



БЪЛГАРСКА АКАДЕМИЯ НА НАУКИТЕ
ИНСТИТУТ ПО ИНФОРМАЦИОННИ И
КОМУНИКАЦИОННИ ТЕХНОЛОГИИ

ПЕНКА ВЪЛКОВА ГЕОРГИЕВА

ИЗСЛЕДВАНЕ НА МОДЕЛИ
НА СОФТ КОМПЮТИНГ
ЗА УПРАВЛЕНИЕ В РЕАЛНО ВРЕМЕ

ДИСЕРТАЦИЯ

ЗА ПРИСЪЖДАНЕ НА ОБРАЗОВАТЕЛНА И НАУЧНА СТЕПЕН „ДОКТОР“

ПРОФЕСИОНАЛНО НАПРАВЛЕНИЕ 4.6. „ИНФОРМАТИКА И КОМПЮТЪРНИ
НАУКИ“

ПО НАУЧНА СПЕЦИАЛНОСТ 01.01.12 „ИНФОРМАТИКА“

НАУЧЕН РЪКОВОДИТЕЛ: АКАД. ИВАН ПОПЧЕВ

София, 2013 г.

СЪДЪРЖАНИЕ

ГЛАВА 1. УВОД, ЦЕЛ И ЗАДАЧИ НА ДИСЕРТАЦИЯТА	3
1.1. ФИНАНСОВО ИНВЕСТИРАНЕ.....	3
1.1.1. Възвращаемост.....	3
1.1.2. Инвестиционен риск.....	4
1.1.3. Ликвидност.....	4
1.1.4. Инвестиционен хоризонт.....	4
1.2. ИНВЕСТИЦИОННИ ТЕОРИИ И МОДЕЛИ.....	5
1.2.1. Портфейлна теория на Х. Марковиц (ПТ).....	5
1.2.2. Модел за оценяване на капиталовите активи (МОКА).....	6
1.2.3. Някои варианти на портфейлната теория.....	8
1.2.4. Сортиращи алгоритми.....	9
1.2.5. Алгоритми за локално търсене.....	10
1.3. НЯКОИ НЕРЕШЕНИ ИЛИ ЧАСТИЧНО РЕШЕНИ ПРОБЛЕМИ В ИНВЕСТИЦИОННИТЕ ТЕОРИИ И МОДЕЛИ.....	11
1.4. СОФТ КОМПЮТИНГ КАТО НАПРАВЛЕНИЕ В ИЗКУСТВЕНИЯ ИНТЕЛЕКТ.....	13
1.4.1. Изкуствен интелект.....	13
1.4.2. Софт компютинг.....	14
1.4.3. Основни направления в софт компютинг.....	15
1.4.4. Приложения на средствата на софт компютинг.....	18
1.4.5. Приложение на софт компютинг във финансовото инвестиране.....	21
1.5. КОМПЮТЪРНИ СИСТЕМИ, РАБОТЕЩИ В РЕАЛНО ВРЕМЕ.....	24
1.6. ЦЕЛ И ЗАДАЧИ НА ДИСЕРТАЦИОННИЯ ТРУД.....	25
ГЛАВА 2. ФИНАНСОВИ АКТИВИ. РАЗМИТИ СИСТЕМИ	27
2.1. ОЦЕНЯВАНЕ НА ФИНАНСОВИ АКТИВИ.....	27
2.1.1. Характеристики на финансов актив.....	27
2.1.2. Инвестиционен портфейл без къси продажби.....	30
2.2. РЕШАВАНЕ НА ПОРТФЕЙЛНАТА ЗАДАЧА В РЕАЛНО ВРЕМЕ.....	32
2.3. РАЗМИТА ЛОГИКА И РАЗМИТИ СИСТЕМИ.....	33
2.3.1. Размити множества.....	33
2.3.2. Размита логика.....	35
2.3.3. Размито моделиране.....	36
ГЛАВА 3. РАЗМИТА СИСТЕМА ЗА ИЗВОДИ	37
3.1. РАЗМИТИ СИСТЕМИ И РАЗМИТИ ЕКСПЕРТНИ СИСТЕМИ.....	37
3.1.1. Принципи на създаването на размита система за изводи.....	37
3.1.2. Видове размити системи.....	39
3.2. РАЗМИТА СИСТЕМА ЗА ИЗВОДИ, ИЗПОЛЗВАЩА МОДЕЛА FLQM ЗА ОЦЕНКА НА ФИНАНСОВО АКТИВИ.....	42
3.2.1. Архитектура на размитата система.....	42
3.2.2. FLQM модел за оценка на финансови активи.....	44
3.3. УПРАВЛЕНИЕ НА ФИНАНСОВ АКТИВ.....	52
ГЛАВА 4. СОФТУЕРНА СИСТЕМА ЗА УПРАВЛЕНИЕ НА ФИНАНСОВИ АКТИВИ FSSAM	54
4.1. ОБЩА СХЕМА НА FSSAM.....	54
4.2. СЪБИРАНЕ И СЪХРАНЯВАНЕ НА ДАННИ (ССД).....	56
4.2.1. Отправяне на заявки.....	57
4.2.2. Извличане на данни.....	58
4.2.3. Попълване на липсващи данни.....	58

4.2.4. Изчисляване на характеристики на активи.....	58
4.3. КОНСТРУИРАНЕ НА ИНВЕСТИЦИОННИ ПОРТФЕЙЛИ С ИЗПОЛЗВАНЕ НА Q-MEASURE	59
4.4. АЛГОРИТЪМ ЗА МАКСИМАЛНО ИЗПОЛЗВАНЕ НА КАПИТАЛА	61
4.5. РЕАЛИЗАЦИЯ НА МОДЕЛИТЕ	61
4.5.1. Обработване на данните	62
4.5.2. База от данни	62
4.5.3. Методи за достъп до база от данни	64
4.5.4. Бизнес логика	64
4.5.5. Автоматично стартиране на системата	68
4.5.6. Реализация на FQLM модела.....	69
4.5.7. Конструирание на портфейли	72
ГЛАВА 5. ТЕСТВАНЕ НА СОФТУЕРНАТА СИСТЕМА FSSAM С РЕАЛНИ ДАННИ	79
5.1. ВРЕМЕ ЗА ИЗПЪЛНЕНИЕ НА ПРОГРАМАТА	79
5.2. РЕЗУЛТАТИ ОТ ПРИЛОЖЕНИЕТО НА МОДЕЛА FLQM ЗА ОЦЕНКА НА ФИНАНСОВИ АКТИВИ	80
5.3. ИЗСЛЕДВАНЕ НА ХАРАКТЕРИСТИКИТЕ И УПРАВЛЕНИЕ НА АКТИВ	83
5.3.1. Изследване на характеристиките на актива 4ЕС.....	83
5.3.2. Изследване на характеристиките на актива E4A	84
5.3.3. Изследване на характеристиките на актива 55B.....	85
5.3.4. Изследване на характеристиките на актива 6F3.....	86
5.3.5. Изследване на характеристиките на актива 3NB	87
5.4. ИНВЕСТИЦИОНЕН ПОРТФЕЙЛ	88
5.4.1. Конструирание на инвестиционен портфейл с дялове на активите, зависещи от Q	88
5.4.2. Изследване на зависимостта между големината на инвестиционния капитал и необходимостта от прилагане на процедура по допълнителна алокация.....	91
5.4.3. Портфейли с фиксиран максимален брой активи	96
ЗАКЛЮЧЕНИЕ - ОСНОВНИ РЕЗУЛТАТИ	99
ДЕКЛАРАЦИЯ ЗА ОРИГИНАЛНОСТ НА РЕЗУЛТАТИТЕ	101
БИБЛИОГРАФИЯ	102
ПУБЛИКАЦИИ ПО ТЕМАТА НА ДИСЕРТАЦИЯТА	106
УЧАСТИЕ В ПРОЕКТИ ПО ТЕМАТА НА ДИСЕРТАЦИЯТА	106
БЛАГОДАРНОСТИ	107
ПРИЛОЖЕНИЕ	108
ПОРТФЕЙЛИ С ФИКСИРАН МАКСИМАЛЕН БРОЙ АКТИВИ ТРИ.....	108
ПОРТФЕЙЛИ С ФИКСИРАН МАКСИМАЛЕН БРОЙ АКТИВИ ПЕТ	112
ПОРТФЕЙЛИ С ФИКСИРАН МАКСИМАЛЕН БРОЙ АКТИВИ СЕДЕМ.....	126

Глава 1. Увод. Цел и задачи на дисертацията

1.1. Финансово инвестиране

Понятието инвестиране в икономиката, управлението и финансите е свързано с отказ от сегашно потребление и придобиване на активи с цел получаване на бъдеща печалба. Уилям Шарп дава следната дефиниция: ”Инвестиция е жертването на сигурна сегашна стойност за (вероятно несигурна) бъдеща стойност.” [82]

Финансовата инвестиция представлява притежаване на финансови активи, които се дефинират като права върху парични суми (пр. банкови депозити) или като права върху доходи и/или прираст на капитала [1] (пр. обикновени акции, привилегирани акции, конвертируеми ценни книжа, взаимни фондове; държавни облигации, общински облигации, корпоративни облигации; съкровищни бонове, търговски ценни книжа, банкови акцепти, евродоларови депозити, краткосрочни общински облигации, деривативи и други).

Инвестирането в ценни книжа обхваща две основни фази: анализ на ценните книжа на капиталовия пазар и управление на портфейл от ценни книжа. Първата фаза обхваща различни анализи за оценяване на активите и прогнози за бъдещото изменение на стойността им, а втората - избор на приемлив за инвеститора портфейл, осигуряване на финансови фондове, закупуване на ценните книжа, мониторинг и анализ на функционирането на портфейла. [1]

В икономическата наука има различни подходи за подпомагане на процеса за вземане на инвеститорски решения, при което се използват предимно следните ключови икономически понятия: възвращаемост, риск, ликвидност и инвестиционен хоризонт.

1.1.1. Възвращаемост

Възвращаемостта често се определя като изменение на цената на даден актив за определен период от време.

За оценка на възвращаемостта на ценни книжа се използват основно два метода: историческа възвращаемост и очаквана възвращаемост.

При определяне на историческата възвращаемост се използват исторически данни за изменение на цените на актива в предходни моменти и следователно историческата възвращаемост е известна величина. Тя се пресмята като средноаритметична или като средногеометрична величина.

За оценка на очакваната прогнозирана възвращаемост се използват вероятностни разпределения, анализ на различни сценарии или експертни мнения. При първия подход се приема, че възвращаемостите имат конкретно вероятностно разпределение: нормално, лог-нормално или друго, т.е. изменението на възвращаемостите е стохастична величина. При анализирането на сценарии анализаторът дефинира няколко сценария за евентуални бъдещи изменения на икономическите условия. На всеки сценарий се приписва субективна вероятност, като тези вероятности се използват за пресмятане на претеглени очаквани възвращаемост и риск. Вероятностите могат да бъдат изведени и от емпирични наблюдения, като периодът, за който има данни, може да бъде разделен на различни подпериоди, откъдето да се пресметнат съответните вероятности. Използването на експертни мнения е подобен подход: анализаторът прави прогнози за очакваните изменения в

икономическото развитие и използвайки интуицията си, предвижда очакваната възвращаемост. Анализаторът може да има достъп до оценки за бъдещото изменение от други експерти и може да ги използва, за да подкрепи или отхвърли своята прогноза; или може да комбинира своята и на други експерти прогнози с получените от историческите данни оценки на възвращаемостта.

Популярно и ефективно средство за оценка на активите е техническият анализ, при който основно се използва предположението, че прогнозите за бъдещи състояния на пазара могат да бъдат изведени от предишни данни за състоянието му. Една от целите на техническия анализ е да се открият дългосрочни, средносрочни и краткосрочни тенденции в изменението на цените на ценните книжа и да се използват за определяне на вероятни бъдещи цени, които стават основа на процеса на вземане на решения при управление на финансови активи.

1.1.2. Инвестиционен риск

За оценка на инвестиционния риск се използва основно отклонението на възвращаемостите от средната възвращаемост [4], [10], [22], [42], [46], [48]. При използване на исторически данни се пресмятат различни статистически измерители – размах, дисперсия, средноквадратично отклонение, β коефициенти, а в случай на използване на очаквани възвращаемости се пресмятат различни моменти на вероятностни разпределения.

1.1.3. Ликвидност

Ликвидността на даден актив отразява степента, в която този актив може да бъде купен или продаден за разумно време, без това да повлияе съществено на неговата цена. Активите с по-висока ликвидност се характеризират и с по-висока степен на търговска активност. Акциите са активи с относително висока ликвидност, като важен елемент от степента на ликвидност е цената на транзакциите.

1.1.4. Инвестиционен хоризонт

Всяка инвестиция е отказ от потребление и следователно отказ от ликвидност. При определени условия инвеститорът прекратява инвестицията, следователно всяка инвестиция има продължителност. При взимане на инвестиционни решения, очакваната продължителност на инвестицията (т.е. времето, за което е планиран отказ от ликвидност) се нарича инвестиционен хоризонт. След достигане на инвестиционния хоризонт, инвеститорът преоценява предходното решение и взема ново такова. [10]

1.2. Инвестиционни теории и модели

1.2.1. Портфейлна теория на Х. Марковиц (ПТ)

Основната концепция в предложената от Хари Марковиц теория [60] е, че възвращаемостите на ценните книжа са случайни величини и за тях могат да се пресметнат математическото очакване и средноквадратичното отклонение, като средноквадратичното отклонение е мярка за инвестиционния риск. В модела на Марковиц са наложени специфични изисквания за инвестиционната среда. По важните от тях са:

- при вземане на решения се използват само две характеристики на активите: математическото очакване на възвращаемостта и средноквадратичното отклонение на възвращаемостта;
- очакваните възвращаемости на отделните ценни книжа имат нормално вероятностно разпределение;
- инвеститорът е несклонен към риск: за дадена очаквана възвращаемост предпочита минимум риск или за даден риск предпочита максимум очаквана възвращаемост;
- не съществуват безрискови активи;
- не се извършват къси продажби;
- няма данъци;
- инфлацията е взета предвид при определяне на очакваната възвращаемост.

Очакваната възвращаемост за портфейл от m актива е:

$$E(R_p) = \sum_{j=1}^m E(r_j) \cdot x_j,$$

очакваният риск на портфейла е:

$$\sigma_p^2 = \sum_{j=1}^m \sigma_j^2 \cdot x_j^2 + \sum_{i=1}^m \sum_{\substack{s=1 \\ s \neq i}}^m \text{cov}(r_i, r_s) \cdot x_i \cdot x_s$$

където:

$E(R_p)$ е очаквана възвращаемост на портфейла;

$E(r_j)$ е очаквана възвращаемост на j -тия актив;

σ_p^2 е дисперсия на възвращаемостта на портфейла;

σ_j^2 е дисперсия на възвращаемостта на j -тия актив;

$\text{cov}(r_i, r_s) = E[(r_i - E(r_i))(r_s - E(r_s))]$ е ковариация между възвращаемостите на i -тия и s -тия активи;

x_j - дял на j -тия актив, като $\sum_{j=1}^m x_j = 1$ и $x_j \geq 0$.

Според ПТ задачата за избор на портфейл може да бъде формулирана като оптимизационна задача върху множеството на реалните числа, с квадратна целева функция и линейни ограничения [61] по следния начин:

$$\min \sigma_p^2 = \sum_{i=1}^m \sum_{s=1}^m \text{cov}(r_i, r_s) \cdot x_i \cdot x_s,$$

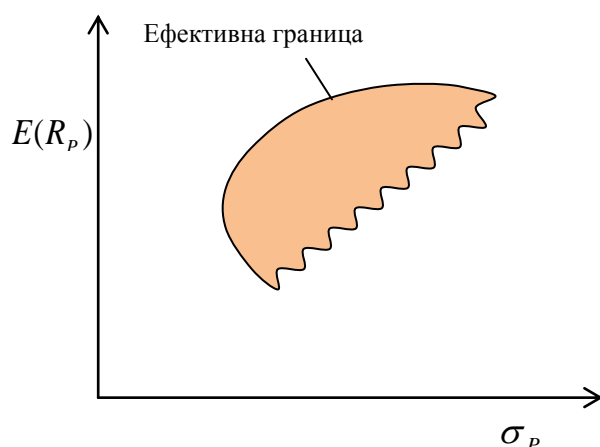
при условия:

$$E(R_p) = \sum_{j=1}^m E(r_j) \cdot x_j,$$

$$\sum_{j=1}^m x_j = 1,$$

$$0 \leq x_j \leq 1 \quad \text{за } j = 1, 2, \dots, m \quad (1.1)$$

Решенията на така формулираната оптимизационна задача формират ефективната граница, като всеки портфейл върху нея се нарича ефективен портфейл (фиг. 1.1).



Фигура 1.1. Ефективна граница.

Джеймс Тобин добавя в модела на Марковиц и безрискова инвестиция, което дава възможност за конструиране на свръхефективни портфейли. [88]

През 1990 Марковиц, съвместно с Мортън Милър и Уилям Шарп, е удостоен с Нобелова награда за станалата изключително популярната теория за конструиране на инвестиционен портфейл.

1.2.2. Модел за оценяване на капиталовите активи (МОКА)

МОКА е предложен от Трейнър [27], [89], Шарп [81], Линтнер [56]. За негова основа служи теорията на Марковиц за диверсификация на риска. Основна концепция на МОКА е пресмятането и използването на β коефициенти, които са мярка за систематичния риск и показват чувствителността на възвращаемостта на даден актив към пазарните изменения. С използването на β коефициентите решаването на портфейлната задача се опростява значително, тъй като се избягва пресмятането на корелациите в модела на Марковиц. Според МОКА β коефициентите отразяват риска, който не може да се диверсифицира и в този смисъл те са по-подходяща мярка за риска от средноквадратичното отклонение. От друга страна β коефициентът е показател за приноса на даден актив в общия риск на портфейла.

Основни предпоставки на МОКА са:

- всички инвеститори са рационално несклонни към риск;
- всички инвеститори имат общ времеви хоризонт за вземане на инвестиционно решение;
- всички инвеститори имат идентични субективни оценки за бъдещата възвръщаемост и риска на всички акции;
- съществуват безрискови активи и всички инвеститори могат да дават или да вземат на заем неограничено количество активи с безрискова лихва;
- всички активи са достъпни в произволно малки обеми, няма разходи по трансакциите или други такси;
- информацията е безплатна и достъпна за всеки инвеститор.

Според МОКА очакваната възвръщаемост на даден актив се получава по формулата:

$$E(r_j) = R_f + \beta_j (E(R_M) - R_f) \quad (1.2)$$

където:

$E(r_j)$ е очакваната възвръщаемост на j -я актив;

R_f е възвръщаемостта на безрисковия актив;

$\beta_j = \frac{\text{cov}(r_j; R_M)}{\sigma_M^2}$ е β коефициентът на j -я актив;

$E(R_M)$ е пазарната възвръщаемост;

σ_M^2 е пазарната дисперсия.

Уравнението (1.2) е известно като линия на пазара на ценни книжа и е главен резултат от МОКА. Ако портфейлът е ефективен, всеки актив от него ще заеме някаква точка от линията на пазара на ценни книжа, което означава, че връзката между очакваната възвръщаемост на даден актив и неговият принос за риска на портфейла, изразен от β , е линейна. Изразът $\{E(R_M) - R_f\}$ се нарича пазарна рискова премия.

В МОКА се приема, че портфейлът има възвръщаемост, която е свързана с пазара точно като при отделен актив, като стойността на β коефициента на портфейла е среднопретеглена на β коефициентите на всички активи от портфейла с тегла, съответно равни на дяловете им. Очакваната възвръщаемост на портфейла се пресмята по формулата:

$$E(R_p) = R_f + \beta_p \{E(R_M) - R_f\} \quad (1.3)$$

Рискът при изменението на възвръщаемостта на портфейл от m инвестиции се пресмята по следната формула:

$$\sigma_p^2 = \beta_p^2 \sigma_M^2 + \sum_{j=1}^m x_j^2 \cdot \sigma_i^2, \quad (1.4)$$

където x_j е делът на j -я актив.

1.2.3. Някои варианти на портфейлната теория

Вариант 1. Марковиц [61] предлага добавяне на ограничително условие за възвращаемостта на портфейла - изискване за предварително зададена минимална желана от инвеститора възвращаемост R^* . Така основната задача (1.1) добива следния вид:

$$\min \sigma_p^2 = \sum_{i=1}^m \sum_{s=1}^m \text{cov}(r_i, r_s) \cdot x_i \cdot x_s,$$

при условия:

$$\sum_{j=1}^m r_j \cdot x_j \geq R^*,$$

$$\sum_{j=1}^m x_j = 1,$$

$$0 \leq x_j \leq 1 \quad \text{за } j = 1, 2, \dots, m \quad (1.5)$$

Вариант 2. В друг вариант към същата целева функция се добавят две допълнителни условия за дяловете на активите. Едното ограничение е за броя на активите в портфейла, т.е. предварително е зададено число $k \leq m$, така че броят на активите, за които $x_j > 0$ е най-много k . Второто ограничение е делът на всеки актив да се изменя в предварително зададен интервал $[\alpha_j; \beta_j]$. Тези две ограничения се моделират чрез добавяне на m бинарни променливи z_j ($j = 1, 2, \dots, m$) и условията:

$$\sum_{j=1}^m z_j \leq k$$

$$\alpha_j \cdot z_j \leq x_j \leq \beta_j \cdot z_j$$

$$z_j \in \{0; 1\} \quad \text{за } j = 1, 2, \dots, m \quad (1.6)$$

като $z_j = 1$ в случай, че j -ят актив участва в портфейла и $z_j = 0$ в случай, че j -ят актив не е включен в портфейла. [79]

Вариант 3. Манчини и Сперанца [59] разглеждат същественото от практическа гледна точка ограничение, че активите могат да се търгуват само на неделими лотове с фиксирана големина. Задачата (1.1) става предмет на целочисленото оптимизиране.

Вариант 4. Коно и Ямазаки [52] предлагат вместо дисперсията за измерител на риска да се използва абсолютното отклонение. Използвайки исторически данни от Токийската фондова борса, те сравняват модела на Марковиц с техния модел и показват, че бързодействието на двата модела е сходно. Въпреки, че така преформулираният модел на Марковиц не изисква ковариационна матрица, през 1997г. Симаан доказва, че цената на ресурсите за това по-прецизно оценяване на риска надвишава ползите от използването на модела [85].

Вариант 5. Марковиц [62] предлага различна целева функция - базирана на полувариацията, при което се използват само отклоненията по-малки от дадена гранична стойност, вместо всички отклонения. Марковиц [61] пръв е отбелязал, че моделите с целева функция, конструирана с вариация имат недостатъка, че много големи и много малки възвращаемости са еднакво нежелани. При използването на полувариация се елиминират много малките възвращаемости.

Вариант 6. Науроки [66], [67] предлага за целева функция да се използва разликата на долните частични моменти (LPM) от втори ред и очакваната възвращаемост, умножена по коефициент:

$$\min \{LPM_{2,P} - \lambda \cdot E(r_p)\},$$

където $LPM_{2,P} = \sum_{j=1}^m \sum_{i=1}^m s_j \cdot s_i \cdot \text{cov}(r_j, r_i) x_j \cdot x_i$ и s_j е квадратният корен на полувариацията за j -тия актив.

Вариант 7. Използване на третия момент е предложено от Пол Самюелсън през 1958 г., но поради изчислителните трудности при пресмятане му за голям брой активи, използването на асиметрията при конструиране на портфейл е пренебрегвано за дълго време. През 1995 г. Коно и Сузуки прилагат линейно приближение на третия момент [51], [53] и така добавя коефициентите на асиметрия на разпределенията на възвращаемостите в модела на Марковиц.

Вариант 8. Йошимото [92] разглежда случай с първоначален портфейл и добавя в модела цената на разходите за транзакции.

1.2.4. Сортиращи алгоритми

При сортиращите алгоритми портфейлът се конструира от предварително подбрани активи, за които са изпълнени дадени условия. Общ подход при тях е задаване на критични стойности C_j за $j = 1, 2, \dots, m$. Върху избраните след сортирането активи се прилага оптимизационната процедура (1.1). Получените решения са приблизително оптимални. [28], [29]

Сортиращите алгоритми са средство за редуциране на неточността при пресмятане на обратната матрица на ковариационната матрица в модела на Марковиц. Тъй като използваната в портфейлната теория ковариационна матрица е силно чувствителна към статистически грешки, то броят на наблюденията трябва значително да надвишава броя на независимите променливи (примерно за 25-30 актива са нужни наблюдения за 48-60 месеца, което е твърде дълъг период и съществено изкривява информацията за текущото състояние на активите). От друга страна се губи ценна информация за ковариациите и така не може да се използва целият потенциал на диверсификацията като методология. [29]

Сортиране с β анализ

При този алгоритъм първо за всеки актив се пресмятат отношенията:

$$\frac{R}{\beta_j} = \frac{E[R_j - R_f]}{\beta_j}$$

Ако $\frac{R}{\beta_j} < C_j$, то се пресмятат величините $z_j = \frac{\beta_j^2}{\sigma_{\varepsilon,j}^2} \cdot \left(\frac{R}{\beta_j} - C^* \right)$, където

$$C^* = \min_j \{C_j\}.$$

Тогава $x_j = \frac{z_j}{\sum z_j}$ за $j = 1, 2, \dots, m$.

В случая възвращаемостта на безрисковия актив R_f е от съществено значение. Колкото R_f е по-голяма, толкова по-малко активи влизат в портфейла. Диверсификация се получава с намаляване на R_f .

Сортиране чрез корелационен анализ

Процедурата е същата като при сортирането с бета анализ, но използваното отношение е:

$$\frac{R}{V_j} = \frac{E[R_j - R_f]}{\sigma_j}$$

Сортиране по отношението възвращаемост/риск

Процедурата е аналогична за отношенията

$$\frac{E[R_j - R_f]}{risk_j}, \text{ където } risk_j \text{ е коя да е оценка на риска.}$$

Сортиране по средноквадратично отклонение

Сортирането е аналогично след пресмятане на $\frac{E[R_j - R_f]}{\sigma_j^2}$.

Сортиране по долни частични моменти (LPM)

Сортирането е аналогично след пресмятане на $\frac{E[R_j - R_f]}{LPM_{n,j}}$ [65]

1.2.5. Алгоритми за локално търсене

Локално търсене е фамилия от техники с общо предназначение за търсене и оптимизация. Тези техники не са изчерпателни в смисъл, че не гарантират да намерят правдоподобно (или оптимално) решение, а търсенето е несистематично и продължава, докато не бъде изпълнено зададено условие. Използването на локално търсене за селектиране на портфейл е предложено от Роналд (1997) и Чанг (2000).

Нека p е случай от задачата P и на нея е съпоставено пространство за търсене S . На всеки елемент $s \in S$ отговаря възможно решение p , наречено състояние на p . Елемент s от S , на който отговаря решение, удовлетворяващо всички ограничения, се нарича приемливо състояние на p . Локалното търсене се задава чрез функция N , зависеща от структурата на P и съпоставяща на всяко $s \in S$ негова околност $N(s) \subseteq S$. Всеки елемент $s' \in N(s)$ се нарича съседен на s .

Локално търсене започва с първоначално състояние s_0 , което се получава с друга техника за търсене или се генерира случайно, след което се обхожда пространството за търсене, като се преминава от едно състояние s_i в съседното s_{i+1} . Обикновено околността се състои от локално изменение на текущото състояние.

Техниките на локалното търсене се различават в зависимост от приложената стратегия от една страна за избор на ход за всяко състояние и от друга - за край на търсенето. Но при всички техники търсенето се управлява от функция на цената f , която е оценка на качеството на състоянието и трябва да бъде минимизирана. [79]

1.3. Някои нерешени или частично решени проблеми в инвестиционните теории и модели

При създаване на приложения за подпомагане на инвестиционните решения, базирани на разгледаните в 1.2 теории и модели възникват редица трудности и проблеми, например: с данните, с необходимите изчислителни ресурси, несъществуване на безрискови активи, кардиналност при конструирането на инвестиционен портфейл, несигурност и неопределеност на финансовите пазари.

А. Данни

При използване на исторически данни основен проблем е достъпът до надеждни данни за цените на активите. Официални данни се публикуват на страниците на съответните финансови институции, но първият възникващ проблем е как да бъдат извлечени тези данни. Втори проблем е в какъв формат да бъдат съхранени тези данни, така че да са подходящи за използване в съответно софтуерно приложение. Не помаловажен е и въпросът какъв да бъде обемът на съхраняваните и използвани данни.

Следващ проблем възниква при оценяването на характеристиките на активите в зависимост от честотата на събиране на данните. Често използваните във финансовата практика средноаритметична величина за оценка на възвращаемостта и съответното средноквадратично отклонение са удобни от статистическа и изчислителна гледна точка, но от инвестиционна не дават информация за истинското изменение на капитала. Използването на средногеометрична величина дава нужната информация, но изисква внимателни и прецизни математически изчисления.

Не на последно място стои проблемът с неравните интервали от време между последователните наблюдения или накратко „липсващи данни”, за чието възникване има поне три различни причини: първо, различните борси имат различни почивни и празнични дни, в които няма сделки; второ, има дни, в които даден актив не е търгуван и трето – активът е заличен от борсата по някаква причина.

Съществуват разнообразни подходи за решаване на проблема с липсващите данни, например: изменяне на минималната сингулярна стойност на ковариационната матрица [12], оценяващи функции, невронни мрежи и много други.

Възможен подход е използване на следната модификация на метода на Надарая-Уотсън [76], при който ако за непрекъснатата функция $K(x)$ са изпълнени условията $K(x) \geq 0$ и $\int K(u)du = 1$, то с помощта на параметър $h > 0$ функцията $K(x)$

се трансформира по следния начин: $K_h(x) = \frac{1}{h} \cdot K\left(\frac{x}{h}\right)$ и $\int K_h(u)du = 1$.

Конструират се тегловни функции $\frac{K_h(x-t)}{\frac{1}{T} \sum_{t=1}^T K_h(x-t)}$, които са коефициенти пред

P_i в $\hat{m}_h(\tau)$. Така функцията

$$\hat{m}_h(\tau) = \frac{\sum_{i=t}^{t+k-1} K_h(\tau-i) P_i}{\sum_{i=t}^{t+k-1} K_h(\tau-i)} : t = 1, \dots, T-k+1;$$

където $K_h(x) = \frac{1}{h\sqrt{2\pi}} e^{-\frac{x^2}{2h^2}}$, може да бъде използвана за получаване на стойности за попълване на липсващи данни.

От практическо значение при избора на подход за решаване на проблема с липсващите данни е доколко избраната процедура води до голям обем допълнителни изчисления, което е пряко свързано с използваните изчислителни ресурси.

Б. Изчислителни ресурси

Бързината, с която инвеститорът има възможност за оценка на ситуацията (времето за анализ, вземане и реализиране на инвеститорското решение в бързо променящата се икономическа среда) е от съществено значение.

В ПТ за портфейл от m на брой актива се пресмятат $\frac{m^2 - m}{2}$ на брой ковариации и тъй като с увеличението на броя на активите в портфейла, съответният брой на ковариациите става значително по-голям от броя на активите (примерно при 90 актива ковариациите са 4005), то решаването на оптимизационната задача за минимизиране на риска при конструиране на портфейл изисква огромни изчислителни ресурси, което от своя страна налага силни ограничения върху броя на използваните активи.

В МОКА пазарният портфейл M трябва да съдържа всички рискови инвестиции и тогава използването на който и да е пазарен индекс е несъстоятелно, защото повечето пазарни индекси се състоят от някои (а не от всички) активи. Наблюдаването на характеристиките на пазарен портфейл, съставен от всички активи е свързано с използването на толкова големи изчислителни ресурси, че е практически невъзможно.

От друга страна, ако за пресмятане на характеристиките на всеки портфейл са нужни μ на брой операции и за конструиране на всеки от възможните портфейли се използват η на брой операции, то за получаване на всички възможни портфейли са нужни η^m операции и тогава общият брой на операциите е $\mu\eta^m$. Необходимото време за извършване на тези операции е равно на $\frac{\mu\eta^m}{v}$, където v е скоростта на използваната машина. Ако се приеме, че $v = 2,4 \times 10^{12} \text{ flop/sec}$ (скоростта на Cray T3E), $\mu \approx 400$, $m = 30$ и $\eta = 100$, то необходимото време е $\frac{400 \cdot 100^{30}}{2,4 \cdot 10^{12}} \approx 1,67 \cdot 10^{50}$ sec или приблизително $5,3 \cdot 10^{42}$ години.

В. Безрискови активи

Важно условие в МОКА е съществуването на безрискови активи. Практиката на финансовите пазари през 2007-2012 показва, че това условие е несъстоятелно.

Г. Кардиналност

Приемането, че няма допълнителни ограничения за частта, която може да се вложи в даден актив е нереалистично. При реално търгуване на фондовата борса не всяка комбинация от ценни книжа е възможна. От една страна с дадена сума не винаги може да се купят целочислен брой активи, а от друга – в даден момент желаният актив

може да не се продава или да се продава в количества, различни от получените в модела.

Д. Несигурност и неопределеност на финансовите пазари

В цените на активите, търгувани на фондовите пазари намират отражение огромен брой фактори: политически и икономически решения на правителствата, финансови интереси на компаниите, теории, стратегии, прогнози, действия и очаквания, психологически реакции на борсовите играчи, не на последно място природни катаклизми и бедствия. Тези фактори постоянно си взаимодействат, защото всички (регулаторни органи, правителства, банкери, инвеститори и анализатори) са участници в този процес на вземане на решения в среда, зависеща от безброй фактори, включително и от техните собствени действия. Несигурността и неопределеността са неизбежен факт от реалността на финансовите пазари. Тази несигурност не може да бъде изучена и избегната напълно, най-малкото поради факта, че е невъзможно да се проведат експерименти при напълно идентични условия - състоянието на глобалната финансова система във всеки момент е уникално и различно от предходното.

В МПТ, МОКА и други модели се използва предположението, че очакваните възвращаемости на отделните ценни книжа имат нормално или друго вероятно разпределение.

Емпиричните изследвания обаче показват, че това предположение не е изпълнено [2], [6], [14], [34]. При проверка на хипотезите за съществуване на нормално или лог-нормално разпределение съответно на възвращаемостите и на техните логаритми могат да бъдат използвани различни статистически тестове. В [2] за проверка на хипотезата за нормално разпределение на логаритмичните възвращаемости са използвани тест на Колмогоров-Смирнов при нива на значимост $\alpha = 0,01; 0,05; 0,1$ и тест на Жарк-Бера. При теста на Колмогоров-Смирнов хипотезата за нормално разпределение на логаритмичните възвращаемости на акциите търгувани на БФБ се отхвърля при всяко от трите нива на значимост. При теста на Жарк-Бера се оценява ексцеса и асиметрията на разпределението на емпиричните данни. Емпиричните данни показват, че и асиметрията и ексцеса на логаритмичните възвращаемости се различават от тези на нормално разпределени величини.

Софт компютинг е едно направление на съвременния изкуствен интелект, предлагащо средства за моделиране на подобни ситуации.

1.4. Софт компютинг като направление в изкуствения интелект

1.4.1. Изкуствен интелект

Изкуственият интелект (ИИ) е направление в компютърната наука. Терминът изкуствен интелект е въведен от Джон Макарти през 1956 като „науката за създаване на интелигентни машини” [63]. Исторически изкуственият интелект възниква от опитите за формализиране на човешкото знание със средствата на математическата логика и първоначално се прилага за доказване на теореми и за моделиране на игри. Основа на изкуствения интелект стават различни науки: философия, математика, алгоритми,

логика, психология, информатика, лингвистика, но постепенно това направление се обогатява с други обекти на изследване и други средства [50].

В настоящия момент изкуственият интелект е съвкупност от традиционния изкуствен интелект, обогатен с различни методологии за числено пресмятане, като обект на изкуствения интелект (за разлика от неизкуствения интелект) стават основно онези задачи, за които липсват директни математически или логически алгоритми и могат да бъдат решавани само интуитивно. Към традиционните базисни за изкуствения интелект науки се добавят нови: компютърни науки, неврология, когнитивна наука, онтология, изследване на операциите, икономика, вероятности, оптимизиране и други.

Областите на приложение на средствата на изкуствения интелект са изключително разнообразни: експертни системи, игри, доказателство на теореми, обработка на естествени езици, разпознаване на образи, роботика, навигация, системи за управление, системи за планиране, дейта майнинг, логистика и много други. [3]

За алтернативно название на научната област „Изкуствен интелект” се използват още изчислителна интелигентност, синтетична интелигентност, интелигентни системи, изчислителна рационалност, като това е опит да се разграничи традиционният ИИ, опериращ със символи (или още GOFAI=Good-Old-Fashioned-Artificial-Intelligence), от съвременното му съдържание.

При традиционните изчисления (hard computing) основните цели са точност, сигурност и строгост. Точно обратно, отправната точка при софт компютинг е тезата, че точността и сигурността имат висока цена и че при извършване на изчисления, разсъждения и вземане на решения трябва да бъде допускана (където и когато е възможно) толерантност към неточност и несигурност. Следвайки тази теза, в софт компютинга се използва за модел човешкия мозък и в същото време целта е формализирането на когнитивните процеси, които хората употребяват толкова ефективно при извършване на различни дейности [94].

Софт компютингът се състои от няколко изчислителни парадигми: размита логика, невронни мрежи, приблизителни изводи и недиференциални методи за оптимизация като генетични алгоритми и симулирано закаляване [47].

Връзката между изкуствения интелект и софт компютинга не е тривиална - средствата на софт компютинг могат да бъдат разглеждани като елементи на изкуствения интелект, защото се използват за решаването на някои проблеми на изкуствения интелект. Дългосрочната цел на изкуствения интелект е създаването на машинна интелигентност и доколкото дадена интелигентна система е в състояние да възприема измененията на средата и да реагира, а тъй като софт компютингът се стреми към създаване на същия модел, то следователно той се развива до известна степен под въздействието на развитието на изкуствения интелект [30].

1.4.2. Софт компютинг

През 1991 Заде въвежда идеята за софт компютинг в опит за създаване на нов вид изкуствен интелект.

На 13.03.1991 на конференцията на Industrial Liason Program (ILP) в Бъркли е създадена организацията Berkley Initiative on Soft Computing (BISC).

Основният принцип на софт компютинга, дефиниран от BISC, е:

Като се използва допустимата степен на неточност, несигурност, частична истинност и приближение, да се постигне ясно, стабилно и на ниска цена решение на фундаментални проблеми, свързани със съвременното технологично развитие и така да се компенсира липсата на

изискваната интелигентност в сегашните информационни технологии, така че да позволи човешка функционалност, т.е. човешкият мозък да бъде използван за модел. [100]

Софт компютингът, според BISC, се състои от

- размити системи, включващи размита логика и играещи водеща роля;

- еволюционно пресмятане, включващо генетични алгоритми;

- изкуствени невронни мрежи;

- машинно обучение;

- вероятностни изводи,

като тези методологии се допълват за създаване на системи с висок MIQ (Machine Intelligence Quotient). [100]

Основните идеи на софт компютинга са основани на концепцията за размити множества, предложени от Заде през 1965 [95]. През 1973 Заде въвежда правила за дефиниране на лингвистични променливи и пресмятане на ако-то правила [97]. Приложение на размитата логика за вземане на решения предлагат Заде и Белман [96]. През 1979 Заде дава идеи за правене на изводи и дедуктивни разсъждения от приблизителна информация [93]; 1981 – публикува теория на възможностите и софтдейта анализ, през 1991 дава идея за софт компютинг; в 1999 – изчисления с думи и единна теория на несигурността, в 2005 публикува обобщена теория на несигурността.

Размитата логика е основата на софт компютинга. Към размитата логика първо са добавени невронни мрежи, а по-късно и еволюционни пресмятания и вероятностни изводи и други.

През 1983 Кр. Атанасов предлага идеята за интуиционистки размити множества [18], в които освен степен на принадлежност е добавена и степен на непринадлежност. Едни от важните резултати са свързани с въведените от Кр. Атанасов оператори: модалните оператори *необходимо*; *възможно*; обобщаващ ги модален оператор и други. Теорията на интуиционистки размитите множества претърпява интензивно и продължаващо развитие.

К. Пеева въвежда и изследва размити крайни автомати, размити линейни системи и е един от създателите на MATLAB Fuzzy Relational Calculus Toolbox [70].

1.4.3. Основни направления в софт компютинг

1.4.3.1. Размити множества и размита логика

Заде предлага концепцията за размити множества в 1965 [95]. Теорията на размитите множества дава система за лингвистична обработка на неточна и непълна информация и извършване на числено пресмятане с лингвистични променливи, описани чрез функции на принадлежност.

Заде предлага размитата логика с цел моделиране на естествените езици и твърди, че размитата логика не бива да се разглежда като отделна теория, а като процес на размиване - обобщение за всяка теория от дискретна към непрекъснатата.

Размитите системи за изводи ефективно моделират човешката експертност в конкретни приложения, като ключовата им компонента е множество от размити „ако-то” правила, дефинирани върху размити множества. Съществуват системи върху размити множества от тип 1, при които резултатът е размита променлива с реални

функции на принадлежност и системи върху размити множества от тип 2 – резултатът е размита променлива със степени на принадлежност, които са размити множества и се използват в ситуации, в които степените на принадлежност не могат да се дефинират точно [64].

Основен недостатък на размитите системи за изводи е липсата на адаптивност за гъвкаво реагиране на промени в условията [47].

