

УНИВЕРЗИТЕТ У БЕОГРАДУ  
МАТЕМАТИЧКИ ФАКУЛТЕТ

Оливера С. Станчић

**ПРОБЛЕМИ ХАБ МАКСИМАЛНОГ  
ПОКРИВАЊА – НОВИ МАТЕМАТИЧКИ  
МОДЕЛИ И МЕТОДЕ РЕШАВАЊА**

докторска дисертација

Београд, 2018.

UNIVERSITY OF BELGRADE  
FACULTY OF MATHEMATICS

Olivera S. Stančić

**HUB MAXIMAL COVERING PROBLEMS -  
NOVEL MATHEMATICAL MODELS AND  
SOLUTION METHODS**

Doctoral Dissertation

Belgrade, 2018.

**Ментор:**

др Зорица СТАНИМИРОВИЋ, ванредни професор  
Универзитет у Београду, Математички факултет

**Чланови комисије:**

др Зорица СТАНИМИРОВИЋ, ванредни професор  
Универзитет у Београду, Математички факултет

др Милан ДРАЖИЋ, ванредни професор  
Универзитет у Београду, Математички факултет

др Татјана ДАВИДОВИЋ, виши научни сарадник  
Математички институт САНУ, Београд

**Датум одбране:**

*Мојим родитељима*

# *Захвалнице*

Захваљујем се свом ментору проф. др Зорици Станимировић на пруженој шанси, несебичној помоћи и стручном усмеравању ка циљу. Током нашег заједничког рада, уз њену велику посвећеност и разумевање, стекла сам непроцењива искуства и знање које ће ми користити у даљем напредовању. Јако сам срећна што сам имала задовољство да сарађујем са њом. Хвала члановима комисије, др Татјани Давидовић и проф. др Милану Дражићу, на корисним запажањима и сугестијама којим су допринели побољшању квалитета ове дисертације. Велику захвалност дугујем свом пријатељу др Раци Тодосијевићу који је својим оптимистичким ставом и увек корисним саветима у значајној мери допринео мом стручном усавршавању. Хвала мојим родитељима и сестри што су ми пружали несебичну подршку, помагали и оправдавали моја честа одсуства са породичних окупљања током досадашњег школовања. Хвала им што разумеју и подржавају моју жељу за напредовањем. Посебно хвала мом оцу који ме је први увео у свет математике и уз чију помоћ сам од детињства решавала „компликоване” задатке. Неизмерно хвала мом мужу и сину што су ми својом позитивном енергијом уливали додатну снагу, враћали осмех на лице и тиме ми у великој мери помагали приликом писања ове дисертације. Захваљујем се и осталим члановима породице, пријатељима и колегама који су саветима, подршком и разумевањем били извор моје мотивације за напредовањем.

**Наслов дисертације:** Проблеми хаб максималног покривања – нови математички модели и методе решавања

**Резиме:** Хаб локацијски проблеми (енгл. *Hub Location Problems - HLP*), због бројних примена у разним областима реалног живота, представљају важну класу проблема оптимизације. Често се срећу у практичним ситуацијама које захтевају успостављање протока од чворова-извора (снабдевача) до чворова-одредишта (корисника) при задатим условима, тако да вредност одређене функције циља буде оптимална. Хабови су посебни објекти (чворови у мрежи) који представљају центре консолидације и колекције протока између две изабране локације – снабдевача и корисника. Како је цена транспорта између хабова по јединици количине нижа у односу на цену између осталих чворова, усмеравање протока на мрежу хабова води ка снижавању транспортних трошкова у мрежи.

Предмет ове докторске дисертације је једна класа хаб локацијских проблема, у литератури позната под називом *проблеми хаб максималног покривања* (енгл. *Hub Maximal Covering Problems - HMCPs*). Циљ ових проблема је одређивање оптималних локација за успостављање унапред задатог броја хабова, тако да укупни проток међу свим покривеним паровима снабдевач-корисник буде максималан, уз претпоставку бинарног или парцијалног покривања. У дисертацији су разматране три варијанте проблема хаб максималног покривања: проблем  $p$ -хаб максималног покривања неограничених капацитета са једноструким алокацијама (енгл. *Uncapacitated Single Allocation  $p$ -hub Maximal Covering Problem – USA $p$ HMCP*), проблем  $p$ -хаб максималног покривања неограничених капацитета са вишеструким алокацијама (енгл. *Uncapacitated Multiple Allocation  $p$ -hub Maximal Covering Problem – UMA $p$ HMCP*) и проблем  $p$ -хаб максималног покривања неограничених капацитета са  $r$ -алокацијском шемом (енгл. *Uncapacitated  $r$ -allocation  $p$ -hub Maximal Covering Problem – UrA $p$ HMCP*). UrA $p$ HMCP до сада није разматран у литератури. За сва три проблема, показано је да припадају класи NP-тешких проблема оптимизације. У случају USA $p$ HMCP, за задати скуп хабова, добијени потпроблем оптималне алокације не-хаб чворова успостављеним хабовима је такође NP-тежак.

У овој дисертацији, предложени су нови математички модели за USA $p$ HMCP са бинарним и парцијалним покривањем. Главна предност нових модела, у односу на постојеће моделе из литературе, огледа се у чињеници да се нови модели малим модификацијама лако трансформишу у нове моделе за проблеме  $p$ -хаб

максималног покривања са различитим алокацијским шемама. Прецизније, из предложених модела за USApHMCP произилазе нови модели за UMApHMCP и UrapHMCP, у оба случаја покривања. Сви предложени модели за USApHMCP и UMApHMCP су у смислу ефикасности упоређени са одговарајућим моделима из литературе, коришћењем егзактног решавача CPLEX 12.6. Приликом поређења ових формулација, у нумеричким експериментима је коришћено неколико хаб скупова података из литературе. Добијени експериментални резултати показују да нови модели UMApHMCP за оба концепта покривања показују најбоље перформансе, у смислу квалитета решења и времена извршавања. За UrapHMCP и оба концепта покривања, предложена су три математичка модела, која су упоређена у смислу ефикасности коришћењем егзактног решавача CPLEX 12.6. Испоставља се да егзактни решавач проналази оптимална или допустива решења само на инстанцама проблема малих димензија.

Имајући у виду сложеност сва три разматрана проблема и резултате добијене егзактним решавачем CPLEX 12.6, закључак је да егзактне методе не дају решења за инстанце проблема већих димензија. Из тог разлога, неходно је имплементирати адекватне хеуристичке или метахеуристичке методе које у кратком времену извршавања обезбеђују квалитетна решења инстанци проблема великих димензија. До сада су у литератури предложене једноставне, али недовољно ефикасне, хеуристичке методе за решавање проблема USApHMCP и UMApHMCP са бинарним покривањем, док варијанте HMCP са парцијалним покривањем нису до сада решаване метахеуристичким методама. Како UrapHMCP са бинарним и парцијалним покривањем до сада није разматран у литератури, не постоје предложене методе за његово решавање. Имајући у виду бројне примере успешне примене методе променљивих околина (VNS) на различите хаб локацијске проблеме, ова метахеуристика је примењена на разматране HMCP проблеме. У овој дисертацији, дизајнирано је и имплементирано неколико варијанти VNS метахеуристике: општа метода променљивих околина (енгл. *General Variable Neighborhood Search - GVNS*) за USApHMCP, основна метода променљивих околина (енгл. *Basic Variable Neighborhood Search - BVNS*) за UMApHMCP и варијанта опште методе променљивих околина (GVNS-R) за UrapHMCP. У случају UrapHMCP, предложене су још две метахеуристике: похлепна стохастичко-адаптивна процедура претраге са методом променљивог спуста (енгл. *Greedy Randomized Adaptive Search Procedure with Variable Neighborhood Descent - GRASP-VND*) и генетски алгоритам (енгл. *Genetic Algorithm*

- GA). Конструктивни елементи свих предложених метахеуристика су прилагођени карактеристикама разматраних проблема.

Извршена су тестирања на хаб скуповима података из литературе, која укључују инстанце проблема величине до 1000 чворова. Добијени резултати показују да предложене метахеуристике за разматране проблеме достижу сва позната оптимална решења претходно добијена CPLEX 12.6 решавачем или успостављају нова најбоља позната решења за значајно краће време извршавања, у поређењу са CPLEX 12.6 решавачем. Предложене GVNS и BVNS метахеуристике брзо достижу сва позната оптимална решења при тестирању инстанци проблема малих димензија редом за USArHMCP и UMArHMCP. У случају инстанци проблема великих димензија, које до сада нису тестиране у литератури за разматране проблеме, предложене GVNS и BVNS метахеуристике налазе решења доброг квалитета, за кратко време извршавања. Резултати тестирања предложених GVNS-R и GRASP-VND метахеуристика за UArHMCP на инстанцама проблема већих димензија указују на њихову ефикасност, у оба случаја покривања. Предложена GA метода се показала успешном само за решавање UArHMCP у случају бинарног покривања, на инстанцама проблема величине до 200 чворова.

Варијанте проблема хаб максималног покривања, које су разматране у овој дисертацији, од великог су теоријског и практичног значаја. Нови математички модели предложени у овој дисертацији за разматране варијанте НМСП, представљају научни допринос теорији хаб локацијских проблема, математичком моделирању и оптимизацији. Развијене и имплементиране метахеуристике за решавање посматраних варијанти НМСП представљају научни допринос области метода оптимизације за решавање локацијских проблема, као и развоју софтвера. Разматране варијанте НМСП имају бројне примене при оптимизацији телекомуникацијских и транспортних система, превозењу путника и робе у авио-саобраћају, система хитних служби, поштанских и других система испоруке, тако да се резултати добијени у овој докторској дисертацији могу применити у пракси, делимично или у потпуности.

**Кључне речи:** математичко моделирање, проблеми хаб максималног покривања, бинарно и парцијално покривање, метахеуристике, метода променљивих околина, похлепна стохастичко-адаптивна процедура претраге, генетски алгоритам



**Научна област:** Математика

**Ужа научна област:** Оптимизација, Операциона истраживања

**Dissertation title:** Hub maximal covering problems - novel mathematical models and solution methods

**Abstract:** Hub Location Problems (HLP) represent an important class of optimization problems due to their numerous applications in many areas of real life. They often arise from practical situations that require routing of the flow from origin node (supplier) to the destination node (customer) under given conditions, such that the value of considered objective function is optimal. Hubs are special objects (nodes in the network) that represent centres for consolidation and flow collection between two selected locations - suppliers and customers. As transportation costs (per unit of flow) along the links that connect hub nodes are lower compared to other links in the network, directing the flow to hubs may lead to significant reductions of transportation cost in the network.

The subject of this doctoral dissertation is one class of hub location problems, denoted as Hub Maximal Covering Problems (HMCPs) in the literature. The goal of HMCPs is to determine optimal locations for establishing certain number of hubs in order to maximize the total flow between all the covered origin-destination pairs, under the assumption of binary or partial covering. Three variants of the hub maximal covering problem are considered: uncapacitated single allocation  $p$ -hub maximal covering problem (USApHMCP), uncapacitated multiple allocation  $p$ -hub maximal covering problem (UMApHMCP) and uncapacitated  $r$ -allocation  $p$ -hub maximal covering problem (UrApHMCP). Note that the UrApHMCP has not been studied in the literature so far. All three considered problems are proven to be NP-hard. In case of USApHMCP, for the given set of hubs, the obtained sub-problem of optimal allocation of non-hub nodes by established hubs is also NP-hard.

In this dissertation, new mathematical models for USApHMCP with binary and partial covering are proposed. The main advantage of the newly proposed models, in respect to existing ones from the literature, is the fact that small modifications of the new models enable their transformation to new models for  $p$ -hub maximal covering problems with different allocation schemes. More precisely, new models for UMApHMCP and UrApHMCP can be obtained from the newly proposed models for USApHMCP in both coverage cases. All proposed models for USApHMCP and UMApHMCP are compared with the existing ones from the literature in the terms of efficiency within the framework of exact CPLEX 12.6 solver. Several hub data sets from the literature are used in numerical experiments when comparing

the formulations. The obtained experimental results indicate that new models for UMApHMCP with both binary and partial coverage show the best performance in terms of solutions' quality and execution times. For UrApHMCP and both coverage criteria, three mathematical models are proposed, and compared in terms of efficiency using the exact CPLEX 12.6 solver. It turns out that the exact solver finds optimal or feasible solutions only for small-size problem instances.

Having in mind the complexity of all three problems under consideration and the results obtained by CPLEX 12.6 solver, the conclusion is that, in practice, exact methods can not provide solutions for large problem dimensions. For this reason, it was necessary to implement adequate heuristic or metaheuristic methods, in order to obtain high-quality solutions in short execution times, even in the case of large problem dimensions. Up to now, only simple but insufficiently effective heuristic methods for solving USApHMCP and UMApHMCP with binary coverage have been proposed in the literature, while the HMCP variants with partial coverage have not been previously solved by using metaheuristic methods. As UrApHMCP with binary and partial coverage has not been previously considered in the literature, no solution methods suggested for this problem existed up to now. Inspired by previous successful applications of variable neighborhood search method (VNS) to other hub location problems from the literature, this metaheuristic approach is applied to the considered HMCP problems. In this dissertation, several variants of VNS metaheuristic are designed and implemented: General Variable Neighborhood Search (GVNS) for USApHMCP, Basic Variable Neighborhood Search (BVNS) for UMApHMCP and a variant of General Variable Neighborhood Search (GVNS-R) for UrApHMCP. In the case of UrApHMCP, two additional metaheuristic methods are proposed: Greedy Randomized Adaptive Search Procedure with Variable Neighborhood Descent (GRASP-VND) and Genetic Algorithm (GA). Constructive components of all proposed metaheuristics are adapted to the characteristics of the considered problems.

Experimental study was conducted on the existing hub data sets from the literature, which include instances with up to 1000 nodes in the network. The obtained results show that the proposed metaheuristics for the considered problems reach all known optimal solutions previously obtained by CPLEX 12.6 solver or establish new best-known solutions in significantly shorter CPU time compared to CPLEX 12.6. The proposed GVNS and BVNS metaheuristics quickly reach all known optimal solutions on small-size problem instances when solving USApHMCP

and UMApHMCP, respectively. In the case of large-size problem instances, which have not been previously used for testing purposes for these problems, the proposed GVNS and BVNS return their best solutions in short execution times. The results obtained by the proposed GVNS-R and GRASP-VND for UrApHMCP on large-size problem instances indicate their effectiveness in both coverage cases. The proposed GA method showed to be successful only for UrApHMCP in binary covering, on instances up to 200 nodes.

The variants of hub maximal covering problems considered in this dissertation are important from both theoretical and practical points of view. The new mathematical models proposed in this dissertation for the considered variants of HMCP, represent a scientific contribution to the theory of hub location problems, mathematical modeling and optimization. Designed and implemented metaheuristic methods for solving the studied variants of HMCP are the scientific contribution to the field of optimization methods for solving location problems, as well as the development of software. The considered variants of HMCP have numerous applications in the optimization of telecommunication and transport systems, air passenger and goods transport, emergency services, postal and other delivery systems, so that the results obtained in this doctoral dissertation can be applied in practice, partially or completely.

**Keywords:** mathematical modeling, hub maximal covering problems, binary and partial coverage, metaheuristics, variable neighborhood search, greedy randomized adaptive search procedure, genetic algorithm

**Research area:** Mathematics

**Research sub-area:** Optimization, Operational research

# Садржај

<b>1</b>	<b>Увод</b>	<b>1</b>
1.1	Проблеми класе NP . . . . .	5
1.2	Метахеуристичке методе . . . . .	7
1.3	Хаб локацијски проблеми . . . . .	27
<b>2</b>	<b>Проблем <math>p</math>-хаб максималног покривања неограничених капа- цитета са једноструким алокацијама</b>	<b>37</b>
2.1	Опис проблема и преглед релевантне литературе . . . . .	37
2.2	Математичке формулације проблема . . . . .	41
2.3	Општа метода променљивих околина за решавање проблема . . .	46
2.4	Експериментални резултати . . . . .	49
<b>3</b>	<b>Проблем <math>p</math>-хаб максималног покривања неограничених капа- цитета са вишеструким алокацијама</b>	<b>62</b>
3.1	Опис проблема и преглед релевантне литературе . . . . .	62
3.2	Математичке формулације проблема . . . . .	64
3.3	Основна метода променљивих околина за решавање проблема . .	67
3.4	Експериментални резултати . . . . .	69
<b>4</b>	<b>Проблем <math>p</math>-хаб максималног покривања неограничених капа- цитета са <math>r</math>-алокацијском шемом</b>	<b>76</b>
4.1	Опис проблема и преглед релевантне литературе . . . . .	76
4.2	Математичке формулације проблема . . . . .	77
4.3	Општа метода променљивих околина за решавање проблема . . .	81
4.4	Похлепна стохастичко-адаптивна процедура претраге са методом променљивог спуста за решавање проблема . . . . .	83
4.5	Генетски алгоритам за решавање проблема . . . . .	86

*САДРЖАЈ*

---

4.6	Експериментални резултати . . . . .	91
<b>5</b>	<b>Закључак</b>	<b>106</b>
	<b>Библиографија</b>	<b>110</b>

# Глава 1

## Увод

Израз *оптимално* потиче од латинске речи *optimus*, што значи најбоље, тако да се оптимизација, у ширем смислу, може дефинисати као процес налажења најбољег решења према задатим критеријумима или управљање системом на најефикаснији начин под задатим условима. Са брзим развојем нових технологија и драстичним променама које доноси данашњица, проблеми оптимизације су све присутнији у науци, инжењерству, војној технологији, менаџменту и економији, али и у другим сферама живота. Проблеми оптимизације су разноврсни, а у пракси се најчешће јављају проблеми планирања транспорта, налажења путева у телекомуникационој мрежи, оптимизације рачунарских мрежа, финансијских улагања и планирања, распоређивања запослених, планирања производње, организације буџета и многи други. Након дефинисања проблема, неопходно је направити одговарајући математички модел који ће представљати добру апроксимацију реалног проблема. У циљу ефикасног решавања проблема, развијају се одговарајуће методе оптимизације које се затим имплементирају у оквиру софтвера. Методе оптимизације морају бити прилагођене специфичностима разматраног проблема, у циљу побољшања квалитета добијених решења и смањења брзине њихових достизања. Тако добијена решења се након анализе могу имплементирати у пракси, делимично или у потпуности. Да би се неко решење сматрало „најбољим”, потребно је увести меру која омогућава поређења решења проблема. Прецизније, неопходно је увести функцију која сваком решењу придружује одговарајућу вредност која представља меру квалитета решења. Та функција се обично назива *функција циља* или *критеријумска функција*.

