

**ПЛОВДИВСКИ УНИВЕРСИТЕТ „ПАИСИЙ ХИЛЕНДАРСКИ“
ФАКУЛТЕТ ПО МАТЕМАТИКА И ИНФОРМАТИКА
КАТЕДРА „КОМПЮТЪРНИ СИСТЕМИ“**

ИВАН МАРКОВ ДИМИТРОВ

**ОПТИМИЗАЦИЯ НА ГЛОБАЛИЗАЦИИТЕ В
INFOSTATION МРЕЖОВА СРЕДА**

АВТОРЕФЕРАТ

на дисертационен труд
за присъждане на образователна и научна степен “доктор”
в област 4. *Природни науки, математика и информатика*,
професионално направление 4.6. *Информатика и компютърни науки*,
докторска програма *Информатика*

Научни ръководители: проф. д-р Станимир Стоянов
доц. д-р Иван Ганчев

Рецензенти: проф. д.т.н. Тодор Стоилов
проф. д-р Асен Рахнев

Пловдив
2014

Дисертационният труд е обсъден и насрочен за защита на разширено заседание на катедра „Компютърни системи“ при Факултета по математика и информатика на ПУ „Паисий Хилендарски“.

Дисертационният труд съдържа 118 страници. Използваната литература включва 156 източника. Списъкът с авторските публикации по темата съдържа 5 заглавия, от които 4 на английски и 1 на български език.

Защитата на дисертационния труд ще се състои на 23.09.2014 г. от 11.00 часа в заседателната зала на Факултета по математика и информатика на ПУ „Паисий Хилендарски“, гр. Пловдив.

Материалите по защитата са на разположение на интересувашите се в секретариата на ФМИ, нова сграда на ПУ, каб. 330, всеки ден от 8:30 до 17:00 часа.

Автор: Иван Марков Димитров

Заглавие: Оптимизация на глобализациите в InfoStation мрежова среда

Тираж: 100 бр.

Пловдив, 2014 г.

Съдържание

Обща характеристика на дисертационния труд	4
Кратко съдържание на дисертационния труд.....	8
ГЛАВА 1. СЪСТОЯНИЕ НА ИЗСЛЕДВАНИЯ ПРОБЛЕМ.....	8
ГЛАВА 2. АБСТРАКТЕН МОДЕЛ	8
2.1. RG проблем.....	8
2.2. Подход за решаване на RG проблема.....	10
2.3. Представяне на картата на разполагане	12
2.4. Граф на информационните ресурси	13
2.5. Граф на картата на разполагане	14
2.6. Трансформация върху DM-Graph	14
2.7. Еволюционен модел	15
2.7.1. Разлагане на абстрактен граф.....	15
2.7.2. Еволюционна стратегия.....	16
2.7.3. Генериране на субоптимално разлагане на графи	16
2.7.4. Оценка на еволюционната стратегия	17
ГЛАВА 3. ОБЕКТНО-ОРИЕНТИРАНА РЕАЛИЗАЦИЯ НА МОДЕЛА.....	18
3.1. Базови структури на OReL ^O	18
3.2. Архитектура на OReL ^O	19
3.3. Провеждане на експерименти с OReL ^O	21
ГЛАВА 4. АГЕНТНО-ОРИЕНТИРАНА РЕАЛИЗАЦИЯ НА МОДЕЛА	23
4.1. Реинженерингова схема.....	23
4.2. Обща архитектура на OReL ^A	23
4.3. Провеждане на експерименти с OReL ^A	24
4.4. Обобщение на резултатите.....	26
Авторска справка за приносите	26
Публикации на автора.....	28
Благодарности.....	28
Библиография	29

ОБЩА ХАРАКТЕРИСТИКА НА ДИСЕРТАЦИОННИЯ ТРУД

За подпомагане на обучението посредством използване на съвременни информационни и комуникационни технологии във ФМИ на Пловдивския университет се разработва инфраструктура, наречена Разпределен център за електронно обучение, познат като DeLC [28]. Предназначението на Центъра е да доставя по един контекстнозависим, адаптивен и персонализиран начин електронни образователни услуги и електронно учебно съдържание, разположени върху физически разделени сървъри [38]. Концептуално DeLC е динамична мрежова структура, състояща се от [39,40,41,14]:

- възли – върху тях могат да бъдат разполагани електронни образователни услуги и хранилища за електронно съдържание;
- релации – специфицират определени зависимости между възлите.

Спрямо предназначението си възлите могат да бъдат от различен тип:

- **образователни възли** – предоставящи електронни образователни услуги и електронно учебно съдържание посредством подходящ потребителски интерфейс;
- **помощни възли** – прозрачни за потребителя възли, задачата на които е да подпомагат функционирането на образователните възли;
- **специализирани възли** – предоставят специфични услуги за потребители със специален статус.

Спрямо достъпа до предлаганите информационни ресурси в модела на DeLC различаваме два вида образователни възли:

- **възли с фиксиран достъп** – предназначени за доставка на услуги и учебно съдържание предимно през фиксирани комуникационни мрежи, без специална поддръжка на мобилен достъп. Потребителският интерфейс на тези възли се разработва под формата на образователен сайт или портал.
- **възли с мобилен достъп** – предназначени за мобилна доставка на услуги и съдържание през InfoStation базирани комуникационни мрежи посредством посредничеството на специализиран софтуер (мидълуер [12,11,36]).

Възлите могат да оперират самостоятелно или динамично да се свързват помежду си, образувайки комплексни виртуални структури, наречени **образователни клъстери**.

В последните години бяха разработени различни компоненти на DeLC за подпомагане на електронното обучение.

Средата **DeLC Test Center (DeTC)** е клъстер с отворена инфраструктура, която предлага услуги за електронно тестване, които са локализирани върху различни възли с възможност за частична и автоматично контролирана интеграция в рамките на предварително дефинирани виртуални структури [53,23,54]. DeTC може да се разглежда като един нов подход за мрежовобазирано електронно тестване, където взаимодействат физически разпределени интерактивни и кооперативни единици, помощни средства, обучаеми, обучаващи и администратори [22,24]. Образователен портал за средното училище, използващ модела и архитектурата на DeLC е представен подробно в [49].

CADeLC (Creativity Assitant) е специализиран възел за изучаване и идентифициране на креативното мислене и действие на обучаемите. Възелът е адаптация на средата Creativity Assistant, разработена в Де Монтфорт университета, Лестър [46]. CADeLC [48] е програмна среда за събиране и съхраняване на информация за настроеността и действията на обучаемите в процеса на усвояване и прилагане на знания. Събраните данни се представят в структури, наречени „**карти на креативност**” (**Creativity Maps**). Разработва се концепция за откриване и използване на подходящи анализатори на картите на креативността.

Цялостната концепция за DeLC като контекстнозависима и адаптивна инфраструктура за електронно обучение е представена в [26]. В съответствие с нея, следващ етап е изграждането на Виртуално образователно пространство (ВОП). В пространството контекстно-зависимостта на образователните услуги ще се поддържа от автономни интелигентни компоненти с реактивно, интерактивно и проактивно поведение. Съществуващите образователни клъстери поетапно ще се трансформират в такива компоненти. Архитектурата на пространството ще бъде напълно децентрализирана. Трансформацията на DeLC във ВОП се предприема поради въздействието върху начина на опериране на интернет базирани приложения (каквото е DeLC), което ще имат в недалечното бъдеще две глобални тенденции – **Интернет**, като мрежа от компютърни мрежи, постепенно се трансформира в **мрежа на нещата** (Internet of Things) [2,6] и в сегашния синтактичен уеб възниква следващият семантичен уеб [4,1]. Първите идеи за създаване на пространството са дадени в [52,47]. Възможностите за трансформация на DeLC във ВОП са разгледани в [50].

За подпомагане на реалния образователен процес във ФМИ на Пловдивския университет бяха изградени два образователни клъстера.

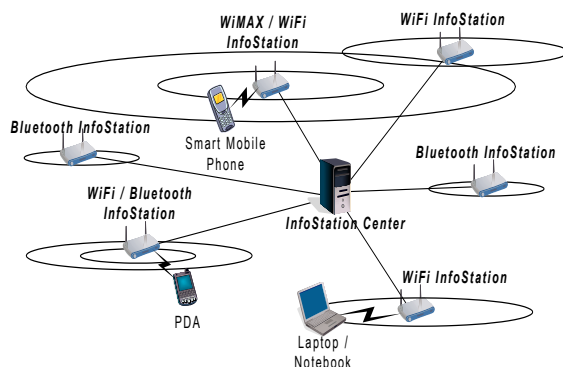
Първият клъстер, наречен **MyDeLC** [50], се използва за организация и провеждане на електронно обучение във ФМИ, като доставя фиксиран достъп до услугите и електронното съдържание посредством специализиран образователен портал. Клъстерът се състои от четири възела:

- **Образователен портал DeLC** – ядрото на клъстера, доставящо достъп на потребителите до електронните услуги и учебното съдържание;
- **eLSE (e-Learning in Software Engineering)** [56] – специализиран възел, в който са интегрирани SREPTICS [55], rLE [30,42] и eLSEBuilder – средства, предназначени за подпомагане на обучението по *Софтуерни технологии* [29,31,16];
- **Agent Village (AV)** [57] – помощен възел, който доставя специализирана помощ на услугите, предоставяни от образователния портал на DeLC, под формата на „асистенти” [32,43]. Асистентите са реализирани като интелигентни агенти [33]. Разширяване на инфраструктурата на Центъра с този възел цели усилване на нейната проактивност, т.е. действие „от името на потребителя”, „поемане на самоинициатива” и „самоактивиране”, когато асистентите „преценят”, че е необходима тяхната намеса. Агентите не са интегрирани директно в портална архитектура, а „населяват” агентно-ориентирания възел AV.
- **Interface Node** – възел за интеграция на клъстера MyDeLC в информационната система на университета.

Вторият клъстер, наречен **InfoStation клъстер** [35,13], осигурява мобилен достъп до услуги и информационни ресурси посредством интелигентни безжични точки на достъп (наричани InfoStations – ISs), разположени около сградата на

университета [27,34,35,15]. Архитектурата на комуникационната мрежа е изградена на три отделни нива – InfoStation Center, InfoStations и мобилни устройства (Фигура 1).

Първият слой обхваща мобилни устройства на потребителите, снабдени с персонални асистенти (реализирани като интелигентни агенти) за осъществяване взаимодействие със средата. Вторият слой се състои от ISs, доставящи локални информационни ресурси на намиращите се в обсега им мобилни устройства. При необходимост една IS става посредник между мобилното устройство и InfoStation Center (ISC). Третият слой е самият ISC, доставящ поделяеми глобални информационни ресурси. Освен това ISC синхронизира работата на отделните ISs. В стандартния вариант на InfoStation архитектурата [9] информационните станции оперират само като посредници между мобилни потребителски устройства и централния сървър (ISC), върху който могат да се разполагат различни приложения. В архитектурата на DeLC правомощията на станциите се разширяват, като върху тях се разполагат информационни ресурси, които могат да бъдат локално използвани. С това децентрализирано разположение на услугите и другите информационни ресурси върху всички възли се цели балансирано натоварване на мрежата и по-добра производителност. Децентрализираното разполагане на информационни ресурси обаче има съществени недостатъци. Системното натоварване, свързано с взаимовръзките на информационните ресурси по време на изпълнение на една заявка, е особено тежко, когато свързаните ресурси са разположени на различни информационни станции. Задачата за намаляване на натоварването е от особена важност в приетия децентрализиран подход. Степента на натоварване зависи от начина на разпределение на информационните ресурси.