Някои важни теоретични постановки на размитата логика, необходими за създаване на размита система за изводи, са разгледани в Глава 2.

1.4.3.2. Еволюционни пресмятания

В еволюционните пресмятания се използват оптимизационни алгоритми, които имитират принципи, наблюдавани в природата и доказали своята пригодност и приложимост. Особено важни са случаите, в които природата е намерила “стабилни острови” в “бурния океан” от възможни решения – като процеса на закаляване на стоманата, централната нервна система, биологичната еволюция и др. [78]

Първите идеи в областта предлагат Фрейзър и Бокс в края на 50-те години на 20-ти век. [58]

От използването на принципите на еволюцията са произлезли оптимизационни методи като:

- генетични алгоритми - Genetic Algorithms (GA);
- еволюционно програмиране - Evolutionary Programming (EP);
- еволюционни стратегии - Evolution Strategies (ES);
- системи за класифициране - Classifier Systems (CFS);
- генетично програмиране - Genetic Programming (GP),

които са елементи на еволюционните алгоритми. Всички те симулират еволюцията на индивидуалните структури чрез процесите селекция, мутация и възпроизвеждане и са базирани върху биологични наблюдения, като теорията на Дарвин за естествения подбор и оцеляване на най-пригодните. [44], [54]

Еволюционните алгоритми представляват популация от структури, която еволюира според правилата на оператори за селекция, търсене, рекомбиниране на индивиди с висока пригодност и мутация, която променя индивидите. Всеки индивид получава мярка за пригодност към средата. [86]

През 1975 Холанд предлага генетични алгоритми, при които представянето на индивидите е с битове и популацията се развива при специфични правила за избор до момент, при който „пригодността” е максимална. През 1989 Голдбърг публикува този подход и успешно го прилага в газопровод. При генетичните алгоритми акцентът е върху рекомбинирането, а мутациите са съвсем слаби. Генетичните алгоритми могат да бъдат разглеждани като алгоритми за търсене, защото при тях се изследват елементите на дадено пространство, като се използват евристични подходи, вдъхновени от природата. За различни приложения са разработени бинарни генетични алгоритми, непрекъснати генетични алгоритми, паралелни генетични алгоритми, симулирано закаляване, „роене”, „колонии от мравки”, генетично програмиране, еволюционни стратегии.

Фундаменталната теорема на генетичните алгоритми на Холанд гласи, че ако дадена хромозома (кодирана информация на даден индивид) с фиксирани гени има пригодност над средната, то се наблюдава тенденция броят на индивидите с този шаблон да се увеличава експоненциално. Доказателство на тази теорема в случай на пропорционална селекция с равномерно разпределение дава Конар. [50]

Еволюционното програмиране и еволюционните стратегии имат за цел получаване на едно оптимално решение и следователно са добри за решаване на оптимизационни проблеми, докато генетичните алгоритми дават няколко оптимални решения.

1.4.3.3. Невронни мрежи

Изкуствената невронна мрежа е модел за обработка на информация, вдъхновен от изучаването на биоелектричните мрежи в мозъка на човека и животните, образувани от неврони и техните синапси. Изкуствените невронни мрежи, често наричани само „невронни мрежи”, се явяват група взаимно свързани изкуствени неврони, използващи математически модели и изчислителни методи за обработка на информация. Невронната мрежа е адаптивна система, чието изменение се осъществява в зависимост от външната и вътрешната (протичаща през мрежата) информация. В този смисъл невронните мрежи симулират интелект [7].

През 1957 Розенблат създава модел на перцептрон с един слой и участва в създаването на първия неврокомпютър Mark I Perceptron. Модел на многослоен перцептрон е създаден през 1960. Невронна мрежа с обратно разпространение на грешката предлага Уеброс през 1974, по-късно обобщен от Румелхарт и Маклелънд (1986). В модела на Хопфийлд (1982) невронната мрежа е с един слой и невроните са напълно свързани помежду си. В машината на Болцман (1984) са комбинирани многослоен перцептрон и мрежа на Хопфийлд. Брумхед предлага модел на радиална мрежа през 1988. През 1982 Кохонен създава самоорганизиращи се мрежи. [5]

Математическият аналог на биологичната невронна мрежа представлява множество от взаимосвързани прости изчислителни елементи (неврони). Всеки неврон приема сигнали от другите и ги сумира, като сумата минава през активационна функция и така се определя степента на активация, която от своя страна бива предадена по изходящите връзки към другите неврони. Всяка връзка има тегло, което умножено със сигнала определя неговата сила. Теглата на връзките са аналогични на силата на синаптичните импулси, предавани между биологичните неврони. Отрицателна стойност на теглото съответства на потискащ импулс, а положителна - на възбуждащ.

Математически е доказано, че всяка невронна мрежа с поне един скрит слой от достатъчно на брой неврони между входния и изходния слой, може да моделира поведението на всяка непрекъсната функция.

Теглата на връзките между невронните определят функционалността и поведението на невронната мрежа. За да бъде невронната мрежа използвана и приложима към даден проблем, тя трябва да бъде предварително обучена. Обучението на невронна мрежа се постига чрез промяна на теглата на връзките между невронните и се осъществява чрез правила, които определят как да се променят тези тегла.

Друга важна особеност на невронните мрежи е възможността за паралелна обработка на информацията. При голям брой междуневронни връзки това значително ускорява процеса на обработка на информацията и дава възможност за моделиране на процеси в реално време. При голям брой междуневронни връзки мрежата е устойчива на възникващи грешки, защото функциите на повредените връзки се поемат от други такива.

1.4.3.4. Размито-невронни мрежи

Адаптивна невронно-размита система за вземане на решения (ANFIS)

При моделите, в които се комбинират невронна мрежа и размита логика, се цели чрез обучение на невронната мрежа така да се изменят параметрите на размитата

система, че да се получи резултат (решение), което е адекватно при всяко изменение на средата. [16]

1.4.4. Приложения на средствата на софт компютинг

Областите на приложения на моделите, построени със средствата на софт компютинг, са многобройни и разнообразни: разпознаване на образи, разпознаване на реч, класификация, оптимизация, предвиждания, планиране, управление, компютърни системи, вземане на решения, управление, икономика, игри, климат, навигация, роботика, космически изследвания и много други. В различните приложения се използват както отделни направления на софт компютинга, така и интегриране на тези направления за създаване на хибридни системи.

Ангелова предлага подробен обзор на състоянието на изследванията в Института по информационни технологии на Българската академия на науките, в които се прилагат средствата на софт компютинг, в [17].

1.4.4.1. Приложения на размитата логика

Размита логика се използва в решаване на задачи, свързани с автоматично управление [21], класификация на данни, разпознаване на образи, анализ и управление на финансови и икономически системи [20], прогнозиране на времеви редове, клъстеризация [11], вземане на решения, роботика и много други.

Силва, Торез и Хадад прилагат размита оптимизация на хетерогенна система от енергийни производства, като е показано приложение за енергийната система на Бразилия. Размити са както целевата функция, така и ограничителните условия на оптимизационната задача [84].

Карлсон и Фулър решават някои конкретни размити оптимизационни задачи – линейни и нелинейни. Размити са правилата и функциите на принадлежност [23].

Подробен обзор на вземането на решения при размити критерии прави Фулър в [33].

Белман, Заде въвеждат използване на размити множества в областта на вземане на решения в неопределена среда [96]. Циерман развива техните идеи [99].

Размитите методи за вземане на решения при много условия се концентрират върху различните начини за оценка на относителната тежест на условията: размити адитивни тегла (Квакернак, Баас, Дюбоа, Прад, Чен), аналитичен йерархичен процес (Саати, Бъкли), размити конюнкции/дизюнкции (Дюбоа, Прад), максиминни методи (Белман, Заде, Ягер).

Приложения на размита логика при многокритериална задача за вземане на решения (MCDM) и подпомагане на процеса за вземане на решения (MCDA) са изложени в [74].

Размито математическо програмиране се развива в множество направления: гъвкаво програмиране (Танака, Циерман, Йоно); програмиране, базирано на теория на вероятностите (Танака, Дюбоа, Прад; линейно програмиране, базирано на теория на вероятностите с размит максимум (Дюбоа, Прад, Танака, Рамик); програмиране, базирано на теория на възможностите с размити предпочитания (Дюбоа, Прад, Негойца), линейно програмиране, базирано на теория на възможностите с размита целева функция (Танака, Бъкли).

Пенева и Попчев предлагат използване на теглови функции вместо константни тегла на размитите критерии и така моделират размити връзки между различните алтернативи в процеса на вземане на решения [73], [71]. Изследванията им върху

квадратни теглови функции за получаване на претеглена агрегация, с които се постига прецизност при решаване на някои многокритериални проблеми в процеса за вземане на решения, е публикувано в [72]. В [74] предлагат съответните алгоритми и програми за прилагане на моделите върху реални данни.

1.4.4.2. Приложения на еволюционни пресмятания

В приложенията, базирани на еволюционния подход се акцентира върху следните основни предимства: оптимизацията е възможна с дискретни и непрекъснати величини, няма нужда от информация относно производните, възможно е едновременно претърсване на голяма област от целевата повърхнина, възможно е опериране с голям брой променливи, има възможност за паралелизация, целевата повърхнина може да е изключително сложна, има възможност за излизане от локален минимум, като резултат се получават няколко оптимални стойности, а не само едно решение. [44]

През 1996 Холанд публикува системи за класификация, базирана на генетични алгоритми, през 1989 Дейвис – адаптивни системи, през 1990 Гарис – генетично програмиране.

През 1991 Белю и Букман представят разпределени еволюционни алгоритми. Тъй като при еволюционните алгоритми операторите се извършват едновременно върху всички индивиди от популацията, то те могат да бъдат реализирани върху паралелни компютърни системи.

1.4.4.3. Приложения на невронни мрежи

Използването на невронните мрежи е в намирането на архитектура на мрежа, подходяща за реализиране на нелинейна функция на няколко променливи за решаване на различни задачи за моделиране, идентификация, обработка на сигнали и др. [9], [19]

При задачите за класификация и разпознаване на образи, невронната мрежа натрупва знания за основните свойства на тези образи, разпределение на главните компоненти или за други характеристики в процеса на обучение. [29]

При обобщенията се акцентира върху различията между образите, което е основата за изработване на класификационните решения.

При задачите за управление на динамични процеси невронната мрежа изпълнява няколко функции. Първо, тя представя като нелинеен модел този процес и идентифицира неговите основни параметри, необходими за изработка на съответстващия управляващ сигнал. Второ, мрежата изпълнява функция на система, следяща за изменение на условията в заобикалящата среда и се адаптира към нея. Тя може да играе ролята на невронен регулатор, заместващ традиционните устройства. Важно значение, особено при управление на робот, има класификацията на текущото състояние и изработване на решение за по-нататъшното развитие на процеса. [49]

В задачите за асоциативност невронната мрежа изпълнява ролята на асоциативно запомнящо устройство. Ако на входа на мрежата се подаде вектор, съдържащ шум или вектор с липсващи отделни фрагменти от данните, то мрежата е способна да възстанови пълен и изчистен от шумове начален вектор. [41],

В областта на прогнозирането задачата на невронната мрежа е формулирането на предположение за бъдещото поведение на дадена система, като се използват нейни предишни състояния в зададена последователност. [15], [32]

Фрийман и Скапура разглеждат приложения на Adaline и Madaline при обработване на сигнали; на невронните мрежи с обратно разпространение на грешката за компресиране на данни, контрол на качество, разпознаване на образи и класификация на образи; на самоорганизиращи се мрежи за разпознаване на реч и записване на речта. [32]

1.4.4.4. Приложения на хибридни системи

В хибридните системи се използват предимствата на даден подход, за да се компенсират недостатъците на друг [16], [21], [24], [25].

Например, при използването на невронни мрежи основен проблем е намирането на оптимална архитектура за дадено приложение и тъй като генетичните алгоритми са основани на идеята за еволюцията, то те се използват за автоматично създаване на такава архитектура. Аналогично ГА се използват и при размитите системи за определяне на оптималния брой размити правила и размити променливи.

Касабов предлага инженерен подход за обясняване на невронните мрежи и размитите системи и приложенията им при създаване на системи, базирани на знания. Различни приложения на невронните мрежи за представяне и обработване на знания са представени и са илюстрирани с решени реални проблеми. Представени са хибридни конекционистки системи за разпознаване на реч и за предвиждане на изменения на цени на фондовата борса. Дискутирано е използването на невронни мрежи и размити системи за предвиждане и управление на хаотични системи [49].

Зилучиан изследва приложенията на невронните мрежи в различни направления: в медицината и биологията за изследване на рака, откриване на биологични сигнали, вземане на решения при планиране на лечение; в инженерните науки при филтриране, при управление на компютърните мрежи, в петролни рафинерии. Освен това предлага различни интелигентни системи, изградени със средствата на СК при навигация на автономни наземни роботи, размита логика за управление на автономни подводни роботи, размита логика за управление на климатични инсталации, приложение на размито-невронни системи за правене на изводи в роботиката; приложение на софтверен компютинг в технологиите за обезсоляване на водата, изчислителна интелигентност за разпознаване на обекти, еволюционни подходи за обработка на образи, еволюционни размити системи за управление на роботи. [98]

Кастило и Мелин анализират проблемите при разпознаване на образи с помощта на интелигентни системи. Освен размити системи, построени чрез средствата на традиционната теория на размитите множества от тип 1, се разглеждат и такива с интуиционистките размити множества и размити множества от тип 2. Описани са размито-невронни мрежи, модулни невронни мрежи, еволюционни алгоритми. Показани са приложенията на софтверен компютинг при клъстеризиране на обекти, разпознаване на човешко лице, разпознаване на пръстови отпечатащи, разпознаване на човешки глас, идентификация на човек чрез биометрични параметри - лице, пръстови отпечатащи и глас. [25]

Кастило и Мелин предлагат еволюционен дизайн на хибридни интелигентни системи, построени с йерархични генетични алгоритми. При йерархични генетични алгоритми, за разлика от стандартните генетични алгоритми, всяка хромозома се състои от два типа гени – за контрол и за връзка. Първите се представят побитово и служат за активиране, а вторите се представят с реални числа и представляват тегла. Основните предимства на този подход са преодоляването на локалните екстремуми и възможност за по-голямо генетично разнообразие. Еволюционният подход е използван

за оптимизиране на топологията на изкуствени невронни мрежи и за оптимизиране на размити ситеми – за оптимизиране на броя на размитите правила и на функциите на принадлежност. Показани са две приложения – в медицината за даване на анестезия на пациент и за предвиждане на времеви ред. Приложенията са реализирани в MATLAB. [24]

Конар създава хибридна система за криминално разследване, която е основана на размита логика, невронни мрежи и генетични алгоритми и се състои от модули за съвпадане на образи, класифициране и разпознаване на пръстови отпечатъци, идентификация на заподозрени по глас, идентификация на заподозрени по описания. [50]

Конар предлага реализация на познавателна система за роботи. Тук се акцентира върху придвижване на роботи и свързаните с това задачи за навигация. Създаването на план за движение и навигацията се извършват с хибридна система, базирана на генетични алгоритми, самоорганизиращи се карти, невронни мрежи с обратно разпространение на грешката. Освен индустриалните приложения е показан и модел на робот, играещ футбол. [50]

Гомез-Скармета, Хименес и Ибанес създават хибридна система, комбинираща средствата на размитата логика и еволюционните техники за идентифициране на характеристиките на променливите в първоначалната обработка на данни в процеса на дейта майнинг. [39]

Гупта предлага много изчерпателно и математически обосновано представяне на изкуствените невронни мрежи – статични и динамични. Представени са модели на обобщени динамични невронни мрежи (ДНМ), изолирани ДНМ, ДНМ с стимулирана и задържана динамика, динамични във времето (непрекъснат и дискретен случай) НМ на Хопфийлд, модел на Пинеда, анализирани са равновесните точки и стабилността (локална и глобална асимптотична) на ДНМ и на обучението и адаптирането на ДНМ. Разгледани са проблемите, свързани с хибридните размито-невронни мрежи и приложението им като универсални апроксиматори. [41]

Ху и Вукович предлагат използване на размити еволюционни алгоритми при генериране на траектория на роботи. [91]

1.4.5. Приложение на софт компютинг във финансовото инвестиране

Големият обем на информация и възможности (огромно количество на финансови средства, големият брой на финансови активи и инструменти, разнообразните софтуерни продукти) са само един от аспектите на неопределеността на финансовите пазари. При прилагане на различните модели финансовите анализатори се сблъскват със ситуации, в които не е очевиден правилният избор на параметри, тъй като икономическата ситуация се променя динамично и значимостта на предварително избраните фактори също се изменя, а дори възникват нови такива.

В такива обстоятелства софт компютингът предлага подходящи средства за справяне с проблеми, които в своята същност са неструктурирани и в случаи на непълна и/или липсваща информация.

1.4.5.1. Приложение на софт компютинг във финансовия инженеринг

Обзор на приложенията в областта е направен от Шапиро в [80].

Уайт (1988) пръв използва невронна мрежа за прогнозиране на пазара. Използва права мрежа за дневните изменения на цените на IBM, като получава твърде оптимистични резултати поради преобучаване на мрежата. Заключениеето му е, че тогавашното ниво на невронните мрежи не ги правят „машина за пари”.

Чанг (1996) използва права мрежа с обратно разпространение на грешката за прогнозиране на едногодишна доходност на взаимни фондове. Получените от него резултати са по-добри от традиционните регресионни модели.

Куо (1996) предлага интелигентна система за предвиждане на пазара с използване на количествени и качествени данни в интегрирана система на невронна мрежа и размит Делфи модел с данни от Тайванския пазар.

Ким и Чън (1998) използват пробабилистична невронна мрежа за предвиждане на фондовия пазар. В техния модел като резултат се получават няколко стойности с различни степени на значимост, а не единствена стойност.

Айкен и Бсат (1999) използват успешно права мрежа с обратно разпространение на грешката, обучена с генетичен алгоритъм за предвиждане на цените на тримесечни съкровищни бонове.

Тансел (1999) сравнява линейна оптимизация, невронна мрежа и генетичен алгоритъм за моделиране на времеви редове по критериите точност, удобство и време за пресмятания. При този анализ най-лоши резултати показва невронната мрежа.

След това други автори прилагат различни архитектури на невронната мрежа и алгоритми за обучение и достигат по-добри резултати.

Абрахам (2001) изследва хибридни техники за автоматизирано предвиждане на пазара и анализ на тренда. В модела са използвани невронни мрежи за едnodневни предвиждания и размито-невронна система за анализ на тенденцията в изменението на цените, като използват 24-месечни данни за изменение на Nasdaq и шест компании на фондовата борса.

Косака (1991) предлага ефективна система за вземане на решение от вида „купува/продава”, построена със средствата на размитата логика и невронните мрежи.

Уилсън (1994) развива напълно автоматична система за търговия на фондовата борса, използваща процедури за моделиране на хаоса, самоорганизиращи се карти (SOFM) за конструиране и управление на портфейл.

Косичич (1997) изследва генетични алгоритми за правила за търгуване, приложени върху времеви редове и концентрирани върху момента на действие.

Баба (2000) конструира система за подпомагане на вземане на решения с невронни мрежи и генетични алгоритми за анализ на TOPIX.

Интересни резултати от прилагане на невронно-хибридни системи СА публикувани в [19], [26] и [40].

1.4.5.2. Приложение на софт компютинг за решаване на портфейлната задача

Основен модел за конструиране на портфейл, използвайки теорията за вземане на решение с помощта на размита логика, е предложението от Рамасуами метод в [77], основан на Теорията за вземане на решение с помощта на размита логика [96].

При използване на тази теория не се прави разграничение между цели и ограничения, т.е. ако с X е означено множеството от алтернативни решения и съществуват m размити цели G_i ($i = 1, \dots, m$) и n размити ограничения C_j ($j = 1, \dots, n$), тогава размитото решение се дефинира като размито множество D с функция на принадлежност:

$$\mu_D(x) = \min(\mu_{G_1}(x), \dots, \mu_{G_m}(x), \mu_{C_1}(x), \dots, \mu_{C_n}(x)).$$

Оптималното решение се определя като неразмито подмножество

$$D^0 = \{x^* \mid x^* \in X, x^* \in \operatorname{argmax} \mu_D\}.$$

Основната идея при метода на Рамасуами е, че инвеститор може да конструира портфейл, базиран на m потенциални сценария от инвестиционната съвкупност от n актива, като x_i^{min} и x_i^{max} са минимален и максимален дял за i -тия актив. Нека R_{ik} представлява възвращаемостта на i -тия актив в k -тия инвестиционен сценарий и с

$$R_k(x) = \sum_{i=1}^n R_{ik}x_i$$

е означена възвращаемостта на портфейла за k -тия инвестиционен сценарий в края на инвестиционния период. Означавайки с R_k^{min} и R_k^{max} минимума и максимума на очакваната възвращаемост съответно на k -тия инвестиционен сценарий, удовлетвореността на инвеститора от портфейла x може да бъде определена със следната функция на принадлежност:

$$\mu_k(R_k(x)) = \begin{cases} 0, & \text{при } R_k(x) \leq R_k^{min} \\ \frac{R_k(x) - R_k^{min}}{R_k^{max} - R_k^{min}}, & \text{при } R_k^{min} < R_k(x) \leq R_k^{max} \\ 1, & \text{при } R_k(x) > R_k^{max} \end{cases}$$

Тогава задачата се състои в намирането на

$$\max(\mu_1(R_1(x)), \dots, \mu_m(R_m(x)))$$

при ограничителни условия

$$\sum_{i=1}^n x_i = 1$$

$$x_i^{min} \leq x_i \leq x_i^{max}, i = 1, \dots, n.$$

Подробен обзор и анализ на съществуващите модели за ребалансиране на портфейл, използващ размита логика за вземане на решения е публикувано в изследването на Йон Фанг и Кин Кун Лей от 2008 година. При оценка на възвращаемостите на активите са използвани тегла, променящи се във времето. Основният принос в моделите е включването на цената на транзакциите. Показани са и модели за ребалансиране на портфейли с разнородни активи. [31]

От съществуващите многобройни приложения на другите направления на софтверен компютинг в портфейлната теория, тук ще бъдат споменати само някои от първите публикации по тематиката.

Лоу през 1994 използва невронни мрежи за оптимизация на портфейл чрез минимизиране на риска.

Уенд през 1995 построява портфейлната ефективна граница чрез генетичен алгоритъм.

Гоу и Хуанг през 1996 използват линейно програмиране, базирано на теория на възможностите за оптимално разпределение на активите, като едновременно максимизират възвращаемостта, минимизират риска и максимизират възможността за постигане на по-висока възвращаемост.

1.5. Компютърни системи, работещи в реално време

Съществуват различни дефиниции за компютърни системи, работещи в реално време. Общото в тези дефиниции е, че системата трябва да спазва определени условия и ограниченията във времето, за да бъдат получени коректни и надеждни резултати от работата ѝ.

Времето между въвеждането на входните данни и получаването на всички изходни резултати се нарича *време за реакция* на компютърната система.

Компютърна система оперира в реално време, ако удовлетвори специфичните ограничения за времето за реакция, а ако не успее - надеждността при използването на тази система е под въпрос, което в някои случаи може да доведе и до сериозни опасности както за самата система, така и за потребителя.

Според [55] съществуват три вида реално време:

- soft real time – при неспазване на ограниченията във времето работата на системата се влошава, но това не води до критични нарушения;
- firm real time - няколко неспазени ограничения във времето не водят до критични нарушения, но повече довеждат до пълна и катастрофална невъзможност на системата да оперира коректно;
- hard real time – едно неспазено ограничение довежда до пълна и катастрофална невъзможност на системата да оперира коректно.

Проектирането и реализирането на системи, работещи в реално време, изисква внимание в няколко посоки:

- избор на хардуер и софтуер и оценка на цената на търсеното решение, включително и възможност за разпределени системи и въпроси, свързани с паралелни алгоритми и синхронизация на процесите;
- разбиране на особеностите на езиките за програмиране, свързани с превода на машинен език;
- максимизиране на толеранса към грешка и максимизиране на надеждността;
- проектиране и администриране на тестове;
- използване на технологията „отворен код“;
- измерване и предвиждане на времето за реакция и намаляването му чрез анализ на операциите.

Съществуват компютърни системи в реално време, реализирани на различни програмни езици: C, C++, C#, Java, Fortran, Pascal, Ada95, асемблер и пр. Изборът на език е пряко свързан с целите на конкретното приложение

1.6. Цел и задачи на дисертационния труд

Дисертационният труд е насочен към създаване и изследване на модели за управление на финансови активи в реално време, в които се използват средства на софт компютинг за моделиране на неопределености от размит вид.

Основните *проблеми* на управлението на финансови активи, могат да бъдат групирани както следва:

- събирането, съхраняването и анализиране на данните за цените на активите трябва да бъде в реално време с използване на достъпни изчислителни ресурси, за да може инвеститорът да вземе адекватно решение при управление на инвестицията;
- при портфейлна инвестиция е важно да се отчете проблемът с кардиналността;
- събитията на финансовите пазари не могат да бъдат повторени и предвидени, а несигурността и неопределеността са неизбежни на финансовите пазари. Софт компютинг предлага средства за моделиране на подобни ситуации.

Целта на дисертационния труд е да се създадат, реализират и тестват модели, основани на средствата на софт компютинг, за конструиране на портфейл с ограничени финансови ресурси.

За постигане на посочената цел са поставени следните *основни задачи*:

- (1) автономно събиране на данни за цени на активи в реално време; отчитане на проблема с липсващите данни; прецизиране на оценките на характеристиките на активи при дневни наблюдения с равни и неравни интервали от време между наблюденията;
- (2) анализ на данните със средствата на размитата логика за оценка на възвращаемостта и риска на активи;
- (3) отчитане на проблема с кардиналността при конструиране на портфейл с ограничени ресурси;
- (4) създаване на софтуерна система;
- (5) тестване на софтуерната система с реални данни.

При реализиране на описаните задачи се налагат следните *ограничения*:

- при решаване на така дефинираните задачи се приема, че редиците от цени на активи съдържат достатъчно информация за моментното състояние на макро и микроикономическите фактори. Тази информация може да се анализира и да се направят достатъчно добри краткосрочни предвиждания за бъдещите изменения на цените, които да подпомогнат процеса на вземане на инвестиционни решения;
- за провеждане на тестовете на моделите и софтуерната система са избрани акции, тъй като те са финансови активи с относително висока ликвидност и могат да са подходящи за краткосрочно инвестиране;

- времевият хоризонт на инвестициите е 1-3 месеца. Това са краткосрочни инвестиции, подходящи за по-консервативни инвеститори, а не за спекулиране на фондовата борса;
- при реалните експерименти наблюдаваните активи са всички акции, търгувани на БФБ-София АД, като данни за цената при затваряне се свалят ежедневно и се използват като входящи данни;
- приема се, че няма ограничения за купуване на желан брой акции.

Глава 2. Финансови активи. Размити системи

2.1. Оценяване на финансови активи

2.1.1. Характеристики на финансов актив

2.1.1.1. Възвращаемост на финансов актив

Нека A е финансов актив, за който са известни историческите данни за котировките в края на всеки от T последователни момента, т.е. известна е редицата $P(1), P(2), \dots, P(T)$, където $P(t)$ е пазарната цена на актива в момент t за $t = 1, 2, \dots, T$, като интервалите от време между моментите $t-1$ и t са равни помежду си. Тогава според [22]:

възвращаемостта¹ на A за периода $[t-1; t]$ е

$$r(t) = \frac{P(t)}{P(t-1)} \quad (2.1)$$

и нормата на възвращаемост^{2,3} за същия период е

$$r_N(t) = r(t) - 1 = \frac{P(t)}{P(t-1)} - 1 = \frac{P(t) - P(t-1)}{P(t-1)} \quad (2.2)$$

за $t = 2, \dots, T$.

При оценяване на възвращаемостта за периода $[1; T]$ може да бъде използвана средно аритметичната стойност на възвращаемостите

$$R = \frac{1}{T-1} \sum_{t=2}^T r(t) = \frac{1}{T-1} \sum_{t=2}^T \frac{P(t)}{P(t-1)} \quad (2.3)$$

или средно аритметичната стойност на нормите на възвращаемост

$$R_N = \frac{1}{T-1} \sum_{t=2}^T r_N(t) = \frac{1}{T-1} \sum_{t=2}^T (r(t) - 1) = \frac{1}{T-1} \sum_{t=2}^T r(t) - \frac{T-1}{T-1} = R - 1. \quad (2.4)$$

Въпреки лекотата, с която се пресмятат средноаритметичните величини, в случая на оценка на финансов актив по-информативна е средногеометричната величина, тъй като показва на инвеститора изменението на инвестираната сума. Тогава за периода $[1; T]$ са в сила следните формули:

1 Величината може да бъде записана в проценти след умножение със 100%.

2 Величината може да бъде записана в проценти след умножение със 100%.

3 Като синоним на „норма на възвращаемост“ се използва и терминът „доходност“.

средногеометричната възвращаемост е равна на

$$R_g = T^{-1} \sqrt[T]{\prod_{t=2}^T r(t)} = T^{-1} \sqrt{\frac{P(T)}{P(1)}} \quad (2.6)$$

средногеометричната норма на възвращаемост е равна на

$$R_{gN} = R_g - 1 = T^{-1} \sqrt[T]{R_T} - 1 \quad (2.7)$$

За пресмятане на средноквадратичното отклонение като мярка на отклонение от средноаритметичната величина е подходящо геометричните възвращаемости да се логаритмуват, при което за периода $[1; T]$ при $t = 2, \dots, T$ се получават следните величини:

логаритмична възвращаемост:

$$\ln(r(t)) = \ln\left(\frac{P(t)}{P(t-1)}\right) \text{ за подпериода } [t-1; t], \quad (2.8)$$

логаритмична средногеометрична възвращаемост

$$\overline{R^*} = \ln(R_g) = \ln\left(T^{-1} \sqrt[T]{\prod_{t=2}^T r(t)}\right) = \frac{1}{T-1} \sum_{t=2}^T \ln(r(t)) \quad (2.9)$$

В сила е равенството

$$R_g = e^{\overline{R^*}}. \quad (2.10)$$

Обща възвращаемост при дневни наблюдения с неравни интервали от време между тях

При използване на дневни наблюдения за цени на акции в редицата $P(1), P(2), \dots, P(T)$ понякога има липсващи данни, т.е. интервалите от време между наблюденията не са равни. Този проблем е дискутиран в 1.3. Един възможен подход за преодоляване на този проблем е формулите да се коригират, както е предложено в [34] и [38]:

$$R_g = k \sqrt[k]{\prod_{t=2}^T r(t)} = k \sqrt[k]{\frac{P(t)}{P(1)}} \quad (2.6')$$

$$R_{gN} = R_g - 1 \quad (2.7')$$

$$\overline{R^*} = \ln(R_g) = \ln\left(k \sqrt[k]{\prod_{t=2}^T r(t)}\right) = \frac{1}{k} \sum_{t=2}^T \ln r(t) \quad (2.9')$$

където Δ_t е броят на дните между наблюденията $P(t-1)$ и $P(t)$, и $k = \sum_{t=2}^T \Delta_t$.

Друг възможен подход е използваният в настоящата дисертация, а именно всяка цена да се „дописва“ до момента, в който е осъществена нова сделка за съответния актив.

Годишна възвращаемост при дневни наблюдения с неравни интервали от време между наблюденията

При вземане на инвестиционни решения е подходящо да се използва сравнителна мярка за оценяване на активите и за целта може да бъде използвана годишната възвращаемост на активите или годишната им норма на възвращаемост.

Тези величини се получават от логаритмичната средногеометрична възвращаемост, като

годишната норма на възвращаемост е равна на

$$ANR = \left(e^{(T-1)\bar{R}} - 1 \right) \cdot \frac{D}{T-1}, \quad (2.11)$$

където D е броят на дните във финансовата година и

годишната възвращаемост е равна на

$$AR = ANR + 1. \quad (2.12)$$

2.1.1.2. Риск на актив

В процеса на вземане на инвестиционно решение освен оценка на постигнатата възвращаемост е важно да има достоверна оценка и за нейното изменение. В инвестиционната теория често използвана мярка за риска е дисперсията на възвращаемостта.

Подходяща мярка на колебанията на възвращаемостите е средноквадратично отклонение s , което се пресмята по следните формули:

при използване на средноаритметичната на възвращаемостите

$$s_a = \sqrt{\frac{1}{T-2} \sum_{t=2}^T (r(t) - R)^2} \quad (2.13)$$

и при логаритмична средногеометрична възвращаемост

$$s_g = \sqrt{\frac{1}{T-2} \sum_{t=2}^T (\ln(r(t)) - \bar{R})^2}. \quad (2.14)$$

Коефициент q -ratio

При сравняване и избор на активи с различни възвращаемост и риск е подходящо да бъде използвана и една допълнителна величина, която да дава възможност да бъде измерена получената възвращаемост за единица поет риск.

Такава величина може да бъде частното на възвращаемостта и риска. Тъй като основна цел на всеки инвеститор е постигане или на максимална възможна възвращаемост или на минимално възможен риск, то частното на двете величини също следва да е максимално възможно.

В настоящата дисертация като такава допълнителна характеристика на активите е използван вариант на коефициента на Шарп [83], а именно коефициентът q – ratio:

$$q = \frac{AR}{s_g}, \quad (2.15)$$

където:

AR е годишната възвращаемост, получена по формула (2.12) от логаритмичната средногеометрична възвращаемост $\overline{R^*}$ и

s_g е средноквадратичното отклонение на възвращаемостите, получено по формула (2.14).

Възможно е използването и на други допълнителни характеристики, като например коефициента на вариация, ковариациите на възвращаемостите и т.н.

2.1.2. Инвестиционен портфейл без къси продажби

Инвестиционният портфейл е специфичен вид инвестиция, която се състои от няколко актива с различни характеристики. Важно е да се подчертае, че инвестиционният портфейл не е просто съвкупност от ценни книжа - значение имат не само характеристиките на отделните ценни книжа, а и взаимовръзките между тези характеристики (пр. ковариациите на възвращаемостите).

Изграждането и управлението на инвестиционен портфейл е метод за управление на инвестиционния риск, като за разлика от диверсификацията, целта не е механично увеличаване на броя на инвестиционните носители, а чрез разглеждане на портфейла като динамична съвкупност от различни активи да се достигне желаната възвращаемост. [10]

В портфейлната теория са предложени различни описания на управлението на инвестиционен портфейл [4], [10], [43], [68], [69]. Общ аспект на тези описания е разглеждането на управлението на инвестиционен портфейл като процес. Етапите на такъв процес могат да бъдат обобщени до следните:

I. Стратегическо конструиране на портфейла, при който се разглеждат активи от общи групи и се избира група (или групи), която да се инвестира.

II. Тактическо конструиране на портфейла, при който се анализират активи от една група. За този етап са предложени разнообразни методи – оптимизационни алгоритми, сортиращи алгоритми, търсещи алгоритми, невронни мрежи, генетични алгоритми и т.н.

III. Избор на портфейл, който максимизира икономическата изгода на инвеститора. Портфейлът може да бъде променен в зависимост от краткосрочните изменения на пазара с опции и фючърси или с включването или изключването на активи от паричните пазари.

IV. Наблюдение и оценка на портфейла – след избора и закупуването на портфейла, се проследява дали представянето му покрива очакванията за възвращаемост.

V. Промяна на портфейла – при промени в икономическите условия, портфейлът се променя, като може да се наложи повтаряне на етапите.

В настоящата дисертация акцентът е върху втория етап.

2.1.2.1. Конструирание на инвестиционен портфейл без къси продажби при ограничени финансови ресурси

За целите на изследването е използвана следната математическа формулировка на задачата за конструирание на портфейл без къси продажби при ограничени финансови ресурси.

Нека в момента t_0 е налична първоначална сума S_0 . Целта на инвестирането е след даден период от време Δt тази сума да нарасне до $S^* = S_0(1 + R^*)$, където R^* е желаната възвращаемост.

Нека A_1, A_2, \dots, A_m са финансови активи.

Задачата е да се конструира съвкупност от тези активи със съответни дялове x_1, x_2, \dots, x_m , така че в момента $t^* = t_0 + \Delta t$ да са изпълнени следните условия:

$$\sum_{j=1}^m v_j \cdot P_{j0} \leq S_0 \quad (1)$$

$$v_j \geq 0, \quad (2)$$

$$v_j - \text{цяло число}, \quad (3)$$

$$\sum_{j=1}^m x_j \cdot r_j^* \geq R^*, \text{ като } x_j = \frac{v_j \cdot P_{j0}}{\sum_{j=1}^m v_j \cdot P_{j0}}; \quad (4)$$

където x_j е относителният дял на участие на актива A_j в портфейла, $j = 1, 2, \dots, m$;

r_j^* е възвращаемостта на актива A_j в момента t^* ;

P_{j0} е цената на актива A_j в момента t_0 ;

v_j е броят на включените в p акции от актива A_j за $j = 1, 2, \dots, m$.

Съвкупността $p = (x_1; x_2; \dots; x_m)$, получена като решение на така формулираната задача в настоящата дисертация се разглежда като финансов портфейл.

Основна роля за определяне на дяловете x_j на съответните активи играят величините r_j^* . Тъй като това са възвращаемости за бъдещ момент, то те са неизвестни величини. Каквито и методи да се използват при оценяване или предвиждане на тези бъдещи възвращаемости, винаги съществува разлика между оценката им и реалните им стойности.

При оценка на характеристиките на инвестиционния портфейл са възможни различни подходи в зависимост от метода на определяне на характеристиките на отделните активи и от използвания теоретичен модел. За целите на настоящото изследване възвращаемостта на портфейла е разглеждана като линейна комбинация на възвращаемостите на включените в него активи, а портфейлният риск е представен като средноквадратично отклонение.

2.1.2.2. Възвращаемост на инвестиционен портфейл

Възвращаемостта R_p на инвестиционен портфейл се пресмята по формулата:

$$R_p = \sum_{j=1}^m x_j \cdot r_j \quad (2.16),$$

където r_j е възвращаемостта на j -тия актив. За целите на настоящата дисертация като оценка на възвращаемостта е използвана средногеометричната възвращаемост R_g , пресметната по формула (2. 6').

2.1.2.3. Риск на инвестиционен портфейл

Средноквадратичното отклонение е равно на:

$$s_p = \sqrt{\sum_{j=1}^m x_j \cdot (r_j - R_p)^2} \quad (2.17)$$

2.1.2.4. Коефициент *q-ratio* за портфейл

Коефициентът *q-ratio* е оценка на отношението на възвращаемостта и риска на портфейла:

$$q_p = \frac{R_p}{s_p} \quad (2.18)$$

2.2. Решаване на портфейлната задача в реално време

Разработките за вземане на инвестиционни решения в реално време са концентрирани предимно върху предоставяне и обработка на финансови данни в реално време, но не предлагат цялостно решение на портфейлната задача.

Съществуват и различни софтуерни системи, които предлагат използването на множество инструменти и различни възможности, които могат да бъдат използвани за подпомагане вземането на инвестиционни решения (пр. [102], [103], [104]), но в нито една от тях не се коментира въпросът за решаване на портфейлната задача в реално време.

В областта на портфейлните инвестиции интерес представлява предложената през 2002 система с обектно-ориентирана база от знания, чийто домейн е описан с йерархични класове и съдържа веб агенти за достъп до нужната информация [90]. В тази система е въведена концепция за спешност при вземане на инвестиционните решение, с помощта на която наличните изчислителни ресурси се преразпределят.

Портфейлната оптимизация като информационна услуга в интернет е публикувана в монографията на Т. Стоилов [12]. В този труд са изследвани тенденциите на развитие и приложение на информационните услуги в Интернет. На тази основа е развит информационен и алгоритмичен модел за обработка на сложно структурирани данни, удовлетворяващ изискванията за изпълнение в реално време. Конкретна цел е създаване на информационна услуга за оптимизиране на финансови инвестиции. За изпълнението на алгоритмичната обработка на данните са проектирани и реализирани методи за квадратично програмиране, известни в йерархичните системи

като неитеративна координация. Математически са изведени модели на портфейлна оптимизация с няколко времеви хоризонта, модел с къси продажби, модел с дълги продажби. Създадена е онлайн система (<http://hs9.iccs.bas.bg/>) за консултантски услуги и портфейлна оптимизация в процеса на финансово инвестиране.

2.3. Размита логика и размити системи

След въвеждането на идеята за размити множества от Л. Заде през 1965 в [95], теорията на размитите множества постоянно се развива. За нуждите на настоящото изследване са използвани предимно литературни източници, предлагащи приложения на тази теория [20], [21], [30], [45], [47], [49], [78], [94], [96], [97], [99] и за тази цел ще бъдат дадени само основни понятия.

2.3.1. Размити множества

Нека U е универсум и $A \subseteq U$ е фиксирано негово подмножество. Множеството

$$\Phi_A = \{ \langle x, \mu_A(x) \rangle \mid x \in U \}$$

се нарича *размито множество* върху U , ако функцията μ_A , наречена *степен на принадлежност*, задава за всяко $x \in U$ точно едно реално число, принадлежащо на интервала $[0, 1]$.

От съображения за опростяване на записа по-надолу вместо „размитото множество Φ_A ” ще бъде използвано „размитото множество A ”.

Размитото множество A има точно толкова елемента, колкото и универсума U .

Обикновеното множество $s_A = \{x \mid x \in U, \mu_A(x) > 0\}$ се нарича *носител* на размитото множество A . Множеството s_A е празно множество, когато $\mu_A(x) = 0$ за всяко $x \in U$.