Нека је  $f : S \rightarrow \mathbb{R}$  реална функција дефинисана на скупу  $S \subseteq \mathbb{R}^n$  и нека је  $D$

подскуп од  $S$  одређен скупом ограничења које елементи скупа  $S$  задовољавају. Прецизније,

$$D = \{x \in S : g_i(x) \leq 0, i = 1, 2, \dots, m\}, \quad (1.1)$$

где је  $x = (x_1, x_2, \dots, x_n)$  потенцијално решење проблема представљено  $n$ -торком променљивих  $x_j, j \in \{1, 2, \dots, n\}$ ,  $g_i : \mathbb{R}^n \rightarrow \mathbb{R}$  су реалне функције које дефинишу ограничења, а  $m$  је укупан број ограничења. Скуп  $S$  се назива *простор решења*, док се скуп  $D$  назива *простор допустивих решења*.

Уопштено говорећи, проблем оптимизације је налажење решења у коме функција циља достиже екстремну вредност (минимум или максимум) на скупу допустивих решења. Уколико је потребно наћи допустиво решење за које функција циља има највећу вредност, у питању је проблем максимизације ( $\max_{x \in D} f(x)$ ), а уколико је потребно наћи допустиво решење за које функција циља има најмању вредност, у питању је проблем минимизације ( $\min_{x \in D} f(x)$ ). Проблеми максимизације функције  $f(x)$  се могу решавати као проблеми минимизације функције  $-f(x)$ , и обрнуто, имајући у виду да важи

$$\max_{x \in D} f(x) = -\min_{x \in D} (-f(x)). \quad (1.2)$$

Уколико треба минимизовати по неком скупу максимум од више задатих функција, у питању је *min-max* проблем оптимизације. Проблеми оптимизације се често називају и проблеми математичког програмирања.

Решење  $x^* \in D$  је *тачка глобалног минимума (максимума)* уколико има најмању (највећу) функцију циља у односу на сва решења из простора решења  $D$ , односно ако  $\forall x \in D$  важи  $f(x^*) \leq f(x)$  ( $\forall x \in D$  важи  $f(x^*) \geq f(x)$ ). Таква решења се називају *оптимална*. Решења која су оптимална у подскупу простора решења називају се *локална оптимална решења*. Прецизније, функција  $f(x)$  има *локални минимум (максимум)* у тачки  $x^* \in D$  ако постоји околина  $U$  решења  $x^*$ , тако да  $\forall x \in U \cap D$  важи  $f(x^*) \leq f(x)$  ( $f(x^*) \geq f(x)$ ). Вредност функције циља  $f^* = f(x^*)$  која одговара оптималном решењу назива се *оптимална вредност функције циља* или *оптимум*<sup>1</sup> (видети [117, 140]).

За класификацију проблема оптимизације могу се користити различити критеријуми:

- У зависности од природе проблема који се разматра, проблеми оптимизације се деле на проблеме *статичке* и *динамичке* оптимизације. Код

---

<sup>1</sup>У литератури се често под оптимумом или оптималним решењем подразумева уређени пар  $(x^*, f^*)$ , што се може сматрати питањем конвенције.



проблема статичке оптимизације, систем или процес који треба оптимизовати се посматра у непроменљивом стању, док се код проблема динамичке оптимизације систем или процес који треба оптимизовати сматра променљивим у времену. Односно, код динамичке оптимизације, функција циља зависи од параметара који су дати у функцији времена;

- На основу броја функција циља, проблеми оптимизације се деле на проблеме: *допустивости*, *једнокритеријумске* и *вишекритеријумске* оптимизације. Код проблема допустивости, циљ је одредити да ли постоји решење које задовољава дата ограничења, али без одређене функције циља. Проблеми једнокритеријумске оптимизације су проблеми са једном функцијом циља, док проблеми вишекритеријумске оптимизације имају више функција циља, које могу бити истог или различитог типа;
- Према критеријуму постојања ограничења, разликују се *проблеми безусловне оптимизације* и *проблеми условне оптимизације*. Код проблема безусловне оптимизације, простор допустивих решења је  $S$ , док се код проблема условне оптимизације уводи скуп ограничења која смањују простор претраживања;
- На основу типа скупа  $D$ , проблеми оптимизације могу бити: *комбинаторни (дискретни)*, *континуални (непрекидни)* и *мешовити*. Уколико је  $D$  коначан или пребројиво бесконачан скуп, реч је о проблему комбинаторне оптимизације, а у случају када је  $D$  небројив и представља подскуп скупа  $\mathbb{R}^n$  ( $n \geq 1$ ), у питању су проблеми континуалне оптимизације. Мешовити проблеми оптимизације имају особине и дискретних и континуалних проблема. У овом случају, неке променљиве решења проблема узимају произвољне реалне вредности, док остале узимају вредности из неког дискретног скупа реалних бројева;
- У зависности од природе улазних параметара, проблеми оптимизације се могу поделити на *детерминистичке* и *недетерминистичке*. Код детерминистичких проблема се за познате улазне параметре добија одређени резултат, са вероватноћом један, за разлику од недетерминистичких проблема, код којих се узима у обзир неодређеност параметара, уз појаву одређених случајних понашања. У том случају, разликују се проблеми *стохастичке* оптимизације и проблеми *робусне* оптимизације. Код про-

блема стохастичке оптимизације, улазни параметри су случајне величине са познатом расподелом, док су код проблема робусне оптимизације параметри случајне величине са непознатом расподелом. Проблеми робусне оптимизације се моделирају тако што одређени улазни подаци могу мењати вредности унутар унапред фиксираног интервала. У овим моделима се разматра најгори могући случај, па функција циља има *min-max* облик;

- Према типу функције циља и ограничења, проблеми оптимизације се могу поделити на *линеарне* и *нелинеарне*. Код линеарних проблема, функција циља и функције садржане у ограничењима су линеарне. Уколико је бар једна од ових функција нелинеарна, реч је о нелинеарном проблему оптимизације. Проблем линеарног програмирања (енгл. *Linear Programming - LP*) је проблем типа:

$$\min_{x \in S} c^T x, \quad S = \{x \in \mathbb{R}^n \mid Ax \leq b\}, \quad (1.3)$$

где је  $x$  вектор променљивих димензије  $n$ ,  $A \in \mathbb{R}^{m \times n}$  је дата матрица типа  $m \times n$ , а  $b \in \mathbb{R}^m$  и  $c \in \mathbb{R}^n$  су дати вектори димензија  $m$  и  $n$ , респективно.

Најзначајније подкласе нелинеарних проблема оптимизације јесу: *квадратни проблеми* (енгл. *Quadratic Problems - QP*), *квадратно условљени проблеми* (енгл. *Quadratically Constrained Problems - QCP*) и *конвексни проблеми* (енгл. *Convex Problems*). Проблем квадратног програмирања је проблем типа:

$$\min_{x \in S} \frac{1}{2} x^T Q x + c^T x, \quad S = \{x \in \mathbb{R}^n \mid Ax \leq b\}, \quad (1.4)$$

где је, уз претходно уведену нотацију,  $Q \in \mathbb{R}^{n \times n}$  дата ненула матрица типа  $n \times n$ . Очигледно је да код QP функција циља укључује квадратне изразе, док су функције које дефинишу ограничења линеарне. Уколико бар једно од ограничења представља квадратни израз, док функција циља може бити линеарна или квадратна, реч је о квадратно условљеном проблему:

$$\min_{x \in S} \frac{1}{2} x^T Q x + c^T x, \quad S = \{x \in \mathbb{R}^n \mid x^T D x + d^T x \leq b\}, \quad (1.5)$$

где је, уз претходно уведену нотацију,  $D \in \mathbb{R}^{n \times n}$  дата матрица типа  $n \times n$ ,  $d \in \mathbb{R}^n$  дати вектор димензије  $n$ , а матрица  $Q$  може бити и нула матрица. Очигледно, уколико је  $Q = 0$ , функција циља је линеарна.

У конвексне проблеме оптимизације спадају проблеми код којих су функција циља и функције које дефинишу ограничења конвексне, што значи да  $\forall x, y \in \mathbb{R}^n, \forall \lambda, \mu \in \mathbb{R}, \lambda + \mu = 1, \lambda \geq 0, \mu \geq 0$  важе неједнакости  $f(\lambda x + \mu y) \leq \lambda f(x) + \mu f(y)$  и  $g_i(\lambda x + \mu y) \leq \lambda g_i(x) + \mu g_i(y), i = 1, 2, \dots, m$ .

## 1.1 Проблеми класе NP

Увођењем појма алгорита у математику и рачунарство [112], доказано је да постоје алгоритами нерешиви и алгоритами решиви проблеми. Проблем се сматра *алгоритами решивим* ако се на њега може применити одговарајући алгоритам описан говорним језиком, алгоритамишким шемом, псеудокодом, програмским језиком итд. У супротном, проблем се сматра *алгоритами нерешивим* (пример: проблем заустављања Тјурингове машине). Формално, проблем је алгоритами решив ако постоји алгоритам који се може применити на сваки његов индивидуални задатак, тј. на сваку инстанцу проблема са конкретним вредностима параметара. Ови проблеми могу бити различите сложености у смислу рачунарских ресурса, процесорског времена и количине меморијског простора потребних за проналажење решења. Временска сложеност неког алгорита представља максимално време извршавања алгорита при налажењу решења проблема. За неке проблеме то време расте брзо (експоненцијално) са порастом димензије проблема и они се тешко могу решити чак и уз помоћ савремених рачунара, док за неке време расте спорије (на пример, полиномски), што је погодније за рад на рачунарима [113].

За процену ефикасности алгорита и асимптотску анализу користи се Bachmann–Landau<sup>2</sup>-ова нотација *велико O* [12, 100]. Нека је  $n \in \mathbb{N}$  димензија улазних података неког проблема, а  $f(n)$  функција која у зависности од димензије проблема изражава време рада алгорита у најгорем случају.

**Дефиниција 1.** Алгоритам има сложеност  $f(n) = O(g(n))$  ако постоје позитивне константе  $n_0$  и  $c$  такве да  $\forall n > n_0$  важи  $f(n) \leq cg(n)$ .

**Дефиниција 2.** Алгоритам има полиномско време извршавања уколико је његова сложеност  $O(p(n))$ , где је  $p(n)$  полиномска функција од  $n$ .

**Дефиниција 3.** Алгоритам има експоненцијално време извршавања уколико је његова сложеност  $O(c^n)$ , где је  $c$  реална константа строго већа од

---

<sup>2</sup>Paul Gustav Heinrich Bachmann (1837 – 1920) и Edmund Georg Hermann Landau (1877 – 1938) су били немачки математичари који су се бавили теоријом бројева и комплексном анализом.

1.

Каже се да је сложеност реда  $O(g(n))$  пропорционална са  $g(n)$ . У пракси, већина алгоритама има неку од следећих сложености:

- $O(1)$  - време извршавања је константно (пример: алгоритам за приступање одређеном елементу низа);
- $O(\log n)$  - време извршавања лагано расте (пример: алгоритам бинарног претраживања);
- $O(n)$  - мала количина обраде над сваким улазним податком (пример: алгоритам линеарног претраживања);
- $O(n \log n)$  - алгоритми који проблем решавају тако што га разбију на пот-проблеме, реше их независно, а затим комбинују решења (пример: алгоритам брзог сортирања низа);
- $O(n^2)$  - алгоритми који обрађују све парове улазних података (пример: елементарне методе сортирања, као што је сортирање избором);
- $O(2^n)$  - са порастом димензије разматраног проблема, време извршавања расте експоненцијалном брзином (генерално непрактични алгоритми за употребу).

**Дефиниција 4.** *Проблем одлучивања је проблем за који се решење добија у облику одговора ДА или НЕ.*

Сваки проблем оптимизације се може свести на проблем одлучивања. На пример, ако је  $M$  минимум функције  $f(x)$  који се достиже у некој тачки  $x_1$ , тада се на следећи начин може формулисати проблем одлучивања: *Да ли за свако  $x$  за које важи  $f(x) \geq M$ , постоји  $x_1$  тако да важи  $f(x_1) = M$ ?*

**Дефиниција 5.** *Проблем припада класи  $P$  и назива се проблемом полиномске сложености уколико постоји алгоритам полиномске сложености који га решава.*

**Дефиниција 6.** *Проблем одлучивања припада класи  $NP$  и назива се проблемом недетерминистичке полиномске сложености уколико се за неко потенцијално решење у полиномском времену може утврдити да ли је заиста решење тог проблема, односно уколико се потврдан одговор може добити у полиномском времену.*

Јасно је да сваки проблем из класе  $P$  припада и класи  $NP$ , али питање које је још увек отворено је да ли је класа  $P$  права поткласа класе  $NP$  (као што се интуитивно очекује). До сада није пронађен проблем који припада класи  $NP$ , а доказано је да не припада класи  $P$ .

Каже се да се проблем одлучивања  $A$  може свести на проблем одлучивања  $B$  у полиномском времену ако се за сваки специјалан случај  $A'$  проблема  $A$  може у полиномском времену пронаћи специјалан случај  $B'$  проблема  $B$ , тако да је за проблем  $A'$  одговор потврдан ако и само ако је и за проблем  $B'$  одговор потврдан.

**Лема 1.** *Ако проблем  $B$  припада класи  $P$  и проблем  $A$  се може свести на проблем  $B$  у полиномском времену, онда и проблем  $A$  припада класи  $P$ . Слично, ако проблем  $B$  припада класи  $NP$  и проблем  $A$  се може свести на проблем  $B$  у полиномском времену, онда и проблем  $A$  припада класи  $NP$ .*

Доказ леме 1 може се наћи у [151].

**Дефиниција 8.** *Проблем одлучивања је  $NP$ -тежак ако се сваки проблем из класе  $NP$  може у полиномском времену свести на њега.*

**Дефиниција 9.** *Проблем одлучивања је  $NP$ -комплетан ако припада класи  $NP$  и при томе је  $NP$ -тежак.*

Може се рећи да  $NP$ -комплетни проблеми представљају најтеже проблеме из класе  $NP$ . Cook је 1971. године изложио теорију  $NP$ -комплетности [31] и показао да је *проблем исказне задовољивости* у математичкој логици  $NP$ -комплетан. Касније се за велики број проблема показало да су  $NP$ -комплетни. Детаљније информације о сложености проблема, као и бројни примери проблема из неких класа сложености могу се наћи у [10, 33, 62, 91].

## 1.2 Метаксхеуристичке методе

У зависности од природе и сложености проблема који треба решити, користе се различите методе оптимизације, за које је неопходно да поседују особине као што су конвергенција, универзалност, прецизност, испуњење свих ограничења проблема, што мање заузимање меморијског простора, задовољавајућа брзина извршавања итд. Методе оптимизације се могу поделити на *егзактне* и *приближне* [140]. **Егзактне методе** су засноване на претраживању целог простора решења. Теоријски, егзактне методе након одређеног броја корака дају оптимално решење проблема, али је у пракси њихова примена најчешће ограничена

на инстанце проблема мањих димензија, услед меморијских или временских ограничења. Егзактне методе се деле на: методе засноване на гранању дрвета претраживања (енгл. *Branching based methods*), програмирање ограничења (енгл. *Constraint Programming*) [9], динамичко програмирање (енгл. *Dynamic Programming*) [15] и  $A^*$  фамилију метода претраге развијену у областима вештачке интелигенције (енгл.  $A^*$ ,  $IDA^*$  - *iterative deepening algorithms*) [96]. Методе гранања су најзаступљеније методе из групе егзактних метода за решавање проблема оптимизације, а најчешће се користе: метода гранања и ограничавања (енгл. *Branch and Bound - BnB*), метода гранања и одсецања (енгл. *Branch and Cut - BnC*), метода гранања и оцењивања (енгл. *Branch and Price - BnP*) и метода гранања, оцењивања и одсецања (енгл. *Branch and Price and Cut - BnPnC*) [86, 101].

Са повећањем димензије проблема, расте број променљивих и ограничења, услед чега се повећавају и потребе за меморијским простором при налажењу решења проблема. Такође, са повећањем димензије проблема, време извршавања NP-тешких проблема (уз претпоставку да је  $P \neq NP$ ) егзактним методама расте експоненцијалном брзином. У овим ситуацијама, јавља се потреба за применом адекватних приближних метода, које у кратком времену извршавања могу дати решења инстанци проблема великих димензија. **Приближне методе** се деле на *хеуристике* и *апроксимативне алгоритме*, а хеуристике се даље деле на *метахеуристике* и хеуристике које су уско специјализоване за решавање конкретних проблема оптимизације (*класичне хеуристике*). У апроксимативне алгоритме спадају апроксимацијске шеме и  $\varepsilon$ -апроксимацијске шеме. Апроксимацијске шеме гарантују да добијено решење не одступа од оптималног решења за више од  $\varepsilon$ , где је  $\varepsilon$  параметар који задаје корисник. Код  $\varepsilon$ -апроксимацијских шема,  $\varepsilon$  је константа која се не може мењати [145].

**Хеуристике** су приближне методе засноване на ефикасним процедурама за налажење допустивог решења разматраног проблема. Основна предност хеуристика је што преко коначног скупа корака воде ка добрим решењима проблема за релативно кратко време. Често достижу оптимална решења, али њихов недостатак је у томе што се оптималност не може доказати. Како су хеуристике најефикаснији, а често и једини начин за решавање проблема оптимизације великих димензија, придаје се пуно пажње њиховом развоју и усавршавању [107].

Класа хеуристичких метода које нису осмишљене искључиво за један конкретан проблем, већ се прилагођавају разноврсним проблемима, називају се

**метахеуристике.** За разлику од класичних хеуристика, које су намењене решавању неких конкретних и специфичних проблема, метахеуристике се уз прилагођавања могу примењивати на широку класу проблема оптимизације. Успешност метахеуристичких метода може се мерити анализом асимптотске конвергенције низа добијених решења ка оптималном, у случају да дозвољено време или број итерација тежи бесконачности [71]. Више о метахеуристикама и њиховим применама може се наћи у [20, 63, 116, 129].