Фигура 1. InfoStation базирана мрежова архитектура

Целта на дисертационния труд е създаване и програмна реализация на формален модел за оптимално разполагане на информационните ресурси на DeLC и неговото разширение (Виртуалното образователно пространство) върху разпределени изчислителни конфигурации (InfoStation станции).

За постигане на целта на дисертационния труд са конкретизирани следните задачи:

- създаване на формализиран модел за разпределеното разполагане на информационните ресурси на DeLC и Виртуалното образователно пространство;
- разработване на метод за решаване на задачата за оптимално разполагане на информационните ресурси;
- прототипна реализация и оценка на обектно-ориентираната версия на модела и метода;

- прототипна реализация и оценка на агентно-ориентираната версия на модела и метода.

Дисертационният труд се състои от увод, четири глави, авторска справка, перспективи за развитие, апробация на резултатите, публикации на автора, декларация за оригиналност и библиография.

Уводът представлява въведение в изследваната тематика. Тук се прави преглед на базовата архитектура на Разпределения център за електронно обучение, наречен **DeLC**. Посочени са неговите основни компоненти и тяхното предназначение. Представен е следващият етап на развитие, който е изграждане на Виртуално образователно пространство (ВОП) и нуждата от управление на настъпилите в него събития. Коментирани са клъстерите MyDeLC, използващ се за организация и провеждане на електронно обучение и InfoStation, осигуряващ мобилен достъп до услуги и информационни ресурси посредством интелигентни безжични точки на достъп. В увода е дефинирана целта на дисертационния труд, като са посочени също задачите за нейното постигане.

В първа глава е направен преглед на състоянието на научноизследователски области, имащи отношение към темата на дисертационния труд. Основно внимание се обръща на класическите методи за решаване на проблеми посредством търсене. Разгледани са основните понятия и характеристики на най-разпространените и широко използваните от тях. Описани са съвременните интерпретации и версии на тези методи. Особено внимание се обръща на еволюционните стратегии и възможностите за тяхната реализация като обектно-ориентиран и агентно-базиран софтуер.

Втората глава, представляваща ядрото на изследването, описва абстрактен модел на решаваната задача. В основата на модела лежи формалната дефиниция на добре дефиниран проблем, наречен **Request Globalization проблем (RG проблем)**. Предложен е подход за решение на RG проблема, структуриран в няколко отделни стъпки. Дефинирани са основните понятия, които се използват за реализиране на описания подход. Разгледан е еволюционен модел за решаване на RG проблема.

Трета глава представя програмната реализация на теоретичния модел, базирана на обектно-ориентирана парадигма, наречена **OReL^O (Optimizing Resource Locations – обектно-ориентирана версия)**. Описана е архитектурата на OReL^O, състояща се от седем базови модула, реализирани като Java класове. Подробно са представени проведените експерименти с четири различни набора (размери на популацията – РП) от входни данни.

Глава четвърта е посветена на програмната реализация на теоретичния модел, базирана на агентно-ориентирана парадигма, наречена **OReL^A (Optimizing Resource Locations – агентно-ориентирана версия)**. Представен е реинженерингов подход, чрез който се реализира агентно-ориентираната версия OReL^A, използвайки основата на обектно-ориентираната версия OReL^O. Описана е архитектурата на OReL^A, базирана на три взаимодействащи си агента. Подробно са представени проведените експерименти с входни данни, аналогични на тези, описани в трета глава.

В авторската справка са отбелязани накратко основните резултати, получени в дисертационния труд.

В перспективите за развитие са посочени възможни бъдещи насоки за продължаване на изследванията по темата на дисертационния труд.

В апробацията на резултатите са изброени международни и национални конференции, на които са докладвани някои от резултатите в дисертационния труд.

Също така са отбелязани научните проекти, по които са изследвани и получени част от резултатите в дисертационния труд.

Библиографията съдържа 156 заглавия на научни статии и монографии, които авторът е използвал при изследванията, свързани с дисертационния труд.

КРАТКО СЪДЪРЖАНИЕ НА ДИСЕРТАЦИОННИЯ ТРУД

ГЛАВА 1. СЪСТОЯНИЕ НА ИЗСЛЕДВАНИЯ ПРОБЛЕМ

В тази глава е направен кратък преглед на състоянието на научно-изследователските области, свързани с проведеното в дисертационния труд изследване.

В съвременните изследвания на еволюционните стратегии и генетичните алгоритми се забелязват изключително интересни тенденции. От една страна, се правят опити техните реализации да се вграждат в обектно-ориентирани и агентно-базирани архитектури. От друга страна, се моделират интересни детайли от биологическата еволюция и живия свят. Трета тенденция е еволюционните стратегии и генетичните алгоритми да се комбинират с други подходи, модели и техники. Така, в специализираната литература се появиха понятия като алгоритми за многокритериална оптимизация; диференциална еволюция (използваща специални репродуктивни механизми), меметични алгоритми (изследва еволюционни модели за предаване на информация и развитието на човешката култура [51]) и културна еволюция, която моделира еволюцията на културата на популацията и нейното влияние върху генетичната и адаптивна еволюция на индивидите; коеволюция и самостоятелно генериране на меметично изчисление, в което индивидите еволюират чрез сътрудничество или конкуренция, или стават годни за оцеляване, чрез помощта на локален механизъм за търсене, обикновено базиран на правила. В дисертационния труд са описани следните методи за търсене, съвременни еволюционни стратегии и генетични алгоритми:

- Еволюиращи агентно-ориентирани симулации;
- Колективна вероятност или разпределени оптимизационни подходи за решаване на проблеми с ограничения;
- Модел на изкуствена имунна система за откриване на ресурси в разпределени среди, базиран на мулти-агентна парадигма;
- Еволюционна оптимизационна рамка за интелигентни агенти;
- Иновативни алгоритми за бързо търсене на образи, базирани на еволюционни агенти;
- Помагащи си агенти с памет или подход, използващ генетично програмирани мрежи;
- Клъстеризиране на методи за оптимизация на разпределението на агенти;
- Търсене в конкурентни пространства;
- Търсене с ограничения.

ГЛАВА 2. АБСТРАКТЕН МОДЕЛ

2.1. RG проблем

В децентрализираната версия на InfoStation мрежа могат да се предават заявки за изпълнение на определени услуги към всяка една от станциите. С това целим

оптимално натоварване на изчислителните ресурси. От гледна точка на ефективната обработка на една заявка идеалният случай е, ако всички необходими за това ресурси са разположени върху същата станция. Тогава казваме, че заявката се обработва в **локален режим**.

В реалния DeLC обаче трябва да очакваме, че в много случаи потребителските заявки за електронни услуги не могат да бъдат обработени изцяло от локалната информационна станция поради различни причини, например съществува недостиг на изчислителни ресурси или необходимите информационни ресурси са разположени на други станции. В тези случаи доставянето на услугата трябва да се **глобализира** по начин, включващ други информационни станции. Необходимостта от глобализация зависи от начина, по който са разположени ресурсите по възлите на мрежата по време на инициализацията на системата. Взимайки под внимание факта, че всяка глобализация изисква допълнително натоварване, проблемът е да се намери такова **разположение на информационните ресурси**, което осигурява ефективна обработка на една заявка. Този проблем ще наричаме **Request Globalization проблем (RG проблем)** [37].

Преди да дадем абстрактно представяне на RG проблема, ще формулираме неформални определения на няколко основни понятия, свързани с глобализацията на заявките.

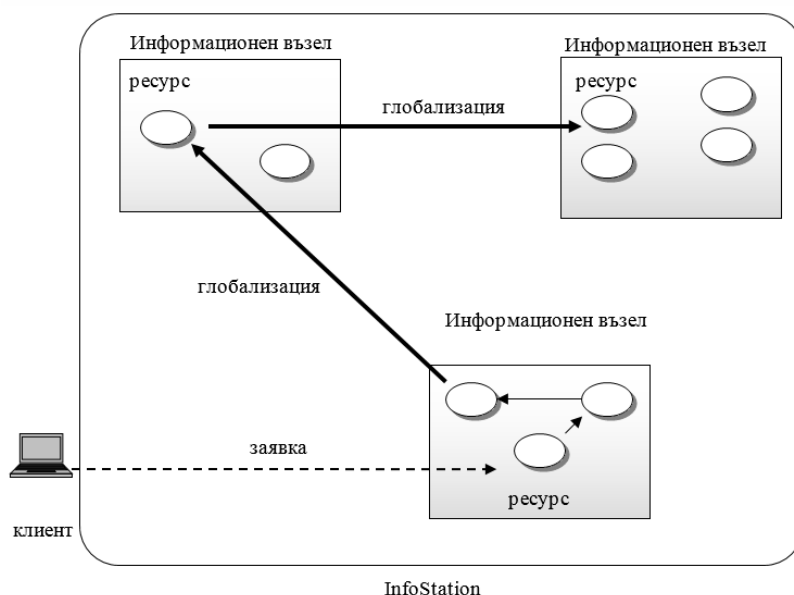
- **Заявка** – информация, изпратена от потребител до системата, която е структурирана съответно правилата за достъп и която предизвиква реакция в системата. Заявката може да бъде изпълнена от възела (локален възел), с който актуално е свързан потребителят. Когато една заявка не може да се изпълни от локалния възел, тя се глобализира и тогава трябва да бъде изпълнена от друг възел или възли.
- **Информационен ресурс** – софтуерен компонент, който може да се активира и използва за обработката на заявки, изпратени на системата. Ресурсите могат да бъдат **активни** (т.е. софтуерни агенти и електронни услуги) или **пасивни** (т.е. бази данни и различни структури от данни). По време на обработката на една заявка могат да възникнат взаимовръзки между определени информационни ресурси.
- **Локален възел** – образователен възел, получил заявката, в обсега на който се намира потребителят.
- **Глобален възел** – образователен възел, различен от локалния, който може да бъде използван при обработката на една заявка. Всички образователни възли на DeLC са потенциално глобални.
- **Глобализация на заявка** – процес на въвличане на глобален възел в изпълнението на една потребителска заявка.
- **Карта на разполагане** – схема на разполагане на информационните ресурси на DeLC върху образователните възли на DeLC.
- **Оптимална карта на разполагане** – карта на разполагане, причиняваща минимален брой глобализации при обработка на една заявка.

За осигуряването на една добра ефективност на опериране на системата стремежът е изпълнението на заявките да се окомплектова по възможност в съответния локален възел, т.е. използвайки само информационните ресурси, които са разположени върху възела, който е приел заявката. При невъзможност за обработка в локален режим стремежът е ефективно изпълнение на заявката, в случая това означава при минимизиран брой на глобализации. Броят на глобализациите зависи основно от

актуално използваната карта на разполагане. Търсенето на оптимална карта на разполагане може да бъде направено на основата на анализа на разпределението на информационните ресурси в InfoStation мрежата [7].

2.2. Подход за решаване на RG проблема

Проблемът за оптимизиране броя на глобализациите в общия случай не е решим с приемливи изчислителни ресурси. Причината за това е, че с линейното нарастване на броя на информационните ресурси пространството на решенията нараства експоненциално и суперекспоненциално. Поради тази причина се отказваме да търсим глобален оптимум (над цялото множество). Задоволяваме се с евристични трансформации, които при запазване на еквивалентността последователно подобряват актуалната карта на разполагане (DM), като редуцират необходимите глобализации. Един такъв подход предвижда разпознаване на определени конфигурации на информационни ресурси и заместването им с еквивалентни структури с по-висока ефективност.

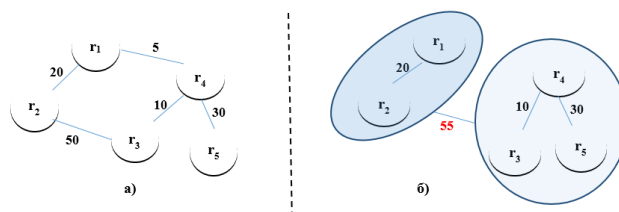


Фигура 2. Пример за изпълнение на заявка

При решаване на RG проблема различаваме два вида оптимизиране:

- **локално оптимизиране** – търси се оптимизиране на времето за изпълнение в рамките на един образователен възел. За тази цел е необходимо да се анализират съдържателните (семантични) взаимосвързки между разположените в него информационни ресурси;
- **глобално оптимизиране** – търси се оптимизиране в рамките на цялата InfoStation мрежа, когато се изисква глобализация на заявка.

Въпреки че на пръв поглед глобалното оптимизиране е от съществено значение, не можем да пренебрегнем проблемите, свързани с локалното оптимизиране. Локалните връзки между ресурсите, разположени в един и същ образователен възел, ни интересуват, понеже те са потенциални глобализации. Например нека е дадено едно множество от информационни ресурси $IRSET = \{r_1, \dots, r_5\}$, разположени в един образователен възел и локалните връзки са оценени както е дадено на Фигура 3 а). При разделяне на ресурсите и разполагането им върху два образователни възела, както е показано на Фигура 3 б), локалните връзки (r_1, r_4) и (r_2, r_3) става глобализация, оценена със сумата от двете локални оценки.



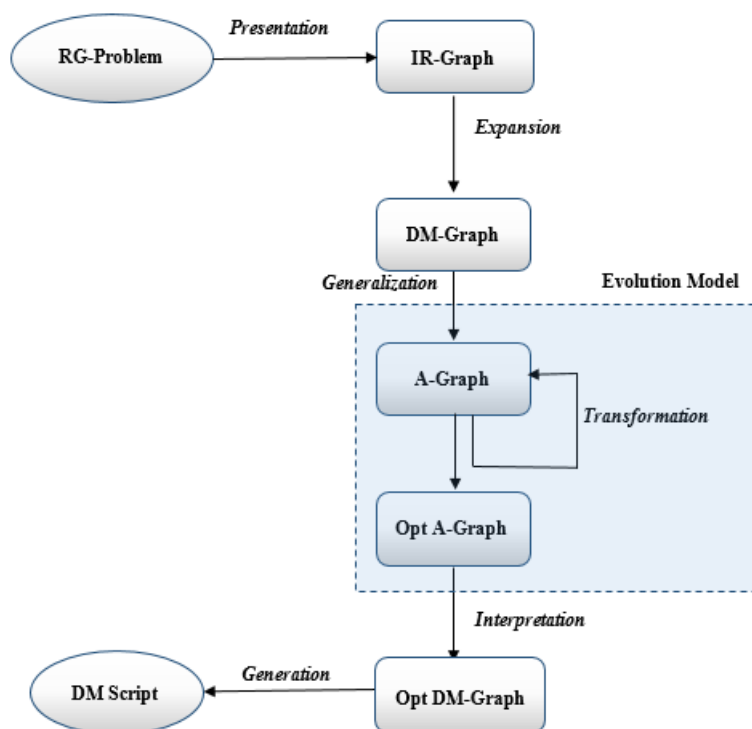
Фигура 3. Потенциални глобализации

Следващ проблем е видът на представяне, което ще използваме за представяне на RG проблема. В дисертационния труд се използват графи. Графите са добро средство за моделиране на структурни и функционални взаимовръзки между елементи на различни постановки на задачи, включително и голяма група от оптимизационни задачи. Принципната схема за използване на графи за решаване на оптимизационни проблеми включва следните две стъпки:

- търси се начин за представяне на оптимизационния проблем чрез подходящ граф;
- търсят се оптималните елементи на този граф (най-преки пътища, минимум разгърнати дървета, подграфи, удовлетворяващи различни условия, и др.) или се генерира множество от подобни графи, в което се търси оптимален граф (например различни разлагания на един граф).

В дисертационния труд предлагаме подход, който се базира на следната схема:

- търсим подходяща евристика за определяне на истинско подмножество от възможни карти на разполагане на информационните ресурси;
- предполагаме, че в това подмножество съществува задоволително решение;
- търсим оптимум в рамките на новото множество (локален оптимум или субоптимум).



Фигура 4. Диаграма на подхода

Предлагаме подход за решение на RG проблема, който включва следните стъпки (Фигура 4):

- формално представяне на RG проблема;
- **Presentation** – формалният RG проблем се представя като граф (IR-Graph);
- **Expansion** – IR-Graph се разлага, в резултат на което получаваме негов породен граф, който моделира картата на разполагане (DM-Graph);
- **Generalization** – DM-Graph се обобщава (по-високо ниво на абстракция). Това се прави по следните две причини:
 - предложеният подход може да се обобщи за решаване не само на RG проблема, но и на по-широк клас проблеми, например, генериране на оптимални препокриващи структури на програми, оптимизиране на достъпа до хранилища с данни, оптимално окабеляване на производствени халета;
 - опростява прилагането на еволюционни алгоритми.
- **Transformation** – върху A-Graph се прилага итеративен еволюционен алгоритъм. При намиране на задоволително решение се генерира графа Opt A-Graph.
- **Interpretation** – Opt A-Graph се интерпретира в термините на картата на разполагане и като резултат се генерира кореспондиращ Opt DM-Graph;
- **Generation** – от Opt DM-Graph се генерира скрипт, описващ търсената карта на разполагане.

2.3. Представяне на картата на разполагане

В този параграф ще дадем формално представяне на картата на разполагане на информационните ресурси, което ще използваме за реализиране на описания по-горе подход.

- $IRSET = \{r_i \mid r_i - \text{информационни ресурси}\}$ – множество на наличните информационни ресурси;
- $IS = \{1, 2, \dots, s\}$ – множество на InfoStations, като приемаме, че всеки InfoStation е разположен на отделен физически сървър;
- $eLN(IRSET) = \{eLN_i \mid eLN_i \subseteq IRSET \vee eLN_i = \emptyset, i = 1, \dots, l\}$ – множество на образователните възли, като приемаме, че всеки образователен възел е разположен върху един InfoStation);
- $R = r_1 r_2 \dots r_t \dots$ – последователност на достъп до ресурсите, където r_t е идентификатор на ресурс, а t е виртуален момент на достъп. Ако r_t -ят ресурс не е в актуалния eLN , тогава казваме, че е възникнала ситуация на необходимост от глобализация на заявката;
- $DM = \{eLN_1, \dots, eLN_{ni}, eLN_{ni+1}, \dots, eLN_p\}$ – карта на разполагане на образователните възли върху InfoStation мрежата;
- $eLN_i \neq \emptyset, i = 1, \dots, ni$, ni ще наричаме ниво на DM;
- $eLN_i = \emptyset, i = ni+1, \dots, p$, $k = p-ni$ ще наричаме k -кратна резерва на DM;
- $\forall r_i \in IRSET (\exists eLN \in DM, r_i \in eLN_i)$;

Важно е да можем да различаваме дали две карти на разполагане са **идентични**. Нека са дадени DM и DM':

- $DM \equiv DM'$, когато $ni^{DM} = ni^{DM'} = ni \wedge \forall eLN_i \in DM (\exists eLN_j' \in DM': eLN_i = eLN_j')$, $i=1, \dots, ni$; $j=1, \dots, ni$.

Множеството на всички възможни разгръщания (разположения) ще обозначаваме с DM . В съответствие с dm , всяко подмножество от ресурси ще бъде разположено върху отделен възел. Освен това изискваме всеки две такива подмножества да бъдат дизюнктивни, така че формално dm дефинира разлагане на множеството IS на отделни подмножества.

Наредено разлагане на множеството IR на p части или накратко p -разлагане на IR се нарича всяка p -торка $(x_1, x_2, \dots, x_p)$ две по две непресичащи се подмножества на IR , обединението на които дава IR . Тези подмножества x_i за $i=1, 2, \dots, p$ можем да разглеждаме като информационните ресурси, разположени върху един информационен възел (станция). Като използваме теорема от комбинаториката „Нека E е множество с n елемента и $k \geq 1$, то съществува точно k^n на брой k -разлагания на E “, можем да заключим, че IR може да се разложи на k^n подмножества, където n е броят на елементите на IR , а k е броят на елементите на IS . В нашия случай едно такова разлагане на елементите на IR ще бъде разположено върху елементите на IS .

Съществуването на огромен брой възможни разположения е обективна основа за оптимизация на глобализациите на заявка – задача, която може да се формулира по следния начин: от множеството DM върху IR да се намери dm такова, че една заявка да се обработи с минимален брой глобализации. Процесът на RG оптимизацията може формално да се представи като:

- $rgnumber: DMSET \rightarrow N^+$ – функция, която връща броя глобализации за всяко възможно разпределение на ресурсите;
- $select: DMSET \rightarrow \{true, false\}$ – функция, която определя условията за допустимост. Едно DM се нарича допустимо, ако $select(DM) = true$.

Целевата функция на RG оптимизацията може да се представи, както следва: да се намери DM^* такова, че:

- DM^* е допустимо разположение;
- $rgnumber(DM^*) = \min\{rgnumber(DM)\}$

$$DM^* \in DMSET$$

$$DM \in DMSET$$

Представената тук дефиниция на възможните разположения на ресурсите върху отделните информационни станции характеризира значението на смислена интерпретация на информационните ресурси, като се взимат под внимание само статични фактори. Такава статична характеристика е достатъчна за задаване на рамка на оптимизационния проблем. От нея обаче не може да се изведе директно практически метод за решаване на проблема. Изпълнението на потребителска заявка се повлиява и от някои динамични фактори. За търсенето на практическо решение на проблема са съществени два въпроса:

- Как можем да допълним статичното характеризиране на разположенията?
- Тъй като е невъзможно да се изследва пълното множество на разположенията, какъв критерий можем да използваме, за да намерим реално подмножество, в което да търсим оптималното разположение?