Обикновеното множество $c_A = \{x \mid x \in U, \mu_A(x) = 1\}$ се нарича *ядро* на размитото множество A .

Размитото множество A се нарича *нормално*, ако $c_A \neq \emptyset$, т.е. съществува поне едно $x_0 \in U$, такова че $\mu_A(x_0) = 1$.

Ако $\mu_A(x) < 1$ за всяко $x \in U$, то размитото множество A се нарича *субнормално* и може да бъде нормализирано, като $\mu_A(x)$ се раздели със $\sup \mu_A(x)$, т.е. множеството

$$A_1 = \left\{ \left\langle x, \frac{\mu_A(x)}{\sup \mu_A(x)} \right\rangle \mid x \in U \right\}$$

е нормално размито множество.

Размит *синглетон* с един елемент $x_0 \in U$ е размито множество:

$$A = \{ \langle x_0, \mu_A(x_0) \rangle \}.$$

Размитото множество A се нарича *изпъкнало* тогава и само тогава, когато за всеки $x_1, x_2 \in U$ и всяко реално число $\lambda \in [0, 1]$ е изпълнено

$$\mu_A(\lambda x_1 + (1 - \lambda)x_2) \geq \min\{\mu_A(x_1); \mu_A(x_2)\}.$$

Нормално и изпъкнало размито множество върху множеството на реалните числа R се нарича *размито число*.

Основни операции с размити множества

За размитите множества $A = \{ \langle x, \mu_A(x) \rangle \mid x \in U \}$ и $B = \{ \langle x, \mu_B(x) \rangle \mid x \in U \}$ се дефинират следните основни операции.

Размитото множество

$$\bar{A} = \{ \langle x, \mu_{\bar{A}}(x) \rangle \mid x \in U \}$$

се нарича *допълнение* на A , ако $\mu_{\bar{A}}(x) = 1 - \mu_A(x)$ за всяко $x \in U$.

Размитото множество

$$C = \{ \langle x, \mu_C(x) \rangle \mid x \in U \}$$

се нарича *сечение* на множествата A и B ($C = A \cap B$), ако $\mu_C(x) = \min\{\mu_A(x); \mu_B(x)\}$ за всяко $x \in U$.

Размитото множество

$$D = \{ \langle x, \mu_D(x) \rangle \mid x \in U \}$$

се нарича *обединение* на множествата A и B ($D = A \cup B$), ако $\mu_D(x) = \max\{\mu_A(x); \mu_B(x)\}$ за всяко $x \in U$.

Размитото множество

$$E = \{ \langle x, \mu_E(x) \rangle \mid x \in U \}$$

се нарича *алгебрично произведение* на A и B ($E = A \cdot B$), ако $\mu_E(x) = \mu_A(x) \cdot \mu_B(x)$ за всяко $x \in U$.

Размитото множество

$$F = \{ \langle x, \mu_F(x) \rangle \mid x \in U \}$$

се нарича *степен* γ на множеството A при $\gamma > 0$, ако $\mu_F(x) = \mu_A^\gamma(x)$ за всяко $x \in U$.

За размити множества се дефинират и обобщени операции: размито допълнение, размито сечение (T -норма) и размито обединение (S -норма).

2.3.2. Размита логика

Основните понятия в размитата логика са *лингвистични променливи*, *лингвистични модификатори*, *съждителна размита логика*, *дедуктивни правила за извод*, *апроксимации*.

2.3.2.1. Лингвистични променливи

Лингвистична променлива се задава с наредената петорка $(\eta, T(\eta), U, G, M)$, където η е името на променливата; $T(\eta)$ е терм множеството от лингвистичните стойности (терми) на променливата η , като всяка от тези лингвистични стойности е размито множество върху универсума U ; G е синтактично правило, което генерира термите на $T(\eta)$ и M е семантично правило, което асоциира всяка лингвистична стойност T с нейния смисъл $M(T)$, като $M(T)$ е размито множество върху U .

2.3.2.2. Съждителна размита логика

Нека $K = \{ \langle x, \mu_K(x) \rangle \mid x \in U \}$ и $\Lambda = \{ \langle x, \mu_\Lambda(x) \rangle \mid x \in U \}$ са размити множества. В размитата логика основно се оперира с три вида съждения:

- (i) $v \in K$ - съждение в канонична форма;
- (ii) $v \in mK$ - модифицирано съждение;
- (iii) ако $v \in K$, то $w \in \Lambda$ - условно съждение.

Нека p и q са съждения от вида (i).

Съждението $p \wedge q$ се нарича *конюнкция* на p и q , ако
$$v(p \wedge q) = \min(\mu_K(x); \mu_\Lambda(y)).$$

Съждението $p \vee q$ се нарича *дизюнкция* на p и q , ако
$$v(p \vee q) = \max(\mu_K(x); \mu_\Lambda(y)).$$

Съждението $p \rightarrow q$ се нарича *импликация*⁴ на p и q , ако
$$v(p \rightarrow q) = \min(1; 1 - \mu_K(x) + \mu_\Lambda(y)).$$

Размити АКО-ТО правила

*Размито АКО-ТО правило*⁵ има вида:

АКО $x \in A$, *ТО* $y \in B$;

където A и B са лингвистични стойности, дефинирани с размити множества върху универсума U . Първата част „ $x \in A$ ” се нарича *предпоставка*, а втората „ $y \in B$ ” - *заклучение*. Използва се и означението $A \rightarrow B$.

⁴ Въпреки, че в размитата логика се използват различно дефинирани импликации, за целите на настоящата дисертация тази дефиниция е достатъчна.

⁵ Освен терминът „размито АКО-ТО правило” се използват и термините „размито правило”, „размита импликация” или „размито условно твърдение”.

2.3.3. Размито моделиране

При проектиране на размита система за вземане на решение като база се използва предварително известно поведение на дадена система, като размитата система трябва да е в състояние да наподобява това поведение.

Процесът на създаване на размита система, който се нарича размито моделиране, има следните характеристики:

- структурата на размитата система трябва да е така проектирана, че опитът на експертите да може лесно да се пресъздаде, т.е. размитото моделиране да се възползва максимално от познаването на изследваната област (което понякога не е възможно при други подходи);
- в случай, че са известни входните и изходни данни, да е възможно прилагането на стандартни техники за идентифициране на системата, т.е. числените данни са важни точно толкова, колкото и при всеки друг математически метод за моделиране.

Основните етапи, които се следват при размитото моделиране, са два.

1. Идентификация на общата структура.

В този етап се използва собствено знание (здрав разум, физични закони и т.н.), информация от експерти или „опити и грешки”. Последователните стъпки са:

- a. избор на подходящи входни и изходни променливи;
- b. избор на конкретна размита система за изводи;
- c. определяне броя на лингвистичните терми на входните и изходните променливи;
- d. проектиране на множеството от „ако-то” правила.

2. Идентификация на структурата в дълбочина чрез описание на лингвистичните терми чрез:

- a. избор на фамилия от функции на принадлежност;
- b. избор на стойности на параметрите за всяка от функциите на принадлежност;
- c. конкретизиране на стойностите на параметрите.

Предимствата на размитите системи и видовете размити системи са разгледани в точка 3.1.

Глава 3. Размита система за изводи

3.1. Размити системи и размити експертни системи

3.1.1. Принципи на създаването на размита система за изводи

Размита система за изводи (fuzzy inference system=FIS) е изчислителна структура, основана на теорията на размитите множества, правила от вида *АКО-ТО* и размитата логика. Тъй като размитите системи за изводи имат разнообразна структура и предназначение, за тях се използват различни наименования като: размита експертна система, размит модел, размита асоциативна памет, контролер с размита логика, размита система и др.

Основната структура на размита система съдържа три концептуални компонента:

- база от правила, където са включени всички размити правила за вземане на решения;
- база от данни (речник), където са дефинирани всички функции на принадлежност, използвани в размитите правила;
- апарат за изводи, който изпълнява процедурата за вземане на решения от правилата и дадените факти, за да се получи коректен изход или заключение;

и задължително съдържа четири основни модула:

- модул за размиване и размити променливи за вход;
- база от размити правила;
- машина за размити изводи;
- модул за деразмиване и размити променливи за изход.

В резултат от прилагането на правилата над входните размити променливи машината за размити изводи присвоява стойности на изходните размити променливи.

Експертни системи, в които се прилагат методите на размитата логика, се наричат размити експертни системи. При тях се използват размити данни, размити правила и размити изводи.

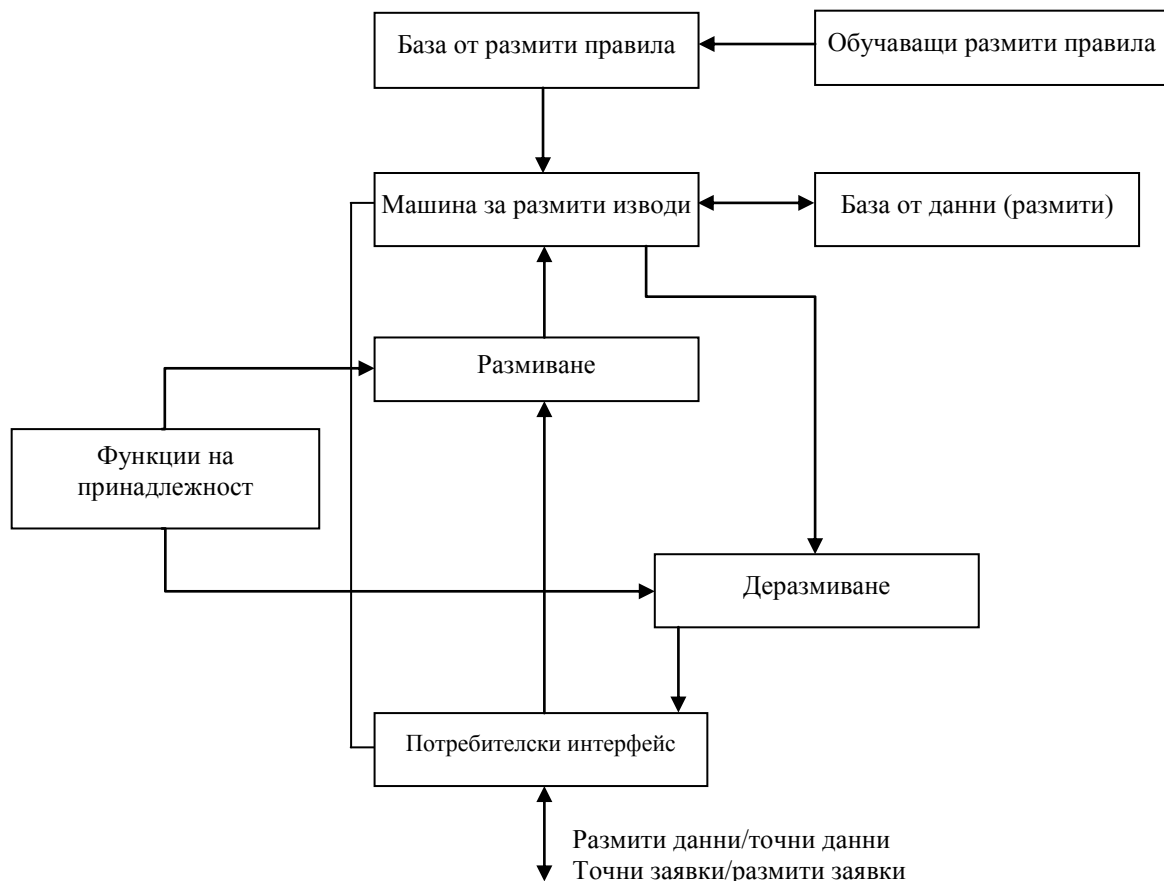
Освен с модул за размиване, база от размити правила, машина за размити изводи и модул за деразмиване, размитата експертна система разполага с потребителски интерфейс и база от данни. На фиг. 3.1 е изобразена принципна схема на размита експертна система.

Потребителският интерфейс дава възможност на системата да комуникира с потребителя или да обменя данни с обкръжаващата я среда.

Модулът за размиване служи за изчисляване степените на принадлежност, с които всяка входна стойност принадлежи на определени размити терми. Тези размити терми представляват състоянията на една от размитите променливи за вход. Функциите за принадлежност са предварително дефинирани.

Базата от размити правила и функциите на принадлежност изграждат базата от знания на размитата експертна система. Размитите правила се използват за правене на изводи от наличните данни.

Базата от данни съдържа данни за текущия проблем, получени от потребителя или от обкръжаващата системата среда. Базата от данни се използва от някои размити системи за съхраняване на данни, придобити при решаването на проблеми. Тези данни се запазват с цел използването им за решаване на подобни проблеми в бъдеще и дори за извличането на нови размити правила. Ако съществува подсистема за обясняване на резултатите, нужните ѝ междинни изводи също се запазват в базата от данни.



Фигура 3.1. Принципна схема на размита експертна система

В машината за размити изводи са реализирани методите за правене на изводи. Съществуват два подхода за правене на изводи от наличните размити правила и входни данни. При единия се изпълняват всички правила с текущите налични данни и се извличат всички възможни изводи. При другия подход се избират правилата, с които може да се достигне до решение. След това се търсят данни, които удовлетворяват условната част на тези правила. Основната разлика е в последователността на изпълнение на всяко от правилата. В единия случай се изпълнява първо условната част, след което частта за извод, докато във втория се започва от частта за извод, а след това евентуално се преминава към условието. Вторият подход често е по-подходящ, тъй като при него се изпълняват само тези правила, с които може да се достигне до решение. По този начин не се извеждат трудните за добиване данни, ако не са необходими за намиране на решението.

Модулът за деразмиване има за цел да изчисли една единствена числена стойност за дадена размита променлива за изход. Това се извършва чрез прилагане на

подходящ метод за деразмиване над изведената функция на принадлежност за размитата променлива.

Блокът обучаващи размити правила не е задължителен и служи за придобиване на нови правила.

Ако съществува подсистема за обясняване на резултатите, то тя трябва да е свързана с машината за размит извод (или с базата от данни) и с потребителския интерфейс. Подходящо място за нея е блокът за потребителски интерфейс.

3.1.2. Видове размити системи

3.1.2.1. Видове размити системи за вземане на решения

Според предназначението си размитите системи се разделят основно на системи за управление и диагностика и системи за подпомагане процеса за вземане на решения.

Системи за управление и диагностика

Размита система за управление и диагностика използва размити правила и машина за размити изводи за решаването на проблем, свързан с управлението или диагностиката. Към проблемите в тази област спадат както сложни задачи като управлението на работи чрез координирането на множество действия, така и по-елементарни като поддържане стойностите на променлива в предварително установени граници.

Популярно название на система за управление е контролер. За разлика от обикновените контролери размитите контролери са проектирани и реализирани така, че да прилагат предварително придобити експертни знания. Размитите контролери се използват основно за решаването на проблеми, за които създаването и използването на точен математически модел е трудно, невъзможно или прекалено скъпо. Причина за тези трудности могат да бъдат: сложно изменящи се във времето процеси, наличие на шум, невъзможност за извършване на точни и надеждни измервания и др.

Принципът на работа на размита система за управление може да се представи като циклично изпълняване на следната последователност от стъпки:

- извършват се нужните измервания, чрез които се получават данни за управлявания процес;
- данните от измерванията се преобразуват, като се пресмятат степени на принадлежност към подходящи размити множества. Този процес се нарича *размиване*;
- размитите данни се използват от машината за размити изводи за изпълняване на правилата. В резултат се получава едно или няколко размити множества, дефинирани над множеството от възможни действия;
- размитото множество се преобразува в единствена точна стойност. Процесът се нарича *деразмиване*, а стойностите представляват действията, които трябва да се приложат според системата за управление.

Всяка размита система за вземане на решения следва описания алгоритъм, но цикличното изпълнение на стъпките е характерно за размитите системи за управление.

Размитите системи за диагностика извършват измервания, които в зависимост от случая могат да са циклични, еднократни или в зависимост от състоянието на системата, подложена на диагностика.

В случай на управление на система с цел поддържане на желана стойност за дадена променлива, независимо от шума от обкръжението на системата, управляващата размита система наблюдава две променливи. Едната е грешката, дефинирана като разликата между текущата и желаната стойност за променливата, а втората – степента на изменение на грешката. Размитата система за управление приема стойностите на тези променливи като входни, а решението на изхода е стойност за управляващата променлива.

Системи за подпомагане процеса за вземане на решения

И тези системи служат за решаване на проблеми от дадена област, но за разлика от системите за управление и диагностика, те не прилагат намереното решение. Тяхната задача е да предложат решение на потребителя, с което той евентуално се съобразява.

Системи, подпомагащи вземането на решения, се създават по няколко причини. Една от тях е наличието на сложни проблеми, за които е невъзможно определянето и снабдяването на системата с всички нужни данни за намиране на търсеното решение. Понякога естеството на даден проблем не позволява дефинирането на множество от възможните алтернативи или решения. Друга причина е нуждата от безпристрастно мнение и препоръка при вземането на важни решения, отговорността за които може да носи само човек. Съществуват случаи, в които намереното решение не се прилага от системата единствено поради липсата на надежден механизъм за това.

Характерна черта в архитектурата на системата за подпомагане процеса за вземане на решения е наличието на удобен потребителски интерфейс, който понякога се комбинира с подсистема за обясняване на резултатите.

3.1.2.2. Видове размити системи според изходните променливи

Размити модели от тип Мамдани

Размит модел за извод от тип Мамдани е сред първите системи, изградени на база теорията на размитите множества. Тя е предложена през 1975 от Ейбрахам Мамдани за управление на парна машина и парен котел чрез лингвистични правила, съобразени с човешкия опит на операторите [57]. Модел от вида Мамдани се характеризира с това, че термите на изходните променливи са размити числа, които се деразмиват по различни методи.

С избор на други оператори и композиции се получават различни варианти на модела, но общото на размитите модели от тип Мамдани е, че изходните резултати са размити множества и следователно е нужна деразмиване. Има различни подходи за деразмиване:

- метод на центъра на тежестта, при който се пресмята

$$x_{CoG} = \frac{\int \mu_A(x).x.dx}{\int \mu_A(x).dx} ;$$

- метод на медианата, при който се пресмята стойността x_{BoA} , такава че

$$\int_{\alpha}^{x_{BoA}} \mu_A(x).dx = \int_{x_{BoA}}^{\beta} \mu_A(x).dx,$$

където $\alpha = \min\{x \mid x \in X\}$ и $\beta = \max\{x \mid x \in X\}$;

- метод на средното на максимумите, при който се пресмята

$$x_{MoM} = \frac{\int_{X'} x.dx}{\int_{X'} dx},$$

където $X' = \{x \mid \mu_A(x) = \mu^* : \mu^* = \max_X(\mu_A(x))\}$;

- метод на минимума на максимумите, при който се пресмята

$$x_{SoM} = \min\{x \mid \mu_A(x) = \mu^* : \mu^* = \max_X(\mu_A(x))\};$$

- метод на максимумата на максимумите, при който се пресмята

$$x_{LoM} = \max\{x \mid \mu_A(x) = \mu^* : \mu^* = \max_X(\mu_A(x))\}.$$

При модел от тип Мамдани на всеки оператор съответства размита операция, като най-често:

- за оператора AND се използва размито сечение;
- за оператора OR се използва размито обединение;
- за оператора за импликация се използва \min оператор при дадени тегла на правилата;
- за оператора за агрегация се използва \min за агрегиране на функциите на принадлежност при генериране на изходна променлива;
- за оператора за деразмиване се използва някой от посочените по-горе подходи за трансформиране на изходната размита променлива до една точна стойност.

Размити модели от тип Сугено

Този модел е предложен от Такаги, Сугено и Канг като опит за систематичен подход за генериране на размити правила от зададени множества от входни и изходни данни [47], [87]. Размитото правило в модел от тип Сугено има вида:

$$\text{ако } x \text{ е } A \text{ и } y \text{ е } B, \text{ то } z = f(x;y),$$

където A и B са размити променливи в предпоставките, а $f(x;y)$ е функция.

Когато $z = f(x;y)$ е константа, моделът се нарича модел на Сугено от нулев ред и изходът е синглетон, а ако е полином от първа степен – модел на Сугено от първи ред.

Моделите от тип Сугено са подходящи за управление на множество от линейни контролери, както и за управление на динамични нелинейни системи. Размитите системи от тип Сугено са изключително подходящи при малки непрекъснати изменения, приложени върху входното пространство и за моделиране на нелинейни системи чрез множество линейни компоненти.

Размити модели от тип Цукамото

В размит модел от тип Цукамото изходната променлива се състои от размити множества с монотонни функции на принадлежност [47]. След изпълнението на всяко разрито правило се получава точна стойност, индуцирана от силата на изпълнение на това правило. Окончателно на изхода се получава точна стойност, която е претеглено средно на изхода от всяко правило. При този модел деразмиване не е нужно.

3.2. Размита система за изводи, използваща модела FLQM за оценка на финансово активи

3.2.1. Архитектура на размитата система

При проектирането на предложената в тази дисертация размита система за изводи са следвани четирите етапа на процеса за получаване на изводи:

- изчисляване на входните данни за всяко правило;
- получаване на извод за всяко правило;
- агрегиране на всички изводи;
- деразмиване.

Ако в базата от размити правила има n на брой правила, α , β и γ са размити променливи, машината за размит извод следва схемата:

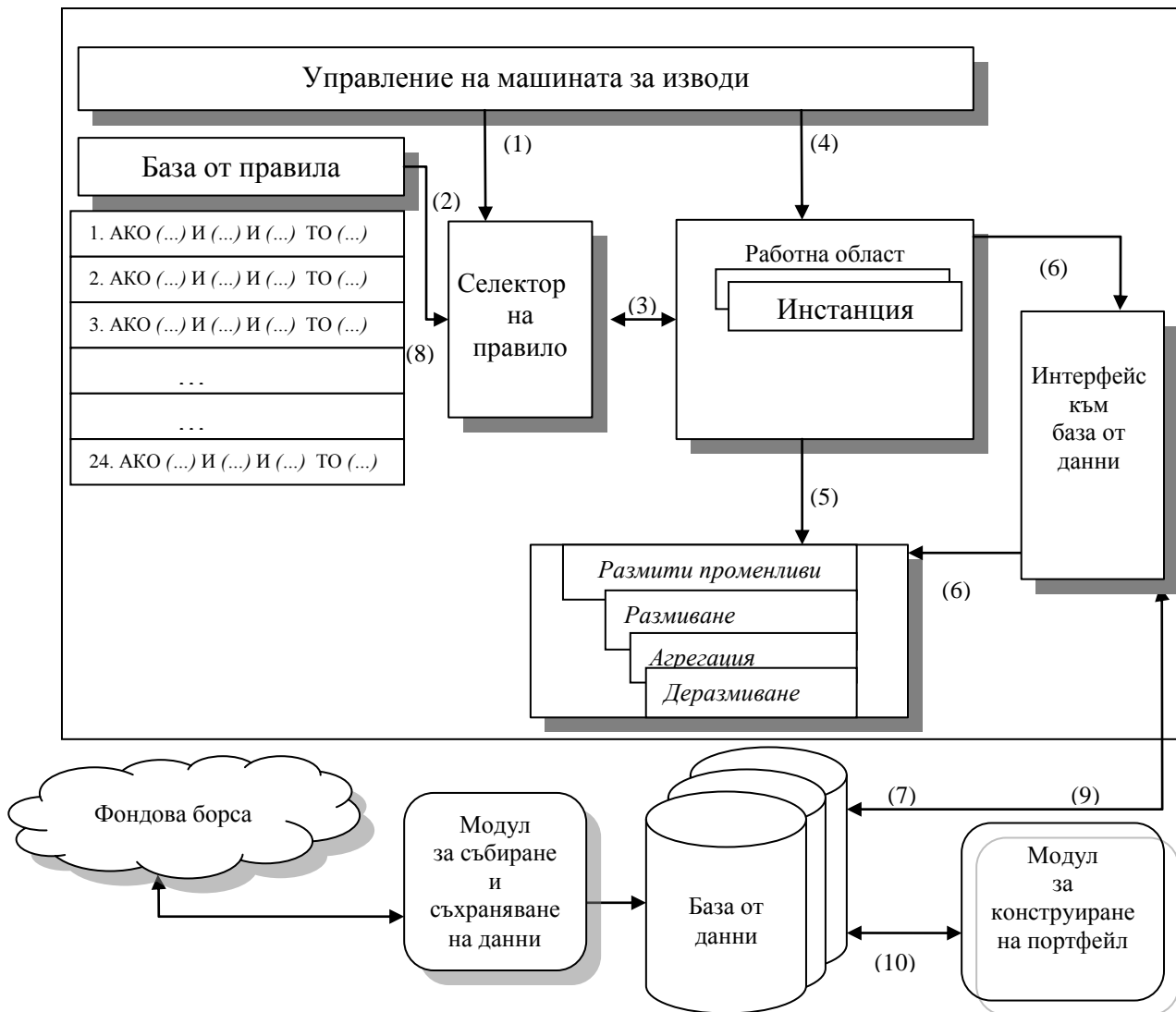
<i>Правило 1:</i>	<i>Ако $\alpha \in A_1$ и $\beta \in B_1$, то $\gamma \in C_1$</i>
<i>Правило 2:</i>	<i>Ако $\alpha \in A_2$ и $\beta \in B_2$, то $\gamma \in C_2$</i>
.....	
<i>Правило n:</i>	<i>Ако $\alpha \in A_n$ и $\beta \in B_n$, то $\gamma \in C_n$</i>
<i>Факт:</i>	<i>$\alpha \in A'$ и $\beta \in B'$</i>
<i>Извод:</i>	<i>$\gamma \in C'$</i>

От наличните правила и фактите ($\alpha \in A'$ и $\beta \in B'$) се прави извода $\gamma \in C'$, където A' , A_i , B' , B_i , C' и C_i са размити множества.

Важен момент е моделирането на функцията за агрегация, която служи за обединяване на направените изводи и дефиниране по този начин на размито множество за всяка размита променлива за изход.

На фигура 3.2 е показана схема на архитектурата на създадената размита система за изводи за вземане на решение, която използва модела FLQM⁶.

⁶ Моделът FLQM е описан в т. 3.2.2.



Фигура 3.2. Управление на размита система за изводи, основана на модела FLQM:

- (1) активиране на селектора;
- (2) избор на правило;
- (3) копиране на шаблона;
- (4) активиране на правилото;
- (5) обръщане към Модул *Размита логика*: *Размити променливи*, *Размиване*, *Агрегация*, *Деразмиване*;
- (6) връзка с интерфейса;
- (7) четене от базата от данни;
- (8) обработване на следващото правило;
- (9) записване на резултата в базата от данни;
- (10) конструиране на портфейл.

Модулът за събиране и съхраняване на данни⁷, се обръща към съответната страница на фондовата борса. Получените от там данни и допълнително изчислените им характеристики се записват в базата данни. След това управлението на машината за

⁷ Модулът за събиране и съхраняване на данните, както и модулът за конструиране на портфейл, е описан в следващата глава.

изводи активира селектора на правило, като в базата от правила има 24 правила, които се избират последователно. Избраното правило се копира в работната област и се създава негова инстанция, като се задействат изчислителните процедури, свързани с размиване на входните променливи и получаването на извод от всяко правило.

След изпълнение на всички правила следва агрегация и деразмиване.

Интерфейсът реализира връзката между базата от данни и размитата машина за изводи.

3.2.2. FLQM модел за оценка на финансови активи

3.2.2.1. Библиотека *Функции на принадлежност*

При реализацията на модела FLQM е предвидено да има възможност за използване на различни функции за принадлежност и различни методи за агрегация и деразмиване, т.е. е създадена като отделен модул библиотека *Функции на принадлежност*. Включените в тази библиотеката функции в таблица 3.1.

<i>Функция на принадлежност</i>	<i>Аналитично представяне</i>
Размит сингелтон	$\mu(x) = \begin{cases} 1, & x = a \\ 0, & x \neq a \end{cases}$
Г – функция	$\Gamma(x, a, b) = \begin{cases} 0, & x < a \\ \frac{x-a}{b-a}, & a \leq x \leq b \\ 1, & x > b \end{cases}$
S – функция	$S(x, a, b, c) = \begin{cases} 0, & x \leq a \\ 2 \cdot \left(\frac{x-a}{c-a} \right)^2, & a < x \leq b \\ 1 - 2 \cdot \left(\frac{x-a}{c-a} \right)^2, & b < x \leq c \\ 1, & x > c \end{cases}$
L – функция	$L(x, a, b) = \begin{cases} 1, & x < a \\ \frac{b-x}{b-a}, & a \leq x \leq b \\ 0, & x > b \end{cases}$

<p>Λ – функция (триъгълна)</p>	$\Lambda(x, a, b, c) = \begin{cases} 0, & x < a, x > c \\ \frac{x-a}{b-a}, & a \leq x < b \\ \frac{c-x}{c-b}, & b \leq x \leq c \end{cases}$
<p>π – функция</p>	$\pi(x, b, c) = \begin{cases} S(x, c-b, c-\frac{b}{2}, c), & x \leq c \\ 1-S(x, c, c+\frac{b}{2}, c+b), & x > c \end{cases}$
<p>Π – функция (трапецовидна)</p>	$\Pi(x, a, b, c, d) = \begin{cases} 0, & x < a, x > d \\ \frac{x-a}{b-a}, & a \leq x < b \\ 1, & b \leq x < c \\ \frac{d-x}{d-c}, & c \leq x \leq d \end{cases}$
<p>Гаусова функция</p>	$gaussian(x, \beta, \alpha) = e^{-\frac{1}{2} \left(\frac{x-\alpha}{\beta} \right)^2}$
<p>Бел функция</p>	$bell(x, \alpha, \beta, \gamma) = \frac{1}{1 + \left \frac{x-\gamma}{\alpha} \right ^{2\beta}}$
<p>Сигмоидална функция</p>	$sig(x, \alpha, \beta) = \frac{1}{1 + e^{-\alpha \cdot (x-\beta)}}$

Таблица 3.1. Функции на принадлежност

3.3.2.2. Лингвистични променливи

Получените от базата от данни стойности на входните променливи за всеки актив се развиват, т.е. се пресмята степента на принадлежност към всеки от термите на съответните входни лингвистични променливи.

В модела FLQM има три входни променливи: възвращаемост, риск и q -ratio и една изходна променлива - Q -measure.

Съответните лингвистичните променливи са:

$$X_1 \triangleq \text{return}$$

$$X_2 \triangleq \text{Risk}$$

$$X_3 \triangleq q\text{-ratio}$$

$$Y \triangleq Q\text{-measure.}$$

Дефиниционните множества на четирите лингвистични променливи съвпадат с множеството от реални числа, т.е. $U_{X_1} = U_{X_2} = U_{X_3} = U_Y = R$.

Терм-множествата на тези лингвистични променливи са съответно $T(X_1) = \{X_{1j}\}$, $T(X_2) = \{X_{2j}\}$, $T(X_3) = \{X_{3k}\}$, $T(Y) = \{Y_j\}$ за $j = 1, \dots, 5$; $k = 1, 2, 3$ и

$$X_{ij} \triangleq \begin{pmatrix} \text{Very low} & i = 1, 2 & j = 1 \\ \text{Low} & i = 1, 2 & j = 2 \\ \text{Neutral} & i = 1, 2 & j = 3 \\ \text{High} & i = 1, 2 & j = 4 \\ \text{Very high} & i = 1, 2 & j = 5 \\ \text{Small} & i = 3 & j = 1 \\ \text{Neutral} & i = 3 & j = 2 \\ \text{Big} & i = 3 & j = 3 \end{pmatrix}; \quad Y_j \triangleq \begin{pmatrix} \text{Bad} & j = 1 \\ \text{Not bad} & j = 2 \\ \text{Neutral} & j = 3 \\ \text{Good} & j = 4 \\ \text{Very good} & j = 5 \end{pmatrix}$$

За дефиниране на функциите на принадлежност (MF) в FLQM модела се използват следните три вида функции:

$$\mu_G(x) = \text{gaussian}(x, \beta, \alpha);$$

$$\mu_B(x) = \text{bell}(x, \alpha, \beta, \gamma);$$

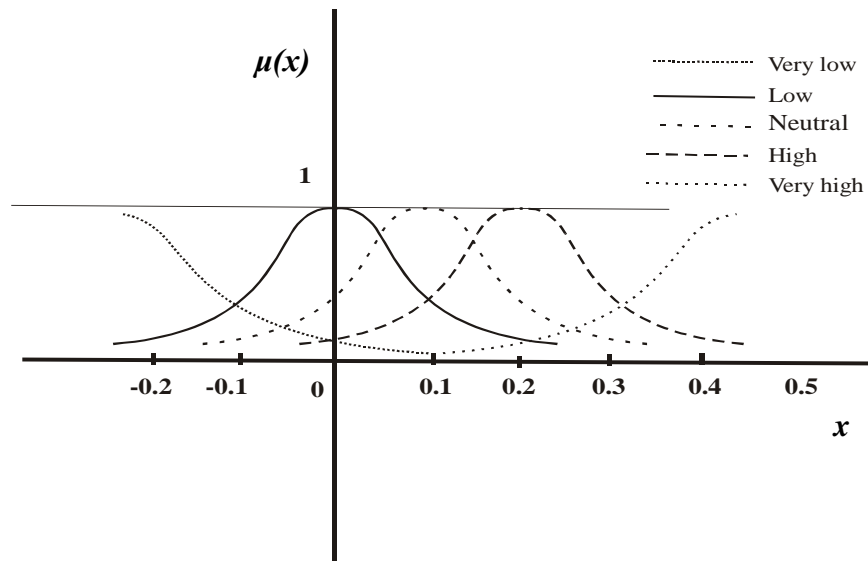
$$\mu_S(x) = \text{sig}(x, \alpha, \beta).$$

Съответните стойности на параметрите на тези функции за съответните терми са показани в таблица 3.2.

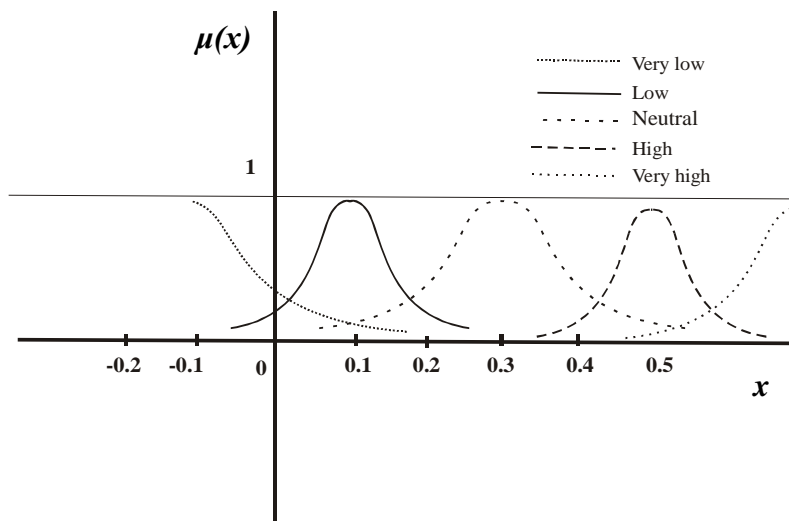
Терми	X_{11}	X_{12}	X_{13}	X_{14}	X_{15}	X_{21}	X_{22}	X_{23}	X_{24}	X_{25}
MF	$\mu_S(x)$	$\mu_G(x)$	$\mu_G(x)$	$\mu_G(x)$	$\mu_S(x)$	$\mu_S(x)$	$\mu_G(x)$	$\mu_G(x)$	$\mu_G(x)$	$\mu_S(x)$
α	-20	0.05	0.08	0.05	20	-2	0.05	0.07	0.05	2
β	0	0	1.1	1.2	1.3	0	0.1	0.3	0.5	0.7
Терми	X_{31}	X_{32}	X_{33}			Y_1	Y_2	Y_3	Y_4	Y_5
MF	$\mu_S(x)$	$\mu_B(x)$	$\mu_G(x)$			$\mu_G(x)$	$\mu_G(x)$	$\mu_G(x)$	$\mu_G(x)$	$\mu_G(x)$
α	-0.3	20	0.3			0.1	0.1	0.1	0.1	0.1
β	20	4	60			0	0.25	0.5	0.75	1
γ		40								

Таблица 3.2. Вид и параметри на функциите на принадлежност на термите на лингвистичните променливи

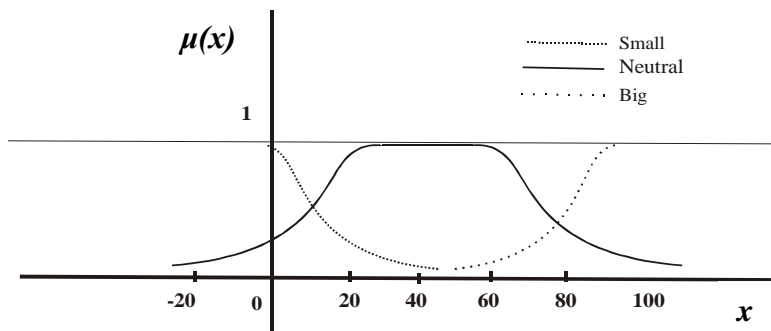
Така дефинираните лингвистични променливи са изобразени съответно на фигури 3.3, 3.4, 3.5 и 3.6.



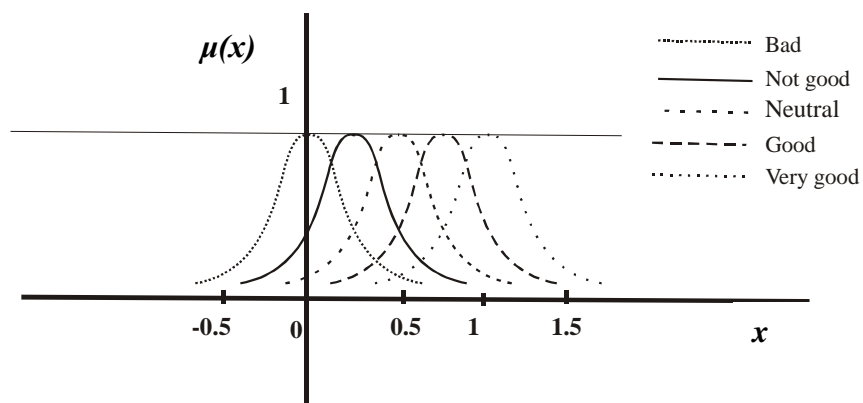
Фигура 3.3. Входна лингвистична променлива възвращаемост : $X_1 \triangleq return$



Фигура 3.4. Входна лингвистична променлива риск: $X_2 \triangleq Risk$



Фигура 3.5. Входна лингвистична променлива коефициент q – ratio: $X_3 \triangleq q\text{-ratio}$



Фигура 3.6. Изходна лингвистична променлива Q -measure: $Y \triangleq Q\text{-measure}$

3.3.2.3. Размити правила

В модела FLQM е използвана размита система за изводи от тип Мамдани (MFIS). Като резултат от действието на такава система се получава размита изходна променлива и това е основната причина, поради която MFIS са широко използвани в приложенията за подпомагане вземане на решения [4], [14].

Размитите правила моделират процеса за вземане на решения и в случая имат вида:

$$\text{АКО } (r^* \text{ е } X_{1i}) \text{ И } (s^* \text{ е } X_{2j}) \text{ И } (q^* \text{ е } X_{3k}) \text{ ТО } (Q - \text{measure} \text{ е } Y_p)$$

за $i=1, \dots, 5; j=1, \dots, 5; k=1, \dots, 3$ и $p=1, \dots, 5$.

Тъй като моделът FLQM има три входни променливи със съответно 5, 5 и 3 терма, то всички възможни правила са 75. От тях са избрани 24 (табл. 3.3), при което е следвано експертно мнение. Въпреки че тези правила адекватно описват най-важните ситуации, които биха могли да възникнат в процеса на вземането на инвестиционно решение, списъкът на размитите правила може да бъде удължен, без да се налага промяна на архитектурата на модела.

№	$X_1 \triangleq \text{return}$	$X_2 \triangleq \text{Risk}$	$X_3 \triangleq \text{q-ratio}$	$Y \triangleq \text{Q-measure}$	Тегло w
1	Very high	Very low	Big	Very good	1
2	Very high	Low	Big	Very good	1
3	High	Very low	Big	Very good	1
4	High	Low	Big	Very good	1
5	Very high	Very low	Big	Good	0,8
6	Very high	Neutral	Big	Good	0,8
7	High	Low	Big	Good	0,8
8	High	Neutral	Big	Good	0,8
9	Very high	Very low	Neutral	Good	0,8
10	Very high	Low	Neutral	Good	0,8
11	Neutral	Neutral	Neutral	Neutral	0,7
12	Neutral	Low	Neutral	Neutral	0,7
13	Low	Neutral	Neutral	Neutral	0,7
14	Neutral	Neutral	Small	Neutral	0,7
15	Very low	Very high	Small	Bad	1
16	Very low	High	Small	Bad	1
17	Low	Very high	Small	Bad	1
18	Low	High	Small	Bad	1
19	Very low	Very high	Small	Not bad	0,8
20	Very low	Very high	Small	Not bad	0,8
21	Low	High	Small	Not bad	0,8

22	<i>Low</i>	<i>Neutral</i>	<i>Small</i>	<i>Not bad</i>	0,8
23	<i>Very low</i>	<i>Very high</i>	<i>Neutral</i>	<i>Not bad</i>	0,8
24	<i>Very low</i>	<i>High</i>	<i>Neutral</i>	<i>Not bad</i>	0,8

Таблица 3.3. Размити правила за вземане на решения.