У литератури постоји више начина класификације метахеуристичких метода. Једна од основних подела разликује метахеуристике засноване на локалној претрази, тј. претраживању дуж неке путање (енгл. *trajectory based*) и метахеуристике засноване на популацији, односно скупу решења (енгл. *population based*). Методе засноване на локалној претрази итеративно побољшавају само једно, текуће решење проблема. У ову групу спадају: локална претрага (енгл. *Local Search - LS*), симулирано каљење (енгл. *Simulated Annealing - SA*), табу претрага (енгл. *Tabu Search - TS*), похлепна стохастичко-адаптивна процедура претраге (енгл. *Greedy Randomized Adaptive Search Procedure - GRASP*), метода променљивих околина (енгл. *Variable Neighborhood Search - VNS*) и друге. Методе засноване на популацији у свакој итерацији покушавају да истовремено побољшају скуп решења или један његов део. У ову групу спадају: генетски алгоритам (енгл. *Genetic Algorithm - GA*), оптимизација ројем честица (енгл. *Particle Swarm Optimization - PSO*), оптимизација мрављим колонијама (енгл. *Ant Colony Optimization - ACO*), оптимизација колонијом пчела (енгл. *Bee Colony Optimization - BCO*) итд. Осим ове, најчешће коришћене поделе, метахеуристике се могу класификовати на основу неког од следећих критеријума: да ли су инспирисане природом или математички засноване, да ли памте информације током претраге или не памте, детерминистичке или стохастичке, итеративне или конструктивне итд [140].

Већина метахеуристика су итеративне природе, тј. полазе од комплетног решења (или популације решења) које побољшавају кроз низ итерација, док конструктивни алгоритми граде решење полазећи од празног скупа. Један од најједноставнијих конструктивних алгоритама је тзв. похлепни алгоритам (енгл. *Greedy Algorithm*). Како у сваком кораку бира локално најбоље решење, похлепни алгоритам је једноставан за имплементацију и брзо се извршава, али не налази увек глобални оптимум. Најчешће се користи као помоћни алгоритам за решавање NP-тешких проблема, којим се смањује простор претраживања при

примени других алгоритама, а самим тим и временска сложеност.

У наставку следи кратак преглед најчешће коришћених метахеуристика заснованих на локалној претрази, односно популацији.

## Метахеуристике засноване на локалној претрази

Главне компоненте метахеуристика заснованих на локалној претрази су *диверсификација* претраге и *интензификација* дела простора решења. Диверсификација претраге подразумева испитивање неистражених делова простора решења у циљу избегавања конвергенције ка локалном оптимуму, док интензификација подразумева фокусирање и детаљно претраживање локалног дела простора у ком се очекују добра решења. У циљу описивања најзначајнијих метахеуристика из ове групе, неопходно је дефинисати растојање између два решења, а затим и појмове околине и суседа решења [6].

**Дефиниција 10.** Нека је  $X$  произвољан скуп тачака. Ако се сваком уређеном пару  $(x, y)$ ,  $x, y \in X$ , придружи реалан број  $d(x, y)$  са особинама:

1.  $\forall x, y \in X, \quad 0 \leq d(x, y) < +\infty,$
2.  $\forall x, y \in X, \quad d(x, y) = 0 \Leftrightarrow x = y,$
3.  $\forall x, y \in X, \quad d(x, y) = d(y, x),$
4.  $\forall x, y, z \in X, \quad d(x, y) \leq d(x, z) + d(z, y),$

каже се да је скуп  $X$  снабдевен метриком  $d$ . Уређен пар  $(X, d)$  назива се метричким простором. Вредност  $d(x, y)$  представља растојање између тачака  $x$  и  $y$ .

**Дефиниција 11.** Скуп  $N_\varepsilon(x) \subseteq X$  свих тачака на растојању до  $\varepsilon > 0$  од тачке  $x \in X$  назива се  $\varepsilon$ -околина тачке  $x$ . Елементи скупа  $N_\varepsilon(x)$  називају се  $\varepsilon$ -суседи тачке  $x$ .

У оптимизацији, свако решење је представљено једном тачком из скупа  $X$ . Под *околином* решења  $x$  се у општем случају подразумева скуп  $N(x) \subseteq X$  решења добијених од  $x$  по неком правилу, односно применом неке, унапред дефинисане, елементарне трансформације. Чланови околине  $N(x)$  су *суседи* решења  $x$ . Решење  $x^* \in N(x)$  је *локални оптимум* у односу на околинину  $N(x^*)$ , уколико не постоји решење  $x \in N(x^*)$  које је боље од решења  $x^*$  у односу на вредност задате функције циља. Решење  $x^* \in X$  је *оптимално решење*



(глобални оптимум), уколико не постоји решење  $x \in X$  које је боље од решења  $x^*$  у односу на вредност задате функције циља [34, 122].

**Локална претрага** (енгл. *Local Search - LS*) се може сматрати најједноставнијом метахеуристиком [140] која полази од неког почетног решења и поставља га за текуће најбоље решење  $x$ . Почетно решење је најчешће генерисано на случајан начин или применом неке помоћне хеуристике. У свакој итерацији, врши се претрага унапред дефинисане околине  $N(x)$  текућег решења  $x$ , са циљем налажења суседа  $x'$  који представља боље решење у односу на вредност задате функције циља. Уколико се такав сусед пронађе, прихвата се за ново текуће најбоље решење из ког се наставља са процедуром локалне претраге. У супротном, локална претрага се зауставља у пронађеном локалном оптимуму.

ОкоLINE решења су дефинисане у складу са конкретним проблемом који се разматра. При томе треба водити рачуна да околине које се користе приликом претраге не буду сувише мале, да би постојала могућност налажења следеће тачке претраге, а ни сувише велике, јер се може догодити да се бирање суседа унутар простора допустивих решења сведе на просту стохастичку претрагу или тзв. Monte-Carlo методу [59].

Постоји више стратегија претраживања околине решења:

- стратегија најбољег побољшања (енгл. *best improvement strategy*), којом се истражује цела околина решења  $x$  у циљу налажења најбољег решења из те околине;
- стратегија првог побољшања (енгл. *first improvement strategy*), код које се претрага околина решења  $x$  зауставља чим се пронађе прво решење из дате околине које има бољу вредност функције циља у односу на решење  $x$ ;
- стратегија  $k$ -тог побољшања (енгл. *k-th improvement strategy*), код које се при истраживању околине решења  $x$  претрага зауставља чим се пронађе  $k$  побољшања решења  $x$ ;
- стратегија случајног избора (енгл. *random selection*), код које се на случајан начин бира неко решење из околине текућег и замењује га уколико је боље (дакле, није обавезно да ће бити одабрано најбоље решење из околине текућег решења).

Која год од наведених стратегија да се примени, главни недостатак процедуре локалне претраге је велика вероватноћа конвергенције ка локалном оптимуму. Да би се превазишао овај проблем и повећала робусност методе, предложене су различите варијанте локалне претраге, које подразумевају итеративно покретање алгорита из различитих почетних решења, прихватање решења која нису боља од текућег, замену околина текућег решења итд. У наставку је дат кратак преглед најпознатијих и највише коришћених метахеуристика заснованих на локалном претраживању.

Метода **вишестартне локалне претраге** (енгл. *Multistart Local Search - MLS*) подразумева понављање процедуре локалне претраге сваки пут из новог, случајно изабраног, почетног решења. Најбољи од свих добијених локални оптимума током извршених понављања представља коначно решење. Процес се зауставља када се задовољи неки критеријум заустављања, а најчешће коришћени су: максимално дозвољено време извршавања, максималан број понављања процедуре локалне претраге, максималан број понављања без поправке решења итд. Ова метода може дати лоше резултате при решавању проблема код којих је простор претраге велики, јер такви проблеми често имају више локалних оптимума сличног квалитета.

**Итеративна локална претрага** (енгл. *Iterated Local Search - ILS*) представља побољшање вишестартне локалне претраге [102]. ILS метода најпре генерише почетно решење  $x_0$  (на случајан начин или на основу информација добијених у претходним итерацијама претраге) и поставља га за текуће најбоље решење. Након извршавања процедуре локалне претраге над почетним решењем и добијања локалног оптимума  $x$ , у свакој итерацији се најпре врши „померање” од пронађеног локалног оптимума у ново решење  $x'$ , а затим и локална претрага над новим решењем  $x'$ . Оператор померања модификује текуће најбоље решење тако што део решења задржава, а део мења на основу података добијених у току извршавања алгорита. Нови локални оптимум  $x''$  (који може бити и лошији од текућег најбољег решења) се само под одређеним условима прихвата за ново текуће најбоље решење, а у супротном се чува старо решење. Процес се понавља до испуњења неког критеријума заустављања (максимално дозвољено време извршавања, максималан број итерација, максималан број итерација без поправке решења итд).

**Симулирано каљење** (енгл. *Simulated Annealing - SA*) је метахеуристика заснована на локалном претраживању, коју су предложили Kirkpatrick и сар.

1983. године [94]. Инспирисана је процесом каљења метала у металургији који започиње топљењем метала наглим загревањем, а затим се метал путем лаганог хлађења стврдњава и доводи до стања ниске енергије. Топлота избацује атоме из њиховог почетног положаја и допушта им слободно кретање на вишим енергетским нивоима. Лагано хлађење омогућава атомима да лакше нађу позиције у којим ће имати мању унутрашњу енергију него пре процеса каљења. Мања унутрашња енергија атома подразумева већу стабилност материје и боље карактеристике. Проблем оптимизације се може посматрати као проблем одређивања стабилног стања једног хипотетичког физичког система, односно стања где свако допустиво решење одговара једном стању хипотетичког физичког система, док функција циља представља енергију тог стања. Циљ је пронаћи глобални оптимум, односно стање у коме систем има најмању енергију. Почетно решење  $x_0$  се генерише на случајан начин или применом неке помоћне хеуристике. У свакој итерацији, метода на случајан начин бира ново решење  $x_{k+1}$  из околине текућег решења  $x_k$ . Уколико је ново решење боље од текућег, ново решење се прихвата и замењује текуће, а у случају да је лошије, ново решење се прихвата са одређеном вероватноћом. Вероватноћа прихватања новог решења које је лошије од текућег зависи од разлике у њиховим вредностима функције циља, као и од глобалног параметра  $T$  који одговара температури током хлађења метала. На пример, уколико се разматра проблем минимизације функције  $f(x)$  при задатим ограничењима, вероватноћа  $p_{k+1}$  прихватања новог решења  $x_{k+1}$  се рачуна на следећи начин:

$$p_{k+1} = \begin{cases} 1, & \text{ако је } f(x_{k+1}) < f(x_k) \\ e^{(f(x_k) - f(x_{k+1})) / T}, & \text{иначе,} \end{cases} \quad (1.6)$$

где је  $f(x)$  функција циља разматраног проблема. Постоје и други начини рачунања вероватноће прихватања решења, али је битно обезбедити да је за веће вредности параметра  $T$ , већа вероватноћа прихватања новог решења, као и да се лошија решења, чије вредности функције циља више одступају од текуће, прихватају са мањом вероватноћом. Параметар  $T$  се током рада алгоритма постепено смањује, што је аналогија са постепеним снижавањем температуре код процеса каљења метала. Уколико вредност параметра  $T$  сувише споро опада, лошија решења се прихватају са великом вероватноћом, што води ка насумичном претраживању великог дела простора решења. Уколико се вредност параметра  $T$  сувише брзо смањује, са великом вероватноћом долази до

брзе конвергенције ка локалном оптимуму. Из наведених разлога, правилан избор стратегије хлађења представља један од важних аспеката алгорита. Детаљни методе SA и примери примене на разне проблеме оптимизације могу се наћи [82, 138].

**Табу претрага** (енгл. *Tabu Search - TS*) је метахеуристика заснована на локалном претраживању коју је 1986. године предложио Glover [64]. Основна идеја ове методе је коришћење тзв. *табу листе* као краткотрајне меморије, у циљу избегавања понављања потеза током претраге простора допустивих решења и конвергенције ка локалном оптимуму. Почетно решење се генерише на случајан начин или применом неке помоћне хеуристике, а затим се добијено почетно решење поставља за текуће најбоље решење. У свакој итерацији TS методе, у табу листи се „памте” допустива решења до којих се дошло током извесног броја претходних итерација. Решења која се налазе у табу листи су означена као „табу” и не могу бити прихваћена у наредном кораку. Дужина табу листе је ограничена и не сме бити премала, јер то може довести до цикличног понављања решења, а ни превелика, јер велики број решења у табу листи смањује простор претраге и тиме повећава могућност конвергенције ка локалном оптимуму. Коришћењем табу листе, околине текућег решења се редукују, јер се искључују решења која се тренутно налазе у табу листи. Најбоље решење добијено локалном претрагом редуковане околине текућег решења постаје ново текуће решење. Након тога, из табу листе се по принципу FIFO (енгл. *First in, First out*) стратегије избацује решење са почетка листе, а убацује текуће решење на крај листе. Алгоритам се зауставља када се задовољи неки критеријум заустављања. Критеријум заустављања може бити: максимално време извршавања методе, максималан број итерација, максимална број итерација између две поправке најбољег решења, достигнуто решење задовољавајућег квалитета итд. У литератури је предложено и унапређење TS методе, код које се табу листе формирају само на основу одређених карактеристика решења, која су означена као „табу”. Тиме се добија на брзини и једноставности претраге табу листе, али и губљењу неких информација. Односно, могуће је да непосећена решења доброг квалитета буду искључена само на основу неке карактеристике. Да би се превазишао овај проблем, уводи се тзв. критеријум аспирације (енгл. *aspiration criteria*) који омогућава, уз задовољавање одређених услова, прихватање неких решења са табу листе. Други начин ослобађања решења из табу листе јесте тзв. *табу време*, које представља максималан број итерација које

једно решење може да проведе у табу листи.

Осим табу листе као краткотрајне меморије, у TS методи постоји и могућност коришћења средњорочне и дугорочне меморије. Средњорочне меморије памте само елементе „добрих” решења до којих се дошло у претходних неколико итерација и на тај начин доприносе експлоатацији делова простора у ком се очекују добра решења. Дугорочне меморије омогућавају диверсификацију претраге, јер чувају информације о елементима присутним у текућем решењу током целог процеса претраге. Детаљи TS методе, њене варијанте и примери примене могу се наћи у [65–67].

**Похлепна стохастичко-адаптивна процедура претраге** (енгл. *Greedy Randomized Adaptive Search Procedure - GRASP*) је метахеуристика заснована на локалном претраживању коју су први представили 1989. године Feo and Bard у раду [55], а касније популаризовали Feo and Resende [56]. GRASP је итеративна метода чија се свака итерација састоји од две фазе: најпре се у конструктивној фази генерише комплетно решење, које се затим побољшава у фази локалног претраживања. У првој фази, примењује се похлепно стохастичка процедура за конструкцију решења (енгл. *Greedy Randomized Construction procedure - GRC*). GRC процедура гради решење додавањем једног по једног елемента парцијалном решењу, све док се не конструише комплетно решење. У свакој итерацији GRC фазе се најпре креира листа кандидата (енгл. *Candidate List - CL*), која се састоји од свих елемената које је могуће додати парцијалном решењу. Затим се формира тзв. ограничена листа кандидата (енгл. *Restricted Candidate List - RCL*), издвајањем елемената из CL са позитивним особинама за конкретан проблем који се решава. У ту сврху, дефинише се функција  $g(\cdot)$  која одражава промене у вредности функције циља које доносе елементи из листе кандидата. Сваком елементу  $i \in CL$  се додељује вредност  $g(i)$ , а затим се RCL формира на следећи начин:

$$RCL = \{i \in CL : g(i) \leq g_{min} + \varphi(g_{max} - g_{min})\}, \quad (1.7)$$

где параметар  $\varphi \in [0, 1]$ , а  $g_{min}$  и  $g_{max}$  су редом најмање и највеће вредности  $g(i)$  на скупу CL. Елементи из RCL се бирају на случајан начин и додају парцијалном решењу, све док се не генерише комплетно решење. Очигледно, за  $\varphi = 0$ , GRC фаза се своди на класичан похлепни алгоритам, док је случај  $\varphi = 1$  еквивалентан конструкцији решења на потпуно случајан начин.

Решење добијено у GRC фази се даље прослеђује фази локалне претраге,

која налази локални оптимум у односу на унапред дефинисане околине. Добијени локални оптимум представља најбоље решење једне GRASP итерације. GRASP итерације се смењују све док се не задовољи неки критеријум заустављања, а најбоље међу решењима добијеним у свакој од GRASP итерација је излаз GRASP алгоритма.

У основној варијанти GRASP методе, параметар  $\varphi$  има константну вредност у свим итерацијама. Код напредног концепта методе, у свакој GRASP итерацији вредност параметра  $\varphi$  се самоподешава на основу квалитета решења добијених у претходним итерацијама. Ова варијанта GRASP методе у литератури је позната под називом реактивна похлепна стохастичко-адаптивна процедура претраге (енгл. *Reactive Greedy Randomized Adaptive Search Procedure - Reactive GRASP*) [124].

Још једна модификација основне GRASP методе је метода поновног повезивања стаза (енгл. *Path Relinking*) [66], којом се постижу побољшања решења претраживањем путања које повезују GRASP тренутна решења и најбоља решења добијена коришћењем неке друге методе (нпр. TS). Постоје и друга побољшања и алтернативе технике које се могу применити у конструктивној фази GRASP методе, као што су: пертурбације трошкова уместо класичног избора случајног елемента (енгл. *cost perturbations in place of randomized selection*), функције пристрасности (енгл. *bias functions*), интелигентна конструкција-меморисање и учење (енгл. *intelligent construction: memory and learning*) и локална претрага над делимично конструисаним решењима (енгл. *local search on partially constructed solutions*). Више о овим варијантама GRASP методе може се пронаћи у [56, 68, 127, 128].

**Метода променљивих околина** (енгл. *Variable Neighborhood Search - VNS*) је метахеуристика коју су 1997. године предложили Мladenović и Hansen у раду [111], за решавање проблема трговачког путника (енгл. *Traveling Salesman Problem - TSP*). Заснована је на идеји систематских промена околина током локалног претраживања и следеће три чињенице:

- локални минимум у односу на једну околину не мора бити и локални минимум у односу на неку другу околину;
- глобални минимум је локални минимум у односу на све околине;
- за већину проблема, локални минимуми у односу на различите околине су међусобно блиски.

Промена околина се може реализовати увођењем различитих величина околина (што се постиже мењањем броја истовремених трансформација над решењем) или применом различитих метрика (односно различитих трансформација над решењем). У зависности од тога да ли се горе наведене три чињенице користе детерминистички, стохастички или комбиновано, у литератури су предложене различите варијанте методе променљивих околина [76]. У наставку је дат кратак опис најчешће коришћених варијанти VNS методе.

**Основна метода променљивих околина** (енгл. *Basic Variable Neighborhood Search - BVNS*) је итеративна метода која користи структуре околина  $N_k, k = 1, 2, \dots, k_{max}$ , дефинисане у складу са карактеристикама разматраног проблема. Максималан број околина  $k_{max}$  представља основни параметар ове методе. BVNS метода започиње рад процедуром која, на случајан начин или применом неке конструктивне хеуристике, генерише почетно решење  $x$  и проглашава га за текуће најбоље решење. Структура сваке BVNS итерације је следећа: након постављања индекса околине  $k$  на 1, основни кораци BVNS методе садржани су у петљи у оквиру које се мења индекс околине  $k$  и извршавају три основне фазе: процедура размрдавања, процедура локалне претраге и померај. У фази размрдавања, на случајан начин се бира неко ново решење  $x'$  из  $k$ -те околине текућег решења  $x$ , а затим се почевши од новог решења  $x'$  примењује локална претрага у односу на унапред дефинисане околине решења  $x'$ . Важно је напоменути да се околине за размрдавање и локалну претрагу могу разликовати на основу растојања (броја трансформација) или на основу метрике (врсте трансформација). Најбоље решење  $x''$  добијено након локалне претраге се затим пореди са текућим решењем  $x$  и замењује га уколико је боље, у односу на вредност функције циља разматраног проблема. Ако је решење  $x''$  боље од текућег решења  $x$ , врши се ажурирање текућег решења  $x = x''$ , индекс околине  $k$  се поставља на 1 и претрага се наставља у околини  $N_1$  текућег најбољег решења  $x$ . У супротном, индекс околине  $k$  се повећава за 1, тако да алгоритам наставља са фазом размрдавања у наредној околини текућег решења  $x$ . Итерације се понављају све док се не задовољи неки критеријум заустављања. Критеријум заустављања може бити максималан укупан број итерација, максималан број итерација између два побољшања, максимално дозвољено време извршавања, достигнут тражени квалитет решења итд.

**Метода променљивог спуста** (енгл. *Variable Neighborhood Descent - VND*) представља детерминистичку варијанту методе променљивих околина и засно-

вана је на следећим чињеницама:

- једно решење не мора бити локални оптимум за две различите околине;
- решење које је локални оптимум у односу на неколико околина има веће шансе да буде глобални оптимум у односу на решење које је локални оптимум у односу на само једну околину.

VND метода претражује задати низ околина  $N_k, k = 1, 2, \dots, k_{max}$ , где  $N_k(x)$  означава  $k$ -ту околину решења  $x$ . Околине се претражују на секвенцијални или угњежден начин, у циљу налажења што бољег решења.

Код *Секвенцијалне VND методе* (енгл. *Sequential Variable Neighborhood Descent - SeqVND*) се најпре почетно решење  $x$  генерише на случајан начин или применом неке конструктивне хеуристике и текуће најбоље решење се поставља на  $x$ . Полазећи од почетног решења, SeqVND метода примењује процедуру локалне претраге редом на дефинисаним околинама  $N_k, k = 1, 2, \dots, k_{max}$ . Прецизније, индекс околине  $k$  најпре узима вредност 1 и SeqVND претражује прву околину  $N_1(x)$  све док у њој постоји могућност поправљања текућег најбољег решења  $x$  и одређује локални оптимум  $x'$  у односу на ту околину. Затим се ажурира текуће најбоље решење тако што се постави да је  $x = x'$ , а претраживање се враћа на околину  $N_1$  текућег најбољег решења. Уколико нема поправке, индекс околине  $k$  се увећава за 1 и локална претрага се наставља у наредној неиспитаној околини  $N_k(x), k = 2, \dots, k_{max}$  текућег најбољег решења. Након поправљања текућег најбољег решења у било којој околини, ажурира се текуће најбоље решење и претраживање се враћа на околину  $N_1$  новодобијеног текућег најбољег решења. SeqVND метода се завршава уколико није могуће поправити текуће најбоље решење ни у једној од унапред задатих  $k_{max}$  околина, што значи да је то решење локални оптимум у односу на све посматране околине. Код SeqVND методе, обично се за  $k_{max}$  узима релативно мали цео број (2, 3 или 4). Уколико је  $k_{max} = 1$ , SeqVND се своди на обичну локалну претрагу. Како комплетна претрага околине стратегијом најбољег побољшања може бити временски веома захтевна, SeqVND метода често користи стратегију првог побољшања.

*Угњеждена VND метода* (енгл. *Nested Variable Neighborhood Descent - Nest-VND*) уместо секвенцијалног поретка околина претражује тзв. угњежене околине. За дефинисан низ околина  $N_k, k = 1, 2, \dots, k_{max}$ , у литератури постоје различите стратегије претраживања околина код Nest-VND методе. Због специ-



фичне стратегије претраживања, околине се називају угњежденим околинама. Ако је, на пример,  $k_{max} = 2$ , тада је једна од могућих стратегија претраживања околина код Nest-VND методе следећа: за свако решење  $x'$  из  $N_1$  околине текућег решења  $x$  примењује се локална претрага у односу на  $N_2$  околину решења  $x'$ . Може се приметити да уколико обе околине имају кардиналност  $l$ , број решења који се посећује при примени SeqVND методе је  $2l$ , док је код Nest-VND методе тај број једнак  $l^2$ . Још једна могућа стратегија претраживања околина код Nest-VND методе је следећа: ако је, на пример,  $k_{max} = 3$ , на свако решење  $x'$  из околине  $N_3$  текућег решења  $x$ , примењује се SeqVND метода редом на дефинисаним околинама  $N_1$  и  $N_2$  решења  $x'$ . Како је претрага угњеждених околина временски захтеван процес, Nest-VND метода најчешће користи стратегију првог побољшања. Комбиновањем идеја секвенцијалне и угњеждене VND методе настаје тзв. *Мешовита VND метода* (енгл. *Mixed Variable Neighborhood Descent*). Више о VND методи и њеним варијантама може се наћи у [77].

**Редукована метода променљивих околина** (енгл. *Reduced Variable Neighborhood Search - RVNS*) је стохастичка варијанта методе променљивих околина у којој је изостављена фаза локалног претраживања, чиме се постиже уштеда процесорског времена. И код RVNS методе најпре је неопходно дефинисати структуре околина  $N_k, k = 1, 2, \dots, k_{max}$  и генерисати почетно решење  $x$ , које се поставља за текуће најбоље решење. Затим се из прве околине текућег решења  $x$  на случајан начин бира решење  $x'$ , које не мора бити локални оптимум. Решење  $x'$  се пореди са текућим решењем  $x$  и замењује га уколико је боље, у односу на вредност функције циља разматраног проблема. Ако је решење  $x'$  боље од  $x$ , ажурира се текуће најбоље решење тако што се постави да је  $x = x'$ , а претраживање се враћа на околину  $N_1$  решења  $x$ . У супротном, индекс околине  $k$  се повећава за 1, тако да се претрага наставља у наредној околини текућег решења  $x$ . RVNS метода је врло ефикасна код примера проблема великих димензија, јер не садржи сложену и временски захтевну процедуру локалног претраживања. Међутим, због великог степена случајности, решења добијена RVNS методом су генерално лошијег квалитета у односу на решења варијанти VNS метода које садрже фазу локалне претраге. Стога се RVNS метода често користи за брзо генерисање почетних решења, за неке друге егзактне или хеуристичке методе.

**Општа метода променљивих околина** (енгл. *General Variable Neighborhood Search - GVNS*) је варијанта VNS методе која настаје из BVNS методе,

тако што се локална претрага замени напреднијом методом променљивог спушта, која истражује неколико структура околина. ОкоLINE које се претражују у оквиру фазе размрдавања и у оквиру VND методе могу бити индиковане различитим метрикама.

**Адаптивна метода променљивих околина** (енгл. *Skewed Variable Neighborhood Search - SVNS*) омогућава премештање претраге у делове претраживачког простора који су удаљени од текућег најбољег решења. Углавном се примењује при решавању проблема код којих се локални оптимуми налазе далеко једни од других, односно када текуће најбоље решење представља локални оптимум у великом делу простора претраге и када размрдавање и локална претрага не доводе до побољшања. Разлика између основне методе променљивих околина и адаптивне методе променљивих околина је у кораку замене околина. За разлику од основне методе променљивих околина, SVNS метода допушта померање и у лошија решења, при одређеним условима. Прецизније, померање из текућег решења  $x$  у лошије решење  $x''$  је дозвољено уколико важи:

$$f(x'') - \alpha \rho(x, x'') < f(x), \quad (1.8)$$

где је  $f(\cdot)$  вредност функције циља разматраног проблема минимизације,  $\alpha$  унапред задати параметар, а  $\rho(x, x'')$  удаљеност решења  $x$  од  $x''$ . Удаљеност два решења може бити њихово растојање, за које се користи иста метрика као за дефинисање околина. Може се приметити да је померање контролисано параметром методе  $\alpha$  - што је вредност параметра мања, мања је и шанса за померањем ка лошијем решењу. Важно је изабрати адекватну вредност параметра  $\alpha$ , која ће омогућити померање у окоLINE које су далеко од  $x$ , а притом избећи прелазак SVNS методе у VNS методу са вишеструким покретањем (енгл. *multistart*) која у сваком покретању започиње претрагу од новог, случајно генерисаног почетног решења.

У литератури су предложене бројне варијанте методе променљивих околина, као што су: метода променљивих околина са декомпозицијом (енгл. *Variable Neighbourhood Decomposition Search - VNDS*), метода променљивих формулација (енгл. *Variable Neighborhood Formulation Space Search - VNFSS*), примално-дуална метода променљивих околина (енгл. *Primal-dual VNS - PD-VNS*), паралелна метода променљивих околина (енгл. *Parallel VNS - PVNS*) итд. Више о VNS методи, њеним варијантама и применама, може се пронаћи у прегледним радовима [26, 72–77].

## Метахеуристике засноване на популацији решења

Метахеуристике засноване на популацији решења раде над скупом потенцијалних решења, који се назива популација. Полазећи од почетне популације, потенцијална решења се кроз низ итерација модификују или комбинују по унапред задатим правилима, у циљу налажења најбољег решења. Неке метахеуристике започињу рад са празним или делимичним скупом потенцијалних решења. Већина метахеуристика из ове групе је инспирисана процесима из природе, а у наставку следи кратак опис неких од најпознатијих.

**Генетски алгоритам** (енгл. *Genetic Algorithm - GA*) припада групи популационих стохастичких метахеуристика и инспирисан је процесима из теорије природне еволуције. Основне поставке GA је изнео John Holland у књизи „Прилагођавање у природним и вештачким системима” (енгл. *Adaptation in natural and artificial systems*) из 1975. године [79]. GA симулира поједностављени процес еволуције коначног скупа јединки које чине популацију применом генетских оператора. Основни генетски оператори су: *селекција*, *укрштање* и *мутација*, а у зависности од карактеристика разматраног проблема, примењују се различите варијанте ових оператора. Свака јединка из популације представљена је генетским кодом и одговара потенцијалном решењу посматраног проблема. Начин кодирања решења је важан аспект генетског алгоритма и треба да одсликава природу проблема који се разматра. Функција прилагођености (енгл. *fitness function*) представља меру квалитета сваког појединачног решења. За успешну примену GA на неки проблем, неопходно је да сви елементи GA буду одабрани у складу са особинама посматраног проблема. Неадекватан избор генетских оператора, начина кодирања, рачунања функције прилагођености, и других аспеката алгоритма, може довести до лоших резултата.

GA започиње рад полазећи од иницијалне популације која садржи  $N_{ind}$  јединки, најчешће генерисаних на случајан начин, што обезбеђује разноврсност генетског материјала. Након израчунавања вредности функције прилагођености за сваку јединку у популацији, следи итеративни процес који се извршава до задовољења неког критеријума заустављања. Свакој GA итерацији одговара једна популација јединки (генерација). *Селекцијом* се бирају јединке које ће учествовати у процесу креирања нове генерације, при чему се најчешће претпоставља да боље прилагођене јединке имају веће шансе да доведу до поправљања текућег најбољег решења. Постоји више варијанти оператора селекције, а најчешће се користе: проста рулет селекција, селекција заснована на рангирању је-

динки, турнирска селекција, фино градирана турнирска селекција и униформна селекција. Међу селекованим јединкама бирају се парови јединки-родитеља на које се примењује оператор *укрштања*. Јединке-родитељи са извесном вероватноћом размењују делове генетског кода и производе јединке-потомке за наредну генерацију јединки. Постоје различити начини укрштања јединки-родитеља, а неки од њих су: једнопозиционо, двопозиционо, вишепозиционо и униформно укрштање. Оператор укрштања се најчешће примењује са одређеном вероватноћом  $p_{cross}$ , која дефинише проценат парова јединки-родитеља који заиста размењују генетски материјал и стварају нове јединке-потомке. На јединке-потомке се даље примењује оператор мутације, који са извесном вероватноћом  $p_{mut}$  мења мале делове генетског кода изабране јединке. Оператор мутације доприноси разноврсности генетског материјала и помаже спречавању преране конвергенције ка локалном оптимуму. Најчешће коришћени оператори мутације су: проста мутација, мутација помоћу биномне, експоненцијалне или нормалне расподеле итд.

Из скупа који чине нове јединке настале применом генетских оператора и јединке текуће генерације, неопходно је изабрати наредну популацију јединки задате величине  $N_{ind}$ . У циљу очувања довољно квалитетних решења, али и разноврсности генетског материјала, одабир јединки за нову генерацију врши се на основу вредности функција прилагођености, чувајући притом одговарајући ниво разноврсности јединки у популацији. Поступак избора јединки које ће чинити нову GA генерацију назива се *замена генерација*. Постоје два основна начина замене - генерацијски и стационарни. Код генерацијског GA текућа генерација се у потпуности замењује новим јединкама, док се код стационарног GA део текуће генерације мења новим јединкама. Код стационарног GA разликује се више стратегија креирања нове генерације: користећи *елитистичку стратегију* (којом одређени број елитних јединки из текуће генерације директно прелази у наредну), користећи *униформну стратегију* (којом се на место одређеног броја најгорих јединки из текуће генерације убацују нове најбоље јединке), користећи *рулет стратегију* (којом се на случајан начин бирају јединке за нову генерацију), користећи *турнирску стратегију* (којом се у нову генерацију уносе најбоље јединке из случајно изабране групе која садржи унапред задати број јединки) итд.

Поступком замене генерација завршава се једна итерација GA. Итерације се понављају до задовољења неког критеријума заустављања, а најчешће се ко-

ристе: максимално дозвољено време извршавања, максималан број генерација, максималан број генерација без поправке текућег најбољег решења, достигнуто решење задовољавајућег квалитета, достигнута максимална сличност јединки у популацији итд.

Свака имплементација генетског алгоритма садржи различите параметре ( $N_{ind}$ ,  $p_{cross}$ ,  $p_{mut}$  итд.) који се за решавање сваког конкретног проблема морају подесити експерименталним путем. Параметри се могу мењати и у току извршавања алгоритма - фиксном променом параметара или адаптивном променом параметара. Фиксном променом параметара се унапред задаје образац по ком ће се вршити промена параметара (смањивање или повећавање) током итерација, док се адаптивном променом параметри могу мењати у зависности од постигнутих резултата и успешности одређеног оператора. Детаљнији опис теоријских и практичних аспеката GA може се наћи у [13, 39, 125, 136].

**Оптимизација ројем честица** (енгл. *Particle Swarm Optimization - PSO*) је стохастичка итеративна метахеуристика коју су предложили Kennedy и Eberhart 1995. године [92], а инспирисана је социјалним интеракцијама између јединки у јату птица или роју инсеката. PSO метода ради над ројем честица, при чему су свакој честици  $i$  из роја додељени вектори  $x^i$  и  $v^i$ , који представљају редом њене векторе положаја и брзине. Координате вектора положаја сваке честице из роја одговарају једном решењу проблема. Свака честица памти своје најбоље пронађено решење и најбоље решење или целог роја или из унапред дефинисане околине честице. Околина честице представља област социјалног утицаја других честица на посматрану честицу, а у граничном случају околина може бити и цео рој.

У иницијалној фази, PSO метода генерише рој на случајан начин, тако што се свакој честици  $i$  додељују случајно одабрани положај  $x_i^0$  и случајно одабрана брзина  $v_i^0$ . У свакој итерацији се најпре врши ажурирање брзина честица, а затим и положаја. Ажурирање брзина честица се обавља по формули:

$$v_{n+1}^i = c_v v_n^i + c_1 r_1 (p_n^i - x_n^i) + c_2 r_2 (s_n - x_n^i), \quad (1.9)$$

где су  $x_n^i$  и  $v_n^i$  положај и брзина кретања честице  $i$  у  $n$ -тој итерацији, респективно. Са  $p_n^i$  је означен текући најбољи положај (до сада најбоље познато решење) честице  $i$ , а са  $s_n$  најбољи текући положај целог роја (односно околине честице  $i$ ) у  $n$ -том кораку. Параметри  $r_1$  и  $r_2$  се у свакој итерацији бирају на случајан начин из интервала  $[0, 1]$ , док су  $c_v$ ,  $c_1$ ,  $c_2$  унапред дефинисани параметри

који се могу мењати током рада алгоритма. Помоћу параметра  $c_v$  контролише се величина претраживачког простора роја, јер се за веће вредности овог параметра координате ажурираног вектора брзине честице значајније мењају у односу на старе, што води ка претраживању удаљенијих делова простора решења у односу на текући положај честице. У супротном, за мање вредности параметра  $c_v$ , фаворизује се локално претраживање у релативно малој околини сваке честице. Параметар  $c_1$  је означен као когнитивни и његова улога је усмеравање честице у правцу текућег најбољег положаја исте честице. Повећавањем вредности параметра  $c_1$ , фаворизује се детаљније испитивање околине најбољег решења до ког је стигла посматрана честица. Параметар  $c_2$  је означен као социјални и његовим повећавањем се фаворизује детаљније испитивање околине најбољег решења до ког је стигао цео рој, односно честице из унапред дефинисане околине посматране честице.

Помоћу вектора брзине се у свакој итерацији ажурирају тренутни положаји честица у роју. Прецизније, нови положаји честица добијају се „померањем” сваке честице из претходног роја на следећи начин  $x_{n+1}^i = x_n^i + v_n^i$ . На основу израчунатих вредности задате функције циља за свако решење (положај), у свакој итерацији се ажурирају и најбоља решења (положаји) сваке честице, као и најбоље решење (положај) на нивоу целог роја, односно посматране околине честице. Итерације се смењују до задовољења унапред дефинисаног критеријума заустављања. Детаљнији опис ове методе може се наћи у [48, 123].

**Оптимизација мрављим колонијама** (енгл. *Ant Colony Optimization - ACO*) је конструктивна метахеуристика коју су увели Марко Дориго и сар. 1997. године у раду [45], инспирисани понашањем мрава у заједницама (мрављим колонијама). Експериментима је уочено да мрави увек проналазе најкраћи пут између хране и своје колоније користећи посредан начин комуникације помоћу феромона - хемијске супстанце коју мрави остављају за собом. Истраживања су показала да мрави са већом вероватноћом прате путању са јачим трагом феромона. Феромонски трагови на свим путевима постепено испаравају, али како временом најкраћим путем пролази све више мрава, на том путу ће траг феромона бити најјачи. Пут који мрави не користе након неког времена остаје без трага феромона и неће привући нове мраве.

Метахеуристика оптимизације мрављим колонијама користи аналогију са описаним процесом и најчешће се користи за налажење најкраћег пута у графу. АСО алгоритам ради над колонијом вештачких мрава (агената), при чему је

сваки мрав задужен за конструкцију једног решења. Свака итерација АСО алгоритма се састоји од конструкције решења и ажурирања феромонских трагова. Итерације АСО алгоритма се смењују до задовољења неког критеријума заустављања.

У свакој АСО итерацији, сваки мрав конструише једно решење применом стохастичке похлепне процедуре која додаје један по један елемент парцијалном решењу, све док се не конструише комплетно решење. На изградњу решења утичу вредности феромонских трагова, који памте карактеристике добрих решења и специфичности разматраног проблема које су најчешће уграђене у помоћну хеуристику. Улога хеуристике је да помаже мравима у избору елемената при конструкцији комплетног решења. Утицај трага феромона и помоћне хеуристике се контролише параметрима АСО алгоритма.

Ажурирање феромонских трагова се врши на основу квалитета добијених решења и одвија се кроз две фазе - испаравања и појачавања. Свим компонентама се смањује ниво феромона унапред одређеном стопом испаравања, док се повећавање вредности феромона компонентама квалитетних решења може реализовати коришћењем различитих стратегија. Прецизније, испаравање феромона се врши на следећи начин:

$$\tau_{ij} \leftarrow \tau_{ij} \cdot (1 - \rho), \quad \forall i, j \in \{1, 2, \dots, n\} \quad (1.10)$$

где је  $\rho \in (0, 1]$  стопа испаравања. Очигледно, код мање стопе испаравања, ниво феромона испарава спорије у односу на случајеве са већом стопом испаравања. Циљ фазе испаравања је елиминација лоших решења и подстицање конвергенције ка тренутно добрим решењима.

У фази појачавања могу бити примењене три различите стратегије: ажурирање феромона се врши у сваком кораку конструкције решења, након што сваки од мрва конструише решење или након што сви мрави конструишу своја решења. Више о стратегијама повећавања вредности феромона може се наћи у [32]. Иако се АСО метахеуристика у почетку највише користила за решавање проблема који се свode на проналажење најкраћег пута у графу, у литератури постоје бројни примери примене АСО методе на различите проблеме оптимизације (видети [43, 44, 46]).

**Оптимизација колонијом пчела** (енгл. *Bee Colony Optimization - BCO*) је метахеуристика коју су 2001. године предложили Лучић и Теодоровић за решавање проблема трговачког путника [104]. Инспирирана је процесом потраге

за храном пчела које припадају једној колонији. Пчеле могу имати различите улоге у колонији. За репродукцију су задужени једна пчела краљица и пар хиљада трутова, док пчеле радилице обављају све остале послове, као што су: нега легла и матице, сакупљање нектара и цветног праха, изградња саћа, брига о температури и чистоћи кошнице, производња меда итд. Пчеле извиђачи (енгл. *scouts*) напуштају кошницу и проналазе локације за сакупљање нектара, а по повратку изводе специфичан плес, који служи за представљање удаљености локације, квалитета и количине пронађеног нектара другим пчелама. На тај начин рекламирају пронађене изворе хране и наводе друге пчеле да их прате. Према аналогiji са колонијом пчела у природи, ВСО метахеуристика користи колонију вештачких пчела, где свака пчела представља једног агента који конструише једно решење проблема. У циљу налажења што бољих решења, вештачке пчеле сарађују и размењују информације. Постоје две варијанте ВСО методе: конструктивна ВСО метода и ВСО са поправком решења (енгл. *Improving Bee Colony Optimization - BCOi*). Код конструктивне ВСО методе, свака пчела полази од празног почетног решења и итеративно конструише комплетно решење [104]. ВСО метода са поправком подразумева генерисање комплетних почетних решења, која се кроз итерације поправљају у циљу налажења што квалитетнијих решења [37]. Већина осталих елемената ВСО и ВСОi метода се поклапају.

Рад ВСО алгоритма се одвија кроз итерације, док се не задовољи неки критеријум заустављања. У оквиру сваке итерације, смењују се две фазе, лет унапред (енгл. *forward pass*) и лет уназад (енгл. *backward pass*). Током лета унапред, вештачке пчеле покушавају да конструишу (односно поправе) своје решење, у зависности од тога која је варијанта ВСО методе у питању. На тај начин оне истражују простор решења, што се може поистоветити са сакупљањем нектара у природи. Током лета уназад, вештачке пчеле размењују информације о квалитету својих решења, прикупљене током претраге. Након што свака пчела заврши конструкцију/трансформацију свог решења, сва решења се евалуирају рачунањем и поређењем вредности задате функције циља за свако решење. У зависности од квалитета конструисаног решења, свака пчела са одређеном вероватноћом одлучује хоће ли остати лојална свом решењу и постати регрутер (енгл. *recruiter*) који рекламира своје решење или ће напустити своје решење и постати неопредељена (енгл. *uncommitted*). Неопредељена пчела преузима неко од понуђених решења регрутера са вероватноћом



која је пропорционална квалитету одговарајућег понуђеног решења. Више детаља о VCO и VCOi методама може се наћи у [38]. У литератури се могу наћи бројни примери успешне примене VCO методе за решавање различитих проблема оптимизације (видети [104, 105, 141, 142]).

Описане метахеуристике је у зависности од природе проблема који се решава могуће унапредити различитим стратегијама паралелизације, као и хибридикације са неким другим хеуристикима или егзактним методама [19]. Како паралелизација омогућава да се више радњи и прорачуна изводи истовремено на различитим процесорима, применом ове стратегије се најчешће смањује укупно време извршавања алгорита, али и побољшава квалитет коначних решења. Наравно, паралелизацију није могуће увек применити, неопходан услов за то је да се посматрани проблем може разложити на потпроблеме који не зависе једни од других. Више о стратегијама паралелизације метахеуристика се може наћи у раду [5].

Како свака метода за решавање разматраног проблема има одређене предности и недостатке, често се приступа процесу хибридикације. Хибридикација подразумева комбиновање две или више различитих метода, тако да се у хибридную методу укључују предности једних и истовремено елиминишу недостаци других метода. Осим међусобно, метахеуристике се могу успешно комбиновати и са: егзактним методама, методама из области вештачке интелигенције (енгл. *artificial intelligence*), методама истраживања података (енгл. *data mining*), машинског учења (енгл. *machine learning*) итд. Детаљније о стратегијама хибридикације метахеуристика може се наћи у [139].

### 1.3 Хаб локацијски проблеми

Хаб локацијски проблеми (енгл. *Hub Location Problems - HLP*) спадају у проблеме оптимизације који су присутни у бројним областима реалног живота. Ови проблеми се срећу при оптимизацији телекомуникационих и транспортних система, превозењу путника и робе у авио-саобраћају, поштанским мрежама, системима брзе испоруке итд. Хаб локацијски проблеми се користе у практичним ситуацијама које захтевају успостављање протока од чворова-извора (снабдевача) до чворова-одредишта (корисника) при задатим условима, тако да вредност одређене функције циља буде оптимална. Хабови су посебни објекти (чворови у мрежи) који представљају центре консолидације и колекције про-

тока између две изабране локације (снабдевача и корисника). Како је цена транспорта између хабова по јединици количине нижа, усмеравање протока на мрежу хабова води ка снижавању трошкова транспорта у мрежи. Код већине хаб локацијских проблема, претпоставља се да мрежа хабова чини комплетан граф. Процент уштеде остварен транспортом између хабова не зависи од количине робе која се преноси и исти је за све парове хабова. Такође, важи претпоставка да директан транспорт између не-хаб чворова није дозвољен, већ се проток од чвора снабдевача до чвора корисника врши преко најмање једног хаба. У многим ситуацијама из праксе је неопходно изоставити неку од наведених општих претпоставки (видети радове [4, 11, 29, 30, 93, 99, 115]).

У литератури постоје класификације хаб локацијских проблема према различитим критеријумима. Имајући у виду да већина хаб локацијских проблема настаје из конкретне реалне ситуације, неретко се дешава да неки хаб локацијски проблем не припада ниједној од постојећих класа или да се истовремено може подвести под више класа. Потпуна класификација хаб локацијских проблема је веома тежак задатак и у досадашњој литератури није предложена класификација која обухвата све постојеће хаб локацијске проблеме. У Табели 1.1 је наведено неколико најчешће коришћених критеријума за класификацију хаб локацијских проблема.

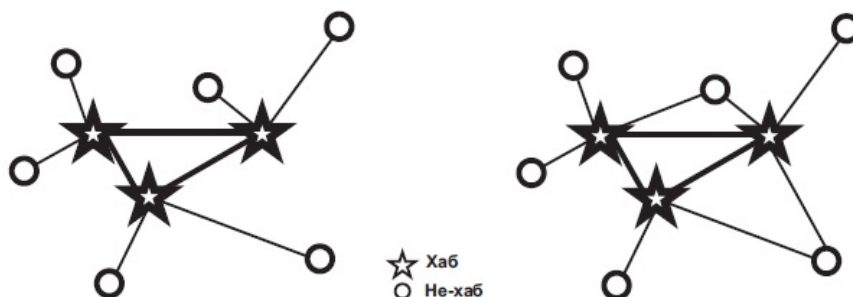
Табела 1.1: Најчешће коришћени критеријуми за класификацију хаб локацијских проблема

топографија простора	капацитет	алокација	функција циља	одређивање броја хабова	број хабова	трошкови
дискрети континуални мрежни	ограничен неограничен	једнострука вишеструка $r$ -алокација	медијана центар покривање скупа максимално покривање	ендогени егзогени	један више	фиксни променљиви

Хаб локацијски проблеми се, **на основу топографије простора** у коме се налазе потенцијалне локације хабова, могу поделити на: дискретне, континуалне и мрежне. Код дискретних проблема, хабове је могуће поставити само на дискретном скупу тачака (у равни или на мрежи), док континуални проблеми дозвољавају лоцирање хабова било где у континуалном простору (раван, сфера итд). Мрежни проблеми представљају комбинацију дискретних и континуалних, јер је успостављање хабова могуће било где на задатој мрежи, под претпоставком да је скуп чворова мреже коначан, а скуп тачака на сваком од лукова мреже који спајају чворове бесконачан.

На основу броја хабова које је неопходно успоставити, хаб локацијски проблеми се деле на проблеме са једним хабом (енгл. *single hub*) и проблеме са више хабова (енгл. *multiple hubs*). Број хабова које треба успоставити може бити унапред задат (ендогени хаб проблеми), или се може одређивати приликом решавања проблема (егзогени хаб проблеми). Код ендогених проблема, број хабова које треба лоцирати представља улазни параметар, који се најчешће означава са  $p$  у називу проблема. У зависности од начина рутирања протока, разликују се два основна алокацијска концепта: шема једноструке алокације (енгл. *single allocation scheme*) и шема вишеструке алокације (енгл. *multiple allocation scheme*). Шема једноструке алокације подразумева да је сваки не-хаб чвор придружен тачно једном, претходно успостављеном, хабу. Код шеме једноструке алокације, целокупан транспорт од почетног чвора до неког не-хаб чвора се реализује искључиво преко хаба коме је дати не-хаб чвор придружен. Шема вишеструке алокације допушта сваком не-хаб чвору да комуницира са једним или више успостављених хабова, што значи да сваки не-хаб чвор може слати и примати проток преко више од једног хаба. На Слици 1.1 илустровани су примери мреже хабова са шемама једноструке и вишеструке алокације. Шема вишеструке алокације омогућава већу флексибилност модела, јер за дати скуп хабова транспорт између снабдевача и корисника може бити реализован путем са најнижом ценом транспорта, независно од осталих чворова. Имајући у виду ситуације које се најчешће дешавају у пракси, већина хаб локацијских проблема подразумева шему са једноструком алокацијом, јер је сваки не-хаб чвор обично придружен свом најближем хабу. Уопштење ова два основна алокацијска концепта је шема  $r$ -алокације (енгл. *r-allocation scheme*), која дозвољава сваком не-хаб чвору да комуницира са највише  $r$  хабова ( $1 \leq r \leq p$ ), где је  $p$  укупан број успостављених хабова.

Према постојању ограничења капацитета у мрежи, разликују се хаб локацијски проблеми неограничених и ограничених капацитета. Код проблема неограничених капацитета, претпоставља се да нема услова који се односе на ограничавање капацитета у посматраној мрежи. Уколико то није случај, ради се о хаб локацијским проблемима ограничених капацитета. Ограничења могу бити постављена на количину протока која се сакупља у хабу или шаље из хаба, количину протока које може да прихвати и рутира целокупна мрежа хабова или појединачне гране између хабова итд. **Цена успостављања хабова или веза између хабова**, у општем случају, не мора бити укључена у укупну цену



Слика 1.1: Пример мреже хабова са шемом једноструке алокације (лево) и шемом вишеструке алокације (десно)

транспорта. Уколико практична ситуација подразумева и трошкове овог типа, они даље могу бити фиксни или променљиви, што води ка хаб локацијским проблемима са фиксним, односно променљивим трошковима.

**У зависности од типа функције циља**, хаб локацијски проблеми се могу поделити на: проблеме хаб медијане (енгл. *hub median problems*), проблеме хаб центра (енгл. *hub center problems*) и проблеме хаб покривања (енгл. *hub covering problems*). Код проблема хаб медијане, циљ је минимизовати укупне суме трошкова (времена) транспорта по свим паровима снабдевач-корисник, док се код проблема хаб центра минимизује максимална цена транспорта међу свим паровима снабдевач-корисник у мрежи. Циљ проблема хаб покривања скупа је одређивање најмањег скупа хабова који је потребан за покривање свих парова снабдевач-корисник при унапред задатом критеријуму покривања. У литератури постоји више варијанти проблема хаб покривања, које се могу сврстати у две групе: проблеми хаб покривања скупова (енгл. *Set Covering Problem - SCP*) и проблеми хаб максималног покривања (енгл. *Hub Maximal Covering Problem - HMCP*). Међутим, у многим случајевима, покривање свих парова снабдевач-корисник није могуће (нпр. због ограниченог буџета). Из тог разлога, уведени су проблеми хаб максималног покривања који за циљ имају одређивање оптималних локација за успостављање унапред задатог броја хабова, тако да укупни проток међу свим покривеним паровима снабдевач-корисник буде максималан. Код ових проблема, функција циља не укључује фиксне трошкове за успостављање хабова.

Код проблема хаб покривања, пар корисник-снабдевач се сматра *покривеним*, уколико је цена транспорта (односно, време неопходно за транспорт, или

растојање) од чвора-снабдевача до чвора-корисника није већа од од унапред задате максималне цене транспорта  $\beta$  (односно, времена или растојања). У супротном, пар корисник-снабдевач *није покривен*. Овај тип покривености је први користио Campbell у [25], а нешто касније и Karimi у [90], док је у раду [121] овај концепт означен терминима *бинарно* или *константно покривање* (енгл. *binary coverage*), који се надаље користе у литератури. Међутим, у пракси постоје проблеми за које коришћење бинарног покривања у одговарајућим математичким моделима води до незадовољавајућих резултата. Један такав пример је оптимизација система брзе испоруке. Ако су трошкови транспорта за пар снабдевач-корисник незнатно нижи у односу на унапред задату максималну вредност  $\beta$  (нпр.  $\beta - \epsilon$ , где је  $\epsilon > 0$  мали позитиван број) у случају бинарног покривања, овај пар се сматра покривеним, док се парови код којих су трошкови транспорта нешто већи од  $\beta$  (нпр.  $\beta + \epsilon$ ,  $\epsilon > 0$ ) сматрају непокривеним. Искључивање свих испорука за које су трошкови транспорта незнатно виши од постављене горње границе може довести до значајног пада профита компаније за брзу испоруку робе. У оваквим ситуацијама, природно се намеће потреба да својство покривености буде функција која зависи од трошкова, односно растојања. Из тог разлога, у раду [121] уведен је концепт *парцијалног (делимичног) покривања* (енгл. *partial coverage*), који укључује „*степен покривености*” (енгл. *degree of coverage*) између парова снабдевач-корисник. Коришћењем концепта парцијалног покривања у моделима система брзих испорука, компаније би могле реализовати изванредан број испорука које су сматране непокривеним са становишта бинарног покривања, уколико се клијенти на датим локацијама одреде за прихватање дужег времена испоруке. Компаније би додатно могле понудити нешто нижу цену испоруке клијентима које се налазе изван области одређене максималним растојањем, односно временом испоруке. На тај начин би компаније привукле већи број клијената и повећале профит.

Због широких могућности примене концепта хабова у разним областима, хаб локацијски проблеми су веома актуелни у литератури и са теоријског и са практичног аспекта. Идеју коришћења хабова у транспортним мрежама је први разматрао Goldman [70], док је прву математичку формулацију предложио O’Kelly за проблем *p*-хаб медијане неограничених капацитета са једноструком алокацијом (енгл. *Uncapacitated Single Allocation p-hub Median Problem – USApHMP*), у раду [114]. O’Kelly је разматрани USApHMP формулисао као проблем квадратног програмирања са  $O(n^2)$  променљивих и  $O(n^2)$  ограничења, где је са  $n$  озна-

чен укупан број чворова у посматраној мрежи. Како је коришћењем квадратне формулације било тешко одредити оптимална решења проблема USApHMP, чак и у случају малих димензија проблема, O’Kelly је предложио и две једноставне похлепне хеуристике за његово решавање. Проучавајући реалне податке о путничкој мрежи авио-компаније између 25 градова у САД, O’Kelly је увео скуп тест примера у литератури познат као САВ инстанце (енгл. *Civil Aeronautics Board*). Прву линеарну формулацију USApHMP, у виду мешовитог целобројног линеарног програма са  $O(n^4)$  променљивих и  $O(n^4)$  ограничења, предложио је Campbell у раду [25]. Модификацију линеарне формулације USApHMP из [25] дали су Skorin Каров и сар. у раду [134]. Ова формулација, добијена релаксацијом одређених ограничења модела из [25], има  $O(n^4)$  променљивих и  $O(n^3)$  ограничења. Ernst и Krishnamoorthy су нешто касније, у раду [52], предложили нову мешовиту целобројну линеарну формулацију USApHMP, са  $O(n^3)$  променљивих и  $O(n^2)$  ограничења. У раду [52], Ernst и Krishnamoorthy су такође увели скуп тест примера познат као AP инстанце (енгл. *Australian Post*), добијен на основу података о поштанском саобраћају у Аустралији. Ebery је у раду [49] први формулисао USApHMP у виду мешовитог целобројног линеарног програма са  $O(n^2)$  променљивих и  $O(n^2)$  ограничења. Иако формулација USApHMP из [49] има мањи број променљивих од формулације предложене у [52], она ипак не омогућава егзактном решавачу достизање бољих решења. Штавише, коришћењем формулације из [49], могуће је добити само решења проблема са највише три успостављена хаба. Прву математичку формулацију за проблем  $p$ -хаб медијане неограничених капацитета са вишеструким алокацијама (енгл. *Uncapacitated Multiple Allocation p-hub Median Problem – UMAPHMP*) предложио је Campbell у [25]. Ова формулација је представљена као мешовити целобројни линеарни проблем са  $O(n^4)$  променљивих и  $O(n^4)$  ограничења. Skorin Каров и сар. су у раду [134] предложили математичку формулацију UMAPHMP са  $O(n^4)$  променљивих и  $O(n^4)$  ограничења, добијену модификацијом одређених ограничења модела из [25]. Нешто касније, Ernst и Krishnamoorthy су у раду [53] предложили нову математичку формулацију UMAPHMP у виду мешовитог целобројног линеарног програма са  $O(n^3)$  променљивих и  $O(n^3)$  ограничења. Voland и сар. су у раду [21] предложили мешовиту целобројну линеарну формулацију UMAPHMP са  $O(n^3)$  променљивих и  $O(n^2)$  ограничења. Garcia и сар. су у раду [61] изложили нову целобројну формулацију UMAPHMP са  $O(n^2)$  променљивих и  $O(n^2)$  ограничења и предложили егзактну методу гранања и

одсецања, која је примењена за решавање AP инстанци до 200 чворова. Yaman је у раду [149] увела појам  $r$ -алокације за  $p$ -хаб проблем медијане. Полазећи од формулација UMAPHMP из [134] и [53], у раду [149], предложене су две мешовите целобројне линеарне математичке формулације за проблем  $p$ -хаб медијане неограничених капацитета са  $r$ -алокацијском шемом (енгл. *Uncapacitated  $r$ -allocation  $p$ -hub Median Problem – UrApHMP*). Прва формулација садржи  $O(n^4)$  променљивих и  $O(n^3)$  ограничења, док друга садржи  $O(n^3)$  променљивих и  $O(n^3)$  ограничења. UrApHMP припада класи NP-тешких проблема, као генерализација USApHMP и UMAPHMP проблема, за које је показано да су NP-тешки [103].

Прве математичке формулације проблема  $p$ -хаб центра неограничених капацитета са једноструким алокацијама (енгл. *Uncapacitated Single Allocation  $p$ -hub Center Problem – USApHCP*) предложио је Campbell у раду [25]. Он је USApHCP формулисао најпре у виду квадратног целобројног програма са  $O(n^2)$  променљивих и  $O(n^2)$  ограничења, а затим је предложио и реформулацију проблема у виду целобројног линеарног програма са  $O(n^4)$  променљивих и  $O(n^4)$  ограничења. Ипак, у раду [25] нису приказани експериментални резултати са предложеним формулацијама. Нешто касније, у раду [87], Kara и Tansel су најпре доказали да је USApHCP NP-тежак проблем оптимизације, а затим предложили нову формулацију USApHCP у виду мешовитог целобројног линеарног програма са  $O(n^2)$  променљивих и  $O(n^3)$  ограничења. У раду [87], приказани су експериментални резултати добијени на САВ инстанцама, поређењем предложених линеарних формулација коришћењем егзактог решавача CPLEX 5.0. Касније су Ernst и сар. [51] доказали да је чак и за фиксиран број хабова, проблем одређивања оптималних алокација за USApHCP, NP-тежак. Аутори су у [51] такође предложили нову мешовиту целобројну линеарну формулацију USApHCP која укључује  $O(n^2)$  променљивих и  $O(n^2)$  ограничења. У раду [51] је први пут разматран проблем  $p$ -хаб центра неограничених капацитета са вишеструким алокацијама (енгл. *Uncapacitated Multiple Allocation  $p$ -hub Center Problem – UMAPHCP*). Најпре је показано да UMAPHCP припада класи NP-тешких проблема оптимизације, а затим је предложена његова мешовита целобројна линеарна формулација са  $O(n^3)$  променљивих и  $O(n^3)$  ограничења. У раду [51] су такође приказани експериментални резултати предложене методе гранања и ограничавања за решавање USApHCP и UMAPHCP на САВ и AP инстанцама величине до 200 чворова.

У односу на остале хаб локацијске проблеме, проблемима хаб покривања (а нарочито хаб максималног покривања) је у литератури посвећено најмање пажње, што упућује на недовољну истраженост ове проблематике (видети [7,90]). Кара и Tansel су у раду [88] предложили целобројну линеарну математичку формулацију за проблем хаб покривања скупа неограничених капацитета са једноструким алокацијама (енгл. *Uncapacitated Single Allocation Hub Covering Problem – USAHCP*) која садржи  $O(n^2)$  променљивих и  $O(n^4)$  ограничења. Такође су доказали да је USAHCP NP-тежак проблем оптимизације. Wagner је у раду [147] предложио целобројну линеарну формулацију USAHCP, добијену релаксацијом једног ограничења модела из [88], која има  $O(n^2)$  променљивих и  $O(n^4)$  ограничења. У раду [147], Wagner је такође предложио мешовиту целобројну линеарну формулацију USAHCP која укључује проценте уштеде, остварене транспортом између хабова, који зависе од количине робе која се преноси. Ова формулација има  $O(n^3)$  променљивих и  $O(n^4)$  ограничења. Ernst и сар. су у раду [50] предложили нову формулацију за USAHCP у виду мешовитог целобројног линеарног програма са  $O(n^2)$  променљивих и  $O(n^2)$  ограничења. Alumur и Кара [8] су решавале једну варијанту транспортног проблема, представљајући га у виду варијанте проблема USAHCP у којем је занемарена општа претпоставка комплетне хаб мреже. Варијанта USAHCP разматрана у [8] је формулисана у виду мешовитог нелинеарног модела са  $O(n^3)$  променљивих и  $O(n^3)$  ограничења.

У овој докторској дисертацији су разматране три варијанте проблема хаб максималног покривања са унапред задатим бројем хабова:

1. проблем  $p$ -хаб максималног покривања неограничених капацитета са једноструким алокацијама (енгл. *Uncapacitated Single Allocation  $p$ -hub Maximal Covering Problem – USA $p$ HMCP*);
2. проблем  $p$ -хаб максималног покривања неограничених капацитета са вишеструким алокацијама (енгл. *Uncapacitated Multiple Allocation  $p$ -hub Maximal Covering Problem – UMA $p$ HMCP*);
3. проблем  $p$ -хаб максималног покривања неограничених капацитета са  $r$ -алокацијском шемом (енгл. *Uncapacitated  $r$ -allocation  $p$ -hub Maximal Covering Problem – UrA $p$ HMCP*).

Циљ USA $p$ HMCP, UMA $p$ HMCP и UrA $p$ HMCP је максимизација протока кроз мрежу чворова са унапред задатим бројем хабова ( $p$ ) и одговарајућом



шемом алокације. За сва три проблема разматрана су два концепта покривања, бинарни и парцијални.

До сада је у литератури предложено неколико математичких формулација проблема  $USArHMCP$  и  $UMArHMCP$ , као и неколико хеуристика за њихово решавање. Први математички модели за  $USArHMCP$  и  $UMArHMCP$ , предложени су у [25] у виду мешовитих целобројних линеарних програма и подразумевали су бинарни концепт покривања. Hwang и Lee [80] су проучавали  $USArHMCP$  у случају бинарног покривања и предложили су нову целобројну линеарну формулацију овог проблема. Weng и сар. [148] су дали нову целобројну линеарну математичку формулацију за  $UMArHMCP$ , у случају бинарног покривања. Рекер и Кара [121] су 2015. године увели нове мешовите целобројне линеарне формулације проблема  $USArHMCP$  и  $UMArHMCP$ , које се могу применити у оба случаја покривања - бинарном и парцијалном. Иако је у литератури концепт парцијалног покривања већ коришћен код других локацијских проблема покривања (видети [2, 17, 18, 28, 60, 89]), Рекер и Кара у [121] први пут уводе концепт парцијалног покривања код хаб локацијских проблема. Проблем  $p$ -хаб максималног покривања неограничених капацитета са  $r$ -алокацијском шемом до сада није разматран у литератури. Детаљнији преглед релевантне литературе за  $USArHMCP$  и  $UMArHMCP$  је изложен редом у Одељцима 2.1 и 3.1, док се систематичан преглед литературе која се односи на различите варијанте хаб локацијских проблема може наћи у радовима [7] и [54].

У овој дисертацији, предложене су нове математичке формулације за  $USArHMCP$  и  $UMArHMCP$  са бинарним и парцијалним покривањем, које су у смислу ефикасности упоређене са постојећим формулацијама из литературе. Такође, по први пут у литератури, уведене су математичке формулације  $UrArHMCP$  за оба концепта покривања.

За проблеме  $USArHMCP$  и  $UMArHMCP$  је познато да припадају класи  $NP$ -тешких проблема [121]. У случају  $USArHMCP$ , чак и за задати скуп хабова, показано је да је добијени потпроблем оптималне алокације не-хаб чворова успостављеним хабовима такође  $NP$ -тежак [121]. Лако се може закључити да и  $UrArHMCP$  припада класи  $NP$ -тешких проблема, као генерализација проблема за које постоји доказ да су  $NP$ -тешки.

С обзиром да проблеми хаб покривања имају највећу примену у оптимизацији телекомуникацијских и транспортних мрежа, које најчешће укључују велики број чворова, решавање инстанци проблема великих димензија је од вели-

ког практичног значаја. Имајући у виду сложеност разматраних проблема, егзактне методе не могу дати решења за инстанце проблема већих димензија. Из тог разлога, у овој дисертацији су имплементиране адекватне метахеуристичке методе које у кратком времену извршавања обезбеђују решења инстанци проблема великих димензија. До сада су у литератури предложене једноставне, али недовољно ефикасне хеуристичке методе за решавање проблема  $USArHMCP$  и  $UMArHMCP$  са бинарним покривањем, док варијанте ових проблема са парцијалним покривањем до сада нису решаване метахеуристичким методама. Како  $UrArHMCP$  са бинарним и парцијалним покривањем до сада није разматран у литератури, не постоје предложене методе за његово решавање.

Остатак дисертације је организован на следећи начин. У Главама 2, 3 и 4 изложени су резултати истраживања редом за проблеме  $USArHMCP$ ,  $UMArHMCP$  и  $UrArHMCP$ . За сваки од проблема су најпре наведене постојеће формулације из литературе (уколико је проблем раније разматран) и предложене нове математичке формулације проблема које се могу применити на оба концепта покривања. Затим су изложени детаљи развијених метахеуристика за решавање датог проблема. На крају сваке од Глава 2–4, презентовани су експериментални резултати добијени упоређивањем наведених формулација (у смислу ефикасности) и резултати предложених метахеуристика имплементираних на постојећим хаб инстанцама малих, средњих и великих димензија. У Глави 5 најпре су истакнути изведени закључци и научни доприноси дисертације, а затим су наведени правци будућих истраживања.

## Глава 2

# Проблем $p$ -хаб максималног покривања неограничених капацитета са једноструким алокацијама

### 2.1 Опис проблема и преглед релевантне литературе

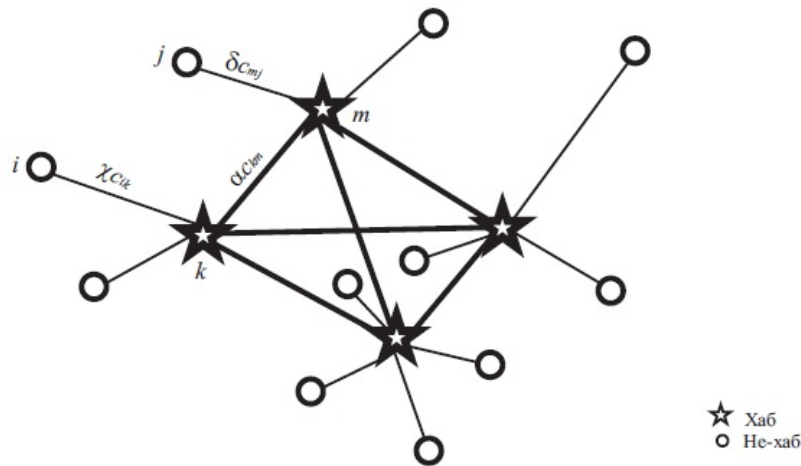
Проблем је дефинисан на комплетном симетричном графу  $G = (N, E)$ , где је  $N = \{1, 2, \dots, n\}$  скуп различитих чворова мреже, а  $E = \{(i, j) : i, j \in N\}$  скуп грана неограниченог капацитета. Сваки чвор представља локацију снабдевача или корисника, као и потенцијалну локацију хаба. Трошкови транспорта по јединици количине протока преко гране  $(i, j) \in E$  су означени са  $c_{ij}$ , при чему је  $c_{ij} = c_{ji}$  и претпоставља се да важи неједнакост троугла:  $c_{ij} \leq c_{ik} + c_{kj}$ , за свако  $i, j, k = 1, \dots, n$ . Ако не постоји грана од одлазног чвора (снабдевача)  $i$  до долазног чвора (корисника)  $j$ , по договору важи  $c_{ij} = \infty$ . Трошкови транспорта између два чвора су обично пропорционални њиховом међусобном растојању, па се за вредност  $c_{ij}$  може узети растојање између  $i$ -тог и  $j$ -тог чвора. За сваки одлазно-долазни (О-Д) пар  $i - j$ ,  $i, j \in N$  позната је потражња  $T_{ij}$  од чвора  $i$  до чвора  $j$ . Број хабова из скупа  $H \subseteq N$  које треба лоцирати је унапред задат -  $p$ . Након одређивања скупа хабова  $\overline{H}$ , потребно је одредити и одговарајуће алокације не-хаб чворова. Претпоставља се да је сваки не-хаб чвор

ГЛАВА 2. ПРОБЛЕМ  $p$ -ХАБ МАКСИМАЛНОГ ПОКРИВАЊА  
НЕОГРАНИЧЕНИХ КАПАЦИТЕТА СА ЈЕДНОСТРУКИМ АЛОКАЦИЈАМА

$i \in N$  придружен тачно једном успостављеном хабу  $k \in \overline{H}$ , што значи да не-хаб чвор  $i$  прима и шаље целокупан проток искључиво преко хаба  $k$ . Сваки хаб чвор је природно придружен самом себи. Директан транспорт између не-хаб чворова није дозвољен, већ се проток од чвора снабдевача до чвора корисника се врши преко најмање једног, а највише два хаба. Цена транспорта по јединици количине протока од чвора  $i \in N$  придруженог хабу  $k \in \overline{H}$  до чвора  $j \in N$  придруженог хабу  $m \in \overline{H}$  се рачуна на следећи начин:

$$d_{ij}^{km} = \chi c_{ik} + \alpha c_{km} + \delta c_{mj}, \quad (2.1)$$

где  $\chi$ ,  $\alpha$  и  $\delta$  редом означавају параметре који множе трошкове колекције (снабдевач–хаб), трансфера (хаб–хаб) и дистрибуције (хаб–корисник) робе по јединици количине (видети Слику 2.1). Имајући у виду да је цена транспорта између хабова нижа, важи да је  $\alpha < \chi$  и  $\alpha < \delta$ . Уколико је  $\chi = \delta = 1$ , тада  $1 - \alpha$  представља коефицијент уштеде за транспорт између хабова. Међутим, ако је  $\chi = \delta = \alpha = 1$ , тада је коефицијент уштеде 0, па се проток између хабова посматра на исти начин као и проток између осталих чворова у мрежи. Циљ USApHMCP је одредити оптималне локације тачно  $p$  хабова, тако да се максимизује укупан проток по свим покривеним паровима корисник–снабдевач, уз претпоставку бинарног или парцијалног покривања.



Слика 2.1: Пример мреже хабова

Проблеми хаб максималног покривања највише налазе примену у оптимизацији система брзе испоруке робе. У овим системима, где су потрошачи генерално јако осетљиви на време и преферирају компаније које испоручују робу у

ГЛАВА 2. ПРОБЛЕМ  $p$ -ХАБ МАКСИМАЛНОГ ПОКРИВАЊА  
НЕОГРАНИЧЕНИХ КАПАЦИТЕТА СА ЈЕДНОСТРУКИМ АЛОКАЦИЈАМА

оквиру унапред дефинисаног времена, најчешће се користи шема са једноструким алокацијама. Компаније су заинтересоване да задовоље потребе породача и покушавају да скрате време испоруке колико год је могуће, при чему је важно да се роба испоручи „на време”. Наравно, ови захтеви не могу бити у потпуности испуњени у свакој ситуацији. Из тог разлога, за циљ оптимизације мреже брзе испоруке робе се узима максимизација укупне количине робе која је испоручена у гарантованом року. У оваквим ситуацијама, предности коришћења структуре хаб мреже и концепта проблема хаб максималног покривања долазе до изражаја. Транспорт робе од почетног до крајњег чвора се врши преко оперативних центара, који се могу поистоветити са хабовима. Уобичајена рута за испоруку има три сегмента: слање робе из почетног чвора до оперативног центра који га опслужује и прикупља робу, затим трансфер до оперативног центра придруженог кориснику, и коначно, дистрибуција робе до крајњег чвора. У реалним ситуацијама, неки чворови на путу  $i \rightarrow k \rightarrow m \rightarrow j$  могу да се покlope. Прецизније, могући су следећи случајеви:

1. случај: Пут  $i \rightarrow k \rightarrow m \rightarrow j$  садржи само један хаб чвор, који се разликује од почетног и крајњег чвора (који су такође различити), тј.  $i \neq j$ ,  $k = m$  и  $i \rightarrow k \rightarrow k \rightarrow j$ . Под претпоставком да су трошкови транспорта од неког чвора до њега самог једнаки нули, важи  $d_{ij}^{kk} = \chi c_{ik} + \delta c_{kj}$ ;
2. случај: Овај случај је сличан претходном, али се почетни и крајњи чвор поклапају, тј.  $i = j$  и  $i \rightarrow k \rightarrow k \rightarrow i$ . Сада је  $d_{ii}^{kk} = \chi c_{ik} + \delta c_{ki} = (\chi + \delta)c_{ik}$ ;
3. случај: Почетни чвор је хаб чвор, а крајњи чвор је не-хаб чвор, тј.  $i = k = m$  и  $i \rightarrow i \rightarrow i \rightarrow j$ . У овом случају,  $d_{ij}^{ii} = \delta c_{ij}$ ;
4. случај: Почетни чвор је не-хаб чвор, а крајњи чвор је хаб чвор, тј.  $k = m = j$  и  $i \rightarrow j \rightarrow j \rightarrow j$ . Тада је  $d_{ij}^{jj} = \chi c_{ij}$ ;
5. случај: Почетни чвор и крајњи чвор су једнаки хаб чвору, тј.  $i = k = m = j$  и  $i \rightarrow i \rightarrow i \rightarrow i$ . У овом случају је  $d_{ii}^{ii} = 0$ ;
6. случај: Почетни и крајњи чвор су хабови, тј.  $i = k = m$  и  $i \rightarrow i \rightarrow i \rightarrow j$  или  $k = m = j$  и  $i \rightarrow j \rightarrow j \rightarrow j$  или  $i = k, m = j$  и  $i \rightarrow i \rightarrow j \rightarrow j$ . Тада је  $d_{ij}^{ii} = \delta c_{ij}$  или  $d_{ij}^{jj} = \chi c_{ij}$  или  $d_{ij}^{ij} = \alpha c_{ij}$ . Како је  $\alpha < \chi$  и  $\alpha < \delta$ , јасно је да се у пракси најчешће разматра случај  $d_{ij}^{ij} = \alpha c_{ij}$ .

ГЛАВА 2. ПРОБЛЕМ  $p$ -ХАБ МАКСИМАЛНОГ ПОКРИВАЊА  
НЕОГРАНИЧЕНИХ КАПАЦИТЕТА СА ЈЕДНОСТРУКИМ АЛОКАЦИЈАМА

---

У односу на остале хаб локацијске проблеме, проблеми  $p$ -хаб максималног покривања су најмање заступљени у литератури. Campbell је у [25] предложио прву формулацију USApHMCP за случај бинарног покривања, у виду мешовитог целобројног програма, са  $O(n^4)$  променљивих и  $O(n^2)$  ограничења. Другу формулацију USApHMCP за концепт бинарног покривања, представљену као целобројни програм, предложили су Hwang и Lee у [80]. Ова формулација има  $O(n^4)$  променљивих и  $O(n^4)$  ограничења. Hwang и Lee су такође предложили две хеуристике за решавање проблема, које користе стратегије алокација према растојању и количини протока. Аутори су у [80] изложили експерименталне резултате добијене тестирањем предложене формулације и хеуристика на САВ инстанцама проблема са 25 чворова. Reker и Kara су 2015. године у [121] увели концепт парцијалног покривања у литературу хаб локацијских проблема и предложили нову формулацију USApHMCP, која се може применити на оба концепта покривања. Ова формулација је представљена као мешовити целобројни програм, који укључује  $O(n^2)$  променљивих и  $O(n^3)$  ограничења. Иако су у [25] и [80] предложене само формулације USApHMCP у случају бинарног покривања, у [121] је показано да се оне лако могу прилагодити и случају са парцијалним покривањем. Тестирањем предложених формулација на САВ инстанцама са 25 чворова и TR (енгл. *Turkish network*) инстанцама са 81 чвором [150], показано је да формулација из [121] даје најбоље резултате, у оба случаја покривања. Reker и Kara су у [121] такође доказали да USApHMCP припада класи NP-тешких проблема. Чак и за задати скуп хабова, показано је да је добијени потпроблем оптималне алокације не-хаб чворова успостављеним хабовима такође NP-тежак. Silva и Cunha су 2017. године у раду [133] предложили нову хеустику за решавање USApHMCP у случају бинарног покривања, засновану на табу претраживању (TS). Хеуристика предложена у [133] је тестирана на САВ инстанцама, а затим су добијени резултати упоређени са резултатима изложеним у [80]. Анализом резултата приказаних у [133] закључено је да је TS хеуристика супериорнија у односу на остале две, јер се једино TS методом достижу сва оптимална решења за релативно кратко време. Silva и Cunha су у [133] такође први пут представили резултате тестирања AP инстанци до 200 чворова за USApHMCP. У досадашњој литератури нису коришћене инстанце већих димензија за тестирање  $p$ -хаб проблема максималног покривања.

## 2.2 Математичке формулације проблема

Математичка формулација проблема  $p$ -хуб максималног покривања неограничених капацитета са једноструким алокацијама, која је предложена у [121], користи следећу нотацију:

- У случају бинарног покривања, параметри  $q_{ij}^{km}$  имају вредност 1 уколико је О–Д пар  $i - j$  са придруженим хабовима  $k, m$  покривен, а 0 иначе. Прецизније,

$$q_{ij}^{km} = \begin{cases} 1, & \text{ако је } \chi c_{ik} + \alpha c_{km} + \delta c_{mj} \leq \beta, \\ 0, & \text{иначе;} \end{cases}$$

- У случају парцијалног покривања, параметри  $r_{ij}^{km}$  су дефинисани на следећи начин:

$$r_{ij}^{km} = \begin{cases} 1, & \text{ако је } \chi c_{ik} + \alpha c_{km} + \delta c_{mj} \leq \beta, \\ f(d_{ij}^{km}), & \text{ако је } \beta \leq \chi c_{ik} + \alpha c_{km} + \delta c_{mj} \leq \eta_{ij}, \\ 0, & \text{иначе,} \end{cases}$$

где је  $\eta_{ij}$  дати ненегативан параметар, а  $f$  нерастућа функција која узима вредности из интервала  $(0, 1)$ ;

- $T_{ij}$  је количина робе коју треба транспортовати од чвора  $i$  до чвора  $j$  односно, „потражња” за О–Д пар  $i - j$ ;
- Параметар  $\lambda_{ij} = \max\{q_{ij}^{km} : k, m \in H\}$  у случају бинарног покривања, док је  $\lambda_{ij} = \max\{r_{ij}^{km} : k, m \in H\}$  за парцијално покривање.

и следећа два скупа променљивих:

- Бинарна променљива  $x_{ik}$  добија вредност 1 уколико је чвор  $i$  придружен хабу  $k$ , а 0 иначе. Променљива  $x_{kk}$  добија вредност 1 уколико је чвор  $k$  хаб, а 0 иначе;
- Ненегативна реална променљива  $Z_{ij}$  означава део протока усмереног од одлазног чвора  $i$  до долазног чвора  $j$ , у случају да је О–Д пар  $i - j$  покривен. У случају бинарног покривања,  $Z_{ij}$  узима вредности из скупа  $\{0, 1\}$ , док у случају парцијалног покривања вредност ове променљиве зависи од претходно дефинисаног параметра  $r_{ij}^{km}$ .

ГЛАВА 2. ПРОБЛЕМ  $p$ -ХАБ МАКСИМАЛНОГ ПОКРИВАЊА  
НЕОГРАНИЧЕНИХ КАПАЦИТЕТА СА ЈЕДНОСТРУКИМ АЛОКАЦИЈАМА

Користећи изложену нотацију, USAрНМСР са концептом бинарног покривања може се формулисати као проблем мешовитог целобројног линеарног програмирања (енгл. *Mixed Integer Linear Programming - MILP*), као што је предложено у раду [121]:

$$\max \sum_{i \in N} \sum_{j \in N} T_{ij} Z_{ij} \quad (2.2)$$

при условима:

$$\sum_{k \in H} x_{ik} = 1, \quad \forall i \in N \quad (2.3)$$

$$x_{ik} \leq x_{kk}, \quad \forall i \in N, k \in H \quad (2.4)$$

$$\sum_{k \in H} x_{kk} = p \quad (2.5)$$

$$Z_{ij} \leq \sum_{k \in H} q_{ij}^{km} x_{ik} + \lambda_{ij}(1 - x_{jm}), \quad \forall i, j \in N, m \in H \quad (2.6)$$

$$x_{ik} \in \{0, 1\}, \quad \forall i \in N, k \in H \quad (2.7)$$

$$Z_{ij} \geq 0, \quad \forall i, j \in N. \quad (2.8)$$

Функција циља (2.2) дефинисана је као укупан покривен проток између свих О–Д парова  $i-j$ , чију вредност треба максимизовати. Услови (2.3) обезбеђују да је сваки чвор придружен тачно једном хабу, док услови (2.4) гарантују да сваки не-хаб чвор може бити алоциран само претходно успостављеном хабу. Ограничење (2.5) указује да је изабрано тачно  $p$  хабова. Услови (2.6) дефинишу део протока за покривени О–Д пар  $i-j$  са придруженим хабовима  $k, m$ . На основу ограничења (2.3), за само један хаб  $m' \in H$  важи  $x_{jm'} = 1$ , тако да се (2.6) своди на једну од следеће две неједнакости:  $Z_{ij} \leq \sum_{k \in H} q_{ij}^{km} x_{ik} + \lambda_{ij}$  ( $\forall m \neq m'$ ) или у  $Z_{ij} \leq \sum_{k \in H} q_{ij}^{km'} x_{ik}$  (за  $m = m'$ ). С обзиром на дефиницију параметра  $\lambda_{ij}$ , прва неједнакост је очигледно задовољена, а из друге следи да је  $x_{ik'} = 1$  за  $k' = \operatorname{argmax}\{q_{ij}^{km'} | k \in H\}$ . На бинарну природу променљивих  $x_{ik}$  указују услови (2.7), док услови (2.8) означавају да реалне променљиве  $Z_{ij}$  морају бити



ГЛАВА 2. ПРОБЛЕМ  $p$ -ХАБ МАКСИМАЛНОГ ПОКРИВАЊА  
НЕОГРАНИЧЕНИХ КАПАЦИТЕТА СА ЈЕДНОСТРУКИМ АЛОКАЦИЈАМА

ненегативне. Формулација (2.2)–(2.8) има  $O(n^2)$  променљивих и  $O(n^3)$  ограничења. Може се приметити да се овај модел лако може адаптирати и на случај парцијалног покривања, простом заменом параметра  $q_{ij}^{km}$  параметром  $r_{ij}^{km}$ .

У циљу представљања нове формулације за USApHMCP са концептом бинарног покривања потребно је дефинисати скуп  $G_{ij}^k = \{m \in H : q_{ij}^{km} = 1\}$ , који садржи све хабове  $m \in H$  за које је успостављен пут  $i \rightarrow k \rightarrow m \rightarrow j$  покривен. Такође, неопходно је увести и бинарне променљиве  $w_{ijk}$  које су придружене скупу  $G_{ij}^k$ , за сваки О–Д пар  $i - j$  и сваки хаб  $k$ . Прецизније, променљива  $w_{ijk}$  узима вредност 1 уколико је почетни чвор  $i$  придружен хабу  $k$  и крајњи чвор  $j$  алоциран неком хабу из скупа  $G_{ij}^k$ . У супротном, вредност променљиве  $w_{ijk}$  је једнака 0.

Користећи нове променљиве  $w_{ijk}$  и претходно наведену нотацију из [121], USApHMCP се у случају бинарног покривања може формулисати као проблем мешовитог целобројног линеарног програмирања на следећи начин:

$$\max \sum_{i \in N} \sum_{j \in N} T_{ij} Z_{ij} \quad (2.9)$$

при условима:

$$\sum_{k \in H} x_{ik} = 1, \quad \forall i \in N \quad (2.10)$$

$$x_{ik} \leq x_{kk}, \quad \forall i \in N, k \in H \quad (2.11)$$

$$\sum_{k \in H} x_{kk} = p \quad (2.12)$$

$$w_{ijk} \leq \sum_{m \in G_{ij}^k} x_{jm}, \quad \forall i, j \in N, k \in H \quad (2.13)$$

$$w_{ijk} \leq x_{ik}, \quad \forall i, j \in N, k \in H \quad (2.14)$$

$$\sum_{k \in H} w_{ijk} \leq 1, \quad \forall i, j \in N \quad (2.15)$$

$$Z_{ij} \leq \sum_{k \in H} w_{ijk}, \quad \forall i, j \in N \quad (2.16)$$

$$x_{ik} \in \{0, 1\}, \quad \forall i \in N, k \in H \quad (2.17)$$

$$Z_{ij} \geq 0, \quad \forall i, j \in N \quad (2.18)$$

$$w_{ijk} \in \{0, 1\}, \quad \forall i, j \in N, k \in H. \quad (2.19)$$

Функција циља (2.9) и услови (2.10)–(2.12) су дефинисани на исти начин као у претходно изложеној формулацији из [121]. Услови (2.13)–(2.14) обезбеђују да бинарна променљива  $w_{ijk}$  узима вредност 1 само ако је чвор  $i$  придружен хабу  $k$ , а чвор  $j$  алоциран неком хабу  $m$  из скупа  $G_{ij}^k$ , односно само у случају да је пут од  $i$  до  $j$  преко хаба  $k$  покривен. Услови (2.15) гарантују да за сваки О–Д пар  $i - j$  постоји највише једна покривена стаза, док услови (2.16) дефинишу део протока за О–Д пар  $i - j$  који је покривен. Типови променљивих  $x_{ik}$ ,  $Z_{ij}$  и  $w_{ijk}$  су дефинисани редом условима (2.17)–(2.19). Може се приметити да предложени модел (2.9)–(2.19) има  $O(n^3)$  променљивих и  $O(n^3)$  ограничења.

Формулација (2.9)–(2.19) се може лако прилагодити за случај USApHMCP са парцијалним покривањем. Прецизније, довољно је дефинисати скуп  $G_{ij}^{kl} = \{m \in H : r_{ij}^{km} = v_{ijkl}\}$ , за сваки О–Д пар  $i - j$  и сваки хаб  $k$ , где  $v_{ijkl} \in V_{ij}^k$ ,  $V_{ij}^k = \{r_{ij}^{km} > 0 : m \in H\}$ , и увести бинарну променљиву  $w_{ijkl}$ , која одговара скупу  $G_{ij}^{kl}$ , на сличан начин као у случају бинарног покривања. Тада се услови (2.13)–(2.16), који важе за случај бинарног покривања, могу преформулисати на следећи начин:

$$w_{ijkl} \leq \sum_{m \in G_{ij}^{kl}} x_{jm}, \quad \forall i, j \in N, k, l \in L_{ij}^k \quad (2.20)$$

$$w_{ijkl} \leq x_{ik}, \quad \forall i, j \in N, k, l \in L_{ij}^k \quad (2.21)$$

$$\sum_{k \in H} \sum_{l \in L_{ij}^k} w_{ijkl} \leq 1, \quad \forall i, j \in N \quad (2.22)$$

$$Z_{ij} \leq \sum_{k \in H} \sum_{l \in L_{ij}^k} v_{ijkl} w_{ijkl}, \quad \forall i, j \in N, \quad (2.23)$$

где је  $L_{ij}^k = \{1, \dots, |V_{ij}^k|\}$ .

ГЛАВА 2. ПРОБЛЕМ  $p$ -ХАБ МАКСИМАЛНОГ ПОКРИВАЊА  
НЕОГРАНИЧЕНИХ КАПАЦИТЕТА СА ЈЕДНОСТРУКИМ АЛОКАЦИЈАМА

---

Коначно, USApНМСП са концептом парцијалног покривања се може формулисати као следећи мешовити целобројни линеарни програм:

$$\max \sum_{i \in N} \sum_{j \in N} T_{ij} Z_{ij} \quad (2.24)$$

при условима:

$$\sum_{k \in H} x_{ik} = 1, \quad \forall i \in N \quad (2.25)$$

$$x_{ik} \leq x_{kk}, \quad \forall i \in N, k \in H \quad (2.26)$$

$$\sum_{k \in H} x_{kk} = p \quad (2.27)$$

$$w_{ijkl} \leq \sum_{m \in G_{ij}^{kl}} x_{jm}, \quad \forall i, j \in N, k, l \in L_{ij}^k \quad (2.28)$$

$$w_{ijkl} \leq x_{ik}, \quad \forall i, j \in N, k, l \in L_{ij}^k \quad (2.29)$$

$$\sum_{k \in H} \sum_{l \in L_{ij}^k} w_{ijkl} \leq 1, \quad \forall i, j \in N \quad (2.30)$$

$$Z_{ij} \leq \sum_{k \in H} \sum_{l \in L_{ij}^k} v_{ijkl} w_{ijkl}, \quad \forall i, j \in N \quad (2.31)$$

$$x_{ik} \in \{0, 1\}, \quad \forall i \in N, k \in H \quad (2.32)$$

$$Z_{ij} \geq 0, \quad \forall i, j \in N \quad (2.33)$$

$$w_{ijkl} \in \{0, 1\}, \quad \forall i, j \in N, k, l \in L_{ij}^k. \quad (2.34)$$

Може се приметити да предложени модел (2.24)–(2.34) има  $O(n^4)$  променљивих и  $O(n^4)$  ограничења.

Иако нови модели USApНМСП са бинарним (2.9)–(2.19) и парцијалним (2.24)–(2.34) покривањем имају већи број променљивих и ограничења у односу на одговарајуће моделе из [121], њихова главна предност се огледа у чињеници

да обједињују моделе за проблеме  $p$ -хаб максималног покривања са различитим алокацијским шемама. Нови модели се малим модификацијама лако могу трансформисати у одговарајуће моделе са вишеструким или  $r$ -алокацијама. Из тог разлога, у дисертацији су предложени модели названи *Општим моделима*.

## 2.3 Општа метода променљивих околина за решавање проблема

До сада су у литератури предложене само једноставне хеуристике за решавање USAрНМСП са бинарним покривањем (видети [80, 133]), док варијанта овог проблема са парцијаним покривањем није решавана хеуристичким методама.

Општа метода променљивих околина (GVNS) је варијанта методе променљивих околина у којој се уместо стандардне локалне претраге користи метода променљивог спуста (VND). У литератури се могу наћи скорије успешне примене GVNS метахеуристике на сличне проблеме:  $p$ -хаб медијане неограничених капацитета са једноструким алокацијама (енгл. *Uncapacitated Single Allocation  $p$ -hub Median Problem*) [81],  $p$ -хаб медијане неограничених капацитета са  $r$ -алокацијском шемом (енгл. *Uncapacitated  $r$ -allocation  $p$ -hub Median Problem*) [143],  $p$ -хаб центра неограничених капацитета са једноструким алокацијама (енгл. *Uncapacitated Single Allocation  $p$ -hub Center Problem*) [24],  $p$ -медијане у равни (енгл. *Planar  $p$ -median Problem*) [47] итд. Како се GVNS метахеуристика показала успешном за решавање неколико хаб локацијских проблема, у овој дисертацији је предложена варијанта GVNS методе прилагођена разматраном USAрНМСП са бинарним и парцијалним покривањем. Експерименталним резултатима је показана ефикасност предложене метахеуристике, као и њена супериорност у односу на постојеће из литературе. У наставку ће бити описани основни елементи предложене GVNS методе.

Основни кораци опште методе променљивих околина за решавање USAрНМСП представљени су Алгоритмом 1. Предложена GVNS метахеуристика има два параметра - први је означен са  $t_{max}$  и представља максимално дозвољено време извршавања, док је други означен са  $k_{max}$  и представља максималан број итерација унутар процедуре размрдавања. GVNS метода започиње рад генерисањем почетног решења  $S$  (које постаје текуће решење) помоћу процедуре *Initial solution*, чији је детаљан опис изложен у наставку. У свакој итерацији

ГЛАВА 2. ПРОБЛЕМ  $p$ -ХАБ МАКСИМАЛНОГ ПОКРИВАЊА  
НЕОГРАНИЧЕНИХ КАПАЦИТЕТА СА ЈЕДНОСТРУКИМ АЛОКАЦИЈАМА

---

се, након постављања реда околине  $k$  на 1, у оквиру петље са променама околина извршавају најпре процедуре размрдавања и побољшања решења, а затим померај, све док је  $k \leq k_{max}$ . Итерације се понављају све док се не достигне постављено максимално време извршавања алгоритма  $t_{max}$ .

---

**Алгоритам 1 Општа метода променљивих околина за USApHMCP**

---

```

1: function GVNS( $k_{max}, t_{max}$ );
2:    $S \leftarrow$  Initial solution;
3:   repeat
4:      $k \leftarrow 1$ ;
5:     while  $k \leq k_{max}$  do
6:        $S' \leftarrow$  Shake( $S, k$ );
7:        $S'' \leftarrow$  Nest-VND( $S'$ );
8:        $k \leftarrow k + 1$ ;
9:       if  $S''$  is better than  $S$  then
10:         $S \leftarrow S''$ ;
11:         $k \leftarrow 1$ ;
12:      end while
13:    $t \leftarrow$  CpuTime();
14: until  $t > t_{max}$ ;
15: return  $S$ ;
```

---

Како су одређивање локација хабова и придруживање не-хаб чворова постављеним хабовима две основне компоненте USApHMCP, природно се намеће захтев да **репрезентација решења** коју користи GVNS садржи обе информације. Прецизније, решење проблема је представљено уређеним паром  $S = (\bar{H}, A)$ , где је  $\bar{H} = \{h_1, h_2, \dots, h_p\}$  скуп  $p$  одабраних хабова, а  $A = (A_k)$   $k = 1, \dots, p$  је матрица чија свака врста  $A_k$  садржи чворове придружене хабу  $h_k$ .

**Иницијално решење** за USApHMCP се конструише процедуром *Initial solution* на следећи начин. Најпре је потребно одредити скуп изабраних хабова  $\bar{H}$  коришћењем похлепне хеуристике, тако да је  $|\bar{H}| = p$ . Похлепна процедура се заснива на запажању да ће минимизација дужина успостављених путева између О–Д парова вероватно водити до максимизације укупног покривеног протока. Из тог разлога, елементи скупа  $\bar{H}$  су чворови чија је максимална удаљеност од свих осталих чворова  $p$  најмања. Прецизније, нека је  $g(h)$  максимална удаљеност чвора  $h \in H$  од свих осталих чворова, тј.

$$g(h) = \max\{c_{ih} : i \in N, i \neq h\}, h \in H. \quad (2.35)$$

Чворови са  $p$  најмањих вредности функције  $g(h)$  су изабрани за иницијални скуп  $\overline{H}$  који садржи тачно  $p$  хабова. Након тога, сваки чвор се придружује свом најближем хабу из скупа  $\overline{H}$ . Оваква стратегија бирања локација хабова, као и алокација не-хаб чворова, повећава шансу успостављања што краћих путева између О–Д парова.

**Структуре околина** које се истражују у оквиру предложене GVNS метахеуристике дефинисане су операторима који мењају скуп изабраних хабова и алокације не-хаб чворова изабраним хабовима. Прецизније, за решење  $S = (\overline{H}, A)$ , дефинисане су две структуре околина:

- $Interchange\_hub(S) = \{S' = (\overline{H}', A) : \overline{H}' \subset H, |\overline{H}' \cap \overline{H}| = p - 1\}$ . Ова околина садржи сва решења која се могу добити из решења  $S$  заменом једног хаб чвора из скупа  $\overline{H}$  неким не-хаб чвором. За рачунање вредности функције циља овако добијеног суседног решења  $S'$ , довољно је израчунати трошкове на О–Д путевима који садрже нов хаб, као и разлику у количини покривеног протока у односу на претходно решење  $S$ .
- $ReAllocate\_node(S) = \{S' = (\overline{H}, A') : |A_i \setminus A'_i| = 1, |A'_j \setminus A_j| = 1, i, j \in \overline{H}, i \neq j\}$ . Ова околина садржи сва решења  $S'$  која се могу добити из решења  $S$  заменом једне не-хаб алокације, тј. премештањем једног чвора из скупа  $A_i$  у скуп  $A_j$ . Уколико се изврши премештање чвора  $l$  из скупа  $A_i$  у скуп  $A_j$ , вредност функције циља новог решења процењује се поновним проверавањем покривености протока за О–Д стазе које садрже чвор  $l$ , чија је алокација промењена.

Претходно описане структуре околина се истражују у оквиру **процедуре размрдавања** и **процедуре побољшања решења**. Процедура размрдавања *Shake*, приказана Алгоритмом 2, користи се у циљу избегавања конвергенције алгоритма ка локалном оптимуму. Према Алгоритму 2, улазни подаци за процедуру *Shake* су решење  $S$  и параметар  $k$ . У свакој од  $k$  итерација процедура размрдавања, решење  $S$  се замењује неким решењем из околине *Interchange\_hub*, изабраним на случајан начин. Решење  $S$  које се добија у  $k$ -тој итерацији је излаз из процедуре размрдавања.

Решење добијено процедуром размрдавања се користи као почетно решење процедуре побољшања, која је заснована на угњежденој методи променљивог спуста (Nest-VND) [81,143]. За свако решење  $S''$  из околине *Interchange\_hub* текућег решења  $S$  примењена је локална претрага у односу на околину *ReAlloca*–

ГЛАВА 2. ПРОБЛЕМ  $p$ -ХАБ МАКСИМАЛНОГ ПОКРИВАЊА  
НЕОГРАНИЧЕНИХ КАПАЦИТЕТА СА ЈЕДНОСТРУКИМ АЛОКАЦИЈАМА

---

$te\_node$  решења  $S''$ . Nest-VND користи стратегију претраживања до првог побољшања, тј. чим пронађе боље решење, поставља га за ново актуелно решење и наставља претрагу. Кораци примењене Nest-VND процедуре представљени су Алгоритмом 3.

---

**Алгоритам 2 Процедура размрдавања**

---

```
1: function SHAKE ( $S, k$ )
2:   for  $i = 1$  to  $k$  do
3:     Select  $S'$  from  $Interchange\_hub(S)$  at random;
4:      $S \leftarrow S'$ ;
   end
```

---

---

**Алгоритам 3 Nest-VND**

---

```
1: function NEST-VND ( $S$ )
2:    $Improve \leftarrow True$ ;
3:   while  $Improve$  do
4:      $Improve \leftarrow False$ ;
5:     for each  $S'' \in Interchange\_hub(S)$  do
6:       Select  $S'$  as the best solution in  $ReAllocate\_node(S'')$ ;
7:       if  $S'$  is better than  $S$  then
8:          $S \leftarrow S'$ ;
9:          $Improve \leftarrow True$ ;
       end
     end
   end
10: return  $S$ ;
```

---

## 2.4 Експериментални резултати

GVNS метахеуристика за решавање USApHMCР имплементирана је у програмском језику C++. Математичке формулације изложене у Одељку 2.2 су упоређене коришћењем егзактног решавача CPLEX 12.6, а све формулације су такође кодирани у програмском језику C++. Сва тестирања су извршена под Windows 10 оперативним системом са процесором Intel Core I7 са 2.8 GHz CPU и RAM меморијом од 8 GB.

Експериментални резултати добијени су тестирањем стандардних CAB и AP хаб инстанци из ORLIB библиотеке [14], као и URAND инстанци великих димензија из [81, 109]. CAB инстанце су малих димензија (до 25 чворова), а

ГЛАВА 2. ПРОБЛЕМ  $p$ -ХАБ МАКСИМАЛНОГ ПОКРИВАЊА  
НЕОГРАНИЧЕНИХ КАПАЦИТЕТА СА ЈЕДНОСТРУКИМ АЛОКАЦИЈАМА

настале су на основу података о путничком авио-саобраћају између 25 градова у САД. AP тест инстанце укључују до 200 чворова, а добијене су из података о поштанском саобраћају у Аустралији. URAND инстанце до 400 чворова су уведене у [109], док су велике инстанце са 1000 чворова предложене у [81]. За ове инстанце,  $x$  и  $y$  координате чворова су генерисане на случајан начин из скупа  $U[0,100000]$ . URAND инстанце су доступне на сајту <http://turing.mi.sanu.ac.rs/~nenad/phub/>. Ове инстанце представљају изазов за решавање, имајући у виду њихову димензију. У досадашњој литератури, URAND инстанце нису коришћене за проблеме  $p$ -хаб максималног покривања. У Табели 2.1 је дат кратак опис скупова инстанци који су коришћени за поређење формулација USApHMCP и метода за његово решавање.

Табела 2.1: Скуп података коришћених за тестирања која се односе на USApHMCP

Назив	Димензија	$\alpha$	Опис
САВ	$n = 25$	од 0.2 до 0.8	Стандардне ORLIB инстанце, уведене у раду [114], настале на основу података о путничком авио-саобраћају између 25 градова у САД.
АР	$10 \leq n \leq 200$	0.75	Стандардне ORLIB инстанце, уведене у раду [53], креиране на основу података о поштанском саобраћају у Аустралији;
URAND	$100 \leq n \leq 1000$	0.75	Инстанце до 400 чворова уведене у [109], а са 1000 чворова предложене у [81]; координате чворова су генерисане на случајан начин из скупа $U[0,100000]$ .

За све разматране инстанце, параметри прикупљања и дистрибуције протока,  $\chi$  и  $\delta$ , постављени су на 1. Трошкови транспорта  $c_{ij}$  су најчешће једнаки или пропорционални растојању између чворова  $i$  и  $j$ . Због тога је, у случају САВ инстанци, растојање између чворова једнако транспортним трошковима датим у САВ скупу података.

Параметри  $q_{ij}^{km}$  и  $r_{ij}^{km}$  коришћени редом за бинарно и парцијално покривање, дефинисани су као у [121]:

$$q_{ij}^{km} = \begin{cases} 1, & \text{ако је } d_{ij}^{km} \leq 0.75\beta, \forall i, j \in N, \forall k, m \in H \\ 0, & \text{иначе,} \end{cases} \quad (2.36)$$



$$r_{ij}^{km} = \begin{cases} 1, & \text{ако је } d_{ij}^{km} \leq 0.75\beta, \\ 0.75, & \text{ако је } 0.75\beta < d_{ij}^{km} \leq 0.8\beta, \\ 0.5, & \text{ако је } 0.8\beta < d_{ij}^{km} \leq 0.85\beta, \quad \forall i, j \in N, \forall k, m \in H \\ 0.25, & \text{ако је } 0.85\beta < d_{ij}^{km} \leq 0.9\beta, \\ 0, & \text{иначе.} \end{cases} \quad (2.37)$$

Наведене вредности параметара  $q_{ij}^{km}$  и  $r_{ij}^{km}$  су коришћене за сва тестирања.

За САВ инстанце, вредности параметра  $\beta$  су преузете из [121], док су за остале инстанце вредности параметра  $\beta$  једнаке одговарајућим вредностима решења проблема  $p$ -хаб центра са вишеструким алокацијама, приказаним у раду [23].

Нове формулације за USApHMCP и оба концепта покривања су упоређене са формулацијама предложеним у [121], коришћењем егзактног решавача CPLEX на САВ и AP инстанцама малих димензија, до 50 чворова. Добијени резултати су упоређени и са резултатима добијеним предложеном GVNS метахеуристиком за решавање USApHMCP на истим инстанцама. Како су тест инстанце великих димензија недостижне за егзактни решавач и све наведене формулације, на њима је тестирана само предложена GVNS метахеуристика. У овој дисертацији URAND инстанце су по први пут коришћене у литератури за тестирање  $p$ -хаб проблема максималног покривања.

При решавању свих тест инстанци, GVNS параметар  $k_{max}$  је постављен на  $p$ , док је вредност параметра  $t_{max}$  прилагођена димензијама решаваних инстанци, као што је приказано у Табели 2.2.

Табела 2.2: Вредности параметра  $t_{max}$  за тестирање GVNS алгоритма

Скуп података	Број чворова	$t_{max}$ (у секундама)
САВ	$n = 25$	1
AP	$10 \leq n \leq 50$	10
AP	$n = 100$	400
AP	$n = 200$	2000
URAND	$100 \leq n \leq 400$	600
URAND	$n = 1000$	1800

Квалитет резултата добијених GVNS методом је процењен упоређивањем са оптималним и субоптималним решењима добијеним егзактним решавачем CPLEX који користи формулације проблема представљене у Одељку 2.2. Време

ГЛАВА 2. ПРОБЛЕМ  $p$ -ХАБ МАКСИМАЛНОГ ПОКРИВАЊА  
НЕОГРАНИЧЕНИХ КАПАЦИТЕТА СА ЈЕДНОСТРУКИМ АЛОКАЦИЈАМА

---

извршавања CPLEX решавача је ограничено на 1h (3600s), за сваку тестирану инстанцу.

У Табелама 2.3–2.4 су приказани резултати добијени коришћењем CPLEX решавача за математичке формулације које су увели Рекер и Кага у [121] и нове формулације предложене у овој дисертацији, за случајеве бинарног и парцијалног покривања. Табеле 2.3–2.4 такође садрже резултате дизајниране GVNS метахеуристике за решавање USApHMCp и оба концепта покривања, на CAB и AP инстанцама мањих димензија. Како CPLEX решавач за инстанце већих димензија не даје чак ни допустива решења, у Табелама 2.5–2.6 су приказани само резултати GVNS метахеуристике за AP инстанце већих димензија и URAND инстанце великих димензија.

Колоне Табела 2.3–2.6 садрже следеће податке:

- $n.p$  - број чворова ( $n$ ) и хабова ( $p$ ) за дату инстанцу;
- $\alpha$  - вредност параметра трошкова по јединици количине робе за трансфер између хабова. Вредност параметра  $\alpha$  варира од 0.2 до 0.8 за инстанце из CAB скупа, док је за AP и URAND инстанце  $\alpha = 0.75$ ;
- Optimal value (%) - вредност функције циља која одговара оптималном решењу добијеним CPLEX решавачем, изражена као проценат укупног покривеног протока. Уколико CPLEX решавач није успео да у максималном времену од 1h нађе допустиво решење, ознака „-” је уписана на одговарајуће место у табели;
- Best value (%) - вредност функције циља која одговара најбољем познатом решењу, добијеном GVNS алгоритмом или CPLEX решавачем, изражена као проценат укупног покривеног протока;
- dev(%) - процентуално одступање вредности функције циља решења од вредности која одговара оптималном или најбољем познатом решењу за дату инстанцу (уколико оптимално није познато). Рачуна се на следећи начин:  $dev = 100 \cdot (value - Optimal\ value) / Optimal\ value$ , ако је познато оптимално решење, у супротном,  $dev = 100 \cdot (value - Best\ value) / Best\ value$ ;
- time(s) - време (изражено у секундама) за које CPLEX даје оптимално, односно допустиво решење. Иста нотација се користи за време које је неопходно GVNS методи да достигне своје најбоље решење.

ГЛАВА 2. ПРОБЛЕМ  $p$ -ХАБ МАКСИМАЛНОГ ПОКРИВАЊА  
НЕОГРАНИЧЕНИХ КАПАЦИТЕТА СА ЈЕДНОСТРУКИМ АЛОКАