваме следните нови резултати (табл. 6 и табл. 7):

Z1	Z2	Z3
19790	-63	30

Таблица 6. Стойности на критериите

$x1$	$x2$	$x3$	$x4$
2	29	24	1

Таблица 7. Стойности на променливите

Получените резултати са задоволителни от практическа гледна точка, но все пак решаваме да изследваме още малко областта на Парето-оптималните решения за да видим какви резултати ще получим. За целта решаваме да запазим текущата стойност на критерий $Z1$, тъй като тя е сравнително близо до идеалната точка. Критерий $Z2$ има не особено добра стойност, съдейки по положението и спрямо идеална-надир точка, затова избираме за него свободно подобряване. Тъй като трябва все пак да „жертваме“ от някой критерий, избираме това да бъде критерий $Z3$, който касае износа, но да го влошим до ниво не по-ниско от 20 (фиг. 40).

МКО-2 Решаване

Решаване Инфо Назад Напред Промениливи Pref. type

Тип решение

☐ Непрекъснато ☐ Най-близко целочислено
☒ Целочислено ☐ Евристично целочислено

☒ Слабо Парето-оптимално
☐ Парето-оптимално

Общо стъпки: 2
Текуща стъпка: 2

Критерии	Идеални точки	Надир точки
Z1=19790 (не влошаване)	25535	1160
Z2=-63 (подобряване)	-1506	-4.54747350886464E-13
Z3=30 (влошаване -> аспирационно ниво: 20)	70.4545454545454	2

Последна задача (DALDI)

min 0 x1 0 x2 0 x3 0 x4 0 a 1 alfa 3 beta

1 a = 1

22 x1 22 x2 35 x3 35 x4 <= 1620

1 x1 1 x2 2 x3 2 x4 <= 81

320 x1 120 x2 650 x3 70 x4 8960 beta >= 8960

Фиг.40. МКО-2.1 – Избор на нови предпочитания за критериите.

Стартираме решаването на задачата с така зададените предпочитания и получаваме следните нови резултати (табл. 8 и табл. 9):

Z1	Z2	Z3
19890	-213	20

Таблица 8. Стойности на критериите

x_1	x_2	x_3	x_4
2	19	26	1

Таблица 9. Стойности на променливите

На тази стъпка нещата изглеждат добре балансирани и виждаме, че критерий Z1 дори се е подобрил малко.

Решаваме да видим дали все пак можем да подобрим малко критериите Z2 и Z3, като все пак запазим стойностите на Z1 над аспирационно ниво от 18000 (фиг. 41).

МКО-2 Решаване

Решаване Инфо Назад Напред Променливи Pref. type

Тип решение

☐ Непрекъснато ☐ Най-близко целочислено ☒ Слабо Парето-оптимално ☐ Парето-оптимално

☒ Целочислено ☐ Евристично целочислено

Общо стъпки: 4
Текуща стъпка: 3

Критерии	Идеални точки	Надир точки
Z1=19890 (влошаване -> аспирационно ниво: 18000)	25535	1160
Z2=-213 (подобряване)	-1506	-4.54747350886464E-13
Z3=20 (подобряване)	70.4545454545454	2

Последна задача (DALDI)

min 0 x1 0 x2 0 x3 0 x4 0 a 1 alfa 3 beta

1 a = 1

22 x1 22 x2 35 x3 35 x4 <= 1620

1 x1 1 x2 2 x3 2 x4 <= 81

320 x1 120 x2 650 x3 70 x4 >= 19790

Фиг.40. МКО-2.1.–. Задаване на новите предпочитания.

Стартираме решаването на задачата с така зададените предпочитания и получаваме следните нови резултати (табл. 10 и табл. 11):

<i>z1</i>	<i>z2</i>	<i>z3</i>
18100	-252	24

Таблица 10. Стойности на критериите

<i>x1</i>	<i>x2</i>	<i>x3</i>	<i>x4</i>
1	23	23	1

Таблица 11. Стойности на променливите

След така проведеното изследване на областта на Парето-оптималните решения, решаваме окончателно, че в контекста на задачата реално за нас най-смислено изглежда решението на стъпка 2, а именно:

- 2 броя от стандартния модел за вътрешния пазар
- 29 броя от стандартния модел за износ
- 24 броя от луксозния модел за вътрешния пазар
- 1 брой от луксозния модел за износ

При това разпределение на производството, решението на нашата задача гарантира математически една балансирана стратегия по отношение на печалба, уплътняване на работното време и изпълнение на условието за засилено производство за износ.

4.4. Сравнителен анализ на резултатите

За целта на сравнителния анализ, задачата е решена и с получени от решаване на задачата за производство на фотоапарати между решаването ѝ със системата МКО-2.1 и добилата голяма популярност уеб базирана система на университета в Яваскула, Финландия – WWW NIMBUS (www.nimbus.it.jyu.fi).

За въвеждане на задачата в NIMBUS се използва стандартна уеб форма (фиг. 41).

Objective functions to be optimized

	name	expression	Lowest Value	Highest Value
Max ▾	f1	$320 \cdot x_1 + 120 \cdot x_2 + 650 \cdot x_3 + 70 \cdot x_4$	1160	25535
Min ▾	f2	$22 \cdot x_1 + 22 \cdot x_2 + 35 \cdot x_3 + 35 \cdot x_4 - 1620 \cdot x_5$	-1506	-4.55
Max ▾	f3	$1 \cdot x_2 + 1 \cdot x_4$	2	70

Use [Standard functions available](#)

Variables

☐ Use [autofill](#)

	lower bound	<= starting point	<= upper bound	Integer values
x1	1	<= 1	<= 100000000	☒
x2	1	<= 1	<= 100000000	☒
x3	1	<= 1	<= 100000000	☒
x4	1	<= 1	<= 100000000	☒
x5	1	<= 1	<= 1	☒

Linear Constraints

	Ax					b						
No1 :	22	*x1 +	22	*x2 +	35	*x3 +	35	*x4 +	0	*x5	<= ▾	1620
No2 :	1	*x1 +	1	*x2 +	2	*x3 +	2	*x4 +	0	*x5	<= ▾	81

Фиг. 41. Дефиниране на задачата в системата WWW NIMBUS

При стартиране на процеса на решаване, получаваме следните резултати (табл. 12 и табл. 13):

<i>z1</i>	<i>z2</i>	<i>z3</i>
10780	-584	28

Таблица 12. Стойности на критериите

[5].

<i>x1</i>	<i>x2</i>	<i>x3</i>	<i>x4</i>
1	27	11	1

Таблица 13. Стойности на променливите

На този етап NIMBUS позволява задаване на предпочитания, единствено чрез указване за всеки критерии дали да се подобрява; влошава или да запази текущата си стойност. По преценка на системата на следваща стъпка може да се изисква указване на аспирационно ниво. Подобно на стъпка 2 от решаването с МКО21, решаваме да зададем критерии f1 и f2 да се подобряват, а критерии f3 да се влоши (фиг. 42).

Function		Current solution Class: < <= = >= >				
Max f1	10780.0	☐	☐	☐	☐	☒
Min f2	-584.0	☐	☐	☐	☐	☒
Max f3	28.0	☐	☒	☐	☐	☐
Function		Lowest Value		Highest Value		
Max f1	1160.0	25535.0				
Min f2	-1506.0	-4.55				
Max f3	2.0	70.0				

Фиг.42. NIMBUS – задаване на предпочитания за търсене на следващо решение

След потвърждаване на избора, системата ни предлага да въведем аспирационно ниво за критерия f3. Задаваме ниво 20 (фиг. 43)

Upper and lower bound functions

Enter boundary values

Max f3 2.0 <= <= 28.0

Select operation

☐ Another problem
 ☐ Save the current problem

☐ Remove a saved problem
 ☒ Calculate a new solution (continue)

☐ Correct Highest or Lowest values
 ☐ Show the whole problem

Фиг.43. NIMBUS – задаване на аспирационно ниво

На тази стъпка системата ни предлага 4 нови алтернативи (табл. 14):

	<i>Z1</i>	<i>Z2</i>	<i>Z3</i>
<i>A1</i>	10780	-584	28
<i>A2</i>	25530	-189	2
<i>A3</i>	19320	-10	2
<i>A4</i>	25210	-211	2

Таблица 14. Нови алтернативи

И съответните стойности на променливите (табл. 15):

	$x1$	$x2$	$x3$	$x4$
$A1$	1	27	11	1
$A2$	2	1	38	1
$A3$	51	19	1	1
$A4$	1	1	38	1

Таблица 15. Стойности на променливите

От така получените резултати, най-смислената алтернатива в контекста на задачата ни изглежда алтернатива A1 а именно:

- 1 брой от стандартния модел за вътрешния пазар
- 27 броя от стандартния модел за износ
- 11 броя от луксозния модел за вътрешния пазар
- 1 брой от луксозния модел за износ

Резултатите от решаването на задачата с двете софтуерни системи са представени обобщено в табличен вид (табл. 16):

<i>Стъпка</i>	<i>Система</i>	<i>X1</i>	<i>X2</i>	<i>X3</i>	<i>X4</i>	<i>Z1</i>	<i>Z2</i>	<i>Z3</i>	<i>Избрано крайно решение</i>
1	МКО-2.1	1	66	1	1	8960	-76	67	
1	Nimbus	1	27	11	1	10780	-584	28	*
2	МКО-2.1	2	29	24	1	19790	-63	30	*
2	Nimbus	2	1	38	1	25530	-189	2	
3	МКО-2.1	2	19	26	1	19890	-213	20	
3	Nimbus	51	19	1	1	19320	-10	2	
4	МКО-2.1	1	23	23	1	18100	-252	24	
4	Nimbus	1	1	38	1	25210	-211	2	

Таблица 16. Сравнение на резултатите получени от МКО-2.1 и NIMBUS

ЗАКЛЮЧЕНИЕ

Многокритериалната оптимизация е единия от двата основни дяла на дисциплината вземане на решения. Тя е свързана с решаване на задачи на многокритериалното математическо оптимизиране. Най-широка популярност за решаване на този клас задачи са добили така наречените интерактивни алгоритми. При тях решаването на тези задачи се свежда до решаване на множество от еднокритериални целочислени задачи, в които лицето вземащо решения е активен участник. Алгоритмите от този клас изискват голям обем от математически изчисления, които не биха били практически възможни без използването на компютърна техника. Това поражда нуждата от програмна реализация на този тип алгоритми, както и за удобството на ползване на базирана на тях система, за да бъде тя приложима в практиката. В дисертационния труд са описани едни от най-използваните методи и алгоритми за решаване на задачи за многокритериална оптимизация чрез скаларизиране. Особено внимание е обърнато на алгоритмите DAL и DALDI, които са разработени в института по информационни и комуникационни технологии към БАН от научен екип и имат пряко отношение към дисертационния труд. Тези класификационно-ориентирани интерактивни алгоритми позволяват генерирането на 12 от най-популярните скаларизиращи задачи, което ги прави изключително полезни при използването им с цел обучение. На базата на тези алгоритми са разработени две софтуерни системи за решаване на многокритериални оптимизационни задачи - МКО-2.1, която е десктоп базирана система, работеща под операционната система Windows, и WebOptim, която е уеб-базирана софтуерна система от ново поколение, използваща най-нови технологии и осигуряваща лесен достъп до широк кръг потребители с различни интереси и нива на познание в областта.

Системата МКО-2.1. предоставя широки възможности на лицето, вземащо решения да задава своите предпочитания за качествата на търсеното най – предпочитано решение.

Потребителският интерфейс позволява със системата да работят лица, вземащи решения с различна степен на квалификация по отношение на използваните оптимизационни методи и програмни средства. МКО-2.1. може да бъде използвана както за учебни цели, така и за решаване на експериментални и изследователски задачи. Един от основните недостатъци на МКО-2.1. е, че изисква локална инсталация и предвидена да работи единствено под операционната система Windows. Работата с нея, основно като средство за извършване на сложни математически изчисления е пряко зависима от ресурсите и изчислителната мощ на машината, на която е инсталирана. Също така не е възможно системата лесно да бъде надграждана и да позволява обмен на данни с други външни системи. Системата WebOptim е естествено и съвременно продължение на МКО-2.1., създадено с основната цел да елиминира всички тези нейни недостатъци.

Поради това, че системата вече е уеб базирана автоматично отпадат двата най-важни недостатъка - нуждата от локална инсталация и изчислителна зависимост ресурсите на локалното устройство. Също така, WebOptim е изградена така, че изчислителните модули да бъдат максимално независими от останалата част на системата. Свързването им със системата става чрез технологията „web services“. Това позволява лесно свързване на нови изчислителни модули, които са разработени от трети страни, на различни платформи и различни програмни езици, и дори физически се намират на отдалечени машини.

СПИСЪК НА ПУБЛИКАЦИИТЕ ПО ДИСЕРТАЦИЯТА

1. B. Staykov, F.Andonov (2019). "Practical decision making". Information Technologies and Control, Vol 4. 2018. ISSN: 1312-2622; Online ISSN: 2367-5357, DOI: 10.1515/itc-2018-0019, pp. 29-38.
2. Kirilov L., Guliashki, V., Staykov B. (2018). "Web-Based Decision Support System for Solving Multiple-Objective Decision-Making Problems", Technological Innovations in Knowledge Management and Decision Support, Chapter 7, ISBN13: 9781522561644, DOI: 10.4018/978-1-5225-6164-4.ch007, pp. 150-175.
3. Staykov, B. (2015). "Solving Multicriteria Optimization Problems with WebOptim Software System", Cybernetics and Information Technologies, Vol. 15, Issue 3, ISSN: 1311-9702, DOI: 10.1515/cait-2015-0049, pp. 165-177.
4. Kirilov, L., Guliashki, V., Genova, K., Zhivkov, P., Staykov, B., Vatov, D. (2015), "Interactive environment WebOptim for solving multiple-objective problems using scalarizing and evolutionary approaches", International Journal of Reasoning-based Intelligent Systems, Vol.7, Issue 1/2, ISSN 1755-0556, pp.4-15., SJR:0.21.
5. Kirilov, L., Guliashki, V., Genova, K., Vassileva, M., Staykov, B. (2013). "Generalized scalarizing model GENS in DSS WebOptim", International Journal of Decision Support System Technology, Vol. 5, Issue 3, ISSN: 1941-6296, DOI: 10.4018/jdsst.2013070101, pp. 1-11.
6. Genova, K., Kirilov, L., Gulisahski, V., Staykov, B., Vatov, D. (2011). "A prototype of a web-based decision support system for building models and solving optimization and decision making problems". CompSysTech '11 Proceedings of the 12th International Conference on

Computer Systems and Technologies, Vienna, Austria - June 16 - 17, 2011, ISBN: 978-1-4503-0917-2, pp. 167-172.

7. Vassileva, M., Vassilev, V., Staykov, B., Dochev, D. (2007). "Generalized Multicriteria Optimization Software System MKO-2.", Proceedings of the Ninth International Conference on Enterprise Information Systems (ICEIS 2007), Vol. AIDSS, ISBN 978-972-8865-89-4, DOI: 10.5220/0002388402830288, pp. 283-288.
8. Stajkov, B. (2006). "Multiobjective Optimization Software System". Problems of Engineering Cybernetics and Robotics, Vol.57, ISSN: 0204-9848, pp.21-30.

СПИСЪК НА ЗАБЕЛЯЗАНИ ЦИТИРАНИЯ НА ПУБЛИКАЦИИТЕ ПО ДИСЕРТАЦИЯТА

Забелязани са следните цитирания на статия номер 3:

1. Feyzioğlu, A., Kar, A. K. (2017). “Axiomatic Design Approach for Nonlinear Multiple Objective Optimizatон Problem and Robustness in Spring Design”, Cybernetics and Information Technologies, Vol. 17, Issue 1, ISSN: 1311-9702, DOI: 10.1515/cait-2017-0005, pp. 63-71. Цитирана под номер 2.
2. T. Tashev, V. Monov, R. Tasheva (2017). “Computer Simulations for Stability Analysis of a Numerical Procedure for Crossbar Packet Switch”, Information Technologies and Control, Vol. 15, Issue 3, ISSN: 1312-2622, pp. 7-11. Цитирана под номер 5.
3. T. Tashev, A. Bakanov, R. Tasheva (2016). “Верхняя граница пропускной способности коммутатора с матричным переключателем для входящего трафика типа модифицированной модели чанг-а”. Conference: University Annual Science Conference of National Military University 2016At: Veliko Turnovo, BulgariaV, ISSN 1314-1937, pp.107-116. Цитирана под номер 11.

Забелязани са следните цитирания на статия номер 6:

1. Sgurev V., Drangajov S. (2018). “Network Flows and Risks”. In: Sgurev V., Jotsov V., Kacprzyk J. (eds) Practical Issues of Intelligent Innovations. Studies in Systems, Decision and Control, Vol 140. Springer, Cham, ISBN: 978-3-319-78437-3, pp. 53-88. Цитирана под номер 5.
2. Sgurev, V., Drangajov, S. (2016). “Decision making in intelligent resource networks under risk”, Proceedings of IEEE 8th International Conference on Intelligent Systems (IS), ISBN: 978-1-5090-1354-8, pp. 27-35. Цитирана под номер 10.

3. Sgurev, V., Drangajov, S. (2016). "Optimal Control of the Flow of Risk on Networks". Information Technologies and Control, Vol. 13, Issue 3-4, ISSN: 1312-2622, pp. 29-33. Цитирана под номер 10.
4. Sgurev V., Drangajov S. (2015). "Intelligent Control of Flows with Risks on a Network. In: Filev D. et al. (eds) Intelligent Systems'2014. Advances in Intelligent Systems and Computing, vol 323. Springer, Cham, ISBN: 978-3-319-11309-8, pp. 27-35. Цитирана под номер 9.
5. Sgurev V., Drangajov S. (2014). "A method for finding optimal network cycles under constraints at a tour of points under surveillance". Proceedings of "Advanced control and optimization: step ahead '2014", Bankya, Bulgaria, ISSN 1314-4634, pp. 34-39. Цитирана на 7-мо място.
6. Sgurev V., Drangajov S., Doukovska, L. (2014). "Maximum Message Flow and Capacity in Sensor Networks". In Proceedings of the Third International Conference on Telecommunications and Remote Sensing (ICTRS 2014), ISBN: 978-989-758-033-8, pp. 74-80. Цитирана на 7-мо място.
7. Sgurev V., Drangajov S. (2014). "Optimal Control of Mobile Agents for Monitoring of Points on a Network". Information Technologies and Control, Vol. 11, Issue 2, ISSN: 1312-2622, pp. 2-6. Цитирана под номер 6.

Забелязани са следните цитирания на статия номер 8:

1. Branke, J., Deb, K., Miettinen, K., Slowinski, R. (2006). "Multiobjective optimization: Interactive and Evolutionary Approaches". ISBN: 3-540-88907-8 Springer Berlin Heidelberg NewYork. Цитирана в глава 12 на 18-то място.

РЕЗЮМЕ НА ПОСТИГНАТИТЕ РЕЗУЛТАТИ

Получените резултати, описани в настоящия дисертационен труд, могат да се обобщят в следните научно-приложни приноси:

1. Систематизирани са множество методи за решаване на задачи за многокритериална оптимизация и са избрани конкретни от тях за алгоритмична и софтуерна реализация.
2. Разработени са синтаксис за дефиниране на линейни и линейно-целочислени задачи за многокритериална оптимизация със съответния програмен парсер за целите на реализираните СПВР.
3. Проектирани и разработени за управляващите и изчислителните модули на системата МКО-21 за работа под операционна система WINDOWS.
4. Проектирана и реализирана е общата архитектура, функционалните възможности и потребителския интерфейс на уеб базираната СПВР WebOptim.
5. За целите на системата WebOptim са разработени комуникационни модули за обмен на информация с външни системи.
6. Проведени са експериментални изследвания с които е доказана работоспособността на разработените системи.

ДЕКЛАРАЦИЯ ЗА ОРИГИНАЛНОСТ НА РЕЗУЛТАТИТЕ

Декларирам, че настоящата дисертация съдържа оригинални резултати, получени при проведени от мен научни изследвания (с подкрепата и съдействието на научния ми ръководител и екип от колеги в звеното). Резултатите, които са получени, описани и/или публикувани от други учени, са надлежно и подробно цитирани в библиографията. Настоящата дисертация не е прилагана за придобиване на научна степен в друго висше училище, университет или научен институт.

Подпис:

/Борис Стайков/

БЛАГОДАРНОСТИ

Описаните в дисертационния труд методи, алгоритми и софтуерни системи за решаване на задачи за многокритериална оптимизация са резултат от дългогодишната съвместна работа в научен екип с ръководител проф. д-р Васил Василев (който за съжаление вече не е между нас) и в състав: доц. д-р Красимира Генова, д-р Филип Андонов, доц. д-р Марияна Василева, доц. д-р Леонид Кирилов, доц. д-р Васил Гуляшки. На тях съм изключително признателен и изказвам специални благодарности. Без техния труд, тази дисертация не би била възможна.

Изказвам благодарности и на г-н Петър Живков, който пренаписа изчислителния модул на системата МКО-2.1. от Visual Basic 6.0 на C# така, че да може да бъде използван в системата WebOptim.

Благодаря на проф. д-р Владимир Монов и доц. д-р Татяна Атанасова, които ми оказаха голяма помощ и подкрепа в последната фаза от разработката на дисертационния труд.

Благодарности дължа и на моите родители, колеги и приятели, които ме подкрепяха и мотивираха през цялото време.

БИБЛИОГРАФИЯ

1. Alves M., Climaco N., (2004). "A Note on a Decision Support System for Multiobjective Integer and Mixed-integer Programming Problems", *European Journal of Operational Research*, Vol. 155, ISSN: 0377-2217, pp. 258-265.
2. Andonov, F. (2009). "Interactive methods for group decision making", *Intelligent Support of Decision Making*, Book 10, supplement to *Information Technologies and Knowledge*, Vol. 3, ISSN 1313-0455, pp. 25-29.
3. Andonov, F. (2009). "Solving Discrete Multicriteria Optimization Problems in Group Environment", *Proceedings of International Conference on SOFTWARE, SERVICES & SEMANTIC TECHNOLOGIES*, October 28-29, 2009, Sofia, Bulgaria, ISBN 978-954-9526-62-2, pp. 239-243.
4. Andonov, F., Petrov, G. (2008). "Applications of the multicriteria method of decision-making for supporting the choice of corporate telecommunication solutions", *Computer Science and Education in Computer Science*, Vol. 4, Issue 1, ISSN: 1313-8624, pp. 8-13.
5. Arrais-Castro A., Leonilde M., Varela R., Putnik D., Ribeiro, R., Dargam C. (2016). "Collaborative Negotiation Platform using a Dynamic Multi-Criteria Decision Model", *International Journal of Decision Support Systems and Technology*, Vol. 7, Issue 1, ISSN: 1941-6296, pp. 1-14.
6. Athan T, Papalambros P. (1996). "A note on weighted criteria methods for compromise solutions in multi-objective optimization", *Engineering Optimization*, Vol. 27, Issue 2, ISSN: 0305-215X, pp. 155-176.
7. Bana C. (1999). "The MACBETH approach: Basic ideas, software and an application", *Advances in Decision Analysis*, Kluwer Academic Publishers, Dordrecht. ISBN: 978-94-017-0647-6, pp. 131-157.
8. Benayoun R., Montgolfier J., Tergny J. and Larichev O. (1971). "Linear Programming with Multiple Objective Functions: Step Method ", *Mathematical Programming*, Vol. 1, ISSN: 0025-5610, pp. 136-375.

9. Berners-Lee R., Cailliau A., Luotonen F., Nielsen A. (1994). "Secret, The World Wide Web", Communications of the ACM 37, Vol. 8, ISSN: 0001-0782, pp.76-82.
10. Beuthe M., Scannella G. (2001). "Comparative Analysis of UTA Multicriteria Methods", European Journal of Operational Research, Vol. 130, Issue 2, ISSN: 0377-2217, pp. 246-262.
11. Bhargava K., King A., McQuay D. (1995). "DecisionNet: an architecture for modeling and decision support over the World Wide Web", Proceedings of 3rd International Conference on DSS, Hong Kong, ISBN: 978-1-4757-4416-3, pp. 22-23.
12. Bhargava K., Krishnan R. (1998). "The World Wide Web: opportunities for operations research and management science", INFORMS Journal on Computing, Vol. 10, ISSN: 1091-9856, pp. 359-383.
13. Bhargava K., Krishnan R., Muller R. (1997). "Decision support on demand: emerging electronic markets for decision technologies", Decision Support Systems 19, ISSN: 0167-9236, pp. 193-214.
14. Bhargava K., Power J. (2001). "Decision Support Systems and Web Technologies: A Status Report", AMCIS 2001 Proceedings. Paper 46. ISSN: 1386-5056, pp. 229-235.
15. Bhargava K., Power J., Sun D. (2007). "Progress in Web-based decision support technologies", Decision Support Systems, Volume 43, Issue 4, ISSN: 0167-9236, pp. 1083-1095.
16. Biron, Paul V., Malhotra A. (2004). "XML Schema Part 2: Datatypes", W3C Recommendation. W3C, second edition, October 2004 (<http://www.w3.org/TR/xmlschema-2/>).
17. Boyko V., Rudnichenko N., Kramskoy S., Hrechukha Y., Shibaeva N. (2017) "Concept Implementation of Decision Support Software for the Risk Management of Complex Technical System", In: Shakhovska N. (eds) Advances in Intelligent Systems and Computing. Advances in Intelligent Systems and Computing, Vol. 512. Springer, Cham. ISBN: 978-3-319-45990-5, pp. 255-269.
18. Brans, J. P., Mareschal, B. (1994). "The PROMCALC & GAIA Decision Support System for Multicriteria Decision Aid", Decision Support System, Vol. 12, ISSN: 0167-9236, pp. 297-310.
19. Buchanan, J. T. (1997). "A Naive Approach for Solving MCDM Problems: The GUESS Method", Journal of the Operational Research Society, Vol. 48, ISSN: 0160-5682, pp. 202-206.