3.3.2.4. Агрегация и деразмиване

При агрегацията на правилата на всяка наредена тройка от стойности на входните променливи (r^*, s^*, q^*) се съпоставя степен на принадлежност към съответния терм на изходната променлива:

$$\theta^* = \min(\mu_{1i}(r^*), \mu_{2j}(s^*), \mu_{3k}(q^*))$$

и съответно претеглените стойности

$$\theta^{**} = w \cdot \theta^*.$$

Агрегирането на седмото правило е илюстрирано на фигура 3.7. В случая r^* е годишната навъзвращаемост и е равна на 1,128, а съответната степен на принадлежност към терма **High** на размитата променлива **Return** е $\mu_{14}(r^*) = 0,8125$. Стойността на s^* е 0,12 със степен на принадлежност $\mu_{22}(s^*) = 0,79$ към терма **Low** на размитата променлива **Risk**. Коефициентът q^* има стойност 93 със степен на принадлежност $\mu_{33}(q^*) = 0,57$ към терма **Big** на размитата променлива **q-ratio**.

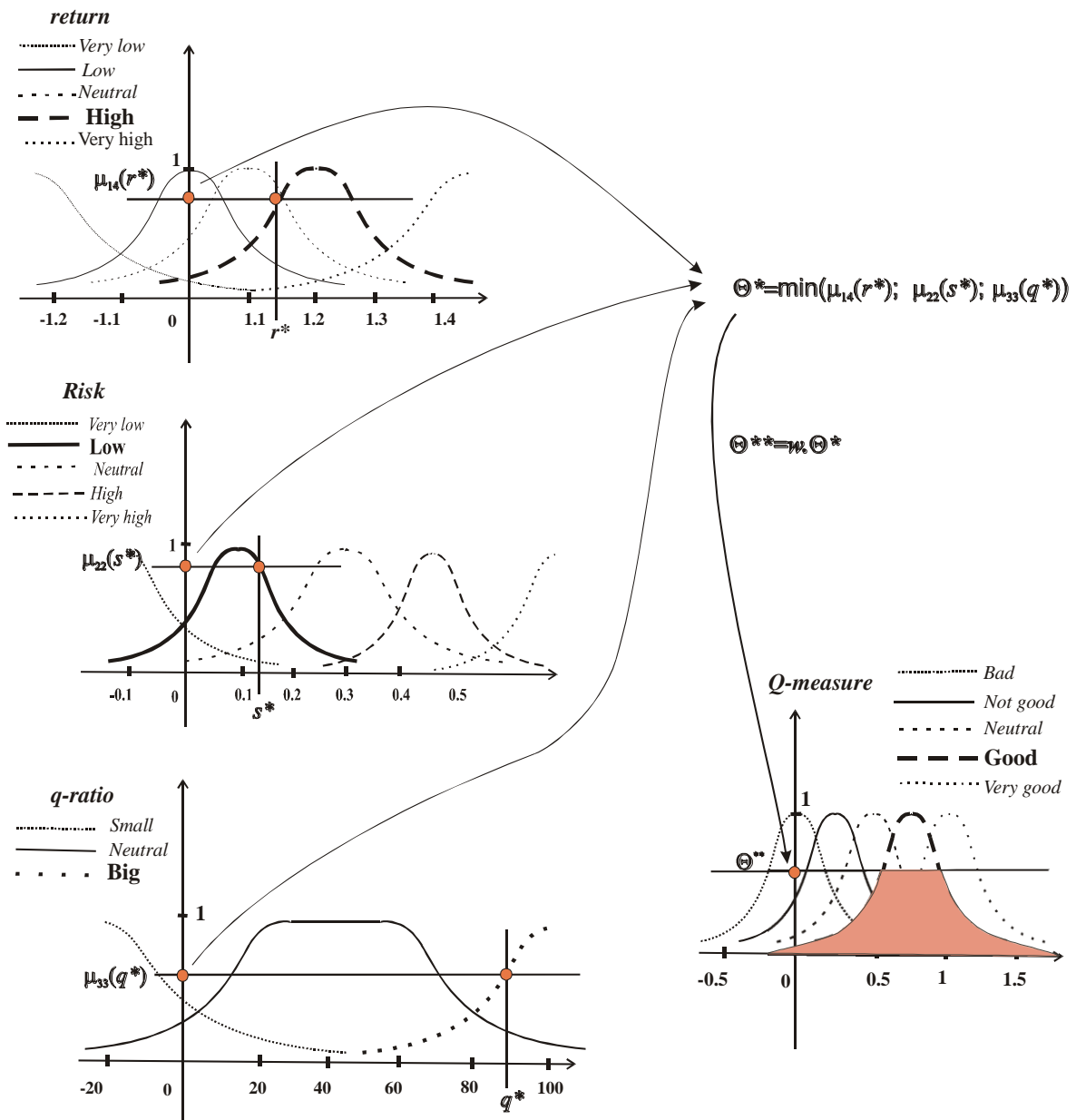
Този слувай

$$\theta^* = \min(0,8125; 0,79; 0,57) = 0,57$$

и следователно

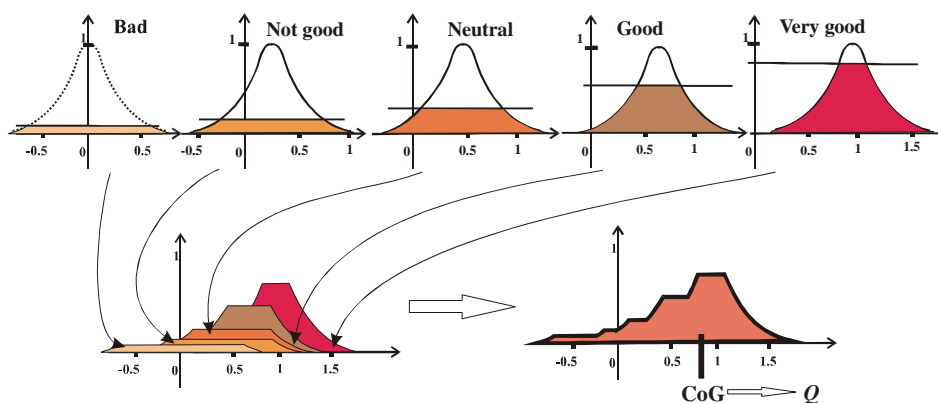
$$\theta^{**} = 0,8 \cdot 0,57 = 0,456,$$

което е и степента на принадлежност към терма **Good** на изходната размита променлива **Q-measure**.



Фигура 3.7. Правило № 7: АКО (*return* е *High*) И (*Risk* е *Low*) И (*q-ratio* е *Big*) ТО (*Q-measure* е *Good*)

След изпълнението на всяко правило са получени няколко степени на принадлежност за всеки от термите на изходната променлива *Q-measure*. При агрегацията от тези стойности се получава изходната величина (фиг. 3.8).



Фигура 3.8. Агрегация и деразвиване за получаване на стойността на Q на финансов актив

Като метод на деразвиване е избран методът на центъра на тежестта, при който се пресмята величината:

$$C_oG(Q) = \frac{\int_{-\infty}^{+\infty} x \cdot Q(x) \cdot dx}{\int_{-\infty}^{+\infty} Q(x) \cdot dx}$$

Тъй като функциите на принадлежност в модела FLQM са гладки, то при програмното реализиране на модела за числено пресмятане на интегралите е използван методът на правоъгълниците.

Пресметнатата точна стойност на Q за всеки актив се записва в базата от данни и след това може да бъде използвана както за оценка на индивидуален актив така и за конструиране на инвестиционни портфейли.

3.3. Управление на финансов актив

Получените точни стойности за Q на активите могат да бъдат използвани от инвеститора в процеса на вземане на решения.

В икономическата наука е прието да се смята, че винаги може да се замени възвращаемост за риск, т.е. по-висока възвращаемост винаги носи по-голям риск и обратно. Емпиричните тестове, проведени с данни от Българска фондова борса потвърждават това схващане. [34,35]

Но за инвеститор, който не извършва спекулативни сделки е важно да проследи не само моментните възвращаемост и риск, а и да успее да прогнозира до каква степен тези две характеристики (възвращаемост и риск) ще бъдат стабилни във времето. Моделът FLMQ за оценка на финансови активи използва и една допълнителна характеристика – отношението на възвращаемостта и риска, което отразява до каква степен поетият риск ще бъде оправдан с по-висока възвращаемост.

Величината Q на актива е комбинация от три величини: възвращаемост, риск и отношението им. След провеждането на разнообразни емпирични тестове с реални данни на модела FLQM в различни интервали от време и за различни активи на БФБ, може да се твърди, че Q е показател за качеството на актива [36], [37], [75], а именно:

- при Q по-малка от 0,4 (независимо какви са възвращаемостта и риска) в краткосрочен (до около 3 месеца) хоризонт настъпва драматично намаление в цената на актива;
- при Q между 0,4 и 0,6 цената на актива не се променя съществено за същия времеви хоризонт и дори да се е повишила или намалила, то цената на транзакциите за купуване или продаване на този актив би надвишила евентуалната полза от извършването на съответното действие;
- при Q по-голяма от 0,6 активът трайно повишава цената си След провеждането на разнообразни тестове на модела FLQM, и е подходящ за купуване.

Тези изводи със съответните инвестиционни решения/действия са показани в таблица 3.4.

Q	Качество	Инвестиционно действие
$Q < 0,4$	лошо	Продава
$0,4 \leq Q \leq 0,6$	неутрално	Задържа
$Q > 0,6$	добро	Купува

Таблица 3.4. Инвестиционно действие според Q на актива

Глава 4. Софтуерна система за управление на финансови активи FSSAM

4.1. Обща схема на FSSAM

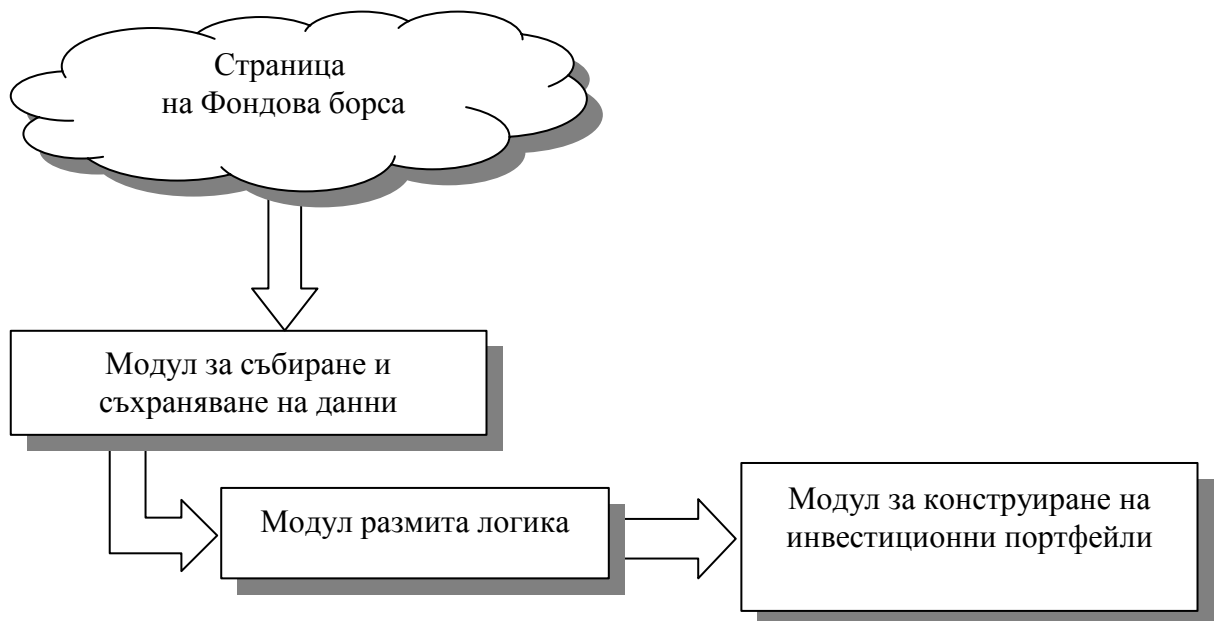
За реализиране на процедурите за събиране и съхраняване на данните, за оценяване на активите и за конструиране на инвестиционните портфейли е подходящо да се създаде цялостна софтуерна система, която да работи самостоятелно.

При създаването на софтуерната ситема е важно предварително да се зложат основни характеристики, определящи нейното качество: коректност, надеждност, стабилност, мащабируемост. [13]

На фигура 4.1 е представен концептуален модел на софтуерната система за управление на финансови активи FSSAM.

Софтуерната система приложение се състои от три модула:

- модул за събиране и съхраняване на данните;
- модул размита логика;
- модул за конструиране на инвестиционни портфейли.



Фигура 4.1. Схема на софтуерна система за управление на финансови активи FSSAM

Модулът за събиране и съхраняване на данни има следните функционалности:

- отправят се заявки към уеб сървъра на фондовата борса;
- извършва се извличане на данните от свалените страници от фондовата борса;

- данните се записват в базата от данни,
- попълват се липсващите данни.

В модула размита логика:

- за всеки актив от базата от данни се извличат неговият код и трите характеристики – възвращаемост, риск и *q-ratio* и се прилага моделът FLQM, като се зктивира ма'ината за изводи, описана в предната глава, а именно:
 - пресмятат се съответните степени на принадлежност за характеристиките към размитите променливи;
 - прилагат се размитите правила за получаване на размита променлива *Q-measure* ;
 - след деразмиване за всеки актив се получават съответните стойности на *Q*, които се добавят в базата от данни.

В модула за конструиране на инвестиционни портфейли:

- потребителят въвежда две стойности:
 - инвестиционна сума *S*;
 - максимален брой активи (*K*), от които да бъдат съставени инвестиционните портфейли;
- за всеки актив от базата от данни се извличат неговият код, характеристики и стойност на *Q*;
- активите се сортират в намаляващ ред според *Q*;
- от така получения списък се генерират комбинации от първите *k* актива за всяко $k \leq K$;
- за всяка от комбинациите се изчислява относителният дял в съответния портфейл за всеки актив в зависимост от *S*;
- получените портфейли се записват в базата от данни;
- извършва се процедура по алокация, ако е необходимо и новополучените портфейли се дописват в базата от данни;
- за всички получени до този момент портфейли се пресмятат характеристиките възвращаемост, риск и *q-ratio*;
- в модула размита логика за всеки портфейл се приприлага моделът FLQM за получаване на *Q* на тези портфейли;
- портфейлите, заедно с техните характеристики, съставлящите ги активи със съответните характеристики и дялове се извеждат на екран.

Автоматичното стартиране на софтуерната система може да бъде постигнато по няколко начина, например чрез:

- Windows service (сервиз);
- Windows Task Scheduler;
- Или с други средства.

Тъй като сервизът стои във фонов режим, то той заема ресурси през цялото време на работа. Сервизите се използват в случаи на предоставяне на услуга на много приложения и изискват някои разширения.

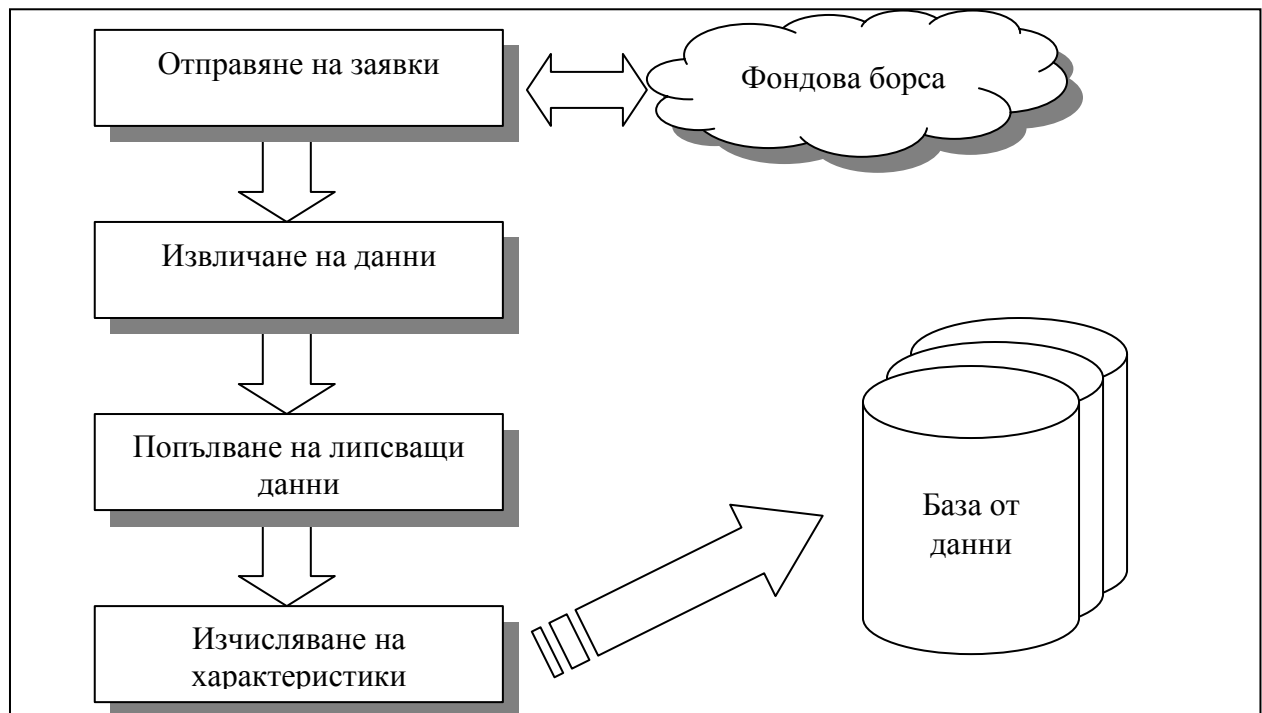
Windows Task Scheduler не изисква допълнителен ресурс и допълнителни разширения, ето защо е по-подходящ при ежедневното стартиране на софтуерната система за управление на финансови активи FSSAM.

4.2. Събиране и съхраняване на данни (ССД)

С оглед програмната обработка на данните в реално време е подходящо процесът на събиране, съхраняване и обработка на данните да бъде реализиран като отделен модул. Концептуалният модел на *Модул за Събиране и съхраняване на данни* се състои от елементи, в които се извършват процедурите: отправяне на заявки, извличане на данни, попълване на липсващи данни, изчисляване на характеристики и запис в база данни (фиг. 4.2).

Първо се отправя заявка към уеб сървър на съответния източник на борсова информация. Тя съдържа методи за извличане на *html* кода на страницата, където е показана информация за финансовите активи. Извлеченият *html* код се подава на частта за извличане на данни, съдържаща методи за неговата обработка. Тези методи извличат необходимите данни за финансовите активи и при нужда се попълват липсващите данни.

След това по алгоритъм се изчисляват характеристиките на всеки актив. Този алгоритъм е реализиран в частта за извършване на изчисления. Данните и резултатите от изчисленията по дадения алгоритъм се съхраняват в база от данни.



Фигура 4.2. Схема на модул за събиране и съхраняване на данни

4.2.1. Отправляне на заявки

Преди проектирането и реализирането на модула за отправяне на заявки към източника на борсова информация е необходимо да се установи, дали в този източник се предоставя XML документ или HTML код и какви методи за отправяне на заявки поддържа даденият уеб сървър – *get*, *post* или друг метод.

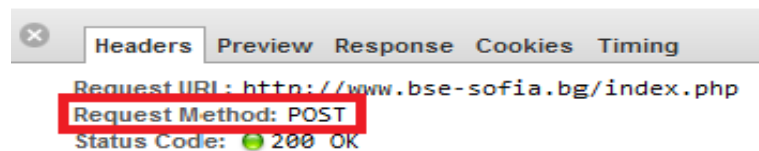
Например, сайтът на Българската фондова борса не предоставя XML документ с информация за активите, които се предлагат. Преглед на кода на страницата показва, че данните, които са нужни се предоставят във вид на таблица (html tag <td>...</td>). Част от кода на страницата, с данни за активите, на Българската фондова борса е показан по-долу.

```
<tr class="title">
  <td align="left" style="width: 70px;">BSE code</td>
  <td align="left" style="width: 420px;">Issuer</td>
  <td align="left" style="width: 85px;">Volume (lots)</td>
  <td align="left">Previous close</td>
  <td align="left">High</td>
  <td align="left">Low</td>
  <td align="left">Average</td>
  <td align="left">Last</td>
  <td align="left">Change (%)</td>
</tr>
<tr>
  <td align="left">
    <a href="/?page=QuotesInfo&code=3JR">3JR</a>
  </td>
  <td align="left">Sopharma AD-Sofia</td>
  <td align="right">20 027</td>
  <td align="right">2.389</td>
  <td align="right">2.400</td>
  <td align="right">2.380</td>
  <td align="right">2.385</td>
  <td align="right">2.380</td>
  <td align="right">-0.38%</td>
</tr>
```

Всяка таблица с данни изглежда по следния начин:

BSE code	Issuer	Volume (lots)	Previous close	High	Low	Average	Last	Change (%)
3JR	Sopharma AD-Sofia	20 027	2.389	2.400	2.380	2.385	2.380	-0.38%
4CF	CB Central Cooperative Bank AD-Sofia	6 950	0.668	0.668	0.663	0.665	0.663	-0.75%
4EH	Eurohold Bulgaria AD-Sofia	7 195	1.049	1.067	1.001	1.055	1.067	1.72%
5F4	CB First Investment Bank AD-Sofia	250	1.789	1.789	1.789	1.789	1.789	0.00%
5MB	Monbat AD-Sofia	378 601	5.140	5.180	4.950	4.952	5.050	-1.75%
6C4	Chimimport AD-Sofia	29 227	1.168	1.179	1.104	1.120	1.120	-4.11%
E4A	Enemona AD-Kozloduy	100	3.020	3.020	3.020	3.020	3.020	0.00%

До уеб сървъра на Българската фондова борса могат да бъдат изпратени заявки посредством *post* и *get* методи:



Следователно, при реализиране на *Отправляне на заявки* е необходимо предварително проучване на източника на информация. При работа с информация от БФБ са реализирани и двата метода за отправяне на заявки към сървъра на фондовата борса. След като кодът е извлечен, той се подава на *Парсване* за последваща обработка.

4.2.2. Извличане на данни

След осъществяване на заявката към уеб сървъра и сваляне на кода на страницата, обработката на кода се извършва с помощта на регулярен израз, при чието конструиране се взема под внимание точно кои данни представляват интерес, ще съществува ли проблем с липсващи данни и начина за съхраняване на данните.

4.2.3. Попълване на липсващи данни

Алгоритъмът за изчисляване на характеристиките на активите не допуска наличие на липсващи данни за активите. Но липсващи данни се появяват по различни причини. Тъй като фондовите борси работят само през делнични дни, то през почивните дни няма информация за никой от активите и котировките при затваряне се запазват до следващия работен ден. От друга страна има работни дни, в които даден актив не е търгуван и в този случай за него също липсват данни, но причината е съвсем различна – търсенето и предлагането не са стигнали до равновесие. Съществува и друга възможност за липсващи данни – заличаване на актива от борсовата търговия за някакъв период от време.

В този модел проблемът с липсващите данни се решава като редицата с цени се допълва, като се “допише” последната котировка съответния брой пъти, което е „подказано” от практиката на самите борси за почивните дни. При използване на средногеометрична като оценка за възвращаемостта, това “дописване” не променя формулата за възвращаемостта. Наистина, ако се добавят липсващите за съответните дни наблюдения чрез повтаряне на последната котировка, се получава подредицата $P(t-1), \underbrace{P(t), \dots, P(t)}_{\Delta_t \text{ на брой}}, P(t+1)$ и при след елементарни преобразования се вижда, че формула (2.9) за логаритмичната средногеометрична възвращаемост е отново в сила.

4.2.4. Изчисляване на характеристики на активи

В модела за оценка на възвращаемостта се използва годишната възвращаемост, получена по формула (2.12), рискът се оценява чрез средноквадратичното отклонение по формула (2.14).

Като допълнителна възможност е предвидено инвеститорът да зададе граница на желана годишна възвращаемост r_0 и допустима граница на риска s_0 .

Така използваната в модела стойност на възвращаемостта е:

$$r^\circ = AR - r_0,$$

стойността на риска е:

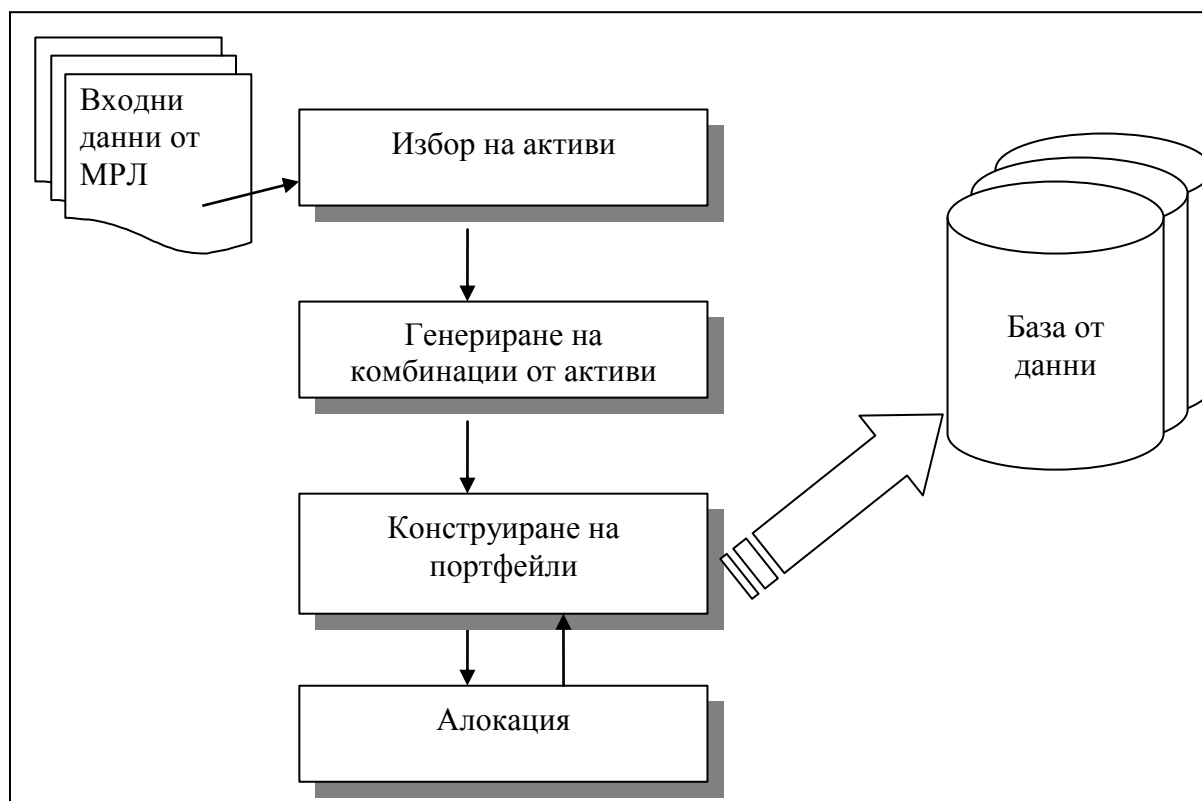
$$\sigma^\circ = s_g - s_0,$$

и коефициентът $q - ratio$ е равен на:

$$q = \frac{r^\circ}{\sigma^\circ}.$$

Данните и получените резултати за характеристиките на всички активи се съхраняват в удобен формат в релационна база от данни.

4.3. Конструирание на инвестиционни портфейли с използване на Q -measure



Фигура 4.3. Схема на модул за конструирание на инвестиционни портфейли.

На фигура 4.3 е показан модулът, в който се конструират инвестиционните портфейли.

Входните данни се извличат от базата от данни. Те представляват характеристиките на финансовите активи, след като е приключила работата на модул Размита логика. Избор на активи селектира определен брой активи спрямо даден критерий, определен от инвеститора. От избраните активи се генерират комбинации от тези активи. След това се конструират портфейли от генерираните комбинации от активи. Тъй като е възможно изборът за инвестицията капитал да не бъде изразходен изцяло, се прилага процедура по алокация. Тази процедура модифицира портфейла с цел изчерпване на по-голямо количество от дадения капитал. Конструираните портфейли се съхраняват в база от данни и инвеститорът може да вземе своето решение.

Нека $A = \{A_1, A_2, A_3, \dots, A_n\}$ е множество от финансови активи. За всеки актив е изчислена Q въз основата на размита правила и размита изходна променлива, дефинирани в модула *размита логика*.

Нека K е максималният брой финансови активи, определени да участват в инвестиционен портфейл.

Целта е конструирание на инвестиционен портфейл (подмножество на A), който притежава възможно най-голяма Q .

Първо елементите на множеството от финансови активи се сортират спрямо тяхната Q . От сортираните активи се избират първите K на брой, след което се генерират комбинации без повторение. Първоначално се генерират комбинациите от 1-ви клас от K елемента, след което се генерират комбинациите от 2-ри клас от K елемента и така до K елемента K -ти клас. Така се изчерпват всички подмножества от избраните активи. Броят на тези подмножества е $2^K - 1$, тъй като не се включва празното множество, т.е. не съществува инвестиционен портфейл без активи. Всяка от генерираните комбинации представлява инвестиционен портфейл.

За всеки актив, участващ в инвестиционен портфейл, се пресмята относителния дял на актива в портфейла. Този относителен дял се означава с x_j и се пресмята по следната формула:

$$x_j = \frac{Q_j}{\sum_{j=1}^N Q_j},$$

където N е броят на финансовите активи в инвестиционния портфейл, а Q_j е стойността на Q за j -я актив.

За всеки от конструираните инвестиционни портфейли се изчисляват характеристиките риск и възвращаемост. Портфейлната възвращаемост R_p се изчислява по следната формула:

$$R_p = \sum_{j=1}^N x_j r_j,$$

където x_j е относителният дял на j – я актив в портфейла, а r_j е възвращаемостта на j – я актив.

Инвестиционният риск на портфейла се оценява като средно претеглено:

$$\sigma_p = \sum_{j=1}^N x_j \sigma_j$$

където x_j е относителният дял на j – я актив в портфейла, а σ_j е рискът на j – я актив.

За определяне на Q на инвестиционния портфейл отново се използва размитата система за вземане на решение. Входните стойности на системата са рискът и възвращаемостта на портфейла, както и тяхното отношение q_p , като:

$$q_p = \frac{R_p}{\sigma_p}.$$

4.4. Алгоритъм за максимално използване на капитала

Броят акции v_j , които трябва да се закупят от актива A_j се изчислява по формулата:

$$v_j = \left\lfloor \frac{S}{x_j P_j} \right\rfloor,$$

където x_j е относителният дял на актива в инвестиционния портфейл, P_j е цената на една акция от този актив, S е първоначалната сума за инвестиране, а със символа $\lfloor u \rfloor$ се означава най-голямото цяло число, по-малко от u .

Използваният на този етап капитал S_u се пресмята по следната формула:

$$S_u = \sum_{j=1}^N v_j P_j$$

В случай че останалият капитал $S - S_u$ е по-голям от предварително определен процент (толеранс) от първоначалния капитал, се прилага процедура по допълнителна алокация. Целта на тази процедура е закупуването на допълнително количество акции за максимално изчерпване на капитала. Първо се сравнява останалият капитал $S - S_u$ с цената на актива с най-голяма стойност на Q и ако тази цена е по-голяма от остатъка, се купуват максималният възможен брой акции от този актив. В обратния случай сравнението се извършва за следващия по ред актив в списъка. Този процес се повтаря, докато останалият капитал е достатъчно малък и не могат да се закупят допълнителни акции.

На всяка стъпка броят на допълнително закупените акции v_{ja} за даден актив се изчислява по формулата:

$$v_{ja} = \left\lfloor \frac{x_j(S - S_u)}{P_j} \right\rfloor,$$

където x_j е дялът на актива в инвестиционния портфейл, преди прилагане на процедурата по алокация, S е началният капитал, S_u е изразходеният до момента капитал, а P_j е цената на една акция от j -я актив.

4.5. Реализация на моделите

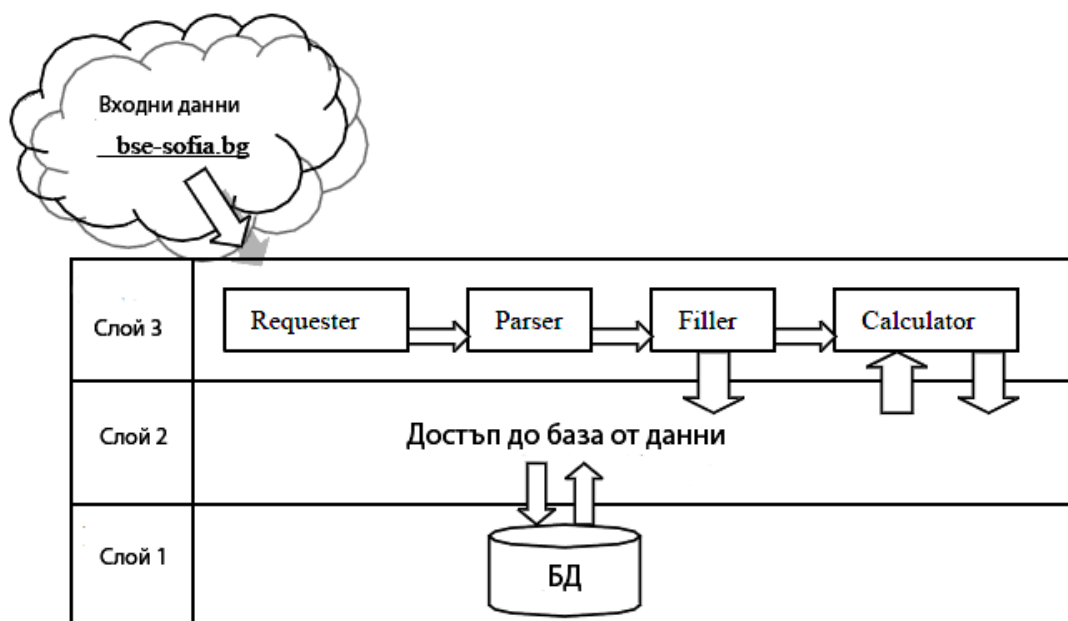
При реализацията на моделите е използвана платформата .NET Framework и езикът за програмиране C#, който е специално проектиран за тази платформа.

Софтуерната система съдържа 2217 реда първичен код.

Различните елементи на системата са реализирани и като отделни модули на C++ и в програмната среда MatLab, като тези допълнителни сорсове са използвани при тестовите за коректност на работа на софтуерната система за управление на финансови активи FSSAM.

4.5.1. Обработване на данните

Приложението, с което се събират, съхраняват и обработват първоначално данните е изградено от три слоя (фиг. 4.4.). Първият слой е база от данни, вторият слой съдържа методи за работа с базата от данни и третият слой представлява бизнес логика. [8]



Фигура 4.4. Трислоен модел на модул Събиране и съхранение на данни.

4.5.2. База от данни

Базата от данни е релационна и съдържа 4 таблици: *Equity*, *EquityData*, *equityportfolio* и *portfolio* (фиг. 4.5).

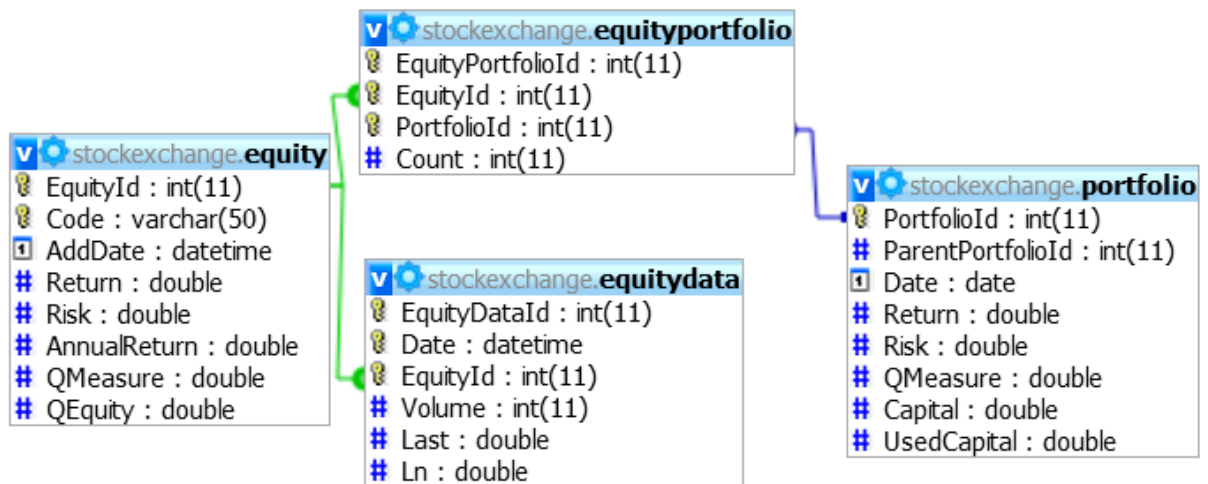
Таблица *equity*

Атрибутът *EquityId* е от тип *int* и е уникален идентификатор за всеки актив. *EquityId* е първичен ключ на таблицата. Атрибутът *Code* е от тип *varchar* и в него се съхранява кодът на актива. *Code* е уникален, т.е не може да има два еднакви кода в таблицата. Атрибутът *AddDate* е от тип *datetime* в него се съхранява датата, на която е добавен съответният актив. Атрибутът *Return* е от тип *double* и съхранява възвращаемостта на дадения актив. Атрибутът *Risk* е от тип *double* и съхранява риска за дадения актив. Атрибутът *AnnualReturn* е от тип *double* и съхранява годишната норма на възвращаемост. Атрибутът *QEQuity* е от тип *double* и съхранява Q-мярката за дадения актив. Атрибутът *QMeasure* е от тип *double* съхранява отношението възвращаемост-

риск за дадения актив. Последните пет атрибута се записват в базата от данни след пресмятането им.

Таблица *equitydata*

Атрибутът *EquityDataId* е от тип *int* и е първичен ключ за таблицата. Атрибутът *Date* е от тип *datetime* и в него се съхранява дата. Атрибутът *EquityId* е от тип *int* и е външен ключ за таблицата. Атрибутът *Volume* е от тип *int* в него се пази количеството търгувани активи. Атрибутът *EquityId* е от тип *double* и в него се пази последната цена след затваряне на фондовата борса за дадения актив. Атрибутът *Ln* е от тип *double* и в него се съхранява изчислената стойност за логаритъма за актива. Двойката атрибути *EquityDataId* и *Date* са уникални за таблицата, т.е за една и съща дата не може да има повече от един запис за дадения актив.



Фигура 4.5. Таблицы на базата от данни.

Таблица *equityportfolio*

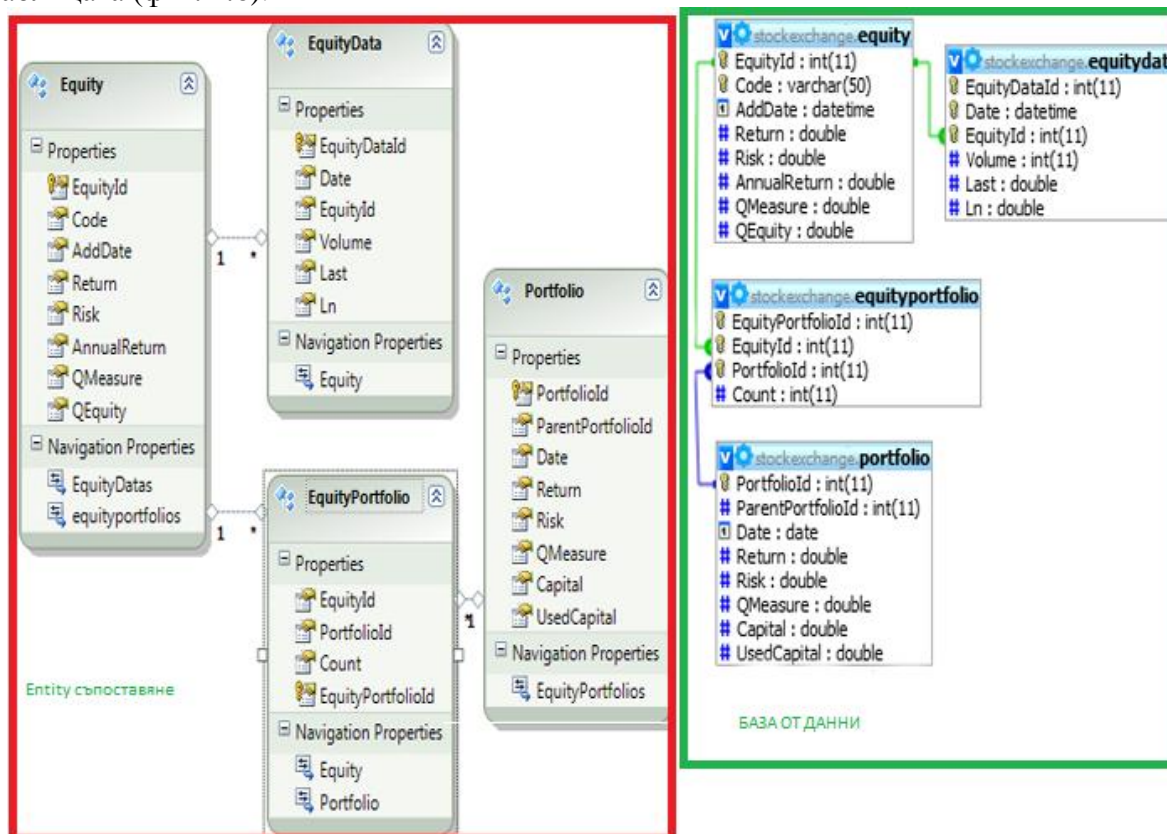
Атрибутът *EquityPortfolioId* е от тип *int* и е първичен ключ в таблицата. Атрибутът *EquityId* е от тип *int* и е външен ключ за таблицата. *PortfolioId* е от тип *int*, защото е уникален идентификатор за портфейл. Атрибутът *Count* е от тип *int* и съдържа броя акции участващи в дадения портфейл.