2.4. Граф на информационните ресурси

Графът на информационните ресурси (IR -Graph) служи за представяне на потенциалните възможности за използване на ресурси при обработка на заявки, т.е. два

информационни ресурса могат да бъдат свързани, ако съществува потенциална възможност единият да използва другия при изпълнение на определена заявка.

$$\text{IR-Graph} = (\text{IRSET}, \text{IRE}),$$

където:

- IRSET – множество на информационните ресурси;
- $\text{IRE} = \text{IRSET} \times \text{IRSET}$ – множество от ребрата на графа такова, че $(r_i, r_j) \in \text{IRE}$, ако в структурата на r_i съществува поне едно извикване (достъп) към r_j .

В нашия случай статичният граф на информационните ресурси IR-Graph не е достатъчен за решаване на RG проблема. Необходима е допълнителна информация, с помощта на която могат да бъдат оценени възлите и ребрата. Оцененият граф дефинираме както следва:

$$\text{EvIR-Graph} = (\text{IR-Graph}, \text{COST}),$$

където:

- IR-Graph – статичен граф на информационните ресурси;
- $\text{COST} = \{\text{cost}_i \mid i = 1, \dots, k\}$ – множество от различни функции за оценка на възлите и ребрата на IR-Graph .

В актуалната версия на модела ще дефинираме следните две оценъчни функции:

- $\text{length}: \text{IRSET} \rightarrow \mathbf{N}^+$ – оценява размера на информационните ресурси;
- $\text{calls}: \text{IRE} \rightarrow \mathbf{N}^+$ – оценява броя на извикванията (достъп) на ресурси. Например $\text{calls}(r_i, r_j)$ е оценката на извикванията (използванията, достъпа) на ресурса r_j от r_i .

EvIR-Graph представя статичното разположение на информационните ресурси върху възлите на InfoStation мрежата. Освен това осигурява информация за съществуващите взаимовръзки между информационните ресурси. При обработка на потребителските заявки обаче могат да възникнат определени ситуации, зависещи от интензивността на връзките между отделните ресурси, които да влияят върху необходимото време за изпълнение. За отчитане на тези динамични отношения се нуждаем от нов вид граф, който се получава като разлагане на EvIR-Graph , както е показано в следващия параграф.

2.5. Граф на картата на разполагане

Графът на картата на разполагане (DM-Graph) моделира разполагането на информационните ресурси върху отделните образователни възли. DM-Graph е породен граф на оценения граф на ресурсите. Той характеризира напълно динамичното поведение на използваната карта на разполагане при обработка на една заявка и като такъв представлява структурата, подлежаща на оптимизация.

2.6. Трансформация върху DM-Graph

Както вече беше споменато в параграф 2.2, за решаване на RG проблема търсим трансформация, с помощта на която последователно трансформираме DM-Graph до достигане на приемливо решение. В този параграф ще представим тази трансформация. Нека е даден следният $\text{DM-Graph} = (\text{ELNODES}, \text{RGE}, \text{RGCOST})$. Дефинираме трансформация върху този граф както следва:

$$\text{dmtr}: \text{ELNODES} \times \text{IRSET} \times \text{ELNODES} \rightarrow \text{DM},$$

така че при $\text{dmtr}(e\text{LN}_q, r^*, e\text{LN}_z)$:

- $eLN_q \in ELNODES$, $eLN_z \in ELNODES$, $r^* \in eLN_q$;
- $DM-Graph' = dmtr(DM-Graph)$ е граф с:
 - $ELNODES' = \{eLN_1', eLN_2', \dots, eLN_q', \dots, eLN_z', \dots, eLN_p'\}$, където:
 - $eLN_q' = eLN_q / r^*$;
 - $eLN_z' = eLN_z \cup r^*$;
 - $eLN_i' = eLN_i$, $\forall i \in \{1, \dots, q-1, q+1, \dots, z-1, z+1, \dots, p\}$
- RGE' – множество на ребрата, което е образувано по следната изчислителна схема:
 1. Задраскване на ребра:
Когато $eLN_q = \{r^*\}$, тогава всички $(eLN_i, eLN_q) \in RGE$;
Ако $eLN_q' \in NIL'$ и $eLN_q' \notin DM'$, тогава:
 - a. Всички $(eLN_i, eLN_q) \in RGE$, с $rgs((eLN_i, eLN_q)) = calls(r_i, r^*)$, $r_i \in eLN_i$;
 - b. Всички $(eLN_q, eLN_i) \in RGE$, с $rgs((eLN_q, eLN_i)) = calls(r^*, r_i)$, $r_i \in eLN_i$;
 2. Добавяне на нови ребра:
Когато $eLN_z \in NIL$, тогава $(eLN_i', eLN_z') \in RGE'$;
 $\forall (r_i, r^*) \in IRE$, $r_i \in IRSET$;
 $eLN_z' \in RGE'$;
 $(eLN_i', eLN_z') \in RGE'$, когато $(eLN_i, eLN_z) \notin RGE$ и $\exists (r_i, r^*) \in IRE$, $r_i \in eLN_i$;
 3. Останалите елементи на RGE се приемат автоматично в RGE' .

2.7. Еволюционен модел

В предишния параграф показвахме, че RG проблемът може да се представи като специални графи. Графът на картата на разположение DM-Graph се получава посредством „разлагане“ на IR-Graph на разположените върху образователните възли на DeLC информационни ресурси. В този параграф ще представим еволюционен модел за решаване на RG проблема, който използва представянето като графи. Моделът използва абстрактен граф, наречен A-Graph.

2.7.1. Разлагане на абстрактен граф

Разлагането на един абстрактен граф може да бъде представено формално, както е дадено в този подпараграф. Нека е даден насоченият оценен граф $G = (V, A, F)$, където:

- V – множество на възлите;
- $A = V \times V$ – множество на ребрата;
- F – функционал (множество от функции), където:
 - $fv \in F: V \rightarrow \mathbf{R}$ е оценъчна функция на възлите;
 - $fa \in F: A \rightarrow \mathbf{R}$ е оценъчна функция на ребрата.

Дадено е също едно положително число p такова, че $0 < fv(i) \leq p$, $\forall i \in V$.

Като **k-кратно разлагане** (k цяло число, по-голямо от 1) на G дефинираме множеството $\{v(1), \dots, v(k)\}$, където:

$$\bigcup_{m=1}^k v(m) = V \wedge (v(n) \cap v(m) = \emptyset, \forall n, m = 1, \dots, k; n \neq m)$$

Едно разлагане ще наричаме **допустимо**, ако

$$\sum_{i \in v(m)} f_v(i) \leq p, \quad \forall m = 1, \dots, k$$

Едно ребро (i, j) ще наричаме **външно**, когато i и j принадлежат към различни $v(m)$, $m=1, \dots, k$. В противен случай реброто (i, j) ще наричаме **вътрешно**.

Разходи на едно разлагане ще дефинираме както следва:

$$\sum_{i \in v(m), j \in v(n)} f_a(i, j), \quad m, n = 1, \dots, k \wedge (i, j) - \text{външни ребра}$$

При така дефинираното разлагане на графа G общият комбинаторен проблем може да бъде формулиран като: **търси се допустимо разлагане на G с минимални разходи**.

2.7.2. Еволюционна стратегия

Възможно решение на преформулирания проблем е методът, представен в [1], който включва следните стъпки:

1. генерираме k -кратно разлагане на G ;
2. опитваме да редуцираме разходите на два подграфа, като разменяме определени възли от тях;
3. втората стъпка се повтаря, докато разлагането стане **по двойки оптимално**, където двойката $(v(i), v(j))$, $v(i), v(j) \subseteq V$ се нарича **по двойки оптимална**, когато $v(i)$ и $v(j)$ са елементи от едно допустимо разлагане и разходите на **външните** ребра между $v(i)$ и $v(j)$ са минимални.

Подвойковата оптималност не е идентична с глобалната оптималност, но в много случаи е едно добро приближение. Когато два възела (подграфа) в разлагането са направени оптимални, това променя подвойковата оптималност в останалите елементи на разлагането, което поражда необходимост от следващи размени. Използвайки този метод за нашия случай, размените могат да се извършват локално, но същевременно да имаме глобален поглед върху тяхната целесъобразност.

Опирайки се върху идеите на този метод, предлагаме еволюционна стратегия за решаване на проблема. Еволюционната стратегия е директен стохастичен метод за търсене, който представя опростена имитация на биологическата еволюция. Основните стъпки на тази стратегия са следните:

1. **Начална стъпка:** от пространството на търсене се избира допустима начална структура E_{old} . Изчисляваме стойността на нейната целева функция $Z(E_{old})$.
2. **Мутационна стъпка:** посредством случайна вариация на параметрите на E_{old} се генерира нова структура E_{new} , при което не се използва информация за поведението на целевата функция.
3. **Селективна стъпка:** изчисляваме $Z(E_{new})$. Ако $Z(E_{new})$ е „по-добра“ от $Z(E_{old})$, тогава $E_{old} \leftarrow E_{new}$.
4. **Управляваща стъпка:** контролираме дали E_{old} е „удовлетворително решение“. Ако не е, тогава се инициира в мутационната стъпка нова итерация. В противен случай методът завършва.

2.7.3. Генериране на субоптимално разлагане на графи

В този подпараграф ще представим използването на еволюционната стратегия за получаване на субоптимално разлагане на абстрактния граф G . Множеството на всички

възможни разлагания на абстрактния граф G ще означим с $EXPSET(G)$. Ще използваме еволюционната стратегия за определяне на едно подмножество $E(Z_a) \subset EXPSET(G)$, в което ще търсим оптимално разлагане. Трябва да бъде гарантирано също, че $E(Z_a)$ съдържа поне едно допустимо разлагане. $E(Z_a)$ ще наричаме **стохастична среда**, а оптималния елемент на тази среда – **субоптимум**.

Сега можем да представим конструктивен алгоритъм, с помощта на който можем да създадем търсеното множество $E(Z_{old})$:

1. **Начало:** случайно генерираме едно допустимо разлагане $Z_{old} \in EXPSET(G)$.
2. **Мутация:** конструираме една среда на началното разлагане Z_a . Елементите на тази среда създаваме като използваме следния изчислителен метод, използващ трансформацията, описана в параграф 2.6:
 - Нека е дадено едно k -кратно разлагане $Z = \{v(1), \dots, v(k)\}$;
 - Трансформацията $tr(v(q), e^*, v(z))$ е дефинирана по следния начин:
 - $v(q) \in Z$ ще наричаме първоизточник, а $v(z) \in Z$ – цел;
 - $e^* \in v(q)$, така че tr отстранява e^* от първоизточника и го приема в целта.
 - Различаваме два гранични случая:
 - $v(q) = \{e^*\}$ – тогава трансформацията генерира $(k-1)$ -кратно разлагане;
 - $v(z) = \emptyset$ – тогава се генерира $(k+1)$ -кратно разлагане.
 - Ще въведем понятието допустимост за трансформация. Една трансформация е допустима, когато:
 - генерира допустими разлагания;
 - $k < k_{max}$, където k е мощността на разлаганията, а k_{max} е предварително зададена цялочислена константа.
 - Елементите на средата $E(Z_{old})$ се генерират по следния начин:
 - чрез мутация (mutation): еднократно прилагане на трансформацията върху Z_{old} : $Z_{new} = tr(Z_{old})$;
 - чрез кръстосване (cross over): двукратно прилагане на трансформацията върху Z_{old} : $Z_{new} = tr(tr(Z_{old}))$;
3. **Селекция:** средата се разширява, когато е генерирано ново разлагане Z_{new} , което е „по-добро“ от Z_{old} . В такъв случай Z_{new} се обявява за начално и следващата трансформация се прилага върху него. При провеждане на оптимизирането средата се променя непрекъснато – колкото повече „добри“ разлагания са генерирани, толкова „по-широка“ става средата.