20. Bui T.X. (1997). "Decision support in the future tense", *Decision Support Systems*, Vol. 19, Issue 3, ISSN: 0167-9236, pp. 149-150.
21. Carlon C1., Critto A., Ramieri E., Marcomini A. (2007). "DESYRE: DEcision Support sYstem for the REhabilitation of contaminated megasites", *Integrated Environmental Assessment and Management*, Vol. 3, Issue 2, ISSN: 1551-3793, pp. 211-222.
22. Chang K., Dasari A., Madduri H., Mendoza A., Mims J. (2004). "Design of an enablement process for on demand applications", *IBM Systems Journal*, Vol. 43, ISSN: 0018-8670, pp. 190-203.
23. Chankong, V., Haimes, Y. (1983). "Multiobjective Decision Making. Theory and Methodology", Elsevier, ISBN: 0-486-46289-7, pp. 285-291.
24. Christensen E., Curbera F., Meredith G., Weerawarana S., (2001). "Web Services Description Language (WSDL) 1.1", (<http://www.w3.org/TR/wsdl>)
25. Clemen T. (1996). "Making Hard Decisions: An Introduction to Decision Analysis" 2nd edition, Duxbury Press, Belmont, ISBN: 978-0534260347, pp. 79-96.
26. Climaco J., Antunes H., Alves C., Maria J. (1997). "From TRIMAP to SOMMIX – Building Effective Interactive MOLP Computational Tools", In *Multicriteria Decision Making. Lecture Notes in Economics and Mathematical Systems*, (Fandel and Gal, Eds.), Vol. 448, Springer Verlag, Berlin, ISSN: 0075-8442, pp. 285-296.
27. Coello C., Christiansen A. (1999). "MOSES: A Multiobjective Optimization Tool for Engineering Design", *Engineering Optimization*, Vol. 31, Issue 3, ISSN: 0305-215X, pp. 337-368.
28. Coello C., Veldhuizen V., Lamont G. (2002). "Evolutionary Algorithms for Solving Multi-Objective Problems", Kluwer Academic Publishers, Boston, MA, ISBN: 978-0-387-33254-3, pp. 515-545.
29. Collette Y., Siarry P. (2013). "Multiobjective Optimization: Principles and Case Studies", Springer Science & Business Media. ISBN: 3662088835, pp. 77-98.
30. Davis G. (1974). "Management Information Systems: Conceptual Foundations, Structure, and Development", New York: McGraw-Hill, ISBN: 9780077550714, pp. 84-96.

31. Dellino G., Laudadio T., Mari R., Mastronardi N., Meloni C. (2017). "A reliable decision support system for fresh food supply chain management", *International Journal of Production Research*, Vol. 56, Issue 4, ISSN: 0020-7543, pp. 1458-1485.
32. Dias C., Clímaco N. (2000). "Additive Aggregation with Variable Interdependent Parameters: the VIP Analysis Software", *Journal of the Operational Research Society*, Vol. 51, Issue 9, ISSN: 0160-5682, pp. 1070-1082.
33. Dias C., Mousseau V. (2003). "IRIS: A DSS for Multiple Criteria Sorting Problems", *Journal of Multi-Criteria Decision Analysis*, Vol. 12, ISSN: 1099-1360, pp. 285-298.
34. Don Box (2000), Inside SOAP, (<http://www.xml.com/pub/a/2000/02/09/feature/index.html>)
35. Dordrecht K. (1999). "Mathematical Modeling: Theory and Applications" Vol. 4, Academic Publishers, ISSN: 1386-2960, pp. 131-157.
36. Dyer J. (2004). "MAUT: Multiattribute Utility Theory", In: *Multiple Criteria Decision Analysis: State of the Art Surveys*. (J. Figueira, S. Greco, and M. Ehrgott, Eds), Springer Verlag, London, ISBN 978-0-387-23081-8, pp. 265-285.
37. Ehrgott M., Wiecek M. (2004). "Multiobjective Programming", In: *Multiple Criteria Decision Analysis: State of the Art Surveys*, (J. Figueira, S. Greco, and M. Ehrgott, Eds), Springer Verlag. London, ISBN 978-0-387-23081-8, pp. 667-722.
38. Farquhar P. (1984). "Utility Assessment Methods. *Management Science*", 30, ISSN: 0025-1909, pp. 1283-1300.
39. Fielding R. (2000). "Architectural Styles and the Design of Network-based Software Architectures", PhD thesis, University of California, Irvine, USA, 2000.
40. Filip G., Zamfirescu B., Ciurea, C. (2017). "Computer-Supported Collaborative Decision-Making Automation", *Collaboration & E-Services Series*, Vol. 4, Springer International Publishing AG 2017, ISBN 978-3-319-47221-8.
41. Fishburn P. C. (1970). "Utility Theory for Decision Making", John Wiley & Sons, New York, ISBN: 471-26060-6.

42. Genova K., Kirilov, L., Guliashki, V., (2013). "New Reference – Neighborhood Scalarization Problem for Multiobjective Integer Programming", *Cybernetics and Information Technologies*, ISSN: 1311-9702, Vol. 13, Issue 1, pp. 104-114.
43. Genova K., Kirilov, L., Guliashki, V., Staykov, B., Vaton, D. (2011). "A Prototype of a Web-based Decision Support System for Building Models and Solving Optimization and Decision Making Problems", *Proceedings of XII International Conference on Computer Systems and Technologies - CompSysTech'11* (Eds. B. Rachev, A. Smrikarov), Wien, Austria, 16-17 June, 2011, ACM ICPS vol. 578, ISBN 978-1-4503-0917-2, pp. 167-172.
44. Genova, K., Vassilev, V., Andonov F., Chipilev, G. (2003). "Multicriteria Analysis Software System", *Proceeding of the International Conference "Automatics and Informatics'03"*, (Eds. Sendov B. et al.), Sofia, Bulgaria, Vol.1, IBN: 027119, pp. 135-138.
45. Genova, K., Vassilev, V., Andonov, F., Vassileva, M., Konstantinova, S. (2004). "A multicriteria analysis decision support system", *Proceeding CompSysTech '04 - 5th international conference on Computer systems and technologies*, ISBN:954-9641-38-4, pp. 1-6.
46. Genova, K., Vassileva, M., Vassilev, V., Andomov, F. (2003). "Linear multicriteria decision support system", *Proceeding CompSysTech '03 - 4th International Conference on Computer Systems and Technologies: e-Learning*, ISBN:954-9641-33-3, pp. 352-358.
47. Goldberg E. (1989). "Genetic Algorithms in Search, Optimization and Machine Learning", Addison-Wesley Publishing Co, ISBN: 0201157675.
48. Gong, X. (2011). "Chebyshev scalarization of solutions to the vector equilibrium problems", *Journal of Global Optimization*, Vol. 49, Issue 4, ISSN: 0925-5001, pp. 607-622.
49. Goul M., Philippakis A., Kiang M., Fernandes D., Otondo B. (1995). "Towards a client/server open-DSS protocol suite for automating DSS deployment on the World Wide Web", *Proceedings of 3rd International Conference on DSS*, Hong Kong, June, 1995, pp. 22-23.
50. Gouljashki, V., Kirilov, L., Vassilev, V., Narula, S. (1997). "An interactive algorithm for solving multiple objective nonlinear integer programming problems", *Multiple Criteria Decision Making* (G. Fandel&T. Gal - Eds.), Springer-Verlag, Berlin, ISBN 978-3-642-59132-7, pp. 308-317.

51. Guliashki V., Genova K., Kirilov L. (2013). "The Decision Support System WebOptim in an E-Learning Context", In: Proceedings of Papers of the International Conference "Automatics and Informatics'2013", 03.-07. October 2013, Sofia, Bulgaria, ISSN: 1313-1850, pp. I-117-I-120.
52. Guliashki V., Genova K., Kirilov L. (2013). "Hybrid Evolutionary Algorithm for Integer Multiple-Objective Optimization Problems", In: Proceedings of Papers of the XLVIII International Scientific Conference on Information, Communication and Energy Systems and Technologies ICEST2013, (Editor Prof. Dr. Mitrovski, C.), June 26 – 29, 2013, Ohrid, Macedonia, Vol. 1, ISBN: 978-9989-786-90-7, pp. 253-256.
53. Guliashki V., Kirilov L. (2011). "An Interactive Evolutionary Algorithm for Multiple Objective Convex Integer Problems", Proceedings of XII International Conference on Computer Systems and Technologies - CompSysTech'11 (Eds. B. Rachev, A. Smrikarov), Wien, Austria, 16-17 June, 2011, ACM ICPS Vol. 578, ISBN 978-1-4503-0917-2, pp. 82-87.
54. Guliashki V., Kirilov L. (2015). "A Reference Point Genetic Algorithm for Multi-Criteria Job Shop Scheduling Problems", In: Proceedings of the International Conference Information Technologies (InfoTech 2015), 29-th issue, (Editor Prof. Radi Romanski), September, 2015, Varna – St. St. Constantine and Elena resort, Bulgaria, ISSN: 1314-1023, pp. 10-18.
55. Guliashki V., Kirilov, L. (2011). "An Interactive Evolutionary Algorithm for Multiple Objective Convex Integer Problems", Proceedings of XII International Conference on Computer Systems and Technologies - CompSysTech'11 (Eds. B. Rachev, A. Smrikarov), Wien, Austria, 16-17 June, 2011, ACM ICPS vol. 578, ISBN 978-1-4503-0917-2, pp. 82-87.
56. Guliashki V., Kirilov, L., Genova, K. (2012). "An evolutionary algorithm for integer multicriteria optimization (EVALIMCO)", In: In: World Scientific Proceedings Series on Computer engineering and Information science-vol.7, Uncertainty Modeling in Knowledge Engineering and Decision Making, Proceedings of the 10th International FLINS Conference, (Eds.: C. Kahraman, E. Kerre, F. Bozbura) Istanbul, Turkey 2012, ISBN 978-981-4417-73-0, pp. 118-123.
57. Guliashki, V., Kirilov, L., (2009). "SPEA - Based Method for MCDM Convex Integer Problems", Cybernetics and Information Technologies, ISSN: 1311-9702, Vol. 9, Issue 4, pp. 93-101.

58. Guliashki, V., Kirilov, L., Genova K. (2013). „An Interactive Evolutionary Algorithm for Multiple Objective Integer Problems”, *International Journal on Information Technologies and Security*, Vol. 2, ISSN: 1313-8251, pp. 45-54.
59. Hanne T., Trinka L. (2003). “KnowCube for MCDM – Visual and Interactive Support for Multicriteria Decision Making”, *Published Reports of the Fraunhofer ITWM*, 50 (<https://kluedo.ub.uni-kl.de/frontdoor/index/index/docId/1507>)
60. Holsapple C., Whinston A. (1995). “Exploring the next generation of decision support”, *Decision Support Systems*, Vol. 14, Issue 3, ISSN: 0167-9236, pp. 185-186.
61. Hosack B., Hall D., Paradise D., Courtney F. (2012). “A look toward the future: Decision Support Systems research is alive and well”, *Journal of the Association for Information Systems*, Vol. 13, ISSN: 1536-9323, pp. 315-340.
62. Ishibuchi H., Murata T. (1998). “Multi-Objective Genetic Local Search Algorithm and Its Application to Flowshop Scheduling”, *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 28, Issue 3, ISSN: 1094-6977, pp. 392-403.
63. Jaskiewicz A., Slowinski R. (1997). “The LBS-Discrete Interactive Procedure for Multiple Criteria Analysis of Decision Problems”, *Multicriteria Analysis (J. Climaco, Ed.)*, Springer-Verlag, Berlin, ISBN: 978-3-642-64500-6, pp. 320-330.
64. Jaskiewicz A., Slowinski R. (1999). “The Light Beam Search Over a Non-Dominated Surface of a Multiple-Objective Programming Problem”, *European Journal of Operational Research* 113(2), ISSN: 0377-2217, pp. 300-314.
65. Jeusfeld A., Bui X. (1995). “Decision support components on the Internet”, *Proceedings of 3rd International Conference on DSS*, Hong Kong, 22–23 June, 1995.
66. Kalakota B., Whinston B. (1995). “Executable documents DSS”, *Proceedings of 3rd International Conference on DSS*, Hong Kong, 22–23 June, 1995.
67. Karaivanova J., Korhonen P., Narula S., Wallenius J., Vassilev V. (1995). “A Reference Direction Approach to Multiple Objective Integer Linear Programming”, *European Journal of Operational Research*, Vol. 24, ISSN: 0377-2217, pp. 176-187.

68. Keeney R., Raiffa H. (1993). "Decisions with Multiple Objectives", Cambridge University Press, ISBN 052143883, pp. 219-223.
69. Kenney C., Garson A., Aggarwal S. (2002). "Place Prioritization for Biodiversity Reserve Network Design: A Comparison of the SITES and ResNet Software Packages for Coverage and Efficiency", Diversity and distributions, Vol. 8, ISSN: 1472-4642, pp. 297-306.
70. Khalili-Damghani K., Abtahi R., Tavana M. (2014). "A Decision Support System for Solving Multi-Objective Redundancy Allocation Problems", Quality and Reliability Engineering International, Vol. 30, Issue 8, ISSN: 0951-8320, pp. 1249–1262.
71. Kiker1 G., Bridges T., Varghese A., Seager T., Linkov I. (2005). "Application of multicriteria decision analysis in environmental decision making", Integrated Environmental Assessment and Management Vol. 1, Issue 2, ISSN: 1551-3777, pp. 95-108.
72. Kirilov L. (1993). "An Interactive Method for Solving Multiple Objective Programming Problems", Proceedings of the International Conference on Mathematical Modelling and Scientific Computation MMSC'93, September 14-18, 1993, Sozopol, Bulgaria; S. Markov (ed.), DATECS Publishing, Sofia, 1993, ISSN: 1573-1340, pp. 129-132.
73. Kirilov L., Guliashki V. (2014). "An extension of flexible job shop problem (FJSP) and method for solving", Proceedings of CompSysTech '14 Proceedings of the 15th International Conference on Computer Systems and Technologies, ACM New York, NY, USA ©2014, ACM International Conference Proceeding Series, Vol. 883, ISBN: 978-1-4503-2753-4, pp. 210-217.
74. Kirilov L., Guliashki, K., Genova, K. (2011). "Interactive Evolutionary Algorithm for Multiple Objective Convex Integer Problems", Proceedings of XLVI International Scientific Conference on Information, Communication and Energy Systems and Technologies - ICEST2011 (Ed. B. Milovanovic), Nis, Serbia, 29 June – 1 July 2011, Vol.2, CS.2, ISBN 978-86-6125-032-3, pp. 421-424.
75. Kirilov L., Guliashki, V. (2011). "An Interactive Evolutionary method FIEM for Solving Integer Multiple Objective Problems", Comptes Rendue de l'Academie Bulgare des Sciences, Vol. 64, Issue 2, ISSN: 1310-1331, pp. 203-212.
76. Kirilov L., Guliashki, V., Genova, K. (2016). "Multicriteria Decision Making in Manufacturing Scheduling", Publuser Obrazovanie/Education/ Ltd, Sofia, 281 pgs., ISBN 978-954-552-074-7.

77. Kirilov L., Guliashki, V., Genova, K., Zhivkov, P., Staykov, B., Vaton, D. (2015). "Interactive environment WebOptim for solving multiple-objective problems using scalarizing and evolutionary approaches", *International Journal of Reasoning-based Intelligent Systems (Special Issue on Applied Formal Methods in Computer, Control, and Communications Systems, Vol. 7, Issue 1/2, ISSN: 1755-0556, pp. 4-15.*
78. Kirilov L., Miettinen, K. (2005). "Solutions' Properties and Numerical Testing of an Interactive Method REF-LEX", *Proceedings of the Int. Conference on Computer Systems and Technologies - CompSysTech'05 (Eds.: B. Rachev, A. Smrikarov), June 2005, Varna, Bulgaria, ISBN 954-9641-42-2, pp. IIIA9.1-III A9.6.*
79. Kirilov L., Vassilev V. (1997). "A Method for Solving Multiple Objective Linear Programming Problems", *Multicriteria Analysis, (J. Climaco - Ed.), Springer-Verlag, Berlin, ISBN 978-3-642-60667-0, pp. 302-309.*
80. Kirilov L., Vassilev, V., Marinchev, I. (2003). "A Program System for Solving Multiple Attribute Decision Making Problems", *IIT Working Papers-164, November 2003, ISSN 1310-652X.*
81. Kirilov, L. (1993). "Interactive Methods for Solving Multiple Objective Mathematical Programming Problems", *Ph. D. Thesis, Institute of Informatics - Bulgarian Academy of Sciences, Sofia, Bulgaria, Central Research and Technical Library, DC5577, inv. NoDC116/93.*
82. Kirilov, L. (1993). „An Interactive Method for Solving Multiple Objective Programming Problems“, *Proceedings of the International Conference on Mathematical Modelling and Scientific Computation MMSC'93, September 14-18, 1993, Sozopol, Bulgaria, S. Markov (ed.), DATECS Publishing, Sofia, 1993, ISSN: 1573-1340, pp. 129-132.*
83. Kirilov, L. (1999). "Reference Direction Method for Solving Nonlinear Multicriteria Problems" – Some Results, *Proceedings of the XXXIV-th scientific conference "Communication, electronic and computer systems '99", Technical University, Sofia, Bulgaria, May, 1999, pp. 376-381.*
84. Kirilov, L. (2000). "A Method for Solving Nonlinear Multiple Objective Problems", *proceedings of 35-th scientific conference "Communication, electronic and computer systems '2000", Sofia, Bulgaria, May, 2000, vol. 2, pp. 143-147.*

85. Kirilov, L., Dokev, N. (2002). "A Lexicographic ordering approach in multiple criteria decision support", Proceedings of 2002 First Int. IEEE Symposium Intelligent Systems (Eds. T. Samad, V. Sgurev), Varna, Bulgaria, September 10-12, vol. 1, pp. 314-319.
86. Kirilov, L., Koynov, S., Toshev, H. (2002). "Multiple Criteria Decision Making Problems and Neural Networks", Proceedings of the XXXVII Int. Scientific Conf. on Information, Communication and Energy Systems and Technologies-ICEST 2002, (Ed. B. Milovanovic), Nis, Yugoslavia, pp. 121-124.
87. Kirilov, L., Miettinen, K. (2003). "Reference Direction Approach Using Implicit Parametrization for Nonlinear Multiobjective Optimization", Publications of the University of Jyväskylä, Finland, Series B. Scientific Computing, No. B16/2003
88. Kirilov, L., Tontchev, N., Monov, Al. (2004). "Multiple Criteria Approach to Solving Problems of Metal Building-up by Welding", Proceedings of the 4-th IFAC Workshop DECOM-TT 2004 – Automatic Systems for Building the Infrastructure in Developing Countries (Regional and Global Aspects), (Eds. V. Sgurev. G. Dimitrovski, M. Hadjiiski), Bansko - Bulgaria, October 2004, pp. 255-260.
89. Kirilov, L., Vassilev, V. (1996). „An Approach for Solving Discrete Multiple Criteria Problems“, Problems of Engineering Cybernetics and Robotics, Vol. 44, ISSN: 0204-9848, pp. 97-104.
90. Kirilov, L., Vassilev, V. (1997). "A Method for Solving Multiple Objective Linear Programming Problems", Multicriteria Analysis (J. Climaco - Ed.), Springer-Verlag, Berlin, ISBN: 978-3-642-64500-6, pp. 302-309.
91. Kirilov, L., Vassilev, V., Narila, S. (2000). "An Approach for Solving MultiCriteria Choice Problems", Proceeding of 4th Balkan Conference on Operational Research (Ed. B. Papathanassiou), Omirus-Graph, Thessaloniki, Greece, Vol. 1, pp. 289-300.
92. Knowles D., Corne W. (2000). "M-PAES: A Memetic Algorithm for Multiobjective Optimization", Proceedings of the 2000 Congress on Evolutionary Computation, vol. 1, Piscataway, New Jersey, ISBN: 0-7803-6375-2, pp. 325-332.
93. Koksalan M., Karwan M., Zionts S. (1984). "An Improved Method for Solving Multiple Criteria Problems Involving Discrete Alternatives", IEEE Transactions on Systems, Man and Cybernetics Vol. 14, Issue: 1, ISSN: 0018-9472, pp 24-34.

94. Korhonen P. (1987). "VIG - A Visual Interactive Support System for Multiple Criteria Decision Making", *Belgian Journal of Operations Research, Statistics and Computer Science*, Vol. 27, Issue 1, ISSN: 0770-0512, pp. 3-15.
95. Korhonen P. (1988). "A Visual Reference Direction Approach to Solving Discrete Multiple Criteria Problems", *European Journal of Operational Research*, ISSN: 0377-2217-34, pp. 152-159.
96. Lewandowski A., Kreglewski T., Rogowski T., Wierzbicki A. (1989). "Decision Support Systems of DIDAS Family", *Lecture Notes in Economics and Mathematical Systems* Vol. 331, ISBN: 978-3-540-51213-4, pp 21-47.
97. Lotfi V., Stewart J., Zionts S. (1992). "An Aspiration-Level Interactive Model for Multiple Criteria Decision Making", *Computers and Operations Research*, Vol. 19, ISSN: 0305-0548, pp. 671-681.
98. Lotov V., Bushenkov A., Chernykh L. (1997). "Multicriteria DSS for River Water–Quality Planning", *Computer-Aided Civil and Infrastructure Engineering*, Vol. 12, Issue 1, ISSN: 1467-8667, pp. 57–67.
99. Marcotte O., Soland R. (1986). "An Interactive Branch and Bound Algorithm for Multiple Criteria Optimization", *Management Science*, Vol. 32, ISSN: 00251909, pp. 61-75.
100. Marinchev I., Kirilov L. (2003). "Interactive Computer System for Solving Problems with Multiple Criteria", *Proceedings of the XXXVIII Int. Conf. on Information, Communication and Energy Systems and Technologies –ICEST’2003*, Heron Press, Sofia, ISBN: 9545801468, pp. 359-362.
101. Marinchev I., Kirilov L. (2003). "Interactive Computer System for Solving Problems with Multiple Criteria", *Proceedings of the XXXVIII Int. Conf. on Information, Communication and Energy Systems and Technologies –ICEST’2003*, Heron Press, Sofia, ISBN: 954-580-146-8, pp. 359-362.
102. Marler R., Arora J. (2004). "Survey of multi-objective optimization methods for engineering", *Structural and Multidisciplinary Optimization*, April 2004, Vol. 26, Issue 6, ISSN: 1615-147X, pp. 369-395.

103. Marler R., Arora J. (2010). "The weighted sum method for multi-objective optimization: New insights", *Structural and Multidisciplinary Optimization*, Vol. 41, Issue 6, ISSN: 1615-147X, pp. 853-862.
104. Metev, B., Vassilev, V. (2003). "A Method for Nadir Point Estimation in MOLP Problems", *Cybernetics and Information Technologies*, Vol. 2, ISSN: 1311-9702, pp. 15-24.
105. Miettinen K. (1994). "On the Methodology of Multiobjective Optimization with Applications", Report 60 (doctoral thesis), University of Jyväskylä, Department of Mathematics, Jyväskylä, pp. 48-56.
106. Miettinen K. (1999). "Nonlinear Multiobjective Optimization", Kluwer Academic Publishers, Boston. ISBN 978-1-4615-5563-6, pp. 146-158.
107. Miettinen K., Makela M. (2000). "Interactive Multiobjective Optimization System WWW-NIMBUS on the Internet", *Computer and Operation Research*, Vol. 27, ISSN: 0305-0548, pp. 709-723.
108. Miettinen K., Makela M. (2001). "Classification and Reference Point Based Scalarizing Functions in Multiobjective Optimization", Technical Report of University of Javaskyla, N8.
109. Miettinen K., Mustajoki, J., Stewart J. (2014). "Interactive Multiobjective Optimization with NIMBUS for Decision Making under Uncertainty", *OR Spectrum*, Vol. 36, Issue 1, ISSN: 0171-6468, pp. 39-56.
110. Miettinen, K., Kirilov, L. (2005). "Interactive Reference Direction Approach Using Implicit Parametrization for Nonlinear Multiobjective Optimization", *Journal of Multi-Criteria Decision Analysis*, Vol.13, ISSN:1099-1360, pp. 115-123.
111. Mitra V. (2007). "The evolution of web-based optimization: From ASP to e-Services", *Decision Support Systems*, Vol. 43, Issue 4, ISSN: 0167-9236, pp. 1096-1116.
112. Moffett A., Garson J., Sarkar S. (2005). "MultCSync: A Software Package for Incorporating Multiple Criteria in Conservation Planning", *Environmental Modelling & Software archive*, Vol. 20, Issue 10, October, 2005, ISSN:1364-8152, pp. 1315-1322.
113. Mustajoki J., Hamalainen P. (2000). "Web-HIPRE: Global Decision Support by Value Tree and AHP Analysis", *INFOR*, Vol. 38, ISBN: 9-7895-1228-5150, pp. 208-220.

114. Naik K., Mohindra A., Bantz F. (2004). "An architecture for the coordination of system management services", IBM Systems Journal, Vol. 43, ISSN: 0018-8670, pp. 78-96.
115. Nakayama H. (1995). "Aspiration Level Approach to Interactive Multi-Objective Programming and its Applications", Advances in Multicriteria Analysis (P. Pardalos, Y. Siskos and C. Zopounidis, Eds.), Kluwer Academic Publishers, ISBN: 1475723830, pp. 147-174.
116. Narula S., Kirilov L., Vassilev V. (1992). "Reference Direction Approach for Solving Multiple Objective Nonlinear Programming Problems", Proceedings of the X International Conference on Multiple Criteria Decision Making, Taipei, Taiwan, July 19-24, 1992, Vol. 2, ISBN 978-1-4612-2666-6, pp. 355-362.
117. Narula S., Kirilov L., Vassilev V. (1994). "An Interactive Algorithm for Solving Multiple Objective Nonlinear Programming Problems", paper from the book Multiple Criteria Decision Making (G. Tzeng, H. Wang, U.Wen, P.L.Yu - eds.), Springer-Verlag, Berlin, ISBN: 978-1-4612-7626-5 pp. 119-128.
118. Narula S., Vassilev V. (1994). "An Interactive Algorithm for Solving Multiple Objective Integer Linear Programming Problems", European Journal of Operational Research, Vol. 79, ISSN: 0377-2217, pp. 443-450.
119. Narula S., Vassilev V., Genova K., Vassileva M. (2003). "A partition-based interactive method to solve discrete multicriteria choice problems", Cybernetics and Information Technologies, Vol. 3, Issue 2, ISSN: 1311-9702, pp. 55-66.
120. Narula, S., Kirilov, L., Vassilev, V. (1992). "Reference Direction Approach for Solving Multiple Objective Nonlinear Programming Problems". Proceedings of the Tenth International Conference on Multiple Criteria Decision Making, Taipei, Taiwan, July 19-24, 1992, Vol. 2, ISBN: 978-1-4612-2666-6, pp. 355-362.
121. Narula, S., Kirilov, L., Vassilev, V. (1994). "An Interactive Algorithm for Solving Multiple Objective Nonlinear Programming Problems", Multiple Criteria Decision Making (G. Tzeng, H. Wang, U.Wen, P.L.Yu - eds.), Springer-Verlag, Berlin, ISBN 978-3-642-87563-2, pp. 119-128.
122. Narula, S., Kirilov, L., Vassilev, V. (1994). "Reference Direction Approach for Solving Multiple Objective Nonlinear Programming Problems", IEEE Transactions on Systems, Man, and Cybernetics, Vol. 24, Issue 5, ISSN: 2168-2216, pp. 804-806.

123. Narula, S., Vassilev, V., Genova, K., Vassileva, M. (2003). "A Partition – Based Interactive Method to Solve Discrete Multicriteria Choice Problems", *Cybernetics and Information Technologies*, Vol. 2, ISSN: 1311-9702, pp. 55-66.
124. Padberg M. (2000). "Linear Optimization and Extensions", *Algorithms and Combinatorics*, Vol. 12. Springer-Verlag, ISBN 978-3-662-12273-0, pp. 114-128.
125. Parsons J. (2002). "Agland Decision Tool: A Multicriteria Decision Support System for Agricultural Property", *iEMSs 2002, Proceedings of the 1st biennial meeting of the International Environmental Modelling and Software Society*, Lugano, Switzerland, June 2002, ISBN 88-900787-0-7, 3, pp. 181-187.
126. Power D. (1998). "Finding DSS Resources on the World-Wide Web", *Journal of Decision Systems*, Vol. 7, ISSN: 1246-0125, pp. 7-20.
127. Power D. (2002). "Decision Support Systems: Concepts and Resources for Managers", *Greenwood/Quorum Books*, Westport, CT, ISBN: 156720497X, pp. 36-49.
128. Power D. (2003). "A Brief History of Decision Support Systems", (<http://DSSResources.COM/history/dsshhistory.html>)
129. Power D. (2008). "Decision support systems: a historical overview", *Handbook on Decision Support Systems*, Springer Berlin Heidelberg. ISBN 978-3-540-48712-8, pp. 121-140.
130. Proos, A., Steven, P., Querin, M., Xie, M. (2001). "Multicriterion evolutionary structural optimization using the weighted and the global criterion methods". *AIAA Journal*, Vol. 39, Issue 10, ISSN: 0001-1452, pp. 2006-2012.
131. Roy B. (1991). "The Outranking Approach and Foundations of ELECTRE Methods", *Theory and Decision*, Vol. 31, Issue 1, ISSN: 0040-5833, pp. 49-73.
132. Roy B. (1996). "Multicriteria Methodology for Decision Aiding", *Kluwer*, ISBN 978-1-4757-2500-1, pp. 86-103.
133. Saaty T. (1980). "The Analytic Hierarchy Process", *McGraw-Hill*, New York. ISBN: 978-0070543713, pp. 181-196.

134. Saaty T. (1994). "Highlights and Critical points in the Theory and Application of the Analytic Hierarchy Process", *European Journal of Operational Research*, 74, ISSN: 0377-2217, pp. 426-447.
135. Sawaragi Y., Nakayama H., Tanino T. (1985). "Theory of Multiobjective Optimization. Elsevier", ISBN: 0080958664, pp. 72-84.
136. Sgurev, V., Vassilev, V., Atanasov, A., Kichovich, M., Kirilov, L., Vladimirov, P., Genova, K., Nikolova, M., Vachkov, G. (1990). "Program product for multicriteria linear programming MOLP-16", Software Products and Systems Corporation – Bulgarian Academy of Sciences, Sofia, 1990 (in English, also in Russian and Bulgarian).
137. Shim J., Warkentin M., Courtney J., Power D., Sharda R., Carlsson C. (2002) "Past, present, and future of decision support technology", *Decision Support Systems*, Special Issue, DSS: Directions for the Next Decade, Vol. 33, Issue 2, ISSN: 0167-9236, pp. 111-126.
138. Sierksma G. (2001). "Linear and Integer Programming: Theory and Practice", ISBN-0-8241-0673-0, pp. 189-216.
139. Singiresu S., Rao S. (2009). "Engineering Optimization: Theory and Practice", 4th edition, ISBN 978-0-470-18352-6, pp. 124-132.
140. Srinivas N., Deb K. (1994). "Multiobjective Optimization using Nondominated Sorting Genetic Algorithms", *Evolutionary Computation*, Vol. 2, Issue 3, ISSN 1063-6560, pp. 221-248.
141. Staykov B., Andonov, F., Vatov, D., Genova, K., Kirilov, L., Guliashki, V. (2011). "Architecture of a flexible web-based framework for building models and solving decision optimization problems", *Proceedings of XLVI International Scientific Conference on Information, Communication and Energy Systems and Technologies - Icest2011* (Ed. B. Milovanovic), Nis, Serbia, 29 June – 1 July 2011, Vol.2, PO6.10, ISBN 978-86-6125-033-0, pp. 857-860.
142. Staykov B., Kirilov, L., Genova, K., Guliashki, V., Vatov. D. (2012), "Modules and inter Modular Communication in a Web-Based System for Solving Decision Problems", In: *Proceedings of the International Conference on Information Technologies (InfoTech-2012)*, Varna, Bulgaria, September 2012, ISSN: 1314-1023, pp. 67-75.

143. Steuer R. (1986). "Multiple Criteria Optimization: Theory, Computation, and Applications", John Wiley & Sons, Inc, ISBN: 047188846X, pp. 313-325.
144. Sun M., Steuer R. (1996). "InterQuad: An Interactive Quad Free Based Procedure for Solving the Discrete Alternative Multiple Criteria Problem", *European Journal of Operational Research*, Vol. 89, ISSN: 0377-2217, pp. 462-472.
145. Tan K., Lee T., Khoo D., Khor E. (2000). "MOEA toolbox for computer aided multi-objective optimization. *Evolutionary Computation*", *Proceedings of the 2000 Congress*, Vol.1, ISBN: 0-7803-6375-2, pp. 38-45.
146. Tonchev N., Kirilov, L. (2007). "Two approaches for solving MCDM problems with an illustrative example", *Problems of Technical Cybernetics and Robotics*, Vol.58, ISSN: 0204-9848, pp.53-63.
147. Tontchev, N., Kirilov, L. (2005). "Two methods for solving multiple criteria decision making (MCDM) problems", *Proceedings of the Int. Conference on Advanced Manufacturing Technologies AMTECH-2005* (Ed. B. Tomov), Technical University, Russe, Bulgaria, Vol. 44, Issue 2, pp. 18-23.
148. Tontchev, N., Kirilov, L., Miettinen, K. (2008). "Modelling of Problems of Metal Building-Up by Welding with Multiple Criteria", *IIT Working Papers*, IIT/WP-247, November 2008, ISSN 1310-652X.
149. Tsangaratos P., Kallioras A., Pizpikis T., Vasileiou E., Ilia I., Pliakas F. (2017). "Multi-criteria Decision Support System (DSS) for optimal locations of Soil Aquifer Treatment (SAT) facilities", *Science of The Total Environment*, Vol. 603, ISSN: 0048-9697, pp. 472-486.
150. Vansnick J. (1986). "On the Problem of Weights in Multiple Criteria Decision Making (the noncompensatory approach)", *European Journal of Operational Research*, Vol. 24 ISSN: 0377-2217, pp. 288-294.
151. Vassilev V., Atanasov A., Sgurev V., Kichovitch M., Deianov A., Kirilov L. (1993). "Software Tools for Multi-Criteria Programming", *User-Oriented Methodology and Techniques of Decision Analysis and Support*, (J. Wessels and A. Wierzbicki, Eds.), Springer- Verlag, Heidelberg, ISBN 3-530-56382-2, pp. 247-257.

152. Vassilev V., Djambov V., Kirilov L., Genova K. (1995). "A Decision Support Software Package for Multiple Objective Nonlinear Integer Programming", IIT Working Papers-2, October 1995, ISSN 1310-652X, pp. 23-30.
153. Vassilev V., Genova K., Andonov F., Staykov B. (2004). "Multicriteria Decision Support System MultiChoice", Cybernetics and Information Technologies, Vol. 4, Issue 1, ISSN: 1311-9702, pp. 65-75.
154. Vassilev V., Genova K., Vassileva M., Narula S. (2004). "An Interactive Method of Linear Mixed Integer Multicriteria Optimization", International Journal on Information Theories and Applications, Vol. 11, Issue 1, Online ISSN: 1097-007X, pp. 73-78.
155. Vassilev V., Narula S., Vladimirov P., Djambov V. (1997). "MOIP: A DSS for Multiple Objective Integer Programming Problems", In: Multicriteria Analysis (Ed.: J. Climaco), Springer-Berlin, ISBN: 978-3-642-64500-6, pp. 259-268.
156. Vassilev, V., Atanassov, A., Sgurev, V., Kichovich, M., Deianov, A., Kirilov, L. (1993). "Software Tools for Multi-Criteria Programming", User-Oriented Methodology and Techniques of Decision Analysis and Support, J. Wessels and A. Wierzbicki (Eds.), Springer-Verlag, Berlin, ISBN 978-3-662-22587-5, pp. 247-257.
157. Vassilev, V., Djambov, V., Kirilov, L., Genova, K. (1995). "A Decision Support Software Package for Multiple Objective Nonlinear Integer Programming", IIT Working Papers-2, October 1995, ISSN 1310-652X.
158. Vassilev, V., Genova, K. (1991). "An Algorithm of Integer Feasible Directions for Linear Integer Programming", European Journal of Operational Research (EJOR), Vol. 52, ISSN: 0160-5682, pp. 203-214.
159. Vassilev, V., Genova, K. (1994). "An Approximate Algorithm of Nonlinear Integer Programming", European Journal of Operational Research, Vol. 74, ISSN: 0377-2217, pp. 170-178.
160. Vassilev, V., Genova, K., Andonov, F., Staikov, B. (2004). "Multicriteria Decision Support System MultiChoice", Cybernetics and Information Technologies, Vol. 1, ISSN: 1311-9702, pp. 65-75.

161. Vassilev, V., Genova, K., Kirilov, L. (1998). "A Method for Solving a Class of Multiple – Criteria Analysis Problems", *Pliska studia mathematica bulgarica* (Eds.: P. Kenderov, J. Revalski), Vol. 12, ISSN 0204-9805, pp. 245-254.
162. Vassilev, V., Gouljashki, V., Kirilov, L. (1996). "An Interactive Algorithm of the Multiple Objective Nonlinear Integer Programming", *Problems of Engineering Cybernetics and Robotics*, Vol. 44, ISSN: 0204-9848, pp. 8-14.
163. Vassilev, V., Kirilov, L. (1995). "An Interactive Method for Solving Multiple Attribute Decision Making Problems", *Proceedings of the International Conference "Automation and Informatics '95"*, November 7-9, Sofia, Bulgaria, pp. 456-459.
164. Vassilev, V., Kirilov, L., Atanasov, A. (1992). "Reference Direction Approach for Solving Multiple Objective Programming Problems", *Problems of Engineering Cybernetics and Robotics*, Vol. 38, ISSN: 0204-9848, pp. 17-23.
165. Vassilev, V., Kirilov, L., Genova, K. (1997). "An Interactive Algorithm for Solving Multiple Attribute Choice Problem", *Proceedings of the Bulgarian-Russian seminar on Methods and Algorithms for Distributed Information Systems Design* (Eds.: Vishnevsky, V., H. Daskalova), RAS, Moscow, pp. 7-24.
166. Vassilev, V., Narula, S. (1993). „A Reference Direction Algorithm for Solving Multiple Objective Integer Linear Programming Problems“, *Journal of the Operational Research Society (JORS)*, Vol. 44, Issue 12, ISSN: 0160-5682, pp. 1201-1209.
167. Vassilev, V., Narula, S., Djambov, V., Vladimirov, P. (1997). "MOIP: A Decision Support System for Multiple Objective Linear Programming Problems", In: *Multicriteria Analysis* (Ed.: J. Climaco), Springer, ISBN: 978-3-642-64500-6, pp. 259-268.
168. Vassilev, V., Sgurev, V., Atanasov, A., Deianov, A., Kichovich, M., Kirilov, L., Djambov, V., Vachkov, G. (1990). "Program product for multicriteria nonlinear programming MONP-16", *User's Guide*, Software Products and Systems Corporation – Bulgarian Academy of Sciences, Sofia, 1990 (in English, also in Russian and Bulgarian).
169. Vassileva M. (2000). "Scalarizing Problems of Multiobjective Linear Programming Problems. *Problems of Engineering Cybernetics and Robotics*", Vol. 50, ISSN: 0204-9848, pp. 54-64.

170. Vassileva M. (2004). "A Learning-Oriented Method of Linear Mixed Integer Multicriteria Optimization. Cybernetics and Information Technologies, Vol 4. Issue 1, ISSN: 1311-9702, pp. 13-25.
171. Vassileva M. (2006). "Generalized Interactive Algorithm of Multicriteria Optimization", Problems of Engineering Cybernetics and Robotics, Vol. 56, ISSN 0204–9848, pp. 69-77.
172. Vassileva, M., Miettinen, K., Vassilev, V. (2005). "Generalized Scalarizing Problem GENWS". In Working Papers of IIT-BAS, N IIT/WP-205, ISSN 1310-652X, pp. 62-74.
173. Vincke P. (1992). "Multicriteria Decision-Aid", John Wiley & Sons, New York, ISBN: 978-0-471-93184-3, pp. 112-137.
174. Von Winterfeldt D., Edwards W. (1986). "Decision Analysis and Behavioral Research", London, Cambridge University Press, ISBN: 0-5212-7304-8, pp. 78-96.
175. White D. (1990) "A Bibliography of the Applications of Mathematical Programming Multiple Objective Methods", Journal of the Operational Research Society, Vol. 41, Issue 8, ISSN: 0160-5682, pp. 669-691.
176. Wierzbicki A. (1980). "The Use of Reference Objectives in Multiobjective Optimization", Multiple Criteria Decision-Making Theory and Application, Lecture Notes in Economics and Mathematical Systems, 177 (G. Fandel and T. Gal, Eds.). Springer-Verlag, Berlin, Heidelberg, ISBN: 978-3-540-09963-5, pp. 468-486.
177. Wiestroffer H., Narula S. (1997). "The state of Multiple Criteria Decision Support Software". Annals of operations Research, Vol. 72, ISSN: 0254-5330, pp. 299-313.
178. Xu D., Yang J. (2003). "Intelligent Decision System for Self-assessment", Journal of Multi-criteria Decision Analysis, Vol. 12, Issue 1, ISSN: 1099-1360, pp .43-60.
179. Yang. Z, Cai, X., Fan, Z. (2014). "Epsilon constrained method for constrained multiobjective optimization problems: some preliminary results", GECCO Comp '14 Proceedings of the Companion Publication of the 2014 Annual Conference on Genetic and Evolutionary Computation, ISBN: 978-1-4503-2881-4, pp. 1181-1186.

180. Zopounidis C., Despotis D., Kamaratou I. (1998). "Profilio Selection Using the Adelais Multiobjective Linear Programming System", Computational Economics, Vol. 11, Issue 3, ISSN: 0927-7099, pp. 189-204.
181. Zopounidis C., Doumpos M. (1998). "Developing a Multicriteria Decision Support System for Financial Classification Problems: The FINCLAS system", Optimization Methods and Software, Vol. 8, ISSN: 1055-6788, pp. 277-304.