Таблица *portfolio*

Атрибутът *PortfolioId* е от тип *int* и е първичен ключ в таблицата. Атрибутът *Date* е от тип *date* и в него се съхранява датата, на която е добавен портфейла. Атрибутът *Return* е от тип *double* и в него се съхранява възвращаемостта за дадения портфейл. Атрибутът *Risk* е от тип *double* и в него се съхранява риска за дадения портфейл. Атрибутът *QMeasure* е от тип *double* и в него се съхранява Q-мярката за дадения портфейл. Атрибутът *Capital* е от тип *double* и в него се съхранява капитала, определен за портфейла. Атрибутът *UsedCapital* е от тип *double* и в него се съхранява стойността за изразходвания капитал за даден портфейл.

4.5.3. Методи за достъп до база от данни

Методите за достъп до базата от данни са реализирани с помощта на технологията *ADO.NET Entity Framework*. Entity Framework е базиран на технологията *ORM (Object-relational mapping)* и предоставя лесен начин за работа с реляционни бази от данни. Интегриран е в *.NET* платформата 4.0. Entity Framework съпоставя таблиците от базата от данни със *C#* класове, т.е. за всяка една таблица на базата от данни Entity Framework създава клас с полета и свойства отговарящи на атрибутите от всяка една таблицата (фиг. 4.6).



Фигура 4.6. Съпоставяне на базата от данни с Entity Framework.

Релациите между таблиците се представят като свойства. Класът, който е реализиран от *Entity Framework* е с име *StockExchangeModel.Designer* и се генерира автоматично. Неговите методи се използват в класа *StockExchangeDAL*. Класовете *StockExchangeModel.Designer* и *StockExchangeDAL* реализират втория слой от приложението. [8]

4.5.4. Бизнес логика

Бизнес логиката е изградена с помощта на класовете, описани по-долу. Пълната реализация на класовете е показана в приложението.

HttpParameter

Класът *HttpParameter* има полета с модификатор за достъп *private*. Полетата имат свойства, които позволяват както промяната на тяхната стойност, така и тяхното извличане. Реализиран е и конструктор, който създава обекти от класа, използвайки подадените му параметри. Той приема два параметъра - по един за всяко от полетата на класа. Параметрите при *post* и *get* метод за заявка към сървъра се подават по следния начин:

`http://www.bse-sofia.bg/?page=SessionResults&month=6;day=3;&year=2012.`

Първият параметър *key* (пр. *page*, *month*) задава ключа в двойката ключ-стойност, а вторият *value* (пр. *SessionResults*, *6*) - стойността.

HttpPostRequest

Класът *HttpPostRequest* служи за отправяне на *post* заявки към даден сървър.

Полето *request* е обект от тип *HttpWebRequest*. Чрез методите на класа *HttpWebRequest* се дава възможност за отправяне на заявка към сървъра, като обектите от този клас съдържат *url* адреса на страницата, метода за отправяне на заявка и други. Второто поле е от тип масив от *HttpParameter*.

Конструкторът на класа приема като параметри *string* и масив от *params HttpParameter*. Ключовата дума *params* означава, че конструкторът може да приема променлив брой аргументи от тип *HttpParameter[]*.

Методът *AddParametersToUrl()* добавя параметрите към *url* адреса. Чрез параметрите се заявява конкретен ресурс.

GetResponse() е метод, който като резултат съхранява отговора, върнат от сървъра. Връщаният резултат е от тип *string*.

```
public class HttpPostRequest
```

HttpGetRequest

Класът *HttpGetRequest* служи за отправяне на *get* заявки към даден сървър. Полетата на класа са с модификатор за достъп *private*. Полето *request* е обект от тип *HttpWebRequest*. Чрез методите на класа *HttpWebRequest* се дава възможност за отправяне на заявка към сървъра. Второто поле *parameters* е масив от обекти на класа *HttpParameter*. Третото поле *url* е от тип *string*. В него се съхранява *url* адреса на уеб сървър.

Конструкторът на класа приема като параметри *string* и масив от *params HttpParameter*.

Методът *AddParametersToUrl()* добавя параметрите към *url* адреса. Чрез параметрите се заявява конкретен ресурс.

GetResponse() е метод, който като резултат съхранява отговора върнат от сървъра. Връщаният резултат е от тип *string*.

IHttpRequest

IHttpRequest представлява интерфейс. В *C#* един интерфейс дефинира какво поведение трябва да има един обект, без да указва как точно се реализира това поведение. В реализирания интерфейс методът *GetResponse()* указва типа на връщаната стойност. Неговите наследници генерират тялото и го използват за техните нужди (пр. класовете *HttpRequest*, *HttpPostRequest*).

Requester

Дефинирани са статичните полета *pageUrl*, *chartUrl* и *pageParam*. Техният модификатор за достъп е *private*. Те са статични, защото се използват само от методите на класа. Понеже методите не са асоциирани с обект, те също са статични. Полето *pageUrl* е инициализирано с *url* адреса на Българската фондова борса. Понеже *url* е просто символен низ, то полето е от тип *string*. Полето *chartUrl* е инициализирано с различен *url* адрес на фондовата борса и то също е от тип *string*. Полето *pageParam* съдържа параметър, който се добавя при отправяне на заявка към уеб сървър на Българската фондова борса.

Методът *BSEPageRequest()* отправя заявка към уеб сървър на борсата. Той ползва *get* метод при отправяне на заявката. Връщаният резултат е от тип *string*.

Методът *BSEPageRequest(DateTime date)* също отправя заявка към уеб сървър на борсата. Той ползва *post* метод при отправяне на заявката. Тази функция приема параметър от тип *DateTime*. Връщаният резултат е от тип *string*.

Методът *BSEChartRequest(string code)* приема параметър от тип *string*. Връщаният резултат е от тип *string*.

Parser

Чрез класа *Parser* се извършва претърсването на страниците. В него има реализирани два метода.

Методът *ICollection<EquityModel> BSEPageParse(string bsePageHtmlCode)*, приема като параметър низа *bsePageHtmlCode*. С помощта на регулярен израз методът претърсва подадения като параметър *bsePageHtmlCode* и търси съвпадение по шаблон и намира характеристиките на активите. Методът връща списък от обекти от тип *EquityModel*.

Вторият метод *ICollection<EquityModel> BSEChartParse(string bseChartData, string code)* приема два параметъра от тип *string*. От подадения като параметър *bseChartData*, методът извлича характеристиките за дадения актив. Типът на връщаната стойност е списък от обекти от тип *EquityModel*.

Filler

В класа *Filler* са реализирани три метода. Първият метод *CheckEquity(List<EquityModel> equityList)* приема като параметър списък от тип *EquityModel*. За всеки от активите в списъка, подаден като параметър, методът проверява дали такъв актив съществува в базата от данни. Ако не съществува, извлича данните за една година назад за дадения актив чрез метода *FillMissingEquities*. Когато методът *FillMissingEquities* е приключил, извлича данните за деня за дадения актив. След като всичко е попълнено се пресмятат логаритмите, като се използва метод на клас *Calculator*.

Методът *FillMissingEquities(List<EquityModel> equities)* приема като параметър списък от *EquityModel*. Методът попълва данните за липсващите активи от списъка, като записва и датата. Математическият модел не допуска липсващи дати.

Когато няма данни за даден актив, методът *CopyEquityDatas(List<EquityModel> equities)* копира данните за актива от предходния ден. Запазва данните в базата от данни, като използва метод на класа *StockExchangeDAL*. Той приема като параметър списък от *EquityModel*.

Calculator

В класа *Calculator* са реализирани два метода.

Методът *LnDate(List<EquityModel> equities, string code)* приема два параметъра. Първият параметър е от тип списък от *EquityModel*, вторият - от тип *string*. Методът пресмята логаритъма за всеки един актив от списъка.

Методът *CalcEvaluation()* пресмята - риск, възвращаемост, годишна норма на възвращаемост и отношението възвращаемост-риск.

EquityModel

Класът *EquityModel* представя финансов актив и неговите характеристики. В него са дефинирани следните полета с модификатор за достъп *private: code, volume, last, date, ln*. Полето *code* е от тип *string*, защото в него се съдържа кодът на актива. Кодът на актива съдържа символи и цифри. Полето *volume* е от базов тип *int*, защото в него се съдържа количеството на търгуваните активи за даден ден. Полето *last* е от тип *double*, защото в него се съдържа цената на актива след затваряне на борсата и тя не е цяло число. Полето *date* е от тип *DateTime*, в него се пази датата за дадения актив. Полето *ln* е от тип *double*, защото в него се пази резултатът от изчислението на логаритъма. Всяко от полетата в класа притежава свойства за модифициране и извличане на тези полета. Свойствата са : *Ln, Code, Volume, Last, Date*.

Класът притежава и конструктор, който създава обектите от дадения клас. Конструкторът *EquityModel()* приема стойностите, с които се инициализират полетата.

StockExchangeDAL

Класът *StockExchangeDAL* реализира програмна логика за работа с базата от данни. Той предоставя методи за извличане и съхранение на данни.

Методът *GetIdByCode(string equityCode)* приема параметър от тип *string*, а типът на връщаната стойност е от тип *int*. Методът намира индекса на актива (*id*- в базата от данни) по зададен код на актива.

Методът *CreateEquity(string equityCode)* приема параметър от тип *string*. Добавя нов запис за актив, като добавя само кода и датата, на която е добавен новият запис.

Методът *CreateEquity(List<EquityModel> equities)* приема параметър списък от *EquityModel*. Методът създава нов запис в базата от данни за всеки елемент от подадения списък, като добавя само кода на актива и датата.

Методът *CreateEquityData(List<EquityModel> equities)* приема като параметър списък от *EquityModel*. Методът добавя в таблицата *EquityData* (таблица в базата от данни) данните за всеки актив от списъка *equities*.

Методът *CreateEquityData(List<EquityModel> equities, string code)* създава запис в таблицата *EquityData*, като се добавят данните за актива от списъкът *equities*. За кой актив се добавят данните се разбира от втория параметър *code*, който е от тип *string*.

SelectYesterdayEquity(string code) е метод, който се използва в класа *Filler*. Методът за даден актив извлича данните за предходния ден. Активът, за който се извличат данните се подава като параметър от тип *string*. Методът връща обект от тип *EquityModel*.

IsEquityExisting(string code) е метод, който се използва в класа *Filler*. По зададен параметър *code* от тип *string*, методът връща булев резултат дали даден актив съществува в базата от данни или не.

CopyEquityDatas(List<string> codes) е метод, който се използва в класа *Filler*. Методът приема като параметър списък от *string*. Списъкът съдържа кодовете на

активите, които са търгувани. Методът копира данните от предходния ден за тези активи, които не са в списъка, но съществуват в базата от данни.

Методът *GetEquities()* извлича данни за активите от базата от данни. Типът на връщаната стойност е списък от *Equity*.

Методът *UpdateEquity(Equity e)* приема като параметър обект от тип *Equity*. Методът актуализира записите за активите в базата от данни.

Program

Методът *main* в класа *Program* реализира програмната логика за извличане и изчисляване характеристиките на активите и тяхното съхранение в следната последователност:

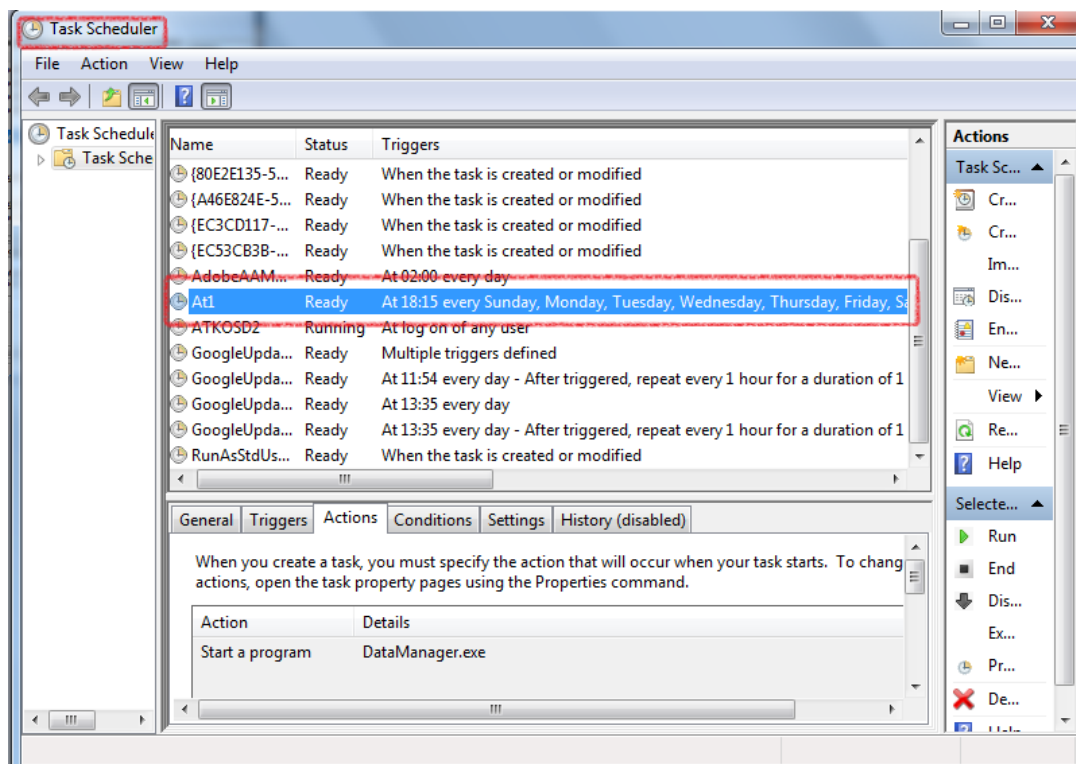
- инициализира се променлива *bsePage*, на която се присвоява резултатът от отговора на сървъра;
- инициализира се списък *equities* от тип *EquityModel*, на който се присвоява резултатът върнат от *Parser*;
- извиква се от клас *Filler*, чийто метод *CheckEquity* приема списъка *equities*, проверява дали има нови активи, ако има - взема предходните данни за тях и ги записва в базата от данни;
- списъкът се подава на метода *CopyEquityDatas* от класът *Filler*, който проверява кои активи не са в списъка (не са търгувани) и за тях копира данните от предходния ден и ги записва в базата от данни;
- методът *CalcQEquities()* от клас *Calculator* пресмята за всички активи от базата от данни - риск, възвращаемост, годишна норма на възвращаемост и отношението възвращаемост-риск и ги съхранява в базата от данни.

4.5.5. Автоматично стартиране на системата

Автоматичното стартиране на системата се постига с помощта на бач (*batch*) файл. След като веднъж е стартиран файлът, той добавя в *Task Scheduler* задачата, която трябва да се изпълнява, а това е именно да стартира приложението ежедневно след като фондовата борса е приключила своята работа. Кодът на бач файла е показан по-долу:

```
at 18:15 /every:m,t,w,th,f,s,su "DataManager.exe"  
del "start.bat"
```

Task Scheduler стартира приложението *DataManager.exe* ежедневно в 18:15 часа (фиг. 4.7)



Фигура 4.7

. Добавяне на задача за изпълнение в Windows Task Scheduler.

4.5.6. Реализация на FQLM модела

FuzzyTerm

В класа **FuzzyTerm** са дефинирани термите на размитата логика като изброени типове. Изброените типове в C# се състоят от множество именуванни константи и се дефинират със запазената дума *enum*. Вътрешно се представят с `int`, но може да се зададе и друг числов тип.

MembershipFunction

Класът *MembershipFunction* е дефиниран с идентификатор на достъп `public` и представлява абстракция на терм, използван в множеството от терми на лингвистичните променливи.

В класа се съхраняват данни в полета, отразяващи състоянието на обекта, които са нужни за работата на методите. Полетата са дефинирани с идентификатор на достъп `public` и `private`. Полетата са *Term*; *Value*; *ParamArray*; *Function*; *truthValue*.

В полето *Term* се съхранява името на дадения терм. *Value* съхранява параметъра, който се използва при изчислението на степента на принадлежност. В *ParamArray* се съдържат останалите параметри, които участват при изчисленията. Полето *truthValue* съхранява изчислената степен на принадлежност спрямо подадените параметри.

За полето *truthValue* е дефинирано свойство. Свойствата са специални елементи, които увеличават функционалността на полетата и дават допълнителна

възможност за обработката на данните преди извличането или записването им в полетата от класа. Свойствата могат да имат `get` и `set` методи за достъп.

Когато се създаде обект от даден клас се извиква метод от класа, наречен конструктор. Конструкторът не връща стойност и носи името на класа, за който е дефиниран. В класа **MembershipFunction** са дефинирани два конструктора, като всеки е с различна сигнатура (брой , подредба и тип на параметрите). Първият конструктор създава обекти от класа, използвайки подадените му параметри. Той приема четири параметъра, които записва в четири полета. Също така записва степента на принадлежност в полето *truthValue*, като извиква метода **Calculate**.

Вторият конструктор създава обекти от класа, като използва подадените му два параметъра. Разликата с първия конструктор е, че при него като параметър се подава вече изчислена степен на принадлежност.

Методът **Calculate** връща изчислената степен на принадлежност, като използва параметрите и функцията за принадлежност. Изчислената стойност е от тип `double`. Идентификаторът на достъп е `private`, тъй като се извиква единствено от първия конструктор.

Дефинирани са функциите за принадлежност спрямо формулите от таблица 1. Спрямо записаната стойност в полето *Function*, методът **Calculate** извиква съответния метод, които изчислява стойността използвайки данните записани в полетата.

LinguisticVariable

Класът **LinguisticVariable** е с идентификатор на достъп `public` и представлява абстракция на лингвистична променлива с множеството от терми.

Класът притежава две полета, които описват лингвистичната променлива. Полетата са : `Name`; `MembershipCollection`.

В полето *Name* се съхранява името на лингвистичната променлива.

MembershipCollection съхранява множеството от терми, с които се описват лингвистичните променливи.

```
public string Name { get; set; }
```

Класът **LinguisticVariable** е дефиниран с два конструктора.

Първият конструктор приема само един параметър – *Name*, името на лингвистичната променлива. Целта му е да създаде абстракция на лингвистична променлива като в последствие се добавят един по един термите към нея.

Вторият конструктор има два параметъра, като първият е отново *Name*. Следващият параметър е множеството от предварително зададени терми.

Методът **FuzzyTermTruthValue** е от тип `double` и връща стойността на принадлежност за избрания терм. Има един параметър *term* от тип *FuzzyTerm* .

FuzzyRule

Класът **FuzzyRule** е с ниво на идентификатор `public` и представлява абстракция на размитите правила дефинирани със съответните им терми.

Класът има пет полета с идентификатор на достъп `public`, които съхраняват данните за съответното размито правило.

Полетата са : `Return`; `Risk`; `QMeasure`; `Quality`; `Weight`.

Return, *Risk*, *QMeasure*, *Quality* са от тип **FuzzyTerm** и в тях се съхраняват термите съответно за Възвръщаемост, Риск, Възвръщаемост-Риск и Качество. *Weight* е от тип `double` и съдържа теглото за съответното правило.

Класът **FuzzyRule** има единствен конструктор, който приема пет параметъра съответно за петте полета.

FuzzyEquity

Класът **FuzzyEquity** е с идентификатор на достъп `public` и представлява абстракция на актива дефиниран със съответните му лингвистични променливи.

Класът има четири полета, които съхраняват данните за актива.

Полетата са : `Name`; `FuzzyReturn`; `FuzzyRisk`; `FuzzyQMeasure`.

В полето *Name* се съхранява BSE кода на дадения актив. Пример за такъв код е „4CF“, отговарящ на „Централна Кооперативна Банка“.

Полетата `FuzzyReturn`, `FuzzyRisk`, `FuzzyQMeasure` са от тип *LinguisticVariable*. Тези полета съхраняват лингвистичните променливи със съответните множества от терми за дадения актив.

Класът е дефиниран с единствен конструктор, който приема четири параметъра за съответните четири полета.

Logic

Класът **Logic** е с идентификатор на достъп `public`. Този клас се използва за входна точка на библиотеката за размита логика. В него се дефинират входните и изходните лингвистични променливи със съответните им параметри.

Класът има единствено поле `qequity` от тип `double` и идентификатор на достъп, което съдържа размитата числова стойност на съответния актив. За полето `qequity` е дефинирано свойството `Qequity`.

Класът има единствен конструктор, който има четири параметъра съответно за код на актива, стойности на възвръщаемост, риск и възвръщаемост-риск изчислени в модула Събиране и съхранение на данни.

В него се дефинират размитите правила, с които работи библиотеката за размитата логика. Размитите правила са зададени в таблица 8.

Дефинират се лингвистичните променливи *Възвръщаемост*, *Риск* и *Възвръщаемост-Риск* за вход и *Качество* за изход със съответните терми – наименование на терма, функция на принадлежност и съответните параметри за функцията.

Инициализира се обект на класа **Engine**, като му се подават като параметри дефинираните правила и лингвистичните променливи за вход и изход.

Стойността върната от метода **CalculateQMeasure** на класа **Engine** се съхранява в свойството `Qequity`.

Engine

Класът **Engine** е с идентификатор на достъп `public`. Класът представлява ядрото на библиотеката за размита логика. В него се извършват действията за размиване и деразмиване до достигане на една размита числена стойност за съответния актив.

Класът има три полета с идентификатор на достъп `private`, които съхраняват размитите правила, размитите входни и изходни променливи.

Полетата са : `RuleList`; `fuzzyEquity`; `ourParams`.

RuleList съхранява множеството от размити правила дефинирани в класа **Logic**.

fuzzyEquity е от тип **FuzzyEquity** и съдържа размитите променливи за вход възвръщаемост, риск и възвръщаемост-риск дефинирани в класа **Logic**.

`ourParams` съдържа размитата променлива за изход дефинирана в класа **Logic**.

Класът **Engine** има един единствен конструктор, който приема три параметъра за съответните три полета:

```
public Engine(List<FuzzyRule> rulelist, FuzzyEquity fuzzyequity, List<ArrayList>
```

Методът **GetQList** е с идентификатор на достъп `private` и не приема параметри. Целта му е да извършва агрегирането за всяко едно правило за съответния актив.

Методът **GetQListMax** е с идентификатор на достъп `private` и не приема параметри. Методът взема максималните стойности за всеки терм от множеството от размити правила.

Методът **Defuzzy** извършва последния етап на размитата логика – деразмиването. Използваният метод за деразмиване е *център на тежестта*. Методът приема един параметър- списък с ограниченията, които ограничават графиките на функциите за принадлежност за изходната размита променлива в интервала [0;1].

Методът **CalculateQMeasure** е с идентификатор на достъп `public`, който не приема параметри. В него се обединяват методите **GetQList**, **GetQListMax** и **Defuzzy**. Резултатът от изпълнението му е една числова стойност, която описва качеството на финансовия актив.

4.5.7. Конструиране на портфейли

За реализацията на модела са създадени следните класове и интерфейси в пространството от имена `StockExchange.PortfolioConstruction`:

- `PortfolioElement`;
- `Portfolio`;
- `IPortfolioConstructionStrategy`;
- `FuzzyPortfolioConstructionStrategy`;
- `PortfolioConstructor`;
- `PortfolioUtilities`.

Също така е дефиниран клас `Combinations` в пространството от имена `StockExchange.Algorithms.Combinatorics`.

За запис на конструираните портфейли се добавят методи в класа `StockExchangeDAL` от пространството от имена `StockExchange.Data`.

Внедряването на функционалността на библиотеката за конструиране на инвестиционни портфейли се извършва в метода `Main` на класа `Program` от пространството на имена `StockExchange.DataManager`.

PortfolioElement

Полето `equityCode` е дефинирано от тип `string`, тъй като предназначението му е да запазва кода на актива, дефиниран от Българската фондова борса. Примери за такива кодове са: „3JR“, отговарящ на „Софарма АД“ и „57B“, отговарящ на „Булгартабак – Холдинг“ АД.

Полето `price` съхранява цената на една акция от дадения актив. Цената на всеки актив се оповестява публично от Българската фондова борса с точност два знака след десетичната запетая, т.е. стойността е реално число. Поради този факт е избран тип `double` за съхранение на този тип променливи.

Полето *get* съхранява стойността на възвращаемостта на даден актив. За тази цел се използва променлива от тип *double*.

Полето *risk* дава възможност на класа да опише инвестиционния риск на даден актив. Рискът е реално число, дефинирано от тип *double*.

Ratio също е поле, дефинирано от *double*. То съхранява отношението, с което даденият актив участва в инвестиционния портфейл.

Полето *qMeasure* е от особена важност в реализирания модел за конструиране на инвестиционен портфейл. То съхранява стойността, която е основен критерий за качеството на даден актив - Q-мярката на актива. Полето е дефинирано от тип *double*.

Класът *PortfolioElement* дефинира свойства за всяко едно от полетата, дефинирани по-горе. Те позволяват както промяна на стойностите на полетата на класа, така и достъп до тези полета. За разлика от полетата, свойствата са дефинирани с идентификатор за достъп *public*. Използвайки модификатора *public*, се указва на компилатора, че това свойство е достъпно от всеки друг клас.

Класът *PortfolioElement* дефинира три конструктора, всеки от които притежава различна сигнатура (брой и подредба на параметрите). Това се прави с цел удобство при работа с класа в различни ситуации. Класът *PortfolioElement* се възползва от възможността на C# да преизползва конструкторите, като един конструктор може да извиква друг конструктор, деклариран в същия клас.

Първият конструктор създава обекти от класа, използвайки подадените му параметри. Той приема седем параметъра - по един за всяко поле на класа.

Вторият конструктор създава обект от класа, като всяко от неговите полета е инициализирано със стойност по подразбиране. В случаят за кода на актива (полето от тип *string*) това е празният низ "", а за останалите полета от тип *double* това е стойността 0.

Третият конструктор на класа създава нов обект от класа на базата на друг, вече създаден обект, подаден като параметър. Стойностите на полетата на новия обект са идентични с тези на параметъра. Ефектът от използването на този конструктор е подобен на клониране на даден обект.

Класът *PortfolioElement* съдържа метод *ToString()*, който представя полетата на класа в текстов формат, удобен за отпечатване на конзола.

Portfolio

Класът *Portfolio* е абстракция на инвестиционен портфейл за целите на създадения софтуерен продукт.

Класът *Portfolio* дефинира няколко полета, описващи инвестиционен портфейл. Всички полета са дефинирани с идентификатор на достъп *private*, което ги прави достъпни само за този клас.

Полето *equities* е списък от активите, взимащи участие в конкретен инвестиционен портфейл. Именно заради това то е дефинирано като списък от *PortfolioElement* (абстракцията за актив, участващ в инвестиционен портфейл, дефинирана по-горе). Списъкът е подходяща структура от данни за съхранение на такъв тип данни, защото предоставя известна степен на гъвкавост, идваща от факта, че не е необходимо предварително да е известен броят на елементите, които ще се съхраняват.

Полето *capital* съхранява капитала, с който се прави инвестиционния портфейл. Това поле е дефинирано от тип *double*.

В полето *risk* се съхранява рискът на инвестиционния портфейл. Тъй като рискът представлява реална стойност, тя би трябвало да се съхрани в поле от тип *double*. При създаване на обекти от класа, задачата на конструктора е да инициализира

полетата на класа. Тъй като характеристики на портфейла като риск, възвращаемост и други са зависими от активите, участващи в портфейла, за инициализирането на полетата представляващи тези характеристики се използва шаблона за дизайн „късна инициализация“. Целта е стойността на тези характеристики да бъде изчислена не при създаването на обект от класа, а тогава, когато стойността е необходима за първи път. За да се реализира това, полетата представляващи тези характеристики са от такъв тип, който освен обичайните за типа стойности може да заема и стойността null, т.е. все още няма стойност. За това полето *risk* се дефинира от тип double - тип заемащ обичайните за тип double стойност, както и стойността null.

Полето *ret* съхранява стойността на възвращаемостта на даден инвестиционен портфейл. За тази цел се използва променлива от тип double.

Полето *qRatio* представя отношението на възвращаемостта и риска на инвестиционния портфейл.

Полето *qMeasure* е от особена важност в реализирания модел за конструиране на инвестиционен портфейл. То съхранява стойността, която е основен критерии за качеството на даден инвестиционен портфейл - Q-мярката на портфейла. Полето е дефинирано от тип double.

Класът дефинира публично свойство за всяко от полетата на класа. Тези свойства притежават единствено метод за четене на стойността на полето, за което отговарят, т.е. стойностите на свойствата не могат да бъдат променяни. При извикване на дадено свойство се проверява дали стойността на полето, за което свойството отговаря, вече е била изчислена, т.е. дали тя null или не. Ако стойността е null се извиква метод, който изчислява нужната стойност и я записва в съответното поле.

Класът дефинира конструктор с два параметъра. Първият параметър задава списък от обекти на класа *PortfolioElement*, участващи в инвестиционния портфейл, който се конструира. Вторият параметър указва капитала на портфейла като реална стойност от тип double.

Класът *Portfolio* съдържа метод ToString(), който представя полетата на класа в текстов формат, удобен за отпечатване на конзола.

Останалите пет метода на класа са дефинирани с идентификатор на достъп private. Те се извикват единствено от свойствата на класа и служат за изчисление на съответстващата им характеристика на инвестиционен портфейл.

Интерфейс *IPortfolioConstructionStrategy*

Тъй като методите на размитата логика не са единственият начин, по който може да се реализира модел за конструиране на инвестиционен портфейл е необходимо създаденият софтуер да работи с различни такива модели. Това става като всеки модел имплементира интерфейса *IPortfolioConstructionStrategy*.

```
public interface IPortfolioConstructionStrategy
{
    List<List<Portfolio>> Construct(IEnumerable<Equity> equityList,
        int portfolioSize, double capital);
}
```

Интерфейсът дефинира един единствен метод Construct, чиято цел е да създаде портфейли. Всеки клас, който имплементира интерфейса, ще предефинира този метод, т.е. ще реализира конкретен модел за конструиране на инвестиционен портфейл. Методът Construct приема три параметъра. Първият е списък от активи, от който ще се избират такива, които ще вземат участие в инвестиционния портфейл. Вторият параметър на функцията е цяло число (тип int), указващо максималния брой активи в

един инвестиционен портфейл. Капиталът, с който се конструират портфейлите, се указва от третия параметър.

FuzzyPortfolioConstructionStrategy

Класът *FuzzyPortfolioConstructionStrategy* представлява абстракция на модел за конкуриране на инвестиционен портфейл. Класът имплементира интерфейса *IPortfolioConstructionStrategy*, което му дава възможност да конструира инвестиционни портфейли по общ, публично известен начин, чрез метода *Construct* на интерфейса. Имплементирайки този метод, класът реализира модела за конструиране на инвестиционен портфейл със средствата на размитата логика.

Класът *FuzzyPortfolioConstructionStrategy* дефинира едно единствено поле *portfoliosList*, което съхранява създадените портфейли. Тъй като полето се използва единственото за вътрешни цели на класа, то няма нужда да бъде достъпно от други класове. Поради това то е дефинирано с идентификатор на достъп *private*. Типът на полето е списък от обекти на класа *Portfolio*.

Класът *FuzzyPortfolioConstructionStrategy* не дефинира свойства.

Тъй като в тялото на класа не присъства изрично дефиниран конструктор, компилаторът създава автоматично празен конструктор по подразбиране (*default constructor*), който присвоява стойност по подразбиране на всички полета на класа. В случая, той задава стойност *null* на полето *portfoliosList*, тъй като то е от референтен тип.

Единственият публичен метод на класа *FuzzyPortfolioConstructionStrategy* е методът *Construct*. Този метод реализира модел за конструиране на инвестиционен портфейл със средствата на размитата логика. Резултат от неговото изпълнение е списък от инвестиционни портфейли. Методът *Construct* приема три параметъра: списък от активи, максималния брой активи в един инвестиционен портфейл и капитала, с който се конструират портфейлите. За извършването на своята задача, методът *Construct* използва редица от помощни методи, дефинирани с идентификатор на достъп *private*.

Методът *ChooseTopEquities* сортира елементите на даден списък от активи, подаден като параметър, спрямо тяхната Q-мярка и избира най-добрите от тях. Списъкът се сортира посредством метода за бързо сортиране (*QuickSort*). Описание на алгоритъма е дадено по-долу. Броят на избраните активи също зависи от параметър. Методът приема два параметъра. Първият е списъкът, от който се избират активи, а вторият е цяло число, определящо броя на избраните от списъка активи.

Алгоритъмът за бързо сортиране започва с избирането на „главен“ елемент от списъка с елементи, които ще бъдат сортирани. След това списъкът се пренарежда така, че всички елементи, които са по-малки от „главния“ се поставят вляво от него, а всички, които са по-големи - вдясно от него. Рекурсивно се повтарят горните стъпки върху списъка с по-малките и списъка с по-големите елементи. Получените списъци се сливат (конкатенация) и се получава сортираният списък.

Генерирането на комбинации от активи се извършва от метода *GenerateEquityCombinations*. Той приема два параметъра: списък от активи, чиито комбинации трябва да се генерират и целочислена стойност, определяща максималния брой активи, участващи в една комбинация.

Методът *GeneratePortfolio* генерира инвестиционни портфейли на базата на комбинация от активи и зададен капитал. Методът приема два параметъра. Първият е комбинацията от активи. Тези активи за записани в списък. Вторият параметър на метода е реално число (тип *double*), задаващо капитала, с който се конструират инвестиционните портфейли. Методът конструира един портфейл, след което се опитва

да го подобри, прилагайки процедура по алокация. Тази процедура създава нови портфейли. Методът връща списък от портфейли.

Проверката, дали даден инвестиционен портфейл подлежи на подобряване, се извършва от метода *HaveToAllocate*. Той връща стойност *true*, ако за портфейла трябва да се извърши алокация. В противен случай методът връща стойност *false*. Методът приема три параметъра от тип *double*. Първият параметър задава използвания от даден портфейл капитал, вторият параметър е началния капитал на портфейла и а третият параметър е толеранса. Толерансът определя минималния праг, при който стойностите на използвания и началния капитал са достатъчно близки и следователно не е необходимо да се прилага алокация.

Самата процедура по алокация е реализирана в метода *Allocate*. Тя подобрява даден инвестиционен портфейл, като разпределя останалия неинвестиран капитал, увеличавайки броя на закупените акции на активите спрямо техния относителен дял.

Методът приема един параметър - списък от портфейли и връща същия този списък, като евентуално е добавил нов елемент в края на списъка.

Методът *GeneratePortfolioElement* преобразува обект от тип *Equity* в обект от тип *PortfolioElement*. Това се извършва с цел редуциране на изискваната памет при работа с абстракция на финансов актив. Класът *Equity* съдържа допълнителни данни, които не са необходими в контекста на инвестиционните портфейли. Методът приема един параметър - обект от класа *Equity*, и връща обект от класа *PortfolioElement*.

PortfolioConstructor

Класът *PortfolioConstructor* дефинира единен начин за конструиране на инвестиционни портфейли. Той параметризира модела, чрез който се създават портфейли. Също така дава възможност, при нужда този модел да се подмени, т.е. не е необходимо създаването на нов обект от типа *PortfolioConstructor*.

Класът *PortfolioConstructor* пази информация за конструиания инвестиционен портфейл в четири полета. Те са дефинирани с идентификатор на достъп *private*, което ги прави достъпни само за дефиницията на този клас.

Полето *PortfolioConstructionStrategy* съхранява модела за конкуриране на инвестиционен портфейл. Типът му е интерфейса *IPortfolioConstructionStrategy*, който имплементират всички модели за конструиране на портфейл.

Второто поле е *equityList*. То е от тип списък от обекти на класа *Equity*, защото целта му е да съхранява активите, от които ще се изберат такива, които ще участват в инвестиционния портфейл.

Полето *portfolioSize* е от тип *int* и съхранява максималния брой финансови активи, които могат да взимат участие в един инвестиционен портфейл.

Капиталът, с който се конструира инвестиционния портфейл се запазва в полето *capital*. То е от тип *double*.

Класът дефинира едно публично достъпно свойство - *ConstructionStrategy*. То дава възможност за подмяна на модела, с който се конструират инвестиционни портфейли. За целта е достатъчно на свойството да бъде присвоен обект на клас, имплементиращ интерфейса *IPortfolioConstructionStrategy*.

Класът дефинира един конструктор с четири параметъра, който инициализира полетата на класа със стойностите подадени като параметри.

Методът с публичен идентификатор за достъп *ConstructPortfolio* на класа конкурира инвестиционни портфейли на базата на стойностите, инициализирани при създаване на обект от класа. Ако е необходимо да се подмени моделът, с който се конструират инвестиционни портфейли, след извършване на подмяната трябва

повторно да се извика методът *ConstructPortfolio* с цел конструиране на портфейлите, използвайки новозададения модел.

PortfolioUtilities

Тъй като съхранението на данни за конструираните инвестиционни портфейли не е функционалност, пряко свързана с класа *Portfolio*, но доколкото такава функционалност е необходима, то тя е изнесена в отделен клас. Това решение подобрява обособеността на елементите на софтуерния продукт, което подобрява неговото качество. Класът *PortfolioUtilities* съдържа метод за запис на списък от инвестиционни портфейли в база от данни.

Класът *PortfolioUtilities* дефинира статичен метод *SavePortfolios*, чиято цел е запис на списък от инвестиционни портфейли в база от данни. Методът приема един параметър - списък от обекти от тип *Portfolio*.

StockExchangeDAL

Описаните до сега класове и интерфейси са част от пространството от имена *PortfolioConstruction*. Те съставят ядрото на библиотеката за конструиране на инвестиционни портфейли. За разлика от тях класът *StockExchangeDAL* е дефиниран в различно пространство от имена, а именно *StockExchange.Data*. В това пространство се намират класове, спомагащи връзката с базата данни. Класът *StockExchangeDAL* е общ и за трите модула на софтуерния продукт и съдържа методи за четене и запис от база от данни.

Методът *GetIdByCode* връща индекса на даден актив, записан в базата от данни, спрямо подаден като параметър код на актива. Методът приема един параметър - *equityCode* от тип *string*, задаващ кода на актива, чиито индекс ще се търси. Търсеният код е от тип *int*. Ако в следствие на липсващ актив методът не е в състояние да върне коректен резултат, то се извежда „изключение“.

Методът *GetLastPriceForEquity* намира последната записана цена за една акция на даден актив, подаден като параметър. Методът приема параметър, указващ актива (обект от тип *Equity*), чиято цена ще се търси и връща цената на актива като тип *double*.

Записът на конструираните инвестиционни портфейли се осъществява от метода *InsertPortfolio*. Той приема параметри, указващи характеристиките на даден портфейл. Методът връща целочислена стойност от тип *int* - индекса на вече записания портфейл.

Записът на активите, взимачи участие в инвестиционен портфейл става чрез използването на метода *InsertPortfolioElement*. Той приема три параметъра. Първият указва индекса на портфейла, към който принадлежи актива. Вторият е кодът на актива. Третият параметър задава броя акции, с които активът участва в инвестиционния портфейл.

Combinations

За генерирането на комбинации от вече избраните активи се дефинира отделен клас.

Полето *elements* е от тип списък от обекти от произволен тип. То съхранява елементите, от които ще се генерират комбинации.

Горната граница на броя елементи в една комбинация е записана в полето *upperIndex*. То е декларирано от целочисления тип *int*.

Полето *option* съхранява информация за начина, по-който се генерират комбинациите - с или без параметри. То е от тип *GenerateOption*. Изброимият тип *GenerateOption* дефинира две константи. Първата указва на класа *Combinations*, че

трябва да се генерират комбинации с повторение (*WithRepetition*), а втората - че трябва да се генерират комбинации без повторение (*WithoutRepetition*). Изброените типове могат да приемат стойности само измежду изброените в типа константи.

Класът дефинира един конструктор с три параметъра, който инициализира полетата на класа, със стойностите подадени като параметри. Първият параметър е списък от обекти от произволен тип, чиито комбинации ще се генерират. Вторият параметър е цяло число, дефиниращо максималния брой обекти в една комбинация. Третият параметър е константа от изброен тип *GenerateOption*, указваща дали комбинациите ще се генерират с повторение или без. По подразбиране стойността на третия параметър е *GenerateOption. WithoutRepetition* определя генерирането на комбинации без повторение.

Публичният метод *Generate* генерира списък от комбинации от обекти. Една комбинация от обекти се представя като списък от тези обекти. На базата стойността на полето *option*, методът *Generate* извиква един от два други метода, генериращи комбинации с повторение и съответно без повторение.

Методът *CombinationsWithoutRepetition* генерира комбинации без повторение. За целите на модела за конструиране на инвестиционни портфейли със средствата на размитата логика се използва именно този метод.

Методът *CombinationsWithRepetition* генерира комбинации с повторение.

Program

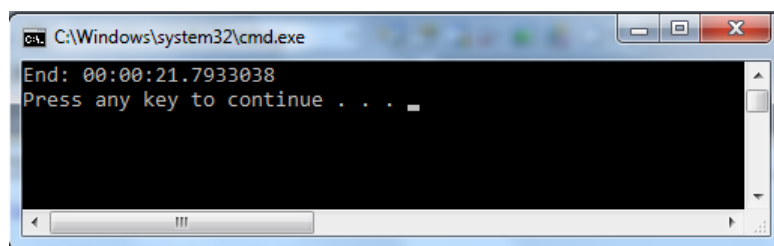
Функционалността на библиотеката за конструиране на инвестиционни портфейли се използва в метода *Main* на програмата. Първоначално се инициализират променливи, които са необходими на конструктора на класа *PortfolioConstructor*. След това се създава обект от този клас и се извиква методът *ConstructPortfolio*, резултатът от което е генериране на списък от инвестиционни портфейли. Този списък се записва в база от данни с помощта на метода *SavePortfolios* на класа *PortfolioUtilities*.

Глава 5. Тестване на софтуерната система FSSAM с реални данни

За тестване на създадената софтуерна система за управление на финансови активи са изследвани следните параметри: време за изпълнение на програмата, достоверност на получените по модела FLQM оценки на единични активи, степен на изчерпване на капитала при различни първоначални суми, поведение на портфейлите в продължение на шест месеца. Използвани са данни от БфБ-София АД [101].

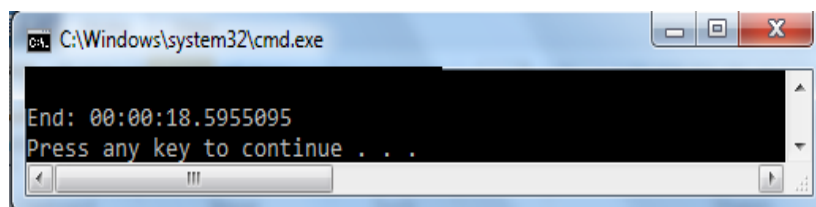
5.1. Време за изпълнение на програмата

При първоначално стартиране на системата всички активи са непознати (т.е. няма данни за тях) и за всеки от тези активите се извличат данни за една година назад. На фигура 5.1 е показано времето, за което системата извлича, пресмята характеристиките на активите и ги съхранява при първоначалното стартиране при един от проведените тестове (на 05.06.2012). В случая са открити данни за 42 актива и времето за работа на системата е 21.7933038 s.



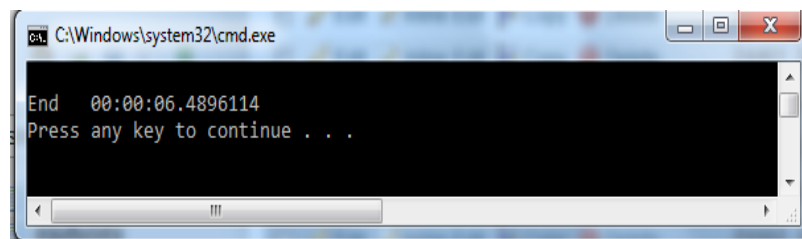
Фигура 5.1. Време за първоначално извличане, пресмятане на характеристиките на активите и съхранението им за 42 актива.

На фигура 5.2 е показано времето за извличане, пресмятане на характеристиките на активите и тяхното съхранение при повторно стартиране на програмата (на следващия ден). Непознатите активи в случая са 22, т.е. системата работи с 64 актива.



Фигура 5.2. Време за извличане, пресмятане на характеристиките на активите и съхранението им за 64 актива, като 22 от тях са непознати за системата.

При третото стартиране на системата се казва, че няма непознати активи. В този случай времето на работа е значително по-малко (фиг. 5.3).



Фигура 5.3. Време за извличане, пресмятане на характеристиките на активите и съхранението им за 64 актива, като нито един от тях не е непознати за системата.

При провеждане на общо 10 теста за бързодействие, полученото време е от подобен порядък – по-малко от 2 минути. Може да бъде направен извод, че бързодействието на системата зависи от броя на непознатите активи за системата в конкретен момент, като при всяко следващо стартиране времето за изпълнение на програмата намалява. Но тъй като това време (2 минути, както е отбелязано по-горе) е пренебрежимо малко в сравнение на периода за сваляне на наблюдения и вземане на решение (24 часа), то системата работи достатъчно бързо, следователно на практика тя работи в реално време.

5.2. Резултати от приложението на модела FLQM за оценка на финансови активи

За демонстриране на резултатите от работата на приложението на модела FLQM с реални данни от Българската фондова борса е избран активът с код 5IC, отговарящ на Застрахователна компания Евро Инс, София. Използвани са данни за цената на актива в периода от 19.10.2011г. до 15.02.2012г.

Стойностите на характеристиките $X_1 \triangleq return$, $X_2 \triangleq Risk$ и $X_3 \triangleq q-ratio$ са показани в таблица 5.1, като степените на принадлежност към съответните терми са в таблици 5.2, 5.3 и 5.4.

$X_1 \triangleq return$	0,038834951
$X_2 \triangleq Risk$	0,035095505
$X_3 \triangleq q-ratio$	1,106550566

Таблица 5.1. Стойностите на трите характеристики за 5IC.

Терм	Степен на принадлежност
<i>Very low</i>	0,480592279
<i>Low</i>	0,002671793
<i>Neutral</i>	0
<i>High</i>	0
<i>Very high</i>	0

Таблица 5.2. Степени на принадлежност за възвращаемост 0,038834951 към термите на размитата променлива X_1 .

Терм	Степен на принадлежност
<i>много малък</i>	0,482459448
<i>малък</i>	0,4306245498
<i>Неопределен</i>	0,0007765479
<i>Голям</i>	0
<i>много голям</i>	0,209190962

Таблица 5.3. Степени на принадлежност на риск 0, 035095505 към термите на размитата променлива X_2 .

Терм	Степен на принадлежност
<i>Small</i>	0,996557245
<i>Neutral</i>	0,004865305
<i>Big</i>	0

Таблица 5.4. Степени на принадлежност на възвращаемост-риск 1,106550566 към термите на размитата X_3 .

Следващата стъпка след размиването на входните променливи е да се приложат правилата за изводи, което е описано подробно в глава 3.

Агрегираните резултати от правилата се използват при формиране на изходната размита променлива $Y \triangleq Q\text{-measure}$.

След деразмиване на изходната променлива като краен резултат от действието на софтуерната система за оценка на актива 5IC се получава стойност 0.192199099, което, според описаните в глава 3 модели, показва доста лошо качество.

Приложението се използва за оценяване на всички активи, за които са свалени данни до конкретния момент. Резултатите от оценяването на 51 актива с данни от Българската Фондова Борса за периода от 19.10.2011г. до 15.02.2012г. са показани в таблица 5.5.

Код	Възвращаемост	Риск	Годишна норма на възвращаемост	Възвращаемост -Риск	Q
3JR	0,86588	0,00878	0,59763	68,0317	0,34158
4CF	0,80388	0,01906	0,41165	21,6024	0,19791
4EH	0,87871	0,02594	0,63614	24,5232	0,20375
5F4	0,84348	0,02959	0,53043	17,9287	0,20056
6C4	0,70206	0,02366	0,10619	4,48764	0,19297
E4A	0,86022	0,01301	0,58065	44,6162	0,28087
1VX	1,17391	0,01598	1,52174	95,2483	0,80900
3JU	0,98535	0,01714	0,95606	55,7954	0,34062
3MZ	0,9636	0,03284	0,89081	27,1293	0,23761
3NB	0,88569	0,02745	0,65706	23,9397	0,20406
3NJ	0,92778	0,03047	0,78333	25,7124	0,21761
3ZL	0,57388	0,03674	-0,2784	-7,5773	0,18919

4BJ	0,92946	0,01994	0,78837	39,5334	0,28443
4I8	0,75111	0,03454	0,25333	7,33382	0,19471
4ID	0,91796	0,02987	0,75389	25,2389	0,21195
4IN	0,73468	0,05748	0,20403	3,54992	0,1938
4L4	1,3125	0,0228	1,9375	84,9622	0,82888
4O1	0,77039	0,04901	0,31116	6,34955	0,19614
52E	0,77956	0,03154	0,33867	10,7391	0,19668
53B	0,76792	0,03119	0,30375	9,73956	0,19594
55B	0,87291	0,01459	0,61872	42,4134	0,27763
57B	1,02665	0,02625	1,07994	41,1446	0,34086
5BN	1,09127	0,01054	1,2738	120,84	0,81218
5IC	0,67961	0,0351	0,03883	1,10655	0,1922
5MH	0,86707	0,01985	0,6012	30,2933	0,25051
5ODE	0,92487	0,02239	0,77462	34,5974	0,27467
5ORG	1	0	1	0	0,19943
5OTZ	0,60976	0,03989	-0,1707	-4,2797	0,19051
5SR	0,7616	0,0267	0,2848	10,6649	0,19536
5V2	1,17043	0,03182	1,51129	47,5021	0,51089
6A8	0,94975	0,00983	0,84925	86,4039	0,78095
6A9	1	0,03503	1	28,5484	0,26337
6AB	0,91379	0,00709	0,74138	104,603	0,80937
6C4P	0,94717	0,02696	0,8415	31,2182	0,26877
6L1	0,69	0,03249	0,07	2,15438	0,19252
6S5	1,30435	0,0211	1,91304	90,6738	0,81145
6S7	0,66667	0,04533	0	0	0,19178
AO0	1,17958	0,04112	1,53874	37,4181	0,50135
C81	0,71667	0,3248	0,15	0,46182	0,20175
E4AP	0,9986	0,01688	0,99581	59,0087	0,35244
G0A	1,13636	0,01319	1,40909	106,855	0,81151
SO5	1,0582	0,01477	1,1746	79,5393	0,75408
4EC	1,72136	0,02378	3,16409	133,044	0,81247
5BD	1,0679	0,03434	1,2037	35,0483	0,36046
5BU	1,22139	0,02292	1,66418	72,6153	0,68653
5H4	1,15873	0,00812	1,47619	181,686	0,81245
6A6	1,12038	0,01152	1,36113	118,107	0,81215
6BM A	0,80769	0,03925	0,42308	10,7795	0,19841
6F3	1,00442	0,03394	1,01327	29,8562	0,279
BLKC	0,96061	0,02072	0,88182	42,5645	0,30396
ZNO A	1,38056	0,07691	2,14167	27,8465	0,68352

Таблица 5.5. Резултати от оценяването на 51 актива.

5.3. Изследване на характеристиките и управление на актив

В тази точка детайлно са показани и анализирани резултатите за 5 от записаните в базата общо 51 актива, за които на Българската фондова борса е имало сделки в периода от 01.06.2012 до 15.06.2012.

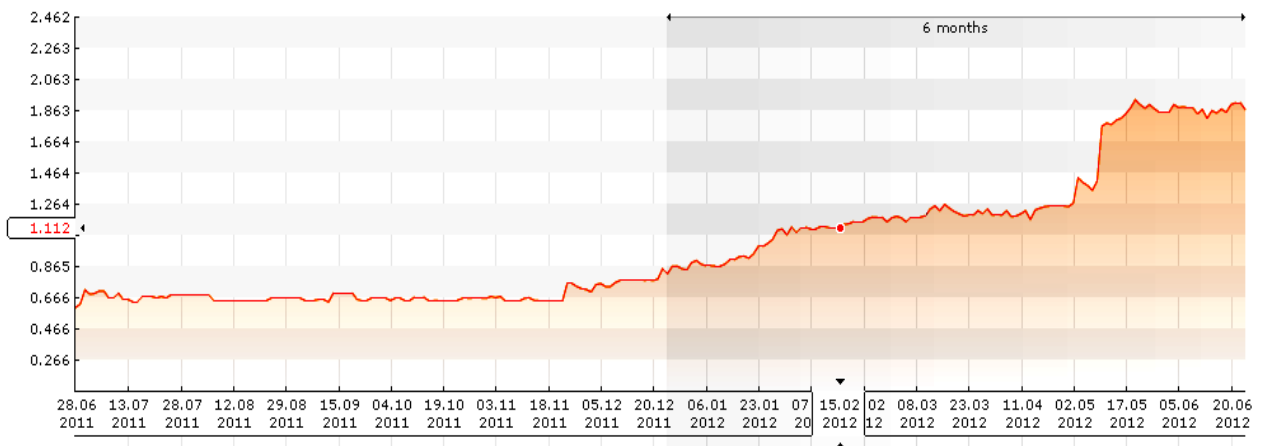
За сравняване на получените резултати с реалното изменение на цените на активите системата е симулирана да работи на предна дата и в случая това е 15.02.2012 (три месеца назад). За изследването са избрани активите с кодове: 4ЕС, Е4А, 55В, 6F3, 3NB, тъй като всеки от тези активи има различно поведение в съответния 3-месечен период.

При симулациите със старата дата са използвани редици с цени, които са с дължина 10, 25, 50, 100, 120, 150 и 200 дни преди конкретната дата. Това е направено с цел да се направят емпирични наблюдения върху зависимостта на достоверността на оценките и дължината на използваната редица от цени

5.3.1. Изследване на характеристиките на актива 4ЕС

ЕЛАРГ Фонд за Земеделска Земя АДСИЦ-София

Като първи актив е избран 4ЕС. За този актив се забелязва значително увеличение на цената за разглеждания 3-месечен период след 15.02.2012 (фиг. 5.4.). Важно е такива печеливши активи да бъдат откривани своевременно.



Фигура 5.4. Едногодишна графика на цените на актива 4ЕС.

В таблица 5.6. са показани получените на 15.02.2012 резултати за възвращаемост, риск, годишна норма на възвращаемост, q -ratio и Q . Стойностите на Q за всички периоди е по-голяма от 0,75, което е индикация за стабилно повишаване на цената му, според заложената в системата и описана в глава 3 методология.

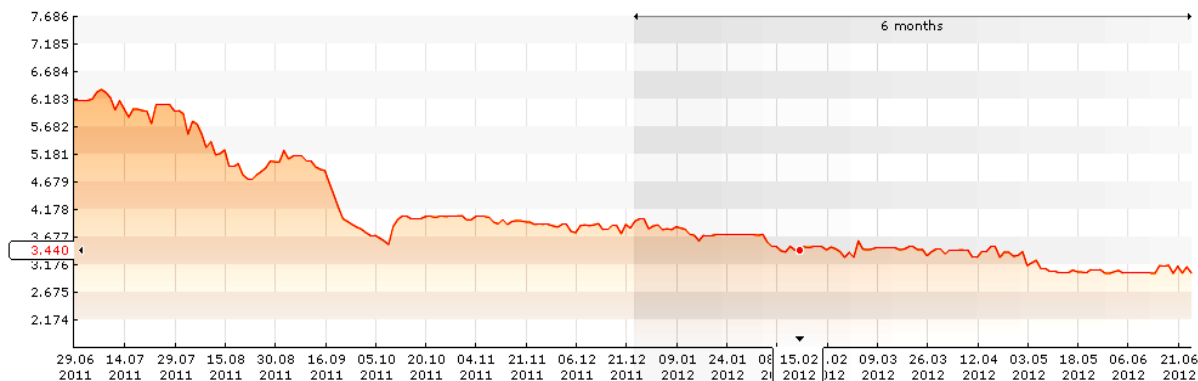
4ЕС					
Период	Възвращаемост	Риск	Годишна норма на възвращаемост	Възвращаемост/Риск	Q
10	1,000000	0,005822621	1,000000	171,7439757	0,81244408
25	1,17921527	0,021573573	3,509013786	162,6533453	0,812479437
50	1,287037037	0,019055353	3,009259259	157,9219874	0,812481545
100	1,732087227	0,025468852	3,196261682	125,4968862	0,812469577
120	1,721362229	0,023782192	3,164086687	133,0443653	0,812474403
150	1,609261939	0,022748406	2,218523878	97,52436543	0,811988257
200	1,637702504	0,021201632	1,637702504	77,24417096	0,755784754

Таблица 5.6. Изменения на характеристиките за различни периоди на актива 4ЕС.

5.3.2. Изследване на характеристиките на актива E4A

Енемона АД-Козлодуй

Следващият избран актив е E4A, показващ трайно намаление на цената си в разглеждания период. На фигура 5.5. е показана графика за едногодишното изменение на цената на актива E4A.



Фигура 5.5. Едногодишна графика на цените на актива E4A.

В таблица 5.7 са показани получените резултати за възвращаемост, риск, годишна норма на възвращаемост, q -ratio и Q . Системата отново е тествана за различни периоди назад от избраната дата 15.02.2012г.: 10, 25, 50, 100, 120, 150 и 200 дни. Стойностите на Q за всички периоди е по-малка от 0,3, което е индикация за намаляване на цената му.

E4A					
Период	Възвращаемост	Риск	Годишна норма на възвращаемост	Възвращаемост/Риск	Q
10	0,960625524	0,015526168	-0,417481151	-26,88887316	0,187568107
25	0,924979833	0,012023871	-0,050282334	-4,181875659	0,190703978
50	0,86000000	0,01324307	0,0200000	1,510223804	0,192185691

100	0,851485149	0,014001146	0,554455446	39,60071919	0,270097175
120	0,860215054	0,013014231	0,580645161	44,61617183	0,280867842
150	0,739784946	0,016090255	0,479569892	29,80499045	0,244071109
200	0,576504106	0,01762915	0,576504106	32,70175295	0,259088987

Таблица 5.7. Изменения на характеристиките за различни периоди на актива Е4А.

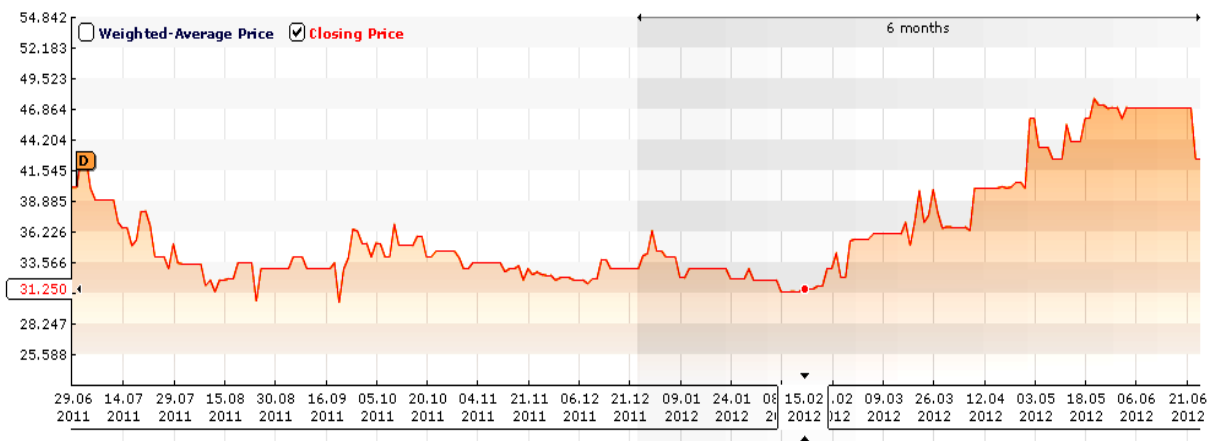
Данните от таблиците и графиките ясно показват, че котировките на актива Е4А се понижават. Интересно е да се отбележи, че този актив не е бил търгуван точно на 15.02.2012. Последната котировка е от 14.02.2012, т.е. тук е използвана процедурата за дописване на данните. По-важно е, че наистина за следващите три месеца цената му от 3,496 на 14.02.2012 пада до 3,098 на 10.05.2012.

Резултатите и в този случай удовлетворяват изискванията на системата и показват нейната правилна функционалност.

5.3.3. Изследване на характеристиките на актива 55В

Благоевград-БТ АД-Благоевград

В таблица 5.8. са показани получените резултати за възвращаемост, риск, годишна норма на възвращаемост, q -ratio и Q . След провеждане на тестове след съответния брой дни Q има стойност по-малка от 0,35, което е индикация за намаляване на цената му.



Фигура 5.6. Едногодишна графика на цените на актива 55В.

55В					
Период	Възвращаемост	Риск	Годишна норма на възвращаемост	Възвращаемост-Риск	Q
10	0,9765625	0,011132527	0,15625	14,03544718	0,193669716
25	0,946969697	0,011855241	0,257575758	21,72674217	0,19505254
50	0,910282552	0,016271766	0,371977862	22,86032478	0,197155223

100	0,932835821	0,01464274	0,798507463	54,53265145	0,304627247
120	0,872905028	0,014587716	0,618715084	42,41343206	0,277629998
150	0,946969697	0,021217156	0,893939394	42,13285695	0,304541603
200	0,889046942	0,022171535	0,889046942	40,09857365	0,298748941

Таблица 5.8. Изменения на характеристиките за различни периоди на актива 55В.

Но на фигура 5.6 ясно се вижда, че в периода след избраната дата, цената на актива рязко се увеличава от 31.032 лв. на 15.02.2012 до 43.50 на 08.05.2012. Разбира се това покачване няма как да бъде уловено от предходните данни, които показват колебание на цената около 32 лв. Точно поради тази причина системата е проектирана и реализирана да се стартира всекидневно и автоматично. За този актив Q започва да е по-голямо от 0,6 още на 12.03.2012, което означава, че системата достатъчно бързо установява предстоящото увеличение.

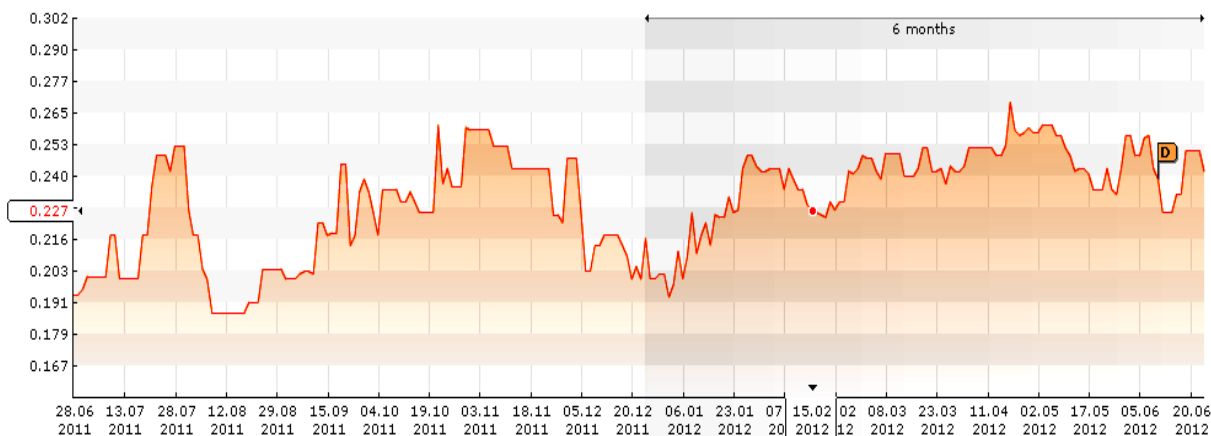
5.3.4. Изследване на характеристиките на актива 6F3

ФеърПлей Пропъртис АДСИЦ-София

Получените резултати за 6F3 са показани в таблица 5.9. Особеното за този актив е, че цената му се покачва и намалява за кратък период от време, т.е. няма стабилна тенденция. За всички тестови периоди (с изключение на 50 дни) Q е по-малка от 0,3, което е индикация за лошото качество на актива.

6F3A					
Период	Възвращаемост	Риск	Годишна норма на възвращаемост	Възвращаемост-Риск	Q
10	0,934156379	0,01946805	-1,37037037	-70,39073546	0,187452133
25	0,978448276	0,02046941	0,698275862	34,11314124	0,267781242
50	1,135	0,028895226	1,945	67,31215726	0,806578552
100	0,900793651	0,031442889	0,702380952	22,33830875	0,206135912
120	0,900793651	0,031442889	0,702380952	22,33830875	0,206135912
150	1,041284404	0,035511066	1,082568807	30,48539287	0,296604838
200	0,900793651	0,033562662	0,900793651	26,83915954	0,234670739

Таблица 5.9. Изменения на характеристиките за различни периоди на актива 6F3.



Фигура 5.7. Едногодишна графика на цените на актива 6F3.

Справка с цената на актива потвърждава видяното на фигура 5.7 изменение на цената и в посока намаление и в посока увеличение. Такъв актив не е подходящ за инвестиция в контекста на ограниченията, описани в 1.5.

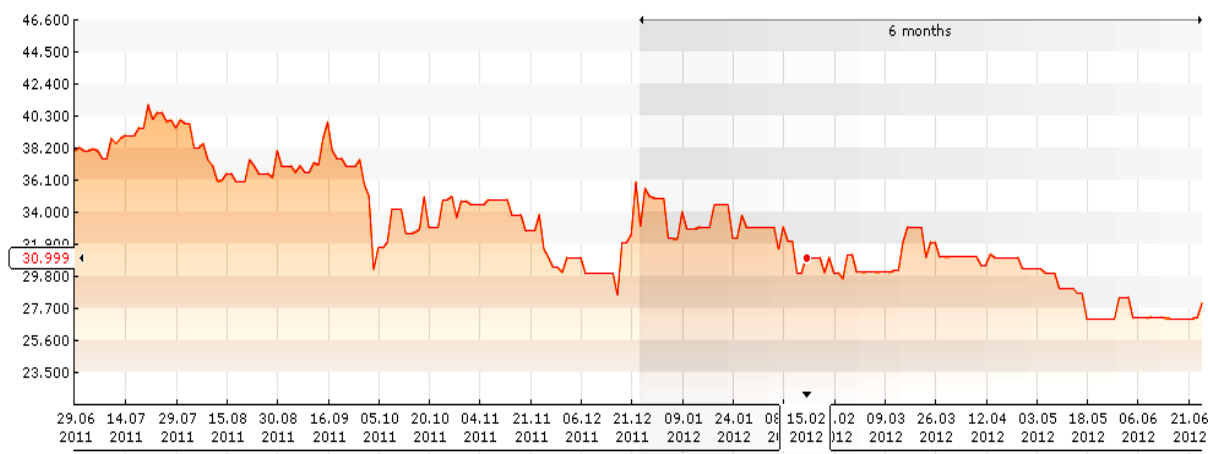
5.3.5. Изследване на характеристиките на актива 3NB

Неохим АД-Димитровград

В таблица 5.10. са показани получените резултати за актива 3NB. При всички тестове Q има стойност 0,25, което е сигурна индикация за намаляване на цената му, което е очевидно и от графиката на фигура 5.8.

3NB					
Период	Възвращаемост	Риск	Годишна норма на възвращаемост	Възвращаемост-Риск	Q
10	0,939363636	0,035897115	-1,182909091	-32,95276236	0,18750712
25	0,899825835	0,027079179	-0,402438316	-14,86154054	0,188015624
50	0,885685714	0,024735916	0,1998	8,077323627	0,194082205
100	0,890775862	0,028540179	0,672327586	23,55723105	0,204556218
120	0,885685714	0,0274463	0,657057143	23,93973464	0,20405607
150	0,815763158	0,029308894	0,631526316	21,54725865	0,203279057
200	0,78478481	0,026752295	0,78478481	29,33523306	0,252526407

Таблица 5.10. Изменения на характеристиките за различни периоди на актива 3NB.



Фигура 5.8. Едногодишна графика на цените на актива 3NB.

Извод: Софтуерната системата за оценка на финансови активи, базирана на модела FLQM коректно и своевременно успява да открие тенденциите в изменението на цените на търгуваните на БФБ акции.

5.4. Инвестиционен портфейл

За първоначално конструиране на всички описани по-надолу портфейли са използвани котировките от 20.06.2012 г. на акциите, търгувани на БФБ.

Описаната в глава 4 софтуерна система FSSAM извежда като резултат от модула за конструиране на инвестиционни портфейли следните данни: кодовете на включените в портфейла акции, цената на всяка акция, брой закупени акции, дела на актива в портфейла, характеристиките на всяка акция, както и характеристиките на самия портфейл.

5.4.1. Конструиране на инвестиционен портфейл с дялове на активите, зависещи от Q

При реализиране на модела за конструиране на портфейл, се извършва сортиране в низходящ ред в зависимост от стойността на Q на търгуваните активи. В таблица 5.11 са показани двадесет и петте актива, за които стойността на Q е най-голяма на 20.06.2012. С тях е конструиран инвестиционен портфейл, като дялове на активите са получени по формула 3.4. Инвестиционният капитал е 100 000 лв.

Име на актив	Цена на една акция	Брой закупени акции	Дял	Риск	Годишна норма на възвращаемост	Q
3JU	45	140	0.0633169	0.0241193	1.99408284	0.8291928
5ORG	90	68	0.0620385	0.0024739	1	0.81245

6A6	1.74	3565	0.0620379	0.0123787	1.85524372	0.8124423
BLKC	0.471	13171	0.0620355	0.0222177	2.52884615	0.8124111
4EC	1.91	3247	0.0620326	0.0275439	2.86830926	0.8123728
SO5	1.846	3360	0.0620319	0.0087443	1.18275862	0.8123643
5BD	0.754	8225	0.0620170	0.0087146	1.04851752	0.8121686
5BN	3.95	1566	0.0618683	0.0116723	1.2027027	0.8102220
57B	50	123	0.0617929	0.0330505	2.6875	0.8092345
55B	42.5	145	0.0617658	0.0271792	1.96134966	0.8088787
4L4	40	143	0.0573309	0.010419	0.85714286	0.7508000
1VX	2.73	1209	0.0330179	0.0183997	1.19921875	0.4323995
4BJ	1.188	2767	0.0328755	0.0260985	1.22826087	0.4305343
5BU	0.5	6447	0.0322369	0.0144816	0.98804781	0.4221720
6F3	0.244	11456	0.0279527	0.018326	1.0373444	0.3660662
5H4	14	177	0.0247831	0.0195123	0.89655172	0.3245579
6C4P	1.707	1243	0.0212330	0.0175938	0.70952381	0.2780657
3NB	27	78	0.0211748	0.0139949	0.59615385	0.2773041
E4A	3.013	671	0.0202315	0.0193644	0.65540541	0.2649496
5V2	5.55	356	0.0197821	0.0318552	0.92105263	0.2590652
6AB	47	41	0.0196215	0.0231535	0.71148632	0.2569616
3JR	2.25	856	0.0192648	0.0135707	0.4333093	0.2522899
4I8	1.461	1246	0.0182115	0.0245231	0.68566176	0.2384963
AO0	1.493	1065	0.0159043	0.0363463	0.73777506	0.2082814
3MZ	0.701	2202	0.0154412	0.0332106	0.59309494	0.2022173

Таблица 5.11. 25 актива с най-голяма стойност на Q

От формули 2.16 и 2.19 за портфейла, конструиран с тези активи се получават следните характеристики:

- Възвращаемост на конструирания портфейл $R_p = 1,4652573$,
- Риск на конструирания портфейл $s_p = 0,0185322$.

След прилагане на модела FLQM се получава, че

- Q на конструирания портфейл 0,7583479.

В случая

- Неизползваният капитал е 232,27 лв.

За сравнение, инвестиционен портфейл, конструиран със същите активи, но с равни относителни дялове ($x_j = 0,04$ за всяко j) има следните характеристики:

- Възвращаемост на конструирания портфейл $R_p = 1,2231736$.
- Риск на конструирания портфейл $s_p = 0,0199578$;
- Q на конструирания портфейл 0,44440461;
- Неизползван капитал 120,65 лв.

След сравняване на резултатите се установява, че инвестиционният портфейл с дялове на активите, зависещи от Q запазва по-висока стойност на Q на портфейла като цяло, независимо от факта, че в портфейла участват активи с ниска стойност на Q .

За проследяване на инвестиционен портфейл във времето на следващия ден е конструиран портфейл само от първите десет актива от таблица 5.11 с инвестиционен капитал отново 100 000 лв. Изменението на капитала след инвестиране в този портфейл е показано в таблица 5.12.

Име на актив	Брой закупени акции	Дял в портфейла	20.06.2012		02.07.2012	
			Цена на една акция	Инвестирана сума	Цена на една акция	Инвестирана сума
3JU	204	0,10196994	50	10200.00	45.012	9182.45
5ORG	111	0,09991100	90	9990.00	90	9990.00
6A6	5742	0,09991006	1.74	9991.08	1.783	10237.99
BLKC	22300	0,09990622	0.448	9990.40	0.48	10704.00
4EC	5400	0,09990151	1.85	9990.00	1.91	10314.00
SO5	5708	0,09990047	1.75	9989.00	1.875	10702.50
5BD	13228	0,09987639	0.755	9987.14	0.738	9762.26
5BN	2588	0,09963701	3.85	9963.80	4.07	10533.16
57B	211	0,09951558	47	9917.00	48.6	10254.60
55B	211	0,09947182	47	9917.00	42.52	8971.72
общо		1		99935.42		100652.68
			разлика	-64.58		717.26

Име на актив	Брой закупени акции	Дял в портфейла	16.07.2012		15.08.2012		14.09.2012	
			Цена на една акция	Инвестирана сума	Цена на една акция	Инвестирана сума	Цена на една акция	Инвестирана сума
3JU	204	0,1019699	45.012	9182.45	49	9996.00	60	12240
5ORG	111	0,0999110	90	9990.00	81	8991.00	81	8991
6A6	5742	0,0999100	1.81	10393.02	2.018	11587.36	2.02	11598.84
BLKC	22300	0,0999062	0.53	11819.00	0.705	15721.50	0.685	15275.5
4EC	5400	0,0999015	1.91	10314.00	1.989	10740.60	2.02	10908
SO5	5708	0,0999004	1.705	9732.14	1.705	9732.14	1.754	10011.832
5BD	13228	0,0998763	0.798	10555.94	0.88	11640.64	0.956	12645.968
5BN	2588	0,0996370	4.5	11646.00	4.75	12293.00	5.28	13664.64
57B	211	0,0995155	47.5	10022.50	52	10972.00	59	12449

55B	211	0,0994718	44.5	9389.50	51.95	10961.45	52.5	11077.5
общо								
	1			112635.69		112635.69		118862.28
разлика				3109.13		12700.27		18926.86

Таблица 5.12. Тримесечно изменение на портфейл, конструиран от 10 актива с най-голяма стойност на Q на 21.06.2012

При конструиране на портфейла на 21.06.2012 използваният капитал $S_u = 99935.42$ лв., т.е. неизползваният капитал е 64.58 лв., което е 0.064568%.

След 15 дни (02.07.2012) цената на портфейла нараства с 717.26 лв. до 100 652.68 лв. Нормата на възвращаемост за периода е 0.72%, а съответната годишна норма на възвращаемост е 17.23%.

След 1 месец (16.07.2012) цената на портфейла е нараснала общо с 3109.13 лв., т.е. нормата на възвращаемост за периода е 3.11% и годишната е 37.33%.

След 2 месеца (15.08.2012) цената на така направената портфейлна инвестиция е нараснала вече с 12 700.27 лв., което е 12.71% норма на възвращаемост за периода и 76.25% за годината.

След 3 месеца (14.09.2012) цената е нараснала с 18 926.86 лв., което означава, че норма на възвращаемост за периода е 18.94% и 75.76% за годината.

Резултатите показват, че моделите са ефективно реализирани и основната цел – увеличаване на капитала в период около 3 месеца е изпълнена. Тук е важно да се отбележи, че по-късно активът 5ORG е заличен от фондовата борса, което само потвърждава хипотезата на автора, че ежедневната проверка на ситуацията с цените на активите е от изключително значение.

5.4.2. Изследване на зависимостта между големината на инвестиционния капитал и необходимостта от прилагане на процедура по допълнителна алокация

За изследване на зависимостта между големината на инвестиционния капитал и необходимостта от прилагане на процедура по допълнителна алокация са конструирани портфейли, съставени от 7 финансови актива. Конструираният портфейл са съставени от едни и същи активи, но с различен по размер инвестиционен капитал. Важно е да се отбележи, че процедура по допълнителна алокация не се прилага, ако неизползваният капитал е под 0,5% от размера на началния капитал.

1) *Инвестиционен портфейл с капитал 1 000 лв.*

Първоначален портфейл

- Възвращаемостта на конструирания портфейл е $R_p = 1.78318324$.
- Рискът на конструирания портфейл е $s_p = 0.01519687$.
- Q на конструирания портфейл 0.81230537.
- Неизползваният капитал е 66.58 лв.

Име на актив	Цена на една акция	Брой закупени акции	Дял	Риск	Годишна норма на възвращаемост	Q
ЗЈУ	45	3	0.14538565	0.0241193	1.99408284	0.82919283
5ОРГ	90	1	0.14245006	0.00247399	1	0.81245
6А6	1.74	81	0.14244872	0.01237877	1.85524372	0.81244236
ВLКC	0.471	302	0.14244326	0.02221776	2.52884615	0.81241119
4ЕC	1.91	74	0.14243653	0.02754392	2.86830926	0.81237285
SO5	1.846	77	0.14243505	0.00874433	1.18275862	0.81236439
5BD	0.754	188	0.14240073	0.00871467	1.04851752	0.81216861

Таблица 5.13. Инвестиционен портфейл с капитал 1 000 лв.

Портфейл след първо прилагане на допълнителна алокация

Първоначално конструираният портфейл е с висок процент на неизползвания капитал (над 0,5%). За да се намали този процент се прилага процедура по допълнителна алокация.

Име на актив	Цена на една акция	Брой закупени акции	Дял	Риск	Годишна норма на възвращаемост	Q
ЗЈУ	45	3	0.135	0.0241193	1.99408284	0.82919283
5ОРГ	90	1	0.09	0.00247399	1	0.81245
6А6	1.74	87	0.15138	0.01237877	1.85524372	0.81244236
ВLКC	0.471	325	0.153075	0.02221776	2.52884615	0.81241119
4ЕC	1.91	79	0.15089	0.02754392	2.86830926	0.81237285
SO5	1.846	82	0.151372	0.00874433	1.18275862	0.81236439
5BD	0.754	202	0.152308	0.00871467	1.04851752	0.81216861

Таблица 5.14. Инвестиционен портфейл с капитал 1000 лв. след първо прилагане на процедура по допълнителна алокация

- Възвращаемостта на портфейла е $R_p = 1.79868443$.
- Рискът на портфейла е $s_p = 0.01556071$
- Q на портфейла е 0.81228124.
- Неизползваният капитал е 15.97 лв.

Рискът на този портфейл е по-малък от риска на първоначалния портфейл, възвращаемостта е по-голяма, но стойността Q е по-малка. Неизползваният капитал е намален до 15.97 лв., но все още е възможна процедура по допълнителна алокация.

Портфейл след второ прилагане на допълнителна алокация

Прилага се процедура по допълнителна алокация за втори път.

Име на актив	Цена на една акция	Брой закупени акции	Дял	Риск	Годишна норма на възвращаемост	Q
ЗЛУ	45	3	0.135	0.0241193	1.99408284	0.82919283
5ORG	90	1	0.09	0.00247399	1	0.81245
6А6	1.74	88	0.15312	0.01237877	1.85524372	0.81244236
BLKC	0.471	331	0.155901	0.02221776	2.52884615	0.81241119
4ЕС	1.91	80	0.1528	0.02754392	2.86830926	0.81237285
SO5	1.846	83	0.153218	0.00874433	1.18275862	0.81236439
5BD	0.754	206	0.155324	0.00871467	1.04851752	0.81216861

Таблица 5.15. Инвестиционен портфейл с капитал 1000 лв. след второ прилагане на процедура по допълнителна алокация

- Възвращаемостта на портфейла е $R_p = 1.81988325$.
- Рискът на портфейла е $s_p = 0.01574007$.
- Q на портфейла е 0.81228817.
- Неизползваният капитал е 4.64 лв.

2) *Инвестиционен портфейл с капитал 10 000 лв.*

Първоначален портфейл

- Възвращаемостта на портфейла е $R_p = 1.78318324$.
- Рискът на портфейла е $s_p = 0.01519687$.
- Q на портфейла е 0.81230537.
- Неизползваният капитал е 92.61 лв.

Име на актив	Цена на една акция	Брой закупени акции	Дял	Риск	Годишна норма на възвращаемост	Q
ЗЛУ	45	32	0.14538565	0.0241193	1.99408284	0.82919283
5ORG	90	15	0.14245006	0.00247399	1	0.81245
6А6	1.74	818	0.14244872	0.01237877	1.85524372	0.81244236
BLKC	0.471	3024	0.14244326	0.02221776	2.52884615	0.81241119
4ЕС	1.91	745	0.14243653	0.02754392	2.86830926	0.81237285
SO5	1.846	771	0.14243505	0.00874433	1.18275862	0.81236439
5BD	0.754	1888	0.14240073	0.00871467	1.04851752	0.81216861

Таблица 5.16. Инвестиционен портфейл с капитал 10000 лв.

Портфейл след първо прилагане на допълнителна алокация

Име на актив	Цена на една акция	Брой закупени акции	Дял	Риск	Годишна норма на възвращаемост	Q
ЗЛУ	45	32	0.144	0.0241193	1.99408284	0.82919283
5ORG	90	15	0.135	0.00247399	1	0.81245
6А6	1.74	825	0.14355	0.01237877	1.85524372	0.81244236
BLKC	0.471	3052	0.14375	0.02221776	2.52884615	0.81241119
4ЕС	1.91	751	0.14344	0.02754392	2.86830926	0.81237285
SO5	1.846	778	0.14362	0.00874433	1.18275862	0.81236439
5BD	0.754	1905	0.14364	0.00871467	1.04851752	0.81216861

Таблица 5.17. Инвестиционен портфейл с капитал 10 000 лв. след първо прилагане на процедура по допълнителна алокация

- Възвращаемостта на портфейла е $R_p = 1.78389321$.
- Рискът на портфейла е $s_p = 0.01523645$.
- Q на портфейла е 0.81230164.
- Неизползваният капитал е 30.04 лв.

3) *Инвестиционен портфейл с капитал 100 000 лв.*

Име на актив	Цена на една акция	Брой закупени акции	Дял	Риск	Годишна норма на възвращаемост	Q
ЗЛУ	45	323	0.14538565	0.0241193	1.99408284	0.82919283
5ORG	90	158	0.14245006	0.00247399	1	0.81245
6А6	1.74	8186	0.14244872	0.01237877	1.85524372	0.81244236
BLKC	0.471	30242	0.14244326	0.02221776	2.52884615	0.81241119
4ЕС	1.91	7457	0.14243653	0.02754392	2.86830926	0.81237285
SO5	1.846	7715	0.14243505	0.00874433	1.18275862	0.81236439
5BD	0.754	18886	0.14240073	0.00871467	1.04851752	0.81216861

Таблица 5.18. Инвестиционен портфейл с капитал 100 000 лв.

- Възвращаемостта на портфейла е $R_p = 1.78318324$.
- Рискът на портфейла е $s_p = 0.01519687$.
- Q на портфейла е 0.81230537.
- Неизползваният капитал е 32.57 лв.

След конструиране на инвестиционни портфейли с различен инвестиционен капитал се забелязва, че съществува зависимост между размера на инвестиционния капитал и неизползвания капитал (табл. 5.19).

Отношение капитал/максимална цена на акция	Неизползван капитал
1000/90 = 11.(1)	6.658 %
10000/90 = 111.(1)	0.9261 %
100000/90 = 1111.(1)	0.1597 %

Таблица 5.19. Зависимост между размера на инвестиционния капитал и неизползвания капитал.

Оказва се, че неизползваният капитал е значително по-голям, когато отношението инвестиционен капитал/максимална цена на акция в портфейла е малко. Количеството на неизползвания капитал намалява при увеличаване на инвестиционния капитал.

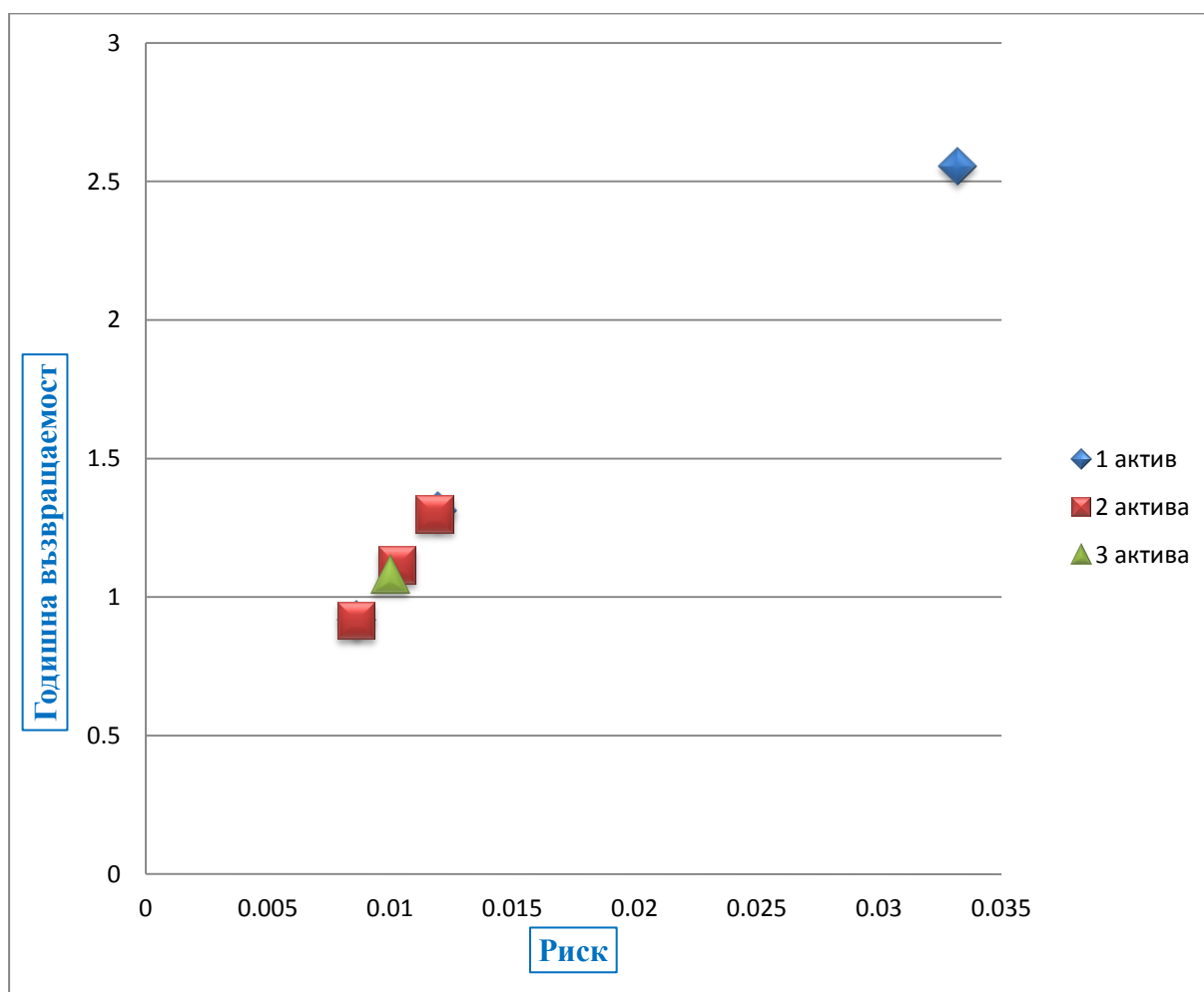
Прилагането на процедура по алокация зависи от процента на неизползвания капитал. В предложените тестове процентът, под който не се прилага допълнителна алокация, е 0,5%. Този процент се използва, за да се избегне правенето на разходи за трансакции в случай, че неизползваният капитал е малък.

5.4.3. Портфейли с фиксиран максимален брой активи

В приложението е предвидена и възможност за предварителен избор на максималния брой активи, от които да се конструира инвестиционния портфейл.

5.4.3.1. Портфейли с максимален брой три актива

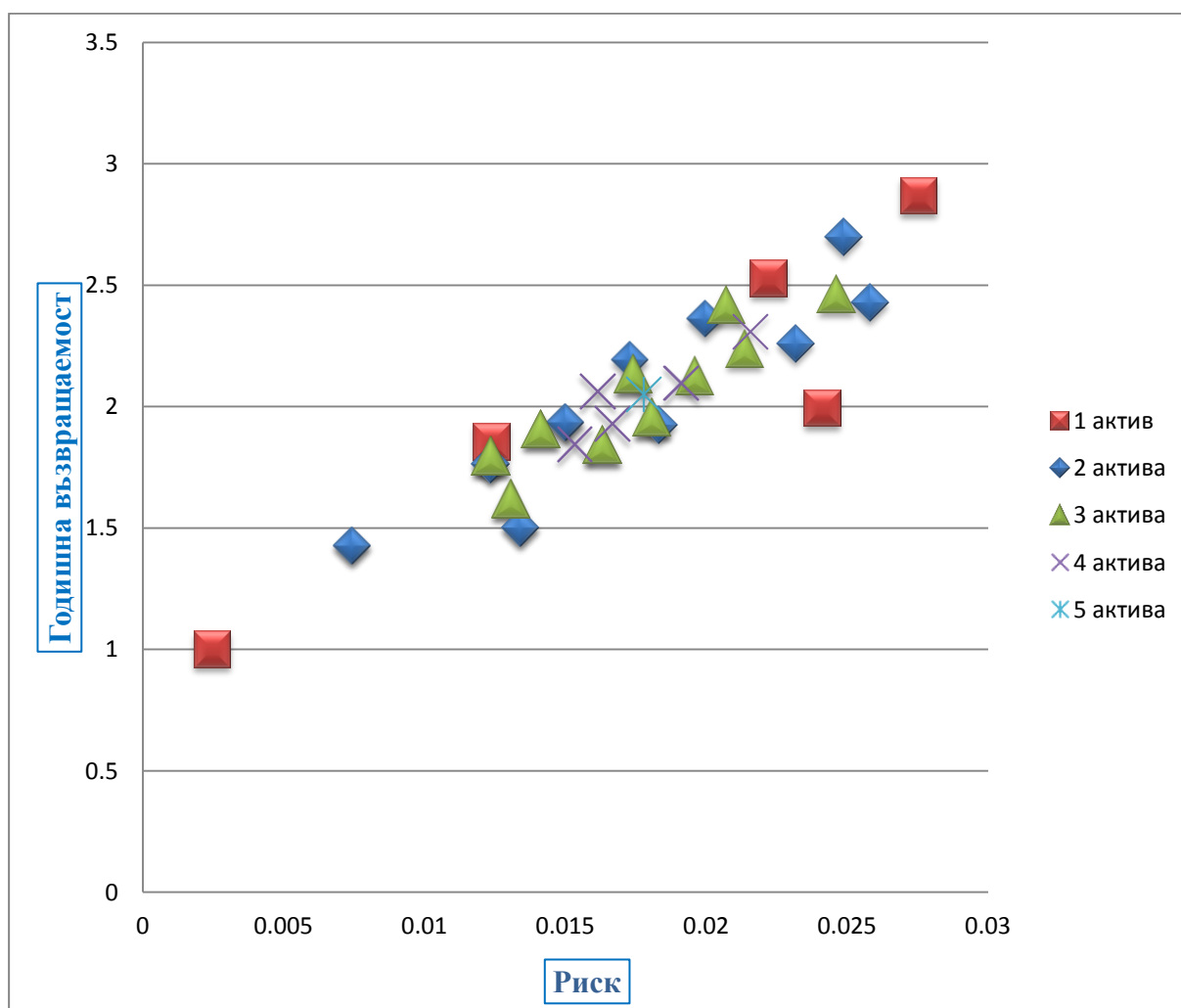
На 12.06.2012 е проведен тест на софтуерната система, като предварително е фиксиран първоначален капитал 100 лв. и три като максимален брой на активите в портфейлите. В портфейлите се включват трите актива с най-висока стойност на Q и това са активите с кодове 6A6, 57B и 5BU. Получените преди и след алокационната процедура портфейли са показани в Приложение 2. На фигура 5.9 е показано разположението на седемте получени портфейла след алокацията за изчерпване на капитала (инвестиционният риск е по абсцисната ос, а възвращаемостта е по ординатната ос).



Фигура 5.9. Инвестиционни портфейли с максимален брой три актива

5.4.3.1. Портфейли с максимален брой пет актива

Нека инвеститорият предпочита да инвестира в портфейл, конструиран най-много от пет актива. След въвеждане на този брой на 18.09.2012, софтуерната система намира най-добрите пет актива (5BN, 6AS, 6A6, 5BD, 4EC) в зависимост от Q и генерира всички възможни комбинации от тях. Получават се общо 31 инвестиционни портфейла (Приложение 2). Първоначалният капитал е 10 000 лв. Всеки един от тях е представен като точка на фигура 5.10.

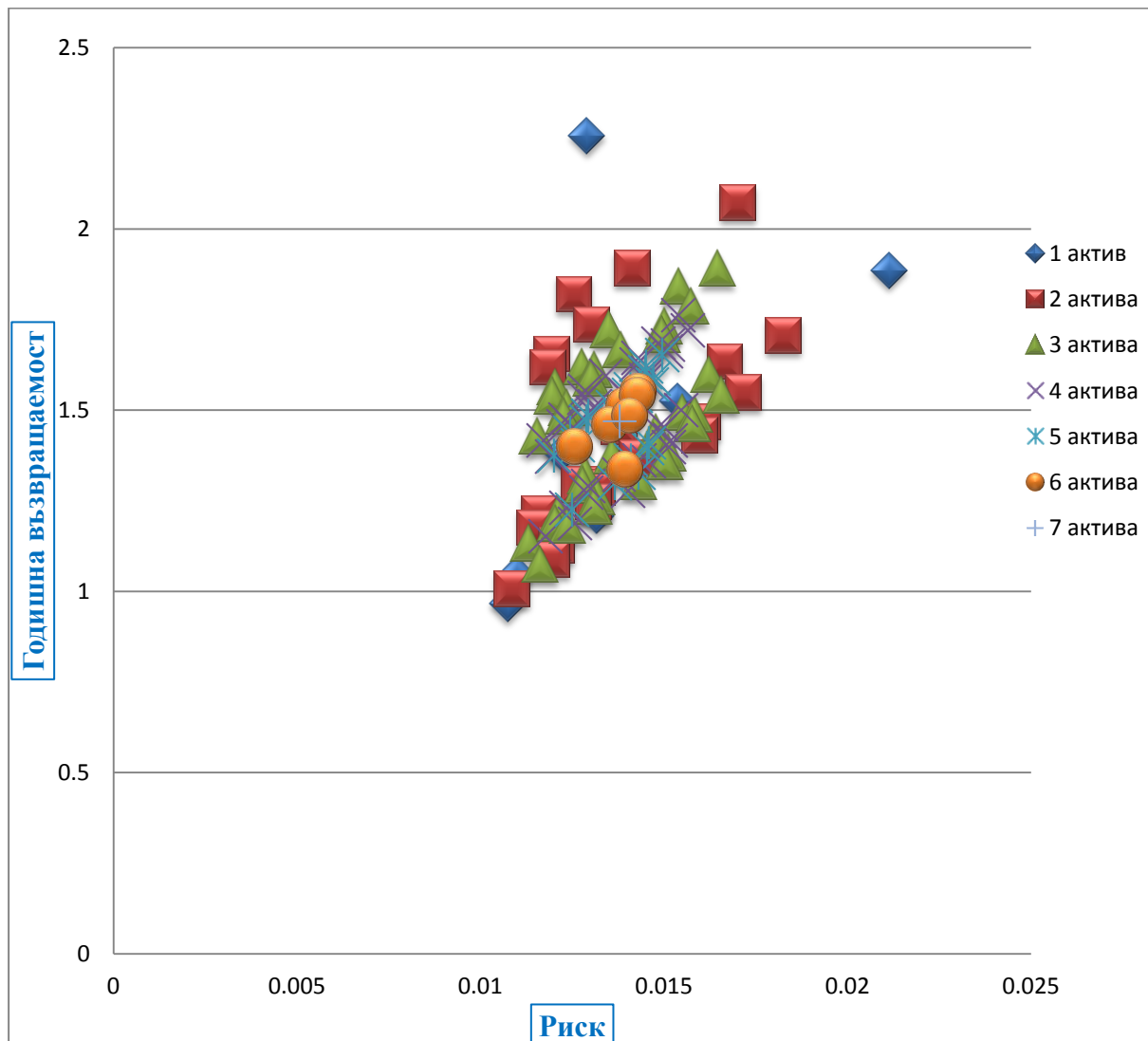


Фигура 5.10. Инвестиционни портфейли с максимален брой пет актива

5.4.3.1. Портфейли с максимален брой седем актива

На 21.11.2012 са конструирани всевъзможните портфейли при капитал 150 000 лв. и максимален брой на активите седем. От най-добрите активи (5BN, 6AS, 6A6, 5BD, 4EC, 3JR, 5ALB) са генерирани всички възможни комбинации, при което са получени

127 портфейла. Поради големия им брой само една част от тях са показани в Приложение 2. Всички портфейли са изобразени на фигура 5.11.



Фигура 5.11. Инвестиционни портфейли с максимален брой седем актива

Всички портфейли са подредени почти около права линия и това не е случайно. Базова постановка във финансовата наука е, че възвращаемостта е право пропорционална на поетия риск. Инвеститорът избира един (или повече) от тези портфейли в зависимост от своето предпочитание, в зависимост както от желаната възвращаемост и риска, така и според броя на активите в портфейла.

Извод. От проведените тестове (част от които са включени в тази глава) може да се направи извод, че разработеният математически модел и програмната реализация на системата работят по очаквания начин. Моделът не определя най-добрия портфейл, а дава възможност на инвеститора да вземе своето решение на базата на информацията за финансовия пазар, предоставена от модела.

Заклучение - основни резултати

Дисертационният труд представлява интердисциплинарно изследване за моделиране на процеса на управление на финансови инвестиции в реално време. Изследването е основано на подходите, концепциите и средствата на софт компютинг с акцент върху размитата логика и размитите системи.

Основната цел, поставена в т. 1.6 е постигната, а именно

създадени, реализирани и тествани са модели, основани на средствата на софт компютинг, за оценяване и управление на финансови инвестиции,

чрез създаване на самостоятелна софтуерна система FSSAM, работеща в режим реално време, която

- събира данни за цени на финансови активи автономно;
- отчита проблема с липсващите данни;
- дава оценки на характеристиките на активи в случай на ежедневни наблюдения чрез средногеометрична на логаритмуваните възвращаемости;
- анализира данните със средствата на размитата логика за получаване на оценка на активи;
- конструира портфейли при предварително фиксирани параметри: първоначален капитал и максимален брой използвани активи.

Резултатите от проведените тестове на софтуерната система FSSAM доказват нейната ефективност и надеждност.

Бъдещото развитие на системата е възможно основно в три направления:

- тестване на системата FSSAM с финансови данни от дуги източници на борсова информация;
- изследване на системата от размити правила за оптималност;
- създаване на хибридни адаптивни системи, използващи модела FLQM.

Основен недостатък на размитите системи е липсата на адаптивност към промени в околната среда, което може да бъде частично компенсирано с невронна мрежа за коригиране на параметрите на системата или с генетичен алгоритъм за оптимизиране броя на правилата и размитите променливи. Създаването на надеждна система, в която са включени средства и от трите области е истинско предизвикателство за всички изследователи, занимаващите се със софт компютинг.

Друг много интересен аспект в областта софт компютинг е систематично изследване на стабилността и скалираемостта на създадените вече хибридни системи.

Важни направления в развитието на софт компютинг са и използването на паралелни и грид компютърни системи за реализирането на хибридните системи.

Основните резултати имат научно-приложен характер и могат накратко да се систематизират така:

1. създаден и реализиран е модел за автономно събиране и анализиране на финансови данни;
2. създаден и реализиран е модел FLQM за използване на размитата логика за подпомагане на вземане на решения, като са използвани характеристики на финансови активи, основани на средно геометрична стойност на логаритмичните възвращаемости, получени при дневни наблюдения;
3. изградена е работеща архитектура на интелигентна система;
4. създаден и реализиран е модел за конструиране на инвестиционен портфейл без къси продажби с ограничени финансови ресурси, използващ резултатите на модела FLQM;
5. създадена, реализирана и тествана с реални данни е самостоятелно работеща софтуерна система FSSAM за подпомагане на вземане на решения при управление на финансови активи.

Получените резултати показват, че средствата на софт компютинга, и в частност средствата на размитата логика са подходящо избрани за постигане на поставените цел и задачите на настоящия дисертационен труд.

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. 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:

БИБЛИОГРАФИЯ

- [1] Георгиев И., *Основи на инвестирането*. София: Университетско издателство "Стопанство", 1999.
- [2] Георгиева П., Оценка на характеристиките на инвестиционен портфейл, конструиран със средствата на софт компютинг. *Съвременни управленски практики VI*, pp. 299-303, 2009.
- [3] Георгиева П., Софт компютинг като направление в изкуствения интелект. *Бургаски свободен университет, Годишник*, vol. том XIX, pp. 317-325, 2008.
- [4] Йорданов Й., *Финансови инвестиции*. Варна: Лотос 23, 2009.
- [5] Калан Р., *Основные концепции нейронных сетей*. Москва: Издательский дом "Вильямс", 2003.
- [6] Костенаров К., Модификации на модела за оценка на капиталови активи, приложими на формиращи се капиталови пазари. НБУ, София, PhD, 2012.
- [7] Оссовский Ст., *Нейронные сети для обработки информации*. Москва: Финансы и статистика, 2004.
- [8] Након С., *Програмиране на .NET Framework*.: Фабер, 2005.
- [9] Пауков Д., *Прогнозирование с помощью искусственных нейронных сетей*. Москва, 2001.
- [10] Пътев П., Н. Канарян, *Управление на портфейла*. Велико Търново: Абагар, 2008.
- [11] Радева И., Приложение на теорията на развити множества в задачи за избор при икономическа клъстеризация. *Корпоративните финанси на формиращите се пазари*. София: Нов български университет, 2012, pp. 186-217.
- [12] Стоилов Т., З. Иванова, К.Стоилова, *Портфейлна оптимизация - информационна услуга в интернет*. София: Академично издателство "М. Дринов", 2005.
- [13] Стоянов Ст., Стоянова-Дойчева А., Трендафилова М., Дойчев Е., *Софтуерни технологии*.: Университетско издателство "Паисий Хилендарски", 2006.
- [14] Цончев Р., Костенаров Кр., Приложими ли са моделът за оценка на капиталовите активи и неговите модификации на формиращите се пазари? *Корпоративните финанси на формиращите се пазари*. София: Нов български университет, 2012, pp. 56-85.
- [15] Abdi H., D. Valentin, B. Edelman, *Neural Networks*. Thousand Oaks: Sage Publications, 1999.
- [16] Altrock C., *Fuzzy Logic & NeuroFuzzy Applications in Business and Finance*. Prentice Hall, 1995.
- [17] Angelova V., "Investigations in the Area of Soft Computing," *CIT*, vol. 9, no. 1, pp. 18-24, 2009.
- [18] Atanassov Kr., Intuitionistic Fuzzy Set. *Fuzzy Sets and Systems*, vol. 20, no. 1, pp. 87-96, August 1986.
- [19] Azoff M., *Neural Network Time Series Forecasting of Financial Markets*. John Wiley&Sons, 1994.
- [20] Bojadziev G., M. Bojadziev, *Fuzzy Logic for Business, Finance and Management*. World Scientific, 1997.
- [21] Brown M., C. Harris, *An Introduction to Fuzzy and Neurofuzzy Systems, Neurofuzzy Adaptive Modelling and Control*. New York: Prentice Hall, 1996.
- [22] Campbell J., A. Lo, C. MacKinlay, *The Econometrics of Financial Markets*. Princeton University Press, 1996.
- [23] Carlsson C., R. Fuller, S. Giove, Optimization under Fuzzy Rule Constraints. in *Eurofuse-SIC'99*, Budapest, 1999, pp. 184-187.
- [24] Castillo O., P. Melin, Evolutionary Design and Applications of Hybrid Intelligent Systems. *Int. J. of Innovative Computing and Applications*, vol. 1, no. 1, pp. 48 - 62, 2007.
- [25] Castillo O.; P. Melin, *Hybrid Intelligent Systems for Pattern Recognition Using Soft Computing: An*

- Evolutionary Approach for Neural Networks and Fuzzy Systems*. Springer, 2005.
- [26] Chin-Shien L., A. Khan, H. Chi-Chung, Can the Neuro Fuzzy Model Predict Stock Indexes Better than its Rivals? Providence University, Discussion papers 2002.
- [27] Craig W., C. French, The Treynor Capital Asset Pricing Model. *Journal of Investment Management*, vol. 1, no. 2, pp. 60-72, 2003.
- [28] Elton E., M. Gruber, T. Urich, Are Betas Best? *The Journal of Finance*, vol. 33, no. 5, pp. 1375-1384, 1978.
- [29] Elton E., M. Gruber; M.Padberg, Simple Criteria For Optimal Portfolio Selection. *The Journal of Finance*, vol. 31, no. 5, pp. 1341-1357, 1976.
- [30] Fakhereddine O., Cl. De Silva, *Soft Computing and Intelligent Systems Design*. Addison Wesley, 2004.
- [31] Fang Y., etc., *Fuzzy Portfolio Optimization*. Springer, 2008.
- [32] Freeman J., D. Skapura, *Neural Networks - Algorithms, Applications and Programming Techniques*. Addison Wesley, 1991.
- [33] Fuller R., C. Carlsson, Multiple Criteria Decision Making: Recent developments. *Fuzzy Sets and Systems*, vol. 78, no. 2, pp. 139–153, March 1996 1996.
- [34] Georgieva P., Distributions of Financial Assets Characteristics. *XVI International Symposium SIELA*, vol. 1, 2009, pp. 101-106.
- [35] Georgieva P., I. Popchev, Application of Q-measure in Real Time Fuzzy System for Managing Financial Assets," *IJSC*, vol. 3, no. 4, pp. 21-38, 2012.
- [36] Georgieva P., I. Popchev, "Cardinality Problem in Portfolio Selection. *ICANNGA'13, LNCS 7824*, pp. 208–217, Lausanna, 2013.
- [37] Georgieva P., I. Popchev, Fuzzy Q-measure Model for Managing Financial Investments. *Compus Rendus Acad. Bulg. Sci.* , vol. 66, 2013.
- [38] Georgieva P., N. Chaney, A. Andonov, K. Dudinov, Managing Real Time Financial Data. *XVIII International Symposium SIELA*, vol. 2, 2012, pp. 379-386.
- [39] Gomez-Skarmeta A., F. Jimenez, J. Ibanez, A Fuzzy Evolutionary Algorithm for Variable Identification in Data Mining. in *CICYT*, Valencia, 2002.
- [40] Gradojevic N., J.Yang, T. Gravelle, Neuro-Fuzzy Decision-Making in Foreign Exchange Trading and Other Applications. in *36 Annual Meeting of the Canadian Economics Association Conference*, Calgary, 2001.
- [41] Gupta M., L.Jin, N.Homma, *Static and Dynamic Neural Networks from Fundamentals to Advanced Theory*. Wiley & Sons, 2003.
- [42] Hanke J., A. Reitsch, *Understanding Business Statistics*. Homewood, 1990.
- [43] Haugen R., *Modern Investment Theory*. NJ: Englewood Cliffs, Prentice-Hall, 2000.
- [44] Haupt R., S. Haupt, *Practical Genetic algorithms*. Wiley, 1998.
- [45] Hoffmann F., M. Köppen, F. Klawonn, R. Roy, *Soft Computing: Methodologies and Applications*. Series: Advances in Soft Computing: Vol. 32, 2005.
- [46] Holton Gl., "Defining Risk," *Financial Analyst Journal*, CFA Institute, vol. 60, 2004.
- [47] Jang R., C. Sun, E. Mizutani, *Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence*. Prentice-Hall, 1997.
- [48] Judge, G., R. C. Hill, W. E. Griffiths, *Introduction to the Theory and Practice of Econometrics*. New York: Wiley , 1988.
- [49] Kasabov N., *Foundations of Neural Networks, Fuzzy Systems and Knowledge Engineering*. England: Massachusetts Institute of Technology, 1996.
- [50] Konar A., *Artificial Intelligence and Soft Computing: behavioral and cognitive modeling of the human brain*. London: CRC Press LLC, 2000.
- [51] Konno H., H. Shirakawa; H. Yamazaki, A mean-absolute deviation-skewness portfolio optimization

- model. *Annals of Operation Research*, vol. 45, pp. 205-220, 1993.
- [52] Konno H., H. Yamazaki, Meanabsolute deviation portfolio optimization model and its applications to Tokio stock market. *Management Science*, vol. 37, no. 5, pp. 519–531, 1991.
- [53] Konno H., K. Suzuki, A meanvariance-skewness portfolio optimization model. *Journal of the Operations Research Society of Japan*, vol. 38, no. 2, pp. 173–187, 1995.
- [54] Lakhmi C., N. Martin, *Fusion of Neural Networks, Fuzzy Systems and Genetic Algorithms: Industrial Applications*. CRC Press LLC, 1998.
- [55] Laplante Ph., *Real-Time Systems Design and Analysis*. IEEE PRESS, Ed.: John Wiley, 2004.
- [56] Lintner J., The valuation of risk assets and the selection of risky investments in stock portfolios and capital budgets. *Review of Economics and Statistics*, vol. 47, pp. 13-37, 1965.
- [57] Mamdani E., S. Assilian, An experiment in linguistic synthesis with a fuzzy logic controller. *International Journal of Man-Machine Studies*, vol. 7, no. 1, pp. 1–13, 1975.
- [58] Man K., K. Tang, S. Kwong, *Genetic Algorithms: Concepts and Designs*. Springer, 1999.
- [59] Mansini R., M. Speranza, Heuristic algorithms for the portfolio selection problem with minimum transaction lots. *European Journal of Operational Research*, vol. 114, pp. 219-233, 1999.
- [60] Markowitz H., Portfolio Selection. *The Journal of Finance*, vol. 7, no. 1, pp. 77-91, 1952.
- [61] Markowitz H., *Portfolio Selection: Efficient Diversification of Investments*. New York: John Wiley and Sons, 1959.
- [62] Markowitz H., P.Todd, G.Xu, Y.Yamane, Computation of mean-semivariance efficient sets by the critical line algorithm *Annals of Operations Research*, vol. 45, pp. 307–317, 1993.
- [63] McCarthy J., *Artificial Intelligence, Logic and Formalizing Common Sense*.: Stanford, 1990.
- [64] Melin P., O. Castillo, E.G. Ramírez, J. Kacprzyk, W. Pedrycz, *Analysis and Design of Intelligent Systems Using Soft Computing Techniques*. Series: Advances in Soft Computing, Vol. 41: Springer, 2007.
- [65] Nawrochi D., K. Staples, A Customized LPM Risk Measure for Portfolio Analysis. *Applied Economics*, vol. 21, pp. 205-218, 1989.
- [66] Nawrocki D., A Comparison of Risk Measures When Used In A Simple Portfolio Selection Heuristic. *Journal of Business Finance and Accounting*, vol. 10, no. 2, pp. 183-194, 1983.
- [67] Nawrocki D., Optimal Algorithms and Lower Partial Moment: ExPost Results. *Applied Economics*, vol. 23, pp. 465-470, 1991.
- [68] Nawrocki D., Portfolio Analysis with a Large Universe of Assets. *Applied Economics*, vol. 28, pp. 1191-1198, 1996.
- [69] Nawrocki D., Tailoring Asset Allocation to the Individual Investor. *International Review of Economics and Business*, vol. 37, pp. 977-990, 1990.
- [70] Peeva K., Y. Kyosev, Fuzzy Relational Calculus: Theory, applications and Software. in *Advances in Fuzzy Systems- Applications and Theory*, Lotfi Zadeh, Ed.: World Scientific Publishing Company, 2004.
- [71] Peneva V., I. Popchev, Fuzzy Criteria Importance with Weighting Functions. *Comp. Rend. Acad. Bulg. Sci.*, vol. 61, no. 3, pp. 293-300, 2008.
- [72] Peneva V., I. Popchev, Multicriteria Decision Making Based on Fuzzy Relations. *CIT, BAS*, vol. 8, no. 4, 2008.
- [73] Peneva V., I. Popchev, Properties of the aggregation operators related with fuzzy relations. *Fuzzy sets and Systems*, vol. 139, no. 3, pp. 615-633, 2003.
- [74] Peneva V., Popchev I., Fuzzy multi-criteria decision making algorithms. *Compus Rendus Acad. Bulg. Sci.*, vol. 63, no. 7, 2010.
- [75] Popchev I., P. Georgieva, A Fuzzy Approach for Solving Multicriteria Investment Problems. *EIAE 07*, New York, 2008, pp. 427-431.
- [76] Powell J. Notes On Nonparametric Regression Estimation. Department of Economics http://emlab.berkeley.edu/users/powell/e241a_sp10/nrnotes.pdf [Online 2010], University of

- California, Berkeley.
- [77] Ramaswamy Sr., Portfolio selection using fuzzy decision theory. Bank for International Settlements, Monetary and Economic Dept, Basel, 59 of BIS working papers 1998.
- [78] Ramík J., *Soft Computing: Overview and Recent Developments in Fuzzy Optimization*. Listopad: Ostravská univerzita, 2001.
- [79] Schaerf A., Local Search Techniques for Constrained Portfolio Selection Problems. *Computational Economics*, vol. 20, pp. 177–190, 2002.
- [80] Shapiro A., *Soft Computing and Financial Engineering*. University Park, 2002.
- [81] Sharp W., Capital asset prices: A theory of market equilibrium under conditions of risk. *Journal of Finance*, vol. 19, no. 3, pp. 425-442, 1964.
- [82] Sharp W., *Investments*. Prentice-Hall: Englewood Cliffs, NJ, 1985.
- [83] Sharp W., The Sharp Ratio. *The Journal of Portfolio Management*, pp. 49-58, Fall 1994.
- [84] Silva L., G. Torres, L.Reis, J.Haddad, Application of Fuzzy Optimization in Energy Saving. *Rev. Ciênc. Exatas*, vol. 5-8, pp. 21- 35, 2002.
- [85] Simaan Y., Estimation risk in portfolio selection: the mean variance model versus the mean absolute deviation model. *Management Science*, vol. 43, no. 10, pp. 1437–1446, 1997.
- [86] Spears W., K. De Jong, An Overview of Evolutionary Computation. in *ECML*, Vienna, 1993.
- [87] Sugeno M., *Industrial applications of fuzzy control*. Elsevier Science Pub., 1985.
- [88] Tobin J., Liquidity preference as behavior towards risk. *The Review of Economic Studies*, vol. 25, pp. 65-86, 1958.
- [89] Treynor J., Towards a theory of market value of risky assets. unpublished manuscript, 1961.
- [90] Tseng C., P. Gmytrasiewicz, Real Time Decision Support System for Portfolio Management. in *35th Annual HICSS*, 2002, pp. 78-87.
- [91] Xu H., G.Vukovich, Fuzzy Evolutionary Algorithms and Automatic Robot Trajectory Generation. *IEEE World Congress on Computational Intelligence*, vol. 2, Orlando, 1994, pp. 595- 600.
- [92] Yoshimoto A., The mean-variance approach to portfolio optimization subject to transaction costs. *Journal of the Operations Research*, vol. 39, no. 1, pp. 100-117, 1996.
- [93] Zadeh L., A Theory of Approximate Reasoning. *Machine Intelligence*, vol. 9, pp. 149-194, 1979.
- [94] Zadeh L., Fuzzy Logic, Neural Networks and Soft Computing. *Communications on the ACM*, vol. 37, no. 3, pp. 77-84, 1994.
- [95] Zadeh L., "Fuzzy Sets," *Information and Control* 8, vol. 8, pp. 338-353, 1965.
- [96] Zadeh L., R. Bellman, Decision-making in A Fuzzy Environment. *Management Science*, vol. 17, no. 4, pp. 141-164, 1970.
- [97] Zadeh, L., Outline of a New Approach to the Analysis of Complex Systems and Decision Processes. *IEEE Trans. on Systems, Man and Cybernetics*, vol. SMC-3, no. 1, pp. 28-44, 1973.
- [98] Zilouchian A. , M. Jamshidi, Eds., *Intelligent Control Systems Using Soft Computing Methodologies*. CRC Press, 2001.
- [99] Zimmermann J., Fuzzy Programming and Linear Programming with Several Objective Functions. *Fuzzy Sets and Systems*, vol. 1, no. 1, pp. 45-55, 1978.
- [100] www-bisc.cs.berkeley.edu
- [101] www.bse-sofia.bg
- [102] www.money.net
- [103] www.personalstockstreamer.com
- [104] www.wikinvest.com/account/portfolio/regx/start

Публикации по темата на дисертацията

1. Popchev I., P. Georgieva. *A Fuzzy Approach for Solving Multicriteria Investment Problems*. In: EIAE 07, New York, pp. 427-431, 2008, ISBN:978-1-4020-8738-7.
2. Георгиева П. *Софт компютинг като направление в изкуствения интелект*. В: Годишник на Бургаски свободен университет, , том XIX, pp. 317-325, 2008, ISSN: 1311-221-X.
3. Георгиева П. *Оценка на характеристиките на инвестиционен портфейл, конструиран със средствата на софт компютинг*. В: Съвременни управленски практики VI, Бургас, pp. 299-304, 2009, ISSN: 1313-8758.
4. Georgieva P. *Distributions of Financial Assets Characteristics*. In: XVI International Symposium SIELA, vol. 1, pp. 101-106, 2009, ISBN: 978-954-323-530-8.
5. Georgieva P., N. Chaney, A. Andonov, K. Dudinov. *Managing Real Time Financial Data*. In: XVIII International Symposium SIELA, vol. 2, pp. 379-386, 2012, ISBN: 978-954-323-530-8.
6. Georgieva P., I. Popchev. *Application of Q-measure in Real Time Fuzzy System for Managing Financial Assets*. - International Journal on Soft Computing, vol. 3, no. 4, pp. 21-38, 2012, ISSN: 2229-7103.
7. Georgieva P., I. Popchev. *Fuzzy Q-measure Model for Managing Financial Investments*. - Compus Rendus Acad. Bulg. Sci. , Vol. 66, 2013, ISSN 1310-1331.
8. Georgieva P., I. Popchev. *Cardinality Problem in Portfolio Selection*. - M. Tomassini et al. (Eds.): ICANNGA'2013, LNCS 7824, pp. 208–217, Springer-Verlag Berlin Heidelberg 2013.

Участие в проекти по темата на дисертацията

1. Ръководител на научноизследователски проект № 5/ 2006 на тема „Конструиране на инвестиционен портфейл със средствата на размита логика и паралелно програмиране”, финансиран по НИД, БСУ.
2. Участник в научноизследователски проект № 3/ 2008 на тема „Изследване на инструментални средства за разработка на програмно осигуряване за

вградени системи при управление на обекти в реално време”, финансиран по НИД, БСУ.

3. Ръководител на научноизследователски проект № 7/ 2008 на тема „Създаване и изследване на универсална библиотека за програмна реализация на размита логика”, финансиран по НИД, БСУ.

4. Участник в проект BG051P0001 - 3.3.04/40 "Изграждане на висококвалифицирани млади изследователи по съвременни информационни технологии за оптимизация, разпознаване на образи и подпомагане вземането на решения" на ИИТ към БАН през 2009-2011 по договор с MOMH по оперативна програма "Развитие на човешките ресурси".

БЛАГОДАРНОСТИ

Благодаря на акад. Иван Попчев, научен ръководител на изследването, за подкрепата и усилията, вложени в продължение на години.

Благодаря за финансовата подкрепа, получена по Проект BG051P0001 - 3.3.04/40 "Изграждане на висококвалифицирани млади изследователи по съвременни информационни технологии за оптимизация, разпознаване на образи и подпомагане вземането на решения" на ИИТ към БАН по договор с MOMH по оперативна програма "Развитие на човешките ресурси".

Благодарна съм на моите студенти: Мартин Петров, Иван Петков, Димитър Тодоров, Иван Миховски, Семо Кискинов, Кръстьо Дудинов, Андон Андонов и Николай Чанев за създаването на част от сорсовете и провеждането на част от тестове.

Приложение

Портфейли с фиксиран максимален брой активи ТРИ

Top of Form

Date :

Bottom of Form

Общ брой портфейли : 10⁸

Портфейли с един актив

PortfolioID	Date	Return	Risk	QMeasure	Capital	UsedCapital		
1	2012-12-06	1.3120706575073593	0.011963235573162563	0.8116382555285433	100	99		
EquityID	Code	Broi Akcii	Cena	Return	Risk	Annual Return	QMeasure	QEquity
12	6A6	44	2.25	0.00083160740685542	0.011963235573162563	1.3120706575073593	109.67523371777125	0.8116382555285433

PortfolioID	Date	Return	Risk	QMeasure	Capital	UsedCapital		
2	2012-12-06	0.9166666666666667	0.0086448370892907	0.8107166770708731	100	99.75		
EquityID	Code	Broi Akcii	Cena	Return	Risk	Annual Return	QMeasure	QEquity
11	5BU	190	0.525	- 0.00023673005854366478	0.008644837089290768	0.9166666666666667	106.03631476204852	0.8107166770708731

PortfolioID	Date	Return	Risk	QMeasure	Capital	UsedCapital
-------------	------	--------	------	----------	---------	-------------

⁸ Всички портфейли, конструирани с най-много три актива са 7, но приложението показва и портфейлите преди процедурата по алокация (пр. портфейли 5 и 6, портфейли 7 и 8, портфейли 9 и 10)

3	2012-12-06	2.554000000000001	0.03320862664619381	0.8048001943210146	100	75.9		
EquityID	Code	Broi Akcii	Cena	Return	Risk	Annual Return	QMeasure	QEquity
8	57B	1	75.9	0.0035075099073398	0.0332086266461938	2.554000000000000	76.90772723637237	0.8048001943210146

Порфейли с два актива

PortfolioID	Date	Return	Risk	QMeasure	Capital	UsedCapital		
4	2012-12-06	1.1144809666736095	0.010304978839088309	0.8114514548113378	100	99.375		
EquityID	Code	Broi Akcii	Cena	Return	Risk	Annual Return	QMeasure	QEquity
12	6A6	22	2.25	0.000831607406855424	0.0119632355731625	1.31207065750735	109.675233717771	0.81163825552854
11	5BU	95	0.525	-0.0002367300585436	0.00864483708929076	0.916666666666666	106.036314762048	0.810716677070873

PortfolioID	Date	Return	Risk	QMeasure	Capital	UsedCapital		
5	2012-12-06	1.9304084459112865	0.02254099364715099	0.8208520911902006	100	49.5		
EquityID	Code	Broi Akcii	Cena	Return	Risk	Annual Return	QMeasure	QEquity
12	6A6	22	2.25	0.000831607406855424	0.011963235573162563	1.3120706575073593	109.67523371777125	0.8116382555285433
8	57B	0	75.9	0.003507509907339819	0.03320862664619381	2.554000000000001	76.90772723637237	0.8048001943210146

PortfolioID	Date	Return	Risk	QMeasure	Capital	UsedCapital		
6	2012-12-06	1.29894995093 22857	0.01184360321 7430937	0.81161742548 40498	100	99		
EquityID	Code	Broi Akcii	Cena	Return	Risk	Annual Return	QMeasure	QEquity
12	6A6	44	2.25	0.00083160 740685542 4	0.01196323 557316256	1.31207065 75073593	109.675233 71777125	0.81163825 55285433
8	57B	0	75.9	0.00350750 990733981 9	0.03320862 664619381	2.55400000 0000001	76.9077272 3637237	0.80480019 43210146

PortfolioID	Date	Return	Risk	QMeasure	Capital	UsedCapital		
7	2012-12-06	1.73233514291 082	0.02088175207 218641	0.79562673613 989	100	49.875		
EquityID	Code	Broi Akcii	Cena	Return	Risk	Annual Return	QMeasure	QEquity
11	5BU	95	0.525	- 0.00023673 005854366	0.00864483 70892907	0.91666666 666666	106.036314 762048	0.81071667 707087
8	57B	0	75.9	0.00350750 990733981 9	0.03320862 664619381	2.55400000 0000001	76.9077272 36372	0.80480019 432101

PortfolioID	Date	Return	Risk	QMeasure	Capital	UsedCapital		
8	2012-12-06	0.91437500000 00007	0.00862322499 656754	0.81070886028 52772	100	99.75		
EquityID	Code	Broi Akcii	Cena	Return	Risk	Annual Return	QMeasure	QEquity

11	5BU	190	0.525	- 0.00023673 005854366 4	0.00864483 70892907	0.91666666 666666	106.036314 762048	0.81071667 707087
8	57B	0	75.9	0.00350750 990733981 9	0.03320862 664619381	2.55400000 0000001	76.9077272 3637237	0.80480019 432101

Порфейли с три актива

PortfolioID	Date	Return	Risk	QMeasure	Capital	UsedCapital		
9	2012-12-06	1.59179911782 93676	0.01789940904 31462	0.80454932452 10112	100	64.575		
EquityID	Code	Broi Akcii	Cena	Return	Risk	Annual Return	QMeasure	QEquity
12	6A6	14	2.25	0.00083160 740685542	0.01196323 55731625	1.31207065 750735	109.675233 717771	0.81163825 552854
11	5BU	63	0.525	- 0.00023673 005854366	0.00864483 70892907	0.91666666 666666	106.036314 762048	0.81071667 707087
8	57B	0	75.9	0.00350750 990733981 9	0.03320862 664619381	2.55400000 0000001	76.9077272 3637237	0.80480019 432101

PortfolioID	Date	Return	Risk	QMeasure	Capital	UsedCapital		
10	2012-12-06	1.08195338567 22274	0.01000962670 132185	0.81145665981 59134	100	97.65		
EquityID	Code	Broi Akcii	Cena	Return	Risk	Annual Return	QMeasure	QEquity
12	6A6	21	2.25	0.00083160 740685542	0.01196323 55731625	1.31207065 750735	109.675233 717771	0.81163825 552854
11	5BU	96	0.525	- 0.00023673 005854366	0.00864483 70892907	0.91666666 666666	106.036314 762048	0.81071667 707087

8	57B	0	75.9	0.00350750 990733981 9	0.03320862 664619381	2.55400000 0000001	76.9077272 3637237	0.80480019 432101
---	-----	---	------	------------------------------	-------------------------	-----------------------	-----------------------	----------------------

Портьейли с фиксиран максимален брой активи ПЕТ

Top of Form

Date :

Bottom of Form

Общ брой портьейли : 31

Портьейли с един актив

PortfolioID	Date	Return	Risk	QMeasure	Capital	UsedCapital		
1	2012-09-18	2.2572944297	0.01290038286 4	0.812449528	10000	9999.15		
Equity ID	Code	Broi Akcii	Cena	Return	Risk	Annual Return	QMeasure	QEquity
15	5BN	1869	5.35	0.00294	0.01290038 2	2.25729442 970	174.978871 0449	0.81244952

PortfolioID	Date	Return	Risk	QMeasure	Capital	UsedCapital		
2	2012-09-18	1.38076923076 92307	0.01220874118 1006254	0.81196512352 11458	10000	9997.16		
EquityID	Code	Broi Akcii	Cena	Return	Risk	Annual Return	QMeasure	QEquity
26	6AS	3412	2.93	0.00100412 5865559	0.01220874 11810062	1.38076923 07692	113.096773 06595394	0.81196512 352114

PortfolioID	Date	Return	Risk	QMeasure	Capital	UsedCapital
3	2012-09-18	1.525305410 122165	0.015369974 343812468	0.810480673 2511	10000	9999

EquityID	Code	Broi Akcii	Cena	Return	Risk	Annual Return	QMeasure	QEquity
40	6A6	4950	2.02	0.00135592 256312942	0.01536997 43438124	1.52530541 0122165	99.2392944 84327	0.81048067 325110

PortfolioID	Date	Return	Risk	QMeasure	Capital	UsedCapital
4	2012-09-18	1.886010362 69430	0.021134839 917027214	0.809590525 57267	10000	10000

EquityID	Code	Broi Akcii	Cena	Return	Risk	Annual Return	QMeasure	QEquity
38	5BD	10000	1	0.00217454 39408181	0.02113483 99170272	1.88601036 26943017	89.2370308 97728	0.80959052 5572678

PortfolioID	Date	Return	Risk	QMeasure	Capital	UsedCapital
5	2012-09-18	1.04663212435 2329	0.010989096866 2783	0.806890565281 0485	10000	9999.92

EquityID	Code	Broi Akcii	Cena	Return	Risk	Annual Return	QMeasure	QEquity
8	4EC	5102	1.96	0.000129617397 6943	0.010989096866 2783	1.04663212435232 9	95.24277900 9809	0.8068905652 8104

Порфейли с два актива

PortfolioID	Date	Return	Risk	QMeasure	Capital	UsedCapital
6	2012-09-18	1.81916252124 96003	0.01255466514 748475	0.81243817648 8536	10000	9992.55

EquityID	Code	Broi Akcii	Cena	Return	Risk	Annual Return	QMeasure	QEquity
15	5BN	934	5.35	0.0029413 576	0.0129003 828	2.2572944 2	174.97887 1	0.8124495
26	6AS	1705	2.93	0.0010041 2586	0.0122087 411	1.3807692 30	113.09677 3	0.8119651

PortfolioID	Date	Return	Risk	QMeasure	Capital	UsedCapital		
7	2012-09-18	1.89174392548 639	0.01413368061 4045352	0.81242474597 34009	10000	9995.69		
EquityID	Code	Broi Akcii	Cena	Return	Risk	Annual Return	QMeasure	QEquity
15	5BN	935	5.35	0.00294135 764242013	0.01290038 2864671	2.25729442 970822	174.978871 044981	0.81244952 8061141
40	6A6	2472	2.02	0.00135592 256312942	0.01536997 43438124	1.52530541 01221	99.2392944 84327	0.81048067 325110

PortfolioID	Date	Return	Risk	QMeasure	Capital	UsedCapital		
8	2012-09-18	2.071979608 241785	0.017010354 37732762	0.827355507 30847	10000	9998.60		
EquityID	Code	Broi Akcii	Cena	Return	Risk	Annual Return	QMeasure	QEquity
15	5BN	936	5.35	0.00294135 76424201	0.01290038 286467136	2.25729442 97082222	174.978871 044981	0.81244952 80611
38	5BD	4991	1	0.00217454 39408181	0.02113483 99170272	1.88601036 26943017	89.2370308 97728	0.80959052 55726

PortfolioID	Date	Return	Risk	QMeasure	Capital	UsedCapital
9	2012-09-18	1.654041292 242276	0.011948020 451119213	0.812423304 5649482	10000	9995.27

EquityID	Code	Broi Akcii	Cena	Return	Risk	Annual Return	QMeasure	QEquity
15	5BN	937	5.35	0.00294135 764242013	0.01290038 2864671	2.25729442 970822	174.978871 044981	0.81244952 8061141
8	4EC	2542	1.96	0.00012961 739769437	0.01098909 68662783	1.04663212 43523	95.2427790 09809	0.80689056 528104

PortfolioID	Date	Return	Risk	QMeasure	Capital	UsedCapital
10	2012-09-18	1.45297119904 7097	0.01378791158 3612037	0.81142075790 11924	10000	9997.88

EquityID	Code	Broi Akcii	Cena	Return	Risk	Annual Return	QMeasure	QEquity
26	6AS	1708	2.93	0.00100412 586555915	0.01220874 11810062	1.38076923 076923	113.096773 065953	0.81196512 352114
40	6A6	2472	2.02	0.00135592 256312942 47	0.01536997 43438124	1.52530541 01221	99.2392944 84327	0.81048067 325110

PortfolioID	Date	Return	Risk	QMeasure	Capital	UsedCapital
11	2012-09-18	1.633019860 4437265	0.016665254 88193734	0.810544287 6472	10000	9996.44

EquityID	Code	Broi Akcii	Cena	Return	Risk	Annual Return	QMeasure	QEquity
26	6AS	1708	2.93	0.00100412 58655591	0.01220874 11810062	1.38076923 0769230	113.096773 065953	0.81196512 352114
38	5BD	4992	1	0.00217454 39408181	0.02113483 99170272	1.88601036 26943017	89.2370308 97728	0.80959052 55726

PortfolioID	Date	Return	Risk	QMeasure	Capital	UsedCapital		
12	2012-09-18	1.214224380 2658506	0.011600830 607284384	0.810671580 8914452	10000	9997.51		
EquityID	Code	Broi Akcii	Cena	Return	Risk	AnnualReturn	QMeasure	QEquity
26	6AS	1711	2.93	0.00100412 586555915	0.01220874 11810062	1.38076923 076923	113.096773 065953	0.81196512 352114
8	4EC	2543	1.96	0.00012961 739769437 6	0.01098909 686627830	1.04663212 435232	95.2427790 098090	0.80689056 528104

PortfolioID	Date	Return	Risk	QMeasure	Capital	UsedCapital		
13	2012-09-18	1.705558791 789317	0.018250823 378511243	0.809266131 1425762	10000	9998.52		
EquityID	Code	Broi Akcii	Cena	Return	Risk	Annual Return	QMeasure	QEquity
40	6A6	2476	2.02	0.00135592 256312942	0.01536997 43438124	1.52530541 0122165	99.2392944 84327	0.81048067 325110
38	5BD	4997	1	0.00217454 394081816 1	0.02113483 991702721 4	1.88601036 2694301	89.2370308 9772	0.80959052 55726

PortfolioID	Date	Return	Risk	QMeasure	Capital	UsedCapital
14	2012-09-18	1.286500027 0795617	0.013184397 761232296	0.808556206 7278724	10000	9997.8

EquityID	Code	Broi Akcii	Cena	Return	Risk	Annual Return	QMeasure	QEquity
40	6A6	2480	2.02	0.00135592 256312942 47	0.01536997 434381246 8	1.52530541 0122165	99.2392944 8432757	0.81048067 32511002
8	4EC	2545	1.96	0.00012961 739769437 66	0.01098909 686627830 3	1.04663212 4352329	95.2427790 0980901	0.80689056 52810485

PortfolioID	Date	Return	Risk	QMeasure	Capital	UsedCapital
15	2012-09-18	1.467022237 7774405	0.016070441 458093526	0.805558969 3685251	10000	9998.16

EquityID	Code	Broi Akcii	Cena	Return	Risk	Annual Return	QMeasure	QEquity
38	5BD	5008	1	0.00217454 394081816 1	0.02113483 991702721 4	1.88601036 26943017	89.2370308 9772843	0.80959052 5572678
8	4EC	2546	1.96	0.00012961 739769437 66	0.01098909 686627830 3	1.04663212 4352329	95.2427790 0980901	0.80689056 52810485

Портфейли с три актива

PortfolioID	Date	Return	Risk	QMeasure	Capital	UsedCapital
16	2012-09-18	1.721349072 4178793	0.013491770 603139538	0.812390062 7864403	10000	9994.33

EquityID	Code	Broi Akcii	Cena	Return	Risk	Annual Return	QMeasure	QEquity
15	5BN	623	5.35	0.00294135 764242013 18	0.01290038 286467136	2.25729442 97082222	174.978871 04498176	0.81244952 8061141

26	6AS	1138	2.93	0.00100412 586555915 63	0.01220874 118100625 4	1.38076923 07692307	113.096773 06595394	0.81196512 35211458
40	6A6	1647	2.02	0.00135592 256312942 47	0.01536997 434381246 8	1.52530541 0122165	99.2392944 8432757	0.81048067 32511002

PortfolioID	Date	Return	Risk	QMeasure	Capital	UsedCapital
17	2012-09-18	1.841397222 9653432	0.015408573 703792254	0.812345750 9670738	10000	9993.39

EquityID	Code	Broi Akcii	Cena	Return	Risk	Annual Return	QMeasure	QEquity
15	5BN	623	5.35	0.00294135 764242013 18	0.01290038 286467136	2.25729442 97082222	174.978871 04498176	0.81244952 8061141
26	6AS	1138	2.93	0.00100412 586555915 63	0.01220874 118100625 4	1.38076923 07692307	113.096773 06595394	0.81196512 35211458
38	5BD	3326	1	0.00217454 394081816 1	0.02113483 991702721 4	1.88601036 26943017	89.2370308 9772843	0.80959052 5572678

PortfolioID	Date	Return	Risk	QMeasure	Capital	UsedCapital
18	2012-09-18	1.562778631 5076853	0.012035091 43594619	0.812384222 6984068	10000	9993.95

EquityID	Code	Broi Akcii	Cena	Return	Risk	Annual Return	QMeasure	QEquity
15	5BN	624	5.35	0.00294135 764242013 18	0.01290038 286467136	2.25729442 97082222	174.978871 04498176	0.81244952 8061141
26	6AS	1139	2.93	0.00100412 586555915 63	0.01220874 118100625 4	1.38076923 07692307	113.096773 06595394	0.81196512 35211458

8	4EC	1693	1.96	0.00012961 739769437 66	0.01098909 686627830 3	1.04663212 4352329	95.2427790 0980901	0.80689056 52810485
---	-----	------	------	-------------------------------	------------------------------	-----------------------	-----------------------	------------------------

PortfolioID	Date	Return	Risk	QMeasure	Capital	UsedCapital
19	2012-09-18	1.889835683 5285974	0.016463803 509638372	0.812335027 2706766	10000	9997.38

EquityID	Code	Broi Akcii	Cena	Return	Risk	Annual Return	QMeasure	QEquity
15	5BN	624	5.35	0.00294135 764242013 18	0.01290038 286467136	2.25729442 97082222	174.978871 04498176	0.81244952 8061141
40	6A6	1649	2.02	0.00135592 256312942 47	0.01536997 434381246 8	1.52530541 0122165	99.2392944 8432757	0.81048067 32511002
38	5BD	3328	1	0.00217454 394081816 1	0.02113483 991702721 4	1.88601036 26943017	89.2370308 9772843	0.80959052 5572678

PortfolioID	Date	Return	Risk	QMeasure	Capital	UsedCapital
20	2012-09-18	1.611100699 3200383	0.013089432 827256206	0.812337674 8746821	10000	9993.66

EquityID	Code	Broi Akcii	Cena	Return	Risk	Annual Return	QMeasure	QEquity
15	5BN	624	5.35	0.00294135 764242013 18	0.01290038 286467136	2.25729442 97082222	174.978871 04498176	0.81244952 8061141
40	6A6	1651	2.02	0.00135592 256312942 47	0.01536997 434381246 8	1.52530541 0122165	99.2392944 8432757	0.81048067 32511002

8	4EC	1694	1.96	0.00012961 739769437 66	0.01098909 686627830 3	1.04663212 4352329	95.2427790 0980901	0.80689056 52810485
---	-----	------	------	-------------------------------	------------------------------	-----------------------	-----------------------	------------------------

PortfolioID	Date	Return	Risk	QMeasure	Capital	UsedCapital
21	2012-09-18	1.731359252 7477768	0.015010093 094045723	0.812253918 9308356	10000	9996.99

EquityID	Code	Broi Akcii	Cena	Return	Risk	Annual Return	QMeasure	QEquity
15	5BN	625	5.35	0.00294135 764242013 18	0.01290038 286467136	2.25729442 97082222	174.978871 04498176	0.81244952 8061141
38	5BD	3333	1	0.00217454 394081816 1	0.02113483 991702721 4	1.88601036 26943017	89.2370308 9772843	0.80959052 5572678
8	4EC	1694	1.96	0.00012961 739769437 66	0.01098909 686627830 3	1.04663212 4352329	95.2427790 0980901	0.80689056 52810485

PortfolioID	Date	Return	Risk	QMeasure	Capital	UsedCapital
22	2012-09-18	1.597123817 5214246	0.016233600 209289895	0.810518788 24117	10000	9996.25

EquityID	Code	Broi Akcii	Cena	Return	Risk	Annual Return	QMeasure	QEquity
26	6AS	1139	2.93	0.00100412 586555915 63	0.01220874 118100625 4	1.38076923 07692307	113.096773 06595394	0.81196512 35211458
40	6A6	1649	2.02	0.00135592 256312942 47	0.01536997 434381246 8	1.52530541 0122165	99.2392944 8432757	0.81048067 32511002
38	5BD	3328	1	0.00217454 394081816 1	0.02113483 991702721 4	1.88601036 26943017	89.2370308 9772843	0.80959052 5572678

PortfolioID	Date	Return	Risk	QMeasure	Capital	UsedCapital		
23	2012-09-18	1.318007934 6767737	0.012858300 836964515	0.810541192 1982749	10000	9995.46		
EquityID	Code	Broi Akcii	Cena	Return	Risk	AnnualReturn	QMeasure	QEquity
26	6AS	1140	2.93	0.00100412 586555915 63	0.01220874 118100625 4	1.38076923 07692307	113.096773 06595394	0.81196512 35211458
40	6A6	1651	2.02	0.00135592 256312942 47	0.01536997 434381246 8	1.52530541 0122165	99.2392944 8432757	0.81048067 32511002
8	4EC	1694	1.96	0.00012961 739769437 66	0.01098909 686627830 3	1.04663212 4352329	95.2427790 0980901	0.80689056 52810485

PortfolioID	Date	Return	Risk	QMeasure	Capital	UsedCapital		
24	2012-09-18	1.438183042 956105	0.014779259 49823531	0.809427067 0902691	10000	9998.33		
EquityID	Code	Broi Akcii	Cena	Return	Risk	AnnualReturn	QMeasure	QEquity
26	6AS	1141	2.93	0.00100412 586555915 63	0.01220874 118100625 4	1.38076923 07692307	113.096773 06595394	0.81196512 35211458
38	5BD	3333	1	0.00217454 394081816 1	0.02113483 991702721 4	1.88601036 26943017	89.2370308 9772843	0.80959052 5572678
8	4EC	1695	1.96	0.00012961 739769437 66	0.01098909 686627830 3	1.04663212 4352329	95.2427790 0980901	0.80689056 52810485

PortfolioID	Date	Return	Risk	QMeasure	Capital	UsedCapital
25	2012-09-18	1.486485826 1453378	0.015836521 391131873	0.807939167 4441688	10000	9998.22

EquityID	Code	Broi Akcii	Cena	Return	Risk	AnnualReturn	QMeasure	QEquity
40	6A6	1653	2.02	0.00135592 256312942 47	0.01536997 434381246 8	1.52530541 0122165	99.2392944 8432757	0.81048067 32511002
38	5BD	3335	1	0.00217454 394081816 1	0.02113483 991702721 4	1.88601036 26943017	89.2370308 9772843	0.80959052 5572678
8	4EC	1696	1.96	0.00012961 739769437 66	0.01098909 686627830 3	1.04663212 4352329	95.2427790 0980901	0.80689056 52810485

Порфейли с четири актива

PortfolioID	Date	Return	Risk	QMeasure	Capital	UsedCapital
26	2012-09-18	1.762436698 2183373	0.015398931 487211293	0.812246313 8909733	10000	9997.74

EquityID	Code	Broi Akcii	Cena	Return	Risk	AnnualReturn	QMeasure	QEquity
15	5BN	468	5.35	0.00294135 764242013 18	0.01290038 286467136	2.25729442 97082222	174.978871 04498176	0.81244952 8061141
26	6AS	854	2.93	0.00100412 586555915 63	0.01220874 118100625 4	1.38076923 07692307	113.096773 06595394	0.81196512 35211458
40	6A6	1236	2.02	0.00135592 256312942 47	0.01536997 434381246 8	1.52530541 0122165	99.2392944 8432757	0.81048067 32511002

38	5BD	2495	1	0.00217454 394081816 1	0.02113483 991702721 4	1.88601036 26943017	89.2370308 9772843	0.80959052 5572678
----	-----	------	---	------------------------------	------------------------------	------------------------	-----------------------	-----------------------

PortfolioID	Date	Return	Risk	QMeasure	Capital	UsedCapital
27	2012-09-18	1.553409930 8201912	0.012868847 34580538	0.812292076 7237802	10000	9992

EquityID	Code	Broi Akcii	Cena	Return	Risk	AnnualReturn	QMeasure	QEquity
15	5BN	468	5.35	0.00294135 764242013 18	0.01290038 286467136	2.25729442 97082222	174.978871 04498176	0.81244952 8061141
26	6AS	854	2.93	0.00100412 586555915 63	0.01220874 118100625 4	1.38076923 07692307	113.096773 06595394	0.81196512 35211458
40	6A6	1237	2.02	0.00135592 256312942 47	0.01536997 434381246 8	1.52530541 0122165	99.2392944 8432757	0.81048067 32511002
8	4EC	1269	1.96	0.00012961 739769437 66	0.01098909 686627830 3	1.04663212 4352329	95.2427790 0980901	0.80689056 52810485

PortfolioID	Date	Return	Risk	QMeasure	Capital	UsedCapital
28	2012-09-18	1.643523390 9600349	0.014308250 075057524	0.812206908 4932855	10000	9996.15

EquityID	Code	Broi Akcii	Cena	Return	Risk	Annual Return	QMeasure	QEquity
15	5BN	468	5.35	0.00294135 764242013 18	0.01290038 286467136	2.25729442 97082222	174.978871 04498176	0.81244952 8061141

26	6AS	855	2.93	0.00100412 586555915 63	0.01220874 118100625 4	1.38076923 07692307	113.096773 06595394	0.81196512 35211458
38	5BD	2498	1	0.00217454 394081816 1	0.02113483 991702721 4	1.88601036 26943017	89.2370308 9772843	0.80959052 5572678
8	4EC	1270	1.96	0.00012961 739769437 66	0.01098909 686627830 3	1.04663212 4352329	95.2427790 0980901	0.80689056 52810485

PortfolioID	Date	Return	Risk	QMeasure	Capital	UsedCapital
29	2012-09-18	1.67980585552 98772	0.01510013315 7283995	0.81211238035 87018	10000	9992.75999999 9998

EquityID	Code	Broi Akcii	Cena	Return	Risk	Annual Return	QMeasure	QEquity
15	5BN	468	5.35	0.00294135 764242013 18	0.01290038 286467136	2.25729442 97082222	174.978871 04498176	0.81244952 8061141
40	6A6	1238	2.02	0.00135592 256312942 47	0.01536997 434381246 8	1.52530541 0122165	99.2392944 8432757	0.81048067 32511002
38	5BD	2499	1	0.00217454 394081816 1	0.02113483 991702721 4	1.88601036 26943017	89.2370308 9772843	0.80959052 5572678
8	4EC	1270	1.96	0.00012961 739769437 66	0.01098909 686627830 3	1.04663212 4352329	95.2427790 0980901	0.80689056 52810485

PortfolioID	Date	Return	Risk	QMeasure	Capital	UsedCapital
30	2012-09-18	1.459983780 4367013	0.014927074 803227799	0.809742071 2761217	10000	9996.07

EquityID	Code	Broi Akcii	Cena	Return	Risk	Annual Return	QMeasure	QEquity
26	6AS	855	2.93	0.00100412 586555915 63	0.01220874 118100625 4	1.38076923 07692307	113.096773 06595394	0.81196512 35211458

40	6A6	1238	2.02	0.00135592 256312942 47	0.01536997 434381246 8	1.52530541 0122165	99.2392944 8432757	0.81048067 32511002
38	5BD	2499	1	0.00217454 394081816 1	0.02113483 991702721 4	1.88601036 26943017	89.2370308 9772843	0.80959052 5572678
8	4EC	1271	1.96	0.00012961 739769437 66	0.01098909 686627830 3	1.04663212 4352329	95.2427790 0980901	0.80689056 52810485

Порфейл с пет актива

PortfolioID	Date	Return	Risk	QMeasure	Capital	UsedCapital		
31	2012-09-18	1.619873801 568916	0.014520648 754384285	0.812085254 0722922	10000	9994.18		
EquityID	Code	Broi Akcii	Cena	Return	Risk	Annual Return	QMeasure	QEquity
15	5BN	374	5.35	0.00294135 764242013 18	0.01290038 286467136	2.25729442 97082222	174.978871 04498176	0.81244952 8061141
26	6AS	684	2.93	0.00100412 586555915 63	0.01220874 118100625 4	1.38076923 07692307	113.096773 06595394	0.81196512 35211458
40	6A6	990	2.02	0.00135592 256312942 47	0.01536997 434381246 8	1.52530541 0122165	99.2392944 8432757	0.81048067 32511002
38	5BD	1998	1	0.00217454 394081816 1	0.02113483 991702721 4	1.88601036 26943017	89.2370308 9772843	0.80959052 5572678
8	4EC	1016	1.96	0.00012961 739769437 66	0.01098909 686627830 3	1.04663212 4352329	95.2427790 0980901	0.80689056 52810485

Порфейли с фиксиран максимален брой активи СЕДЕМ

Top of Form

Date :

Bottom of Form

Общ брой порфейли: 127

Порфейли с един актив

PortfolioID	Date	Return	Risk	QMeasure	Capital	UsedCapital		
1	2012-11-21	2.25729442970 82222	0.01290038286 467136	0.81244952806 1141	150000	149997.949999 99998		
EquityID	Code	Broi Akcii	Cena	Return	Risk	Annual Return	QMeasure	QEquity
15	5BN	28037	5.35	0.00294135 764242013 18	0.01290038 286467136	2.25729442 97082222	174.978871 04498176	0.81244952 8061141

PortfolioID	Date	Return	Risk	QMeasure	Capital	UsedCapital		
2	2012-11-21	1.38076923076 92307	0.01220874118 1006254	0.81196512352 11458	150000	149998.42		
EquityID	Code	Broi Akcii	Cena	Return	Risk	Annual Return	QMeasure	QEquity
26	6AS	51194	2.93	0.00100412 586555915 63	0.01220874 118100625 4	1.38076923 07692307	113.096773 06595394	0.81196512 35211458

PortfolioID	Date	Return	Risk	QMeasure	Capital	UsedCapital		
3	2012-11-21	1.52530541012 2165	0.01536997434 3812468	0.81048067325 11002	150000	149999.14		
EquityID	Code	Broi Akcii	Cena	Return	Risk	Annual Return	QMeasure	QEquity
40	6A6	74257	2.02	0.00135592 256312942 47	0.01536997 434381246 8	1.52530541 0122165	99.2392944 8432757	0.81048067 32511002

PortfolioID	Date	Return	Risk	QMeasure	Capital	UsedCapital		
4	2012-11-21	1.886010362 6943017	0.021134839 917027214	0.809590525 572678	150000	150000		
EquityID	Code	Broi Akcii	Cena	Return	Risk	AnnualReturn	QMeasure	QEquity
38	5BD	150000	1	0.00217454 394081816 1	0.02113483 991702721 4	1.88601036 26943017	89.2370308 9772843	0.80959052 5572678

PortfolioID	Date	Return	Risk	QMeasure	Capital	UsedCapital		
5	2012-11-21	1.04663212435 2329	0.01098909686 6278303	0.80689056528 10485	150000	149998.8		
EquityID	Code	Broi Akcii	Cena	Return	Risk	Annual Return	QMeasure	QEquity
8	4EC	76530	1.96	0.00012961 739769437 66	0.01098909 686627830 3	1.04663212 4352329	95.2427790 0980901	0.80689056 52810485

PortfolioID	Date	Return	Risk	QMeasure	Capital	UsedCapital		
6	2012-11-21	1.20805369127 5167	0.01316769688 7302946	0.80195710987 62698	150000	149998.79		
EquityID	Code	Broi Akcii	Cena	Return	Risk	Annual Return	QMeasure	QEquity
1	3JR	62761	2.39	0.00056346 334351165 45	0.01316769 688730294 6	1.20805369 1275167	91.7437348 0908739	0.80195710 98762698

PortfolioID	Date	Return	Risk	QMeasure	Capital	UsedCapital		
7	2012-11-21	0.96605795950 77403	0.01073925428 4325216	0.79812401943 11958	150000	149997.834		
EquityID	Code	Broi Akcii	Cena	Return	Risk	Annual Return	QMeasure	QEquity
14	5ALB	30114	4.981	- 0.00009561 767932161 6	0.01073925 428432521 6	0.96605795 95077403	89.9557766 2387393	0.79812401 94311958

Портфейли с два актива

PortfolioID	Date	Return	Risk	QMeasure	Capital	UsedCapital
8	2012-11-21	1.819162521 2496003	0.012554665 14748475	0.812438176 488536	150000	149993.47

EquityID	Code	Broi Akcii	Cena	Return	Risk	Annual Return	QMeasure	QEquity
15	5BN	14022	5.35	0.00294135 764242013 18	0.01290038 286467136	2.25729442 97082222	174.978871 04498176	0.81244952 8061141
26	6AS	25589	2.93	0.00100412 586555915 63	0.01220874 118100625 4	1.38076923 07692307	113.096773 06595394	0.81196512 35211458

PortfolioID	Date	Return	Risk	QMeasure	Capital	UsedCapital
9	2012-11-21	1.89174392548 63955	0.01413368061 4045352	0.81242474597 34009	150000	149994.91

EquityID	Code	Broi Akcii	Cena	Return	Risk	Annual Return	QMeasure	QEquity
15	5BN	14035	5.35	0.00294135 764242013 18	0.01290038 286467136	2.25729442 97082222	174.978871 04498176	0.81244952 8061141
40	6A6	37083	2.02	0.00135592 256312942 47	0.01536997 434381246 8	1.52530541 0122165	99.2392944 8432757	0.81048067 32511002

PortfolioID	Date	Return	Risk	QMeasure	Capital	UsedCapital
10	2012-11-21	2.071979608 241785	0.017010354 37732762	0.827355507 3084777	150000	149997.05

EquityID	Code	Broi Akcii	Cena	Return	Risk	Annual Return	QMeasure	QEquity
15	5BN	14043	5.35	0.00294135 764242013 18	0.01290038 286467136	2.25729442 97082222	174.978871 04498176	0.81244952 8061141
38	5BD	74867	1	0.00217454 394081816 1	0.02113483 991702721 4	1.88601036 26943017	89.2370308 9772843	0.80959052 5572678

PortfolioID	Date	Return	Risk	QMeasure	Capital	UsedCapital		
11	2012-11-21	1.65404129224 2276	0.01194802045 1119213	0.81242330456 49482	150000	149993.779999 99997		
EquityID	Code	Broi Akcii	Cena	Return	Risk	Annual Return	QMeasure	QEquity
15	5BN	14066	5.35	0.00294135 764242013 18	0.01290038 286467136	2.25729442 97082222	174.978871 04498176	0.81244952 8061141
8	4EC	38133	1.96	0.00012961 739769437 66	0.01098909 686627830 3	1.04663212 4352329	95.2427790 0980901	0.80689056 52810485

PortfolioID	Date	Return	Risk	QMeasure	Capital	UsedCapital		
12	2012-11-21	1.73608369482 59255	0.01303317120 68284	0.81241453618 5104	150000	149993.789999 99998		
EquityID	Code	Broi Akcii	Cena	Return	Risk	Annual Return	QMeasure	QEquity
15	5BN	14109	5.35	0.00294135 764242013 18	0.01290038 286467136	2.25729442 97082222	174.978871 04498176	0.81244952 8061141
1	3JR	31176	2.39	0.00056346 334351165 45	0.01316769 688730294 6	1.20805369 1275167	91.7437348 0908739	0.80195710 98762698

PortfolioID	Date	Return	Risk	QMeasure	Capital	UsedCapital		
13	2012-11-21	1.61741875111 08982	0.01182942982 9500077	0.81241710684 34243	150000	149996.512999 99998		
EquityID	Code	Broi Akcii	Cena	Return	Risk	Annual Return	QMeasure	QEquity

Следват още 60 страници за портфейли с три, четири, пет и шест актива

Част от портфейлите със шест актива

PortfolioID	Date	Return	Risk	QMeasure	Capital	UsedCapital
121	2012-11-21	1.51226970196 83091	0.01389831212 5307394	0.81183881278 11518	150000	149993.696

EquityID	Code	Broi Akcii	Cena	Return	Risk	Annual Return	QMeasure	QEquity
15	5BN	4697	5.35	0.00294135 764242013 18	0.01290038 286467136	2.25729442 97082222	174.978871 04498176	0.81244952 8061141
26	6AS	8571	2.93	0.00100412 586555915 63	0.01220874 118100625 4	1.38076923 07692307	113.096773 06595394	0.81196512 35211458
40	6A6	12410	2.02	0.00135592 256312942 47	0.01536997 434381246 8	1.52530541 0122165	99.2392944 8432757	0.81048067 32511002
38	5BD	25041	1	0.00217454 394081816 1	0.02113483 991702721 4	1.88601036 26943017	89.2370308 9772843	0.80959052 5572678
8	4EC	12733	1.96	0.00012961 739769437 66	0.01098909 686627830 3	1.04663212 4352329	95.2427790 0980901	0.80689056 52810485
14	5ALB	4956	4.981	- 0.00009561 767932161 6	0.01073925 428432521 6	0.96605795 95077403	89.9557766 2387393	0.79812401 94311958

PortfolioID	Date	Return	Risk	QMeasure	Capital	UsedCapital
122	2012-11-21	1.53946519251 61325	0.01426191454 1696072	0.81180766529 1671	150000	149990.661

EquityID	Code	Broi Akcii	Cena	Return	Risk	Annual Return	QMeasure	QEquity
15	5BN	4701	5.35	0.00294135 764242013 18	0.01290038 286467136	2.25729442 97082222	174.978871 04498176	0.81244952 8061141
26	6AS	8580	2.93	0.00100412 586555915 63	0.01220874 118100625 4	1.38076923 07692307	113.096773 06595394	0.81196512 35211458
40	6A6	12423	2.02	0.00135592 256312942 47	0.01536997 434381246 8	1.52530541 0122165	99.2392944 8432757	0.81048067 32511002
38	5BD	25066	1	0.00217454 394081816 1	0.02113483 991702721 4	1.88601036 26943017	89.2370308 9772843	0.80959052 5572678
1	3JR	10389	2.39	0.00056346 334351165 45	0.01316769 688730294 6	1.20805369 1275167	91.7437348 0908739	0.80195710 98762698
14	5ALB	4961	4.981	- 0.00009561 767932161 6	0.01073925 428432521 6	0.96605795 95077403	89.9557766 2387393	0.79812401 94311958

PortfolioID	Date	Return	Risk	QMeasure	Capital	UsedCapital		
123	2012-11-21	1.39939070540 22587	0.01256730770 1379454	0.81188252629 79619	150000	149992.644		
EquityID	Code	Broi Akcii	Cena	Return	Risk	Annual Return	QMeasure	QEquity
15	5BN	4704	5.35	0.00294135 764242013 18	0.01290038 286467136	2.25729442 97082222	174.978871 04498176	0.81244952 8061141
26	6AS	8585	2.93	0.00100412 586555915 63	0.01220874 118100625 4	1.38076923 07692307	113.096773 06595394	0.81196512 35211458
40	6A6	12429	2.02	0.00135592 256312942 47	0.01536997 434381246 8	1.52530541 0122165	99.2392944 8432757	0.81048067 32511002
8	4EC	12753	1.96	0.00012961 739769437 66	0.01098909 686627830 3	1.04663212 4352329	95.2427790 0980901	0.80689056 52810485
1	3JR	10395	2.39	0.00056346 334351165 45	0.01316769 688730294 6	1.20805369 1275167	91.7437348 0908739	0.80195710 98762698
14	5ALB	4964	4.981	- 0.00009561 767932161 6	0.01073925 428432521 6	0.96605795 95077403	89.9557766 2387393	0.79812401 94311958

PortfolioID	Date	Return	Risk	QMeasure	Capital	UsedCapital		
124	2012-11-21	1.45969077234 76986	0.01353089126 8427075	0.81169676582 15678	150000	149990.004		
EquityID	Code	Broi Akcii	Cena	Return	Risk	Annual Return	QMeasure	QEquity
15	5BN	4705	5.35	0.00294135 764242013 18	0.01290038 286467136	2.25729442 97082222	174.978871 04498176	0.81244952 8061141
26	6AS	8586	2.93	0.00100412 586555915 63	0.01220874 118100625 4	1.38076923 07692307	113.096773 06595394	0.81196512 35211458

38	5BD	25085	1	0.00217454 394081816 1	0.02113483 991702721 4	1.88601036 26943017	89.2370308 9772843	0.80959052 5572678
8	4EC	12756	1.96	0.00012961 739769437 66	0.01098909 686627830 3	1.04663212 4352329	95.2427790 0980901	0.80689056 52810485
1	3JR	10397	2.39	0.00056346 334351165 45	0.01316769 688730294 6	1.20805369 1275167	91.7437348 0908739	0.80195710 98762698
14	5ALB	4964	4.981	- 0.00009561 767932161 6	0.01073925 428432521 6	0.96605795 95077403	89.9557766 2387393	0.79812401 94311958

PortfolioID	Date	Return	Risk	QMeasure	Capital	UsedCapital
125	2012-11-21	1.48392077930 76261	0.01406071564 3556179	0.81149807445 1878	150000	149990.106

EquityID	Code	Broi Akcii	Cena	Return	Risk	Annual Return	QMeasure	QEquity
15	5BN	4706	5.35	0.00294135 764242013 18	0.01290038 286467136	2.25729442 97082222	174.978871 04498176	0.81244952 8061141
40	6A6	12436	2.02	0.00135592 256312942 47	0.01536997 434381246 8	1.52530541 0122165	99.2392944 8432757	0.81048067 32511002
38	5BD	25093	1	0.00217454 394081816 1	0.02113483 991702721 4	1.88601036 26943017	89.2370308 9772843	0.80959052 5572678
8	4EC	12759	1.96	0.00012961 739769437 66	0.01098909 686627830 3	1.04663212 4352329	95.2427790 0980901	0.80689056 52810485
1	3JR	10400	2.39	0.00056346 334351165 45	0.01316769 688730294 6	1.20805369 1275167	91.7437348 0908739	0.80195710 98762698
14	5ALB	4966	4.981	- 0.00009561 767932161 6	0.01073925 428432521 6	0.96605795 95077403	89.9557766 2387393	0.79812401 94311958

PortfolioID	Date	Return	Risk	QMeasure	Capital	UsedCapital		
126	2012-11-21	1.33676613218 95747	0.01394477724 1991174	0.80795580131 9099	150000	149992.036		
EquityID	Code	Broi Akcii	Cena	Return	Risk	Annual Return	QMeasure	QEquity
26	6AS	8590	2.93	0.00100412 586555915 63	0.01220874 118100625 4	1.38076923 07692307	113.096773 06595394	0.81196512 35211458
40	6A6	12437	2.02	0.00135592 256312942 47	0.01536997 434381246 8	1.52530541 0122165	99.2392944 8432757	0.81048067 32511002
38	5BD	25095	1	0.00217454 394081816 1	0.02113483 991702721 4	1.88601036 26943017	89.2370308 9772843	0.80959052 5572678
8	4EC	12761	1.96	0.00012961 739769437 66	0.01098909 686627830 3	1.04663212 4352329	95.2427790 0980901	0.80689056 52810485
1	3JR	10401	2.39	0.00056346 334351165 45	0.01316769 688730294 6	1.20805369 1275167	91.7437348 0908739	0.80195710 98762698
14	5ALB	4966	4.981	- 0.00009561 767932161 6	0.01073925 428432521 6	0.96605795 95077403	89.9557766 2387393	0.79812401 94311958

Портфейл със седем актива

PortfolioID	Date	Return	Risk	QMeasure	Capital	UsedCapital		
127	2012-11-21	1.46910062730 55007	0.01379463584 7101864	0.81157826966 72564	150000	149989.142		
EquityID	Code	Broi Akcii	Cena	Return	Risk	Annual Return	QMeasure	QEquity
15	5BN	4030	5.35	0.00294135 764242013 18	0.01290038 286467136	2.25729442 97082222	174.978871 04498176	0.81244952 8061141
26	6AS	7355	2.93	0.00100412 586555915 63	0.01220874 118100625 4	1.38076923 07692307	113.096773 06595394	0.81196512 35211458
40	6A6	10649	2.02	0.00135592 256312942 47	0.01536997 434381246 8	1.52530541 0122165	99.2392944 8432757	0.81048067 32511002
38	5BD	21488	1	0.00217454 394081816 1	0.02113483 991702721 4	1.88601036 26943017	89.2370308 9772843	0.80959052 5572678

8	4EC	10926	1.96	0.00012961 739769437 66	0.01098909 686627830 3	1.04663212 4352329	95.2427790 0980901	0.80689056 52810485
1	3JR	8906	2.39	0.00056346 334351165 45	0.01316769 688730294 6	1.20805369 1275167	91.7437348 0908739	0.80195710 98762698
14	5ALB	4252	4.981	- 0.00009561 767932161 6	0.01073925 428432521 6	0.96605795 95077403	89.9557766 2387393	0.79812401 94311958