2.7.4. Оценка на еволюционната стратегия

Основните характеристики на предложената еволюционна стратегия са:

- тъй като е евристичен метод, не може да се докаже, че тя намира глобален оптимум – задоволяваме се с локален оптимум;
- за да може да бъде приложена, тя изисква наличието на начално решение;
- в най-лошия случай стратегията не разваля началното решение.

Заклучение

Както може да се види от горните дефиниции, множеството на възможните разположения нараства много бързо с увеличаване на броя на информационните

ресурси. Например възможните разположения за 20 информационни ресурса, разположени само върху два възела на InfoStation мрежата, са над 10^6 . В общия случай проблемът за намиране на оптимално разполагане не е решим с помощта на приемливи изчислителни ресурси (например компютърно време и памет). Поради тази причина търсенето на глобален оптимум се оказва неподходящо. Вместо това в дисертацията е разработен подход, при който:

- предлагаме подходяща евристика за ограничаване на пространството на картите на разполагане – така получаваме едно подмножество, което може да бъде изследвано с помощта на предложената еволюционна стратегия;
- задоволяваме се с търсене на оптимум в новото множество, който ще бъде локален – не можем да докажем, че е намерен глобален оптимум;
- експериментите с разработените прототипи могат да демонстрират коректността на евристиката – доставят като резултат възможни карти на разполагания, които оптимират RG проблема.

ГЛАВА 3. ОБЕКТНО-ОРИЕНТИРАНА РЕАЛИЗАЦИЯ НА МОДЕЛА

OReL (Optimizing Resource Locations) е програмната реализация на теоретичния модел, представен във втората глава на дисертационния труд. Програмният компонент е създаден в две версии [8]:

- OReL^O – обектно-ориентирана версия, реализирана на езика за програмиране Java [44] в средата Eclipse. Този софтуер може да се интегрира в обектно-ориентирани приложения, които трябва да решават такъв тип оптимални задачи (описан в предишната глава);
- OReL^A – агентно-ориентирана версия, реализирана на езика за програмиране Java в средата JADE [45,3]. Тази версия на компонента ще бъде интеграционна част от виртуалното образователно пространство.

В тази глава ще бъде представена детайлно версията OReL^O. OReL^A е описана в следващата глава на дисертацията.

3.1. Базови структури на OReL^O

В обектно-ориентираната реализация на предложения теоретичен модел информационните ресурси се представят като обекти със следните основни атрибути:

- **име на ресурса** – служи за разграничаването на различните възли и като ключ за достъп до обекта. Обектите се съхраняват в структура от тип HashMap, като достъпът до тях се осъществява посредством двойката (ключ, стойност). Основно предимство на тази структура е ускоряване на търсенето в сравнение, например със списъчни структури.
- **позиция** – използва се за съхраняване информация за местоположението на ресурса в InfoStation мрежата. Това поле съдържа цяла положителна стойност, съответстваща на така наречената „точка на достъп“ (InfoStation), която характеризира местоположението на съответния ресурс.
- **връзки на ресурса** – списък, съдържащ обекти от тип „връзка“, включително и тежест на връзката (цяло положително число).

За подобряване ефективността на алгоритъма, реализиращ еволюционната стратегия, е дефинирана помощна матрична структура. Матрицата моделира актуалното разполагане на ресурсите в InfoStation мрежата. За да се избегне повторение на информация и излишни изчисления се използва триъгълна матрица (матрица, чиито

елементи под главния диагонал са 0). Инициализирането на матрицата (представя началното разпределение на ресурсите) е еднократно, в началото на алгоритъма. Подреждането на информацията е в зависимост от броя на възлите на графа, т.е. при 9 възела се използва 9x9 матрица, както следва:

- (n,m) – елементите на матрицата (където $n \leq$ броя на възлите, $n+1 < m \leq$ броя на възлите) съдържат информация за връзката между n -тия и m -тия възел, като:
 - съдържат 0, ако двата възела нямат връзка;
 - съдържат положително число, равно на тежестта на връзката им, ако тя е глобална (ако се намират на различни InfoStations);
 - съдържат отрицателно число, равно на отрицанието на тежестта на връзката им, ако тя е локална (ако се намират на един InfoStation);
- (n,n) – елементите на матрицата (тъй като един възел не може да има връзка със себе си, използваме главния диагонал, за да съхраняваме допълнителна информация, която улеснява алгоритъма) съдържат стойността на полето „позиция” на n -тия възел;
- останалите елементи (елементите под главния диагонал) са равни на 0.

Върху матрицата се извършват всички изчислителни процедури, предвидени в алгоритъма, реализиращ еволюционната стратегия. Например оценката на глобалните връзки става, като се съберат стойностите на тежестите на глобалните връзки, т.е. сумата на положителни числа над главния диагонал. За да се направи оптимизация на разпределението без загуба на информация за предишното разпределение се използва копие на оригиналната матрица (неин наследник), което се обработва с CrossOver алгоритъм (кръстосване).

Същността на този алгоритъм се състои в размяната на позициите на два случайно подбрани ресурса. След тази размяна съвсем естествено е връзките на тези ресурси да се променят, т.е. от глобални някои да са станали локални и обратно. За да се реализира това, върху матрицата наследник се прилага алгоритъм, оценяващ актуалните глобализации (сборът от тежестите на глобалните връзки). Целта е намаляване на тази оценка. Ако оценката на наследника е число по-малко от това на родителя, казваме, че има подобрение на разпределението; в противен случай – нямаме промяна (резултатът е по-лош или същият). При подобрение родителят се унищожава, наследникът заема мястото му и същият алгоритъм се прилага отново до достигане на локален оптимум. При неуспех наследникът отново става копие на матрицата родител и се повтаря същият процес на оценка. За да се избегне зацикляне на програмата и за да се въведе условие за край (при достигане на локален оптимум), алгоритъмът използва брояч на неуспешните итерации. Ако дадена итерация е „подобрене”, тогава този брояч се нулира; в противен случай се увеличава с единица. Когато тази променлива достигне граничната стойност, определена от потребителя, програмата прекратява работата си. Последната матрица родител моделира едно субоптимално разпределение на ресурсите.

3.2. Архитектура на OReL⁰

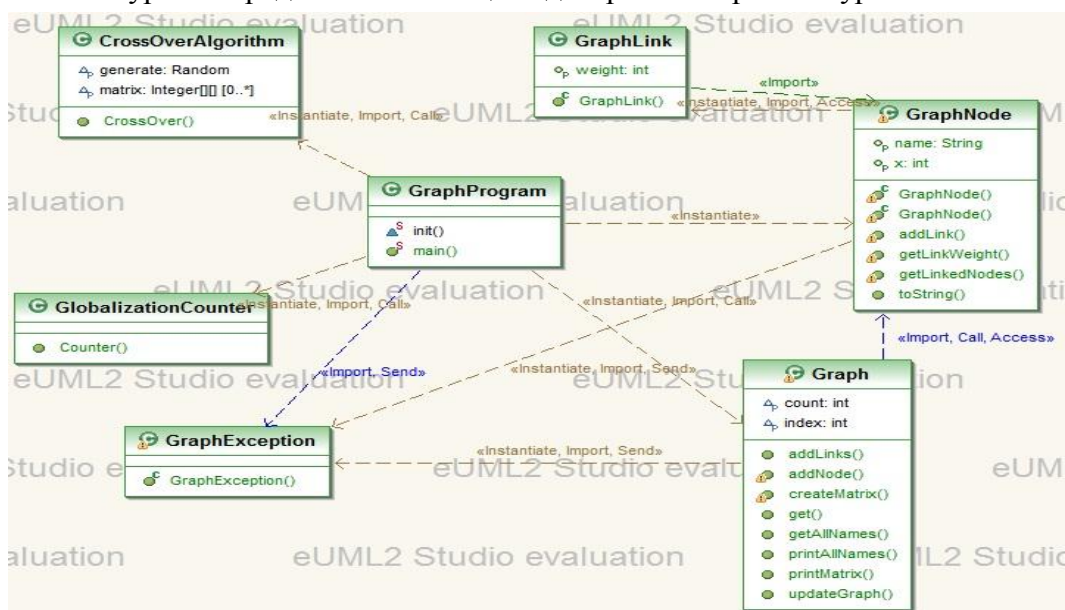
Архитектурата на OReL⁰ се състои от седем базови модула, реализирани като Java класове.

- **Graph** – този клас е своеобразен контейнер, съдържащ множество от възли (GraphNodeS). Тези възли се съхраняват в структурата HashMap (ключ-

стойност структура), за да могат да бъдат достъпвани по име, като по този начин се осигурява по-бързо търсене.

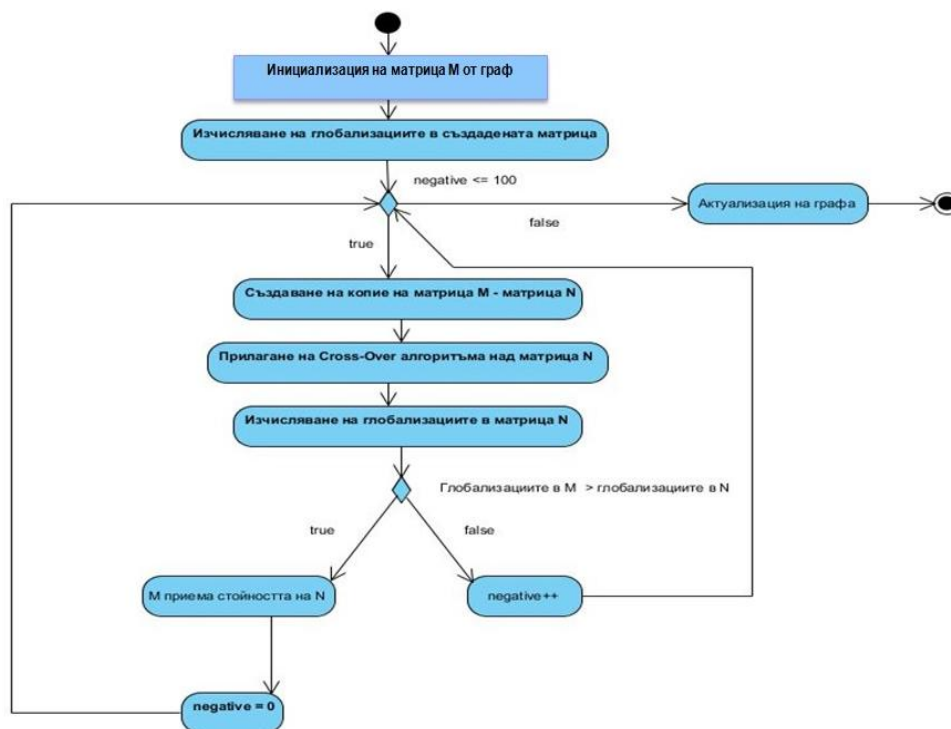
- **GraphNode** – този клас обособява потребителски тип, отговарящ на възлите в един граф, а методите му описват допустимите действия на този тип. Полета на класа са: име (String name) – това е името на възела; връзки (ArrayList links) – списък с връзките на възел към други възли; x (int x) – позиция на възела (индекс на станцията, на която се намира съответстващият на възела ресурс).
- **GraphLink** – Този клас се използва за обозначаване на възлите, които са свързани с даден възел. Класът има две полета: свързан възел (GraphNode relatedNode) – обект, съдържащ информацията за възел, свързан с текущия; тежест на връзката (int weight) – съдържа стойността на тежестта на връзката.
- **GraphException** – този клас е наследник на вградения в Java клас Exception, т.е. връща съобщение под формата на грешка. Класът има само един метод, който е конструктор с име GraphException(). Конструкторът приема един параметър от тип String с име msg.
- **CrossOverAlgorithm** – този клас се използва за размяна на позициите на два възела. Това става като се вземат два произволни възела и се разменят техните елементи за позиция в генерираната матрица. Например ако са дадените възли x и y, разположени съответно на станции k и l – $x \in k$, $y \in l$, то техните позиции се разменят и съответно $x \in l$ и $y \in k$. Класът има две полета: generate от тип Random – използва се за генериране на произволно число в определени граници; matrix от тип Integer[][] – записва се копие от оригиналната генерирана от графа матрица.
- **GlobalizationCounter** – този клас се използва за брояч на глобалните връзки и техните тежести.
- **GraphProgram** – това е основният клас на програмата, който задейства алгоритъма. В него се инициализират всички възли и връзките между тях, както и се правят проверки на получените резултати.

На Фигура 5 е представена обобщена диаграма на архитектурата на OReL⁰.



Фигура 5. Обща клас-диаграма на $OReL^0$

Обща диаграма на алгоритъма, реализиращ еволюционната стратегия, е представена на Фигура 6.



Фигура 6. Еволюционен алгоритъм

В началото на алгоритъма се инициализира графът, който ще бъде оптимизиран. Чрез извикване на `init()` метода се инициализират ръчно въведените в него данни за всеки един възел и неговите връзки. След приключване на инициализацията от графа се извличат необходимите за изграждане на матрицата *M* данни. Използвайки матрицата *M*, изчисляваме актуалните глобализации (чрез методите на класа `GlobalizationCounter`). За реализиране на `CrossOver` алгоритъма се използва копие на матрицата. В конкретния случай условие за край са 100 негативни итерации, т.е. 100 итерации без подобрение на разпределението в матрицата. Ако условието `negative <= 100` е изпълнено, то тогава се създава копието – матрица *N*, след което преминава през `CrossOver` алгоритъма и се изчисляват глобализациите в новополучената матрица. Ако глобализациите в нея са по-малко от тези в матрицата *M*, то имаме по-добро разпределение, вследствие на което матрица *M* се презаписва с данните от матрица *N* и `negative = 0`. Ако глобализациите в матрица *N* са повече от тези в матрица *M*, то тогава нямаме подобрение в разпределението и старата матрица се запазва и `negative` се увеличава с единица. След достигане на условието за край (`negative > 100`) цикълът приключва работата си и графът се обновява с новата информация за разпределението на възлите/ресурсите.

3.3. Провеждане на експерименти с OReL⁰

За провеждане на експериментите бяха избрани четири различни набора (размери на популацията – РП) входни данни (Таблица 1). За всеки един от тях разпределението на ресурсите по InfoStation-ите е на случаен принцип – единствено е следвано условието за равномерно разпределение. Създадените връзки между ресурсите също са на случаен принцип за всеки от наборите. За сравнение на влиянието и на броя InfoStations са проведени експерименти с всеки от наборите съответно за 3

InfoStations и 5 InfoStations, като не са променяни създадените връзки между ресурсите, а само тяхното разпределение (отново на случаен принцип).

	РП 1	РП 2	РП 3	РП 4
Размер на популацията	9 ресурса	15 ресурса	30 ресурса	45 ресурса

Таблица 1. Таблица на размери на популацията

Резултатите от експеримента са обобщени в таблицата на Таблица 2.

	Условие за край (неуспешни итерации)	100 итерации	1000 итерации	5000 итерации	10000 итерации
9 ресурса 3 InfoStations	Начална стойност на глобалните връзки	17	17	17	17
	Локален оптимум	10	10	10	14
9 ресурса 5 InfoStations	Начална стойност на глобалните връзки	27	27	27	27
	Локален оптимум	16	16	16	16
15 ресурса 3 InfoStations	Начална стойност на глобалните връзки	51	51	51	51
	Локален оптимум	35	37	36	35
15 ресурса 5 InfoStations	Начална стойност на глобалните връзки	70	70	70	70
	Локален оптимум	52	50	50	49
30 ресурса 3 InfoStations	Начална стойност на глобалните връзки	96	96	96	96
	Локален оптимум	51	49	45	52
30 ресурса 5 InfoStations	Начална стойност на глобалните връзки	120	120	120	120
	Локален оптимум	65	68	67	61
30 ресурса 10 InfoStations	Начална стойност на глобалните връзки	112	112	112	112
	Локален оптимум	71	58	58	62
45 ресурса 3 InfoStations	Начална стойност на глобалните връзки	154	154	154	154
	Локален оптимум	57	44	49	45
45 ресурса 5 InfoStations	Начална стойност на глобалните връзки	175	175	175	175
	Локален оптимум	88	69	59	65
45 ресурса 10 InfoStations	Начална стойност на глобалните връзки	178	178	178	178
	Локален оптимум	112	79	80	93

Таблица 2. Резултати от експеримента с OReL^o

ГЛАВА 4. АГЕНТНО-ОРИЕНТИРАНА РЕАЛИЗАЦИЯ НА МОДЕЛА

OReL^A е наименованието на софтуерен компонент, реализиращ представения във втора глава теоретичен модел като мулти-агентна система. OReL^A се създава като част от виртуалното образователно пространство. Активните компоненти на пространството са интелигентни автономни, реактивни, интерактивни и проактивни компоненти, реализирани като агенти, които са в състояние да взаимодействат помежду си [8].

4.1. Реинженерингова схема

Идеята е OReL^A да се реализира на основата на обектно-ориентираната версия OReL^O посредством реинженерингов подход. Алгоритъмът позволява кода да се декомпозира на отделни функционални модули, които да бъдат капсулирани в агентна архитектура в съответствие с технологията JADE, следвайки следната схема:

- **Основният клас на програмата**, извършващ трансформацията на входните данни в еволюционната матрица, ще бъде реорганизиран като самостоятелен агент.
- **Стартирането на метода** (както и самият метод) за прилагане на еволюционна стратегия ще се вгради в агент, наречен **CrossOver**.
- **Агентът**, който ще приеме в себе си оценъчния алгоритъм, т.е. ще извършва мутация, ще наричаме **Rating агент**.
- Трите агента ще работят в един контейнер и ще могат да комуникират помежду си чрез съобщения – при необходимост агентите могат да разменят като съдържание на съобщенията еволюционната матрица (или части от нея).

4.2. Обща архитектура на OReL^A

Агентно-ориентираната система OReL^A включва три взаимодействащи си агенти: Transformer агент, Rating агент и CrossOver агент.

Transformer агент има две поведения. Едното конструира еволюционна матрица от граф, представящ информационните ресурси, разположени върху InfoStation мрежата. Това поведение използва два класа – GraphFactory и MatrixFactory. Резултатът е обект от тип EnvironmentModel, съдържащ матрицата, който може да се изпрати към Rating агента за оценяване. Rating агентът от своя страна връща оптимизираната матрица на Transformer агента за обновяване на графа.

Основните функции на Transformer агента са следните:

- извлича данни от граф, съдържащ информация за разпределението и връзките на ресурсите в една InfoStation мрежа;
- генерира и актуализира еволюционна матрица, предоставяща възможност за ефективно прилагане на еволюционен алгоритъм за търсене на оптимално разпределение;
- актуализира графа с оптимизираното разпределение на ресурсите.

Rating агент реализира селекционната стъпка на теоретичния модел, използвайки класа RatingAlgorithm. Агентът избира популацията, която ще се използва в следващата еволюционна стъпка. Матрицата, представяща актуалното субоптимално разпределение, се изпраща към Transformer агента.

Rating агентът има много важна роля в процеса за разпределение на информационните ресурси. Той изпълнява алгоритъма за оценка и сравнение и има няколко основни задачи:

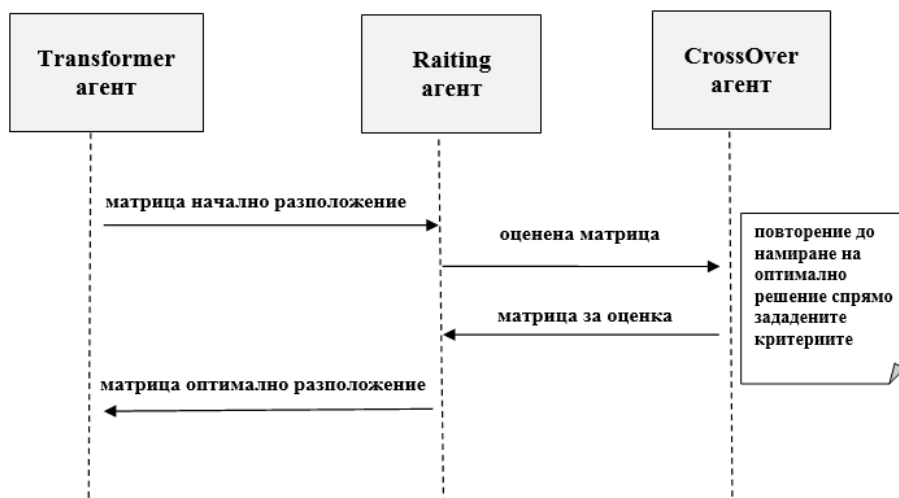
- да запомни текущото състояние на околната среда;
- да поиска ново представяне от CrossOver агента;
- да направи оценка на текущото и новото представяне на средата;
- да сравни новото представяне с текущото и да определи кое от двете е по-добро;
- да запише по-доброто представяне като текущо и да определи кога процесът на еволюция да приключи.

CrossOver агент получава съобщения единствено от Rating агента. В съобщенията се съдържа матрицата, върху която CrossOver агентът прилага еволюционни оператори (мутация и кръстосване). Новополучената матрица се поставя в обект от тип EnvironmentModel и се изпраща на Rating агента, където се извършва селекция.

CrossOver агентът изпълнява алгоритъм за реализиране на еволюционни оператори. Агентът се състои от три части:

- основен клас (Agent клас);
- поведение за получаване на матрица;
- поведение за изпращане на матрица.

Последователността от размяна на съобщения между трите агента може да бъде онагледена с диаграма на Фигура 7.



Фигура 7. Sequence диаграма на размяна на съобщения между разглежданите агенти

4.3. Провеждане на експерименти с OReL^A

Целите на проведените общо четири опита са аналогични на тези, дадени в предишната глава 3. За целите на експеримента е разработен агентно-ориентиран прототип, с помощта на който са проведени планираните експерименти. Този прототип притежава аналогични характеристики с тези на обектно-ориентираната реализация. За да могат да се направят коректни сравнения между двете версии експериментите са проведени със същата хардуерна конфигурация.

В експериментите са използвани същите четири набора входни данни, представени в Таблица 1. За всеки един от тях разпределението на ресурсите по InfoStations и на създадените връзки между ресурсите са на случаен принцип. Използват се също 3 и 5 InfoStations. Резултатите от проведените опити са обобщени в таблица 19, представена в тази глава от дисертационния труд.

	Условие за край (неуспешни итерации)	100 итерации	1000 итерации	5000 итерации	10000 итерации
9 ресурса 3 InfoStations	Начална стойност на глобалните връзки	17	17	17	17
	Локален оптимум	10 10 при OReL ^o	10 10 при OReL ^o	10 10 при OReL ^o	14 14 при OReL ^o
9 ресурса 5 InfoStations	Начална стойност на глобалните връзки	27	27	27	27
	Локален оптимум	16 16 при OReL ^o	16 16 при OReL ^o	16 16 при OReL ^o	16 16 при OReL ^o
15 ресурса 3 InfoStations	Начална стойност на глобалните връзки	51	51	51	51
	Локален оптимум	33 38 при OReL ^o	35 34 при OReL ^o	34 34 при OReL ^o	34 34 при OReL ^o
15 ресурса 5 InfoStations	Начална стойност на глобалните връзки	67	67	67	67
	Локален оптимум	48 45 при OReL ^o	45 45 при OReL ^o	46 47 при OReL ^o	45 52 при OReL ^o
30 ресурса 3 InfoStations	Начална стойност на глобалните връзки	96	96	96	96
	Локален оптимум	54 52 при OReL ^o	45 43 при OReL ^o	45 44 при OReL ^o	46 46 при OReL ^o
30 ресурса 5 InfoStations	Начална стойност на глобалните връзки	120	120	120	120
	Локален оптимум	80 70 при OReL ^o	69 63 при OReL ^o	64 58 при OReL ^o	66 62 при OReL ^o
30 ресурса 10 InfoStations	Начална стойност на глобалните връзки	143	143	143	143
	Локален оптимум	91 93 при OReL ^o	86 84 при OReL ^o	86 82 при OReL ^o	82 83 при OReL ^o
45 ресурса 3 InfoStations	Начална стойност на глобалните връзки	154	154	154	154
	Локален оптимум	44 66 при OReL ^o	45 50 при OReL ^o	41 56 при OReL ^o	45 43 при OReL ^o
45 ресурса 5 InfoStations	Начална стойност на глобалните връзки	175	175	175	175
	Локален оптимум	94 75 при OReL ^o	75 61 при OReL ^o	67 62 при OReL ^o	74 68 при OReL ^o
45 ресурса 10 InfoStations	Начална стойност на глобалните връзки	178	178	178	178
	Локален оптимум	125 97 при OReL ^o	81 85 при OReL ^o	78 81 при OReL ^o	83 84 при OReL ^o

Таблица 3. Резултати от експеримента с OReL^A

4.4. Обобщение на резултатите

Резултатите от проведените експерименти с компонентите OReL^O и OReL^A могат да бъдат обобщени както следва:

- При увеличаването на размера на входните данни се влошава и качеството на решението на алгоритмите.
- При увеличаването на размера на входните данни се увеличава и времето за изпълнение на алгоритмите.
- И двата алгоритъма при увеличаване на негативните итерации увеличават и времето за изпълнение на програмата. При по-малки входни данни може да се намали броят на негативните итерации. При по-големи входни данни е необходимо да се увеличат негативните итерации поради увеличаването на вариациите при кръстосване и размяна.
- АОП алгоритъмът доставя по-добри оптимизации в сравнение с ООП алгоритъма поради следните причини:
 - мобилност на агентите;
 - лесно вграждане на агентите в голяма система;
 - лесно пренаписване на Transformer агента с цел свеждане на друг тип данни до матрица.

Предимствата на агентно-ориентирания алгоритъм имат своята „цена“. От направените тестове се вижда, че АО алгоритъмът е по-бавен в сравнение с ОО алгоритъма. От изпълнението на теста с 45 информационни ресурса и 10 станции при 10000 негативни итерации се вижда, че ОО алгоритъмът се изпълнява за 281 милисекунди. Същият тест, направен чрез АО алгоритъма, се изпълнява за 30 секунди. По наше мнение времето за изпълнение на двата алгоритъма е бързо, съответно бързодействието на АО алгоритъма не представлява проблем.

АВТОРСКА СПРАВКА ЗА ПРИНОСИТЕ

В дисертационния труд са изследвани проблеми, свързани с разработване на формален модел за намиране на оптимално разпределение на информационни ресурси върху разпределени изчислителни конфигурации от InfoStation тип (регионална локална компютърна мрежа). За разлика от стандартната версия в DeLC се използва разширена InfoStation архитектура, която позволява информационните станции да оперират не само като посредници между мобилните устройства на потребителите и централния сървър (върху който могат да се разполагат различни приложения, електронни услуги и глобални бази данни), но също върху тях се разполагат и локално управляват различни информационни ресурси. С едно такова децентрализирано разположение на услугите и другите информационни ресурси върху всички възли може да се постигне добро балансиране на натоварване на мрежата и с това по-добра производителност. Същевременно, децентрализираното разполагане на информационни ресурси има един съществен недостатък. Системното натоварване, свързано с взаимовръзките на информационните ресурси по време на изпълнение на една заявка е особено тежко, когато свързаните ресурси са разположени на различни информационни станции. Така задачата за оптимизиране на разполагането на информационните ресурси става особена важна за ефективното функциониране на приетия в DeLC децентрализиран подход. Степента на натоварване зависи от начина на разпределение на информационните ресурси.

Приносите на автора, представени в дисертационния труд, могат да бъдат обобщени както следва:

1. Създаден е теоретичен модел на разполагането на информационните ресурси върху разпределена InfoStation мрежова архитектура, наречен Request Globalization проблем (RG проблем).
2. Създадена е еволюционна стратегия за намиране на оптимално разполагане на информационните ресурси върху разпределени изчислителни конфигурации (InfoStations), която използва представянето на информацията като графи. Еволюционната стратегия е интегрирана в теоретичния модел.
3. Разработен е програмен прототип на теоретичния модел и еволюционната стратегия с обектно-ориентирана архитектура, наречен OReL^O.
4. Разработен е програмен прототип на теоретичния модел и еволюционната стратегия с агентно-ориентирана архитектура, наречен OReL^A.

Междинни резултати и отделни части от работата са публикувани в 5 публикации и са представени и в отчетите на 2 научноизследователски проекта, в които е участвал авторът.

Следната таблица показва връзката между резултатите, структурата на дисертацията и направените публикации:

Резултат	Задача	Публикация	Глава
1	1	1	II
2	2	3	II
3	3	4, 5	III
4	4	4, 5	IV

Таблица 4. Връзка между резултати, структура на дисертационния труд и публикации

За бъдещото развитие на получените в дисертацията резултати предвиждаме следните насоки:

- Написване на оценъчен алгоритъм реализиран в самообучаващ се агент – той също ще е важна част за постигането на добри резултати при оптимизацията на разпределение на ресурси. Целта му е да събира необходимата информация, за да се направи статистическо проучване и заключение коя услуга е най-често използвана, за да се оптимизира разпределението спрямо нея;
- Изграждане на GUI (Graphical User Interface) – подходящият интерфейс би помогнал за достъпността на прототипната реализация. Чрез него въвеждането на необходимата информация ще става лесно и бързо, без да се изискват допълнителни познания за кода на прототипа. Извежданите резултати биха били доста по-прегледни и подредени, вследствие на което и по-лесни за разбиране;
- Реализиране на други подобни алгоритми решаващи оптимизационни задачи, като „Алгоритъма на мравките“ и „Теория на грубите множества“. На базата на разгледания Еволюционен алгоритъм и на предложените алгоритми би

могъл да се реализира агентно-ориентиран модул, решаващ различни оптимизационни задачи. Този модул може да се интегрира в Разпределения център за електронно обучение DeLC и във Виртуално образователно пространство (ВОП).

ПУБЛИКАЦИИ НА АВТОРА

1. Stoyanov, S., I. Ganchev, I. Popchev and **I. Dimitrov**, “Request Globalization in an InfoStation Network”, *Compt. Rend. Acad. Bulg. Sci*, vol. 63, no. 6, 2010, pp. 901 – 908, ISSN 1310-1331, **(IF: 0.219)**.
2. Стоицов, Г., **И. Димитров** и О. Рахнева, „Представителен слой на услугата за търсене в DeLC“, *Юбилейна научна конференция '2007*, Смолян, 30.11 – 1.12.2007, 43 – 52, ISBN: 978-954-8767-24-8.
3. **Dimitrov, I.**, S. Stoyanov and I. Popchev, “Optimal Resource Deployment in an InfoStation-Based Network”, *Proc. of International Conference REMIA 2010*, University of Plovdiv, 2010, pp. 247 – 253, ISBN: 978-954-423-648-9
4. Ganchev, I., S. Stoyanov, D. Meere, **I. Dimitrov** and M. O’Droma, “The Infostation Paradigm – Past, Present and Future Developments“, *Proc. of International Conference “From DeLC to VelSpace”*, Plovdiv, Bulgaria, 26 – 28 March 2014.
5. **Dimitrov, I.**, V. Valkanov and A. Stoyanova-Doycheva, “Optimizing the globalizations in InfoStation architecture“, *Proc. of International Conference “From DeLC to VelSpace”*, Plovdiv, Bulgaria, 26 – 28 March 2014.

БЛАГОДАРНОСТИ

Искам да изкажа своите благодарности към хората, без които резултатите от този дисертационен труд не биха били възможни.

На първо място благодаря на научния си ръководител проф. д-р Станимир Стоянов за непрекъснатата подкрепа, ценни съвети, градивни критики и разбира се насоките за правия път.

Също така бих искал да благодаря на колегите си, чиито труд също допринесе съществено за резултатите от моята работа.

Не на последно място благодаря на семейството си за подкрепата и търпението, което имаха към мен през цялото време.

БИБЛИОГРАФИЯ

- [1] Allemang, D. и J. Hendler, *Semantic Web for the Working Ontologist.*: Elsevier, 2011, ISBN 978-0-12-385965-5.
- [2] Ashton, K. , "That 'Internet of Things' Thing", *RFID Journal*, 2009.
- [3] Bellifemine, F. L., G. Caire и D. Greenwood, *Developing Multi-Agent Systems with JADE.*: Wiley, 2007.
- [4] Berners-Lee, T. , J. Handler и O. Lassila, "The Semantic Web", *Scientific American* , no. 284, pp. 34-43, May 2001.
- [6] Dieter, U. , H. Mark и M. Florian, "Architecting the Internet of Things", *Springer*, 2011, ISBN 978-3-642-19156-5.
- [7] Dimitrov, I. , S. Stoyanov и I. Popchev, "Optimal Resource Deployment in an InfoStation-Based Network" в *International Conference REMIA 2010*, University of Plovdiv, 2010, pp. 247-253.
- [8] Dimitrov, I. M., V. Valkanov и A. Stoyanova-Doycheva, "Optimizing the globalizations in InfoStation architecture" в *International conference "From DeLC to VelSpace"*, Plovdiv, Bulgaria, 26-28 march 2014.
- [9] Frenkiel, R. H. и T. Imielinski, "InfoStations: The joy of 'many-time, many-where' Communications", *WINLAB, Technical Report TR-119*, Apr. 1996.
- [11] Ganchev, I. , S. Stojanov, D. Meere и M. O'Droma, "InfoStation-based mLearning System Architectures: Some Development Aspects" в *The 8th IEEE International Conference on Advanced Learning Technologies (ICALT'08)*, Santander, Spain, 2008, pp. 504-505.
- [12] Ganchev, I. , S. Stojanov, M. O'Droma и D. Meere, "An InfoStation-Based University Campus System Supporting Intelligent Mobile Services", *Journal of Computers*, vol. 2, no. 3, pp. 21-33, 2007.
- [13] Ganchev, I. , S. Stoyanov, D. Meere, I. Dimitrov и M. O'Droma, "THE INFOSTATION PARADIGM – PAST, PRESENT AND FUTURE DEVELOPMENTS" в *International conference "From DeLC to VelSpace"*, Plovdiv, Bulgaria, 26-28 march 2014.
- [14] Ganchev, I. , S. Stoyanov и M. O'Droma, "Consumer-oriented DeLC Service Architecture" в *In Proc. of the 3rd International Conference on Education and Information Systems, Technologies and Applications (EISTA 2005)*, Orlando, Florida, USA, ISBN 980-6560-34-5 , 2005, pp. 213-218.
- [15] Ganchev, I. , S. Stoyanov, V. Valkanova и M. O'Droma, "Service-oriented and Agent-based Architecture Supporting Adaptable Context-Aware Provision of Mobile e-Learning Services" в *IADIS e-Learning 2010 Conference*, Freiburg, Germany, 26 - 29 July 2010, pp. 97-104.
- [16] Glushkova, T. и A. Stoyanova, "Interaction and adaptation to the specificity of the subject domains in the system for e-Learning and distance training DeLC" в *International Scientific Conference "Informatics in scientific knowledge"*, June 2008, Varna Free University "Chernorizets Hrabar".

- [22] Rahneva, O. , A. Rahnev и N. Pavlov, "Functional Workflow and Electronic Services In a Distributed Electronic Testing Cluster – DeTC" в *Proceedings 2nd International Workshop on eServices and eLearning*, Otto-von-Guericke Universitaet Magdeburg, ISBN 3-929757-76-1 , 2004, pp. 147-157.
- [23] Rahnev, A. , N. Pavlov и O. Rahneva, "Architecture & Design of Distributed Electronic Testing Cluster (DeTC) based on Microsoft.NET Framework" в *IMAPS CS International Conference*, Brno, Czech Republic, ISBN 80-214-2990-9 , 2005, pp. 417-422.
- [24] Rahnev, A. и O. Rahneva, "Algorithms for Generation of Circuitries and Drafts in Distributed e-Testing Cluster (DeTC)", *Scientific Works*, vol. 35, no. Book 3, 2007 – Mathematics, pp. 85-100, 2007, ISSN 0204–5249.
- [26] Stoyanov, S. , "Context-Aware and Adaptable eLearning Systems", STRL, De Montfort University, Leicester, UK, PhD Thesis 2012.
- [27] Stoyanov, S.; Ganchev, I.; O'Droma, M.; Zedan, H.; Meere, D.; Valkanova, V., "Semantic Multi-Agent mLearning System" in *Semantic Agent Systems: Foundations and Applications*, A. Elci, M. T. Kone, and M. A. Orgun, Eds.: Springer Verlag, 2011, vol. 344, ISBN: 978-3-642-18307-2.
- [28] Stoyanov, S.; Popchev, I.; Doychev, E.; Mitev, D.; Valkanov, V.; Stoyanova-Doycheva, A.; Valkanova, V.; Minov, I., "DeLC Educational Portal", *Cybernetics and Information Technologies (CIT)*, vol. 10, no. 3, pp. 49-69, 2010.
- [29] Stoyanova-Doycheva, A. , "A Software Refactoring Process and the Supported Tools" в *Fourth International Conference for Informatics and Information Technology*, Bitola, Macedonia, 11-14 Dec 2003, pp. 70-77, ISBN 9989-668-45-0.
- [30] Stoyanova-Doycheva, A. , "Development of Refactoring Learning Environment" в *Research and Education in Mathematics, Informatics and their Applications*, Plovdiv, Bulgaria, December 2010.
- [31] Stoyanova, A. и T. Glushkova, "Model for implementation of interactive distance learning" в *International Scientific Conference "Informatics in scientific knowledge"*, Varna, June 2010, Varna Free University "Chernorizets Hrabar".
- [32] Stoyanov, S. , G. Cholakov, V. Valkanova и M. Sandalski, "Personalized, Reactive and Proactive Providing of e-Learning Services" в *EdiLib Conference , AACE E-Learn 2011 – World Conference on E-Learning in Corporate, Government*, Honolulu, Hawaii, USA, 17-21 October, 2011.
- [33] Stoyanov, S. , E. Doychev, A. Stoyanova-Doycheva, V. Valkanova и V. Valkanov, "Education Cluster Supporting eTesting and eLearning in Software Engineering" в *2nd Annual International Conference on Web Technologies & Internet Applications*, Bali, Indonesia, 2012, pp. 23-28.
- [34] Stoyanov, S. , I. Ganchev, I. Popchev, M. O'Droma и V. Valkanova, "Agent-Oriented Middleware for InfoStation-based mLearning Intelligent Systems" в *5th IEEE International Conference on Intelligent Systems IS'10*, London, IEEE Catalog Number: CFP10802-CDR, ISBN:978-1-4244-5164-7, Library of Congress:2009934065 , 07.07 – 09.07.2010, pp. 91-95.
- [35] Stoyanov, S. , I. Ganchev, D. Mitev, V. Valkanov и M. O'Droma, "Service-oriented and

- Agent-based Architecture Supporting Adaptable, Scenario-based and Context-aware Provision of Mobile e-Learning Services", *International Journal of Computer Information Systems and Industrial Management Applications*, vol. 3, pp. 771-779, 2011, ISSN 2150-7988.
- [36] Stoyanov, S. , I. Ganchev, M. O'Droma, D. Mitev и I. Minov, "Multi-Agent Architecture for Context-Aware mLearning Provision via InfoStations" в *The 5th International Conference on Soft Computing as Transdisciplinary Science and Technology*, Paris, 2008, pp. 549-552.
- [37] Stoyanov, S. , I. Ganchev, I. Popchev и I. Dimitrov, "Request Globalization in an InfoStation Network", *Compt. Rend. Acad. Bulg. Sci*, vol. 63, no. 6, pp. 901-908.(IF), 2010.
- [38] Stoyanov, S. , I. Ganchev, I. Popchev и M. O'Droma, "An Approach for the Development of a Context-Aware and Adaptive eLearning Middleware" in *Intelligent Systems: From Theory to Practice*, V. Sgurev et al., Ed. Berlin Heidelberg: Springer-Verlag, 2010, pp. 519-535, ISBN: 978-3-642-13427-2.
- [39] Stoyanov, S. , I. Ganchev, I. Popchev и M. O'Droma, "An Approach for the Development of InfoStation-Based eLearning Architectures", vol. 61, no. 9, pp. 1189-1198, 2008.
- [40] Stoyanov, S. , I. Ganchev, I. Popchev, M. O'Droma и R. Venkov, "DeLC – Distributed eLearning Center" в *1st Balkan Conference in Informatics BCI'2003*, Thessaloniki, Greece, ISBN 960-287-045-1 , 2003, pp. 327-336.
- [41] Stoyanov, S. и I. Popchev, "Evolutionary Development of an Infrastructure Supporting the Transition from CBT to e-Learning", *Cybernetics and Information Technologies (CIT)*, vol. 2, pp. 101-114, 2006, Bulgarian Academy of Sciences.
- [42] Stoyanov, Stanimir , Asya Stoyanova-Doycheva и Mihail Chigrichenko, "Improving the creativity in software engineering education with Refactoring Learning Environment" в *Second International Conference Creativity and Inovation in Software engineering*, Ravda, Bulgaria, 2009.
- [43] Stoyanov, S. , V. Valkanova, G. Cholakov и M. Sandalski, "Education Portal for Reactive and Proactive Service Provision" в *The Third International Conference on Advanced Cognitive Technologies and Applications (COGNITIVE 2011)*, Rome, Italy, September 25-30, 2011.
- [44] Sun,. (2014, May) <<http://java.sun.com/products/rmi-iiop/>>
- [45] TILAB,. (2014, May) TILAB. <<http://jade.tilab.com>>
- [46] Zedan, H. и др., "Mapping Human Creativity" in *STRL Internal Monograph (STRL-2008-09)*. Leicester, UK: De Montfort University, 2008.
- [47] Вълканов, Веселина , "Изследване за ВОП в средното училище", ИИКТ, София, 2013.
- [48] Вълканова, В. , С. Стоянов, Х. Зедан и И. Попчев, "Модел за изследване на креативното мислене и действие на ученици" в *39-та пролетна конференция на СМБ*, Албена, 2-6 Април, 2010, pp. 274-280.

- [49] Глушкова, Т. , "Адаптивна среда за електронно обучение в средните училища", ФМИ на ПУ "П. Хилендарски", Дисертационен труд 2011.
- [50] Дойчев, Емил , *Среда за електронни образователни услуги*. Пловдив: Пловдивски университет "П. Хилендарски", 2013.
- [51] (2013, Oct.) Меметика. <<http://bg.wikipedia.org/wiki/меметика>>
- [52] Орозова, Д. , С. Стоянов и И. Попчев, "Виртуално образователно пространство" в *Научна конференция с международно участие „Знанието – традиции, иновации, перспективи”*, Бургас, 2013, приета за печат.
- [53] Рахнева, О. , "Разпределен клъстер за електронно тестване", Пловдивски университет, дисертация 2006.
- [54] Стоицов, Г. , И. Димитров и О. Рахнева, "Представителен слой на услугата за търсене в DeLC" в *ЮБИЛЕЙНА НАУЧНА КОНФЕРЕНЦИЯ'2007*, Смолян, 30.11-1.12.2007.
- [55] Стоянова-Дойчева, А. , "Автоматизирано средство за навигация на процеси за refactoring" в *Национална научна конференция "Информатиката в научното познание"*, юни 2004, Варненски Свободен Университет "Черноризец Храбър".
- [56] Стоянова-Дойчева, А. , "Дефиниране на процес и средства за рефакторинг в обучението по софтуерни технологии", Пловдивски университет, дисертационен труд 2011.
- [57] Чолаков, Г. , "Хибридна архитектура за изграждане на Разпределен център за електронно обучение (DeLC)", Пловдивски университет, дисертация 2013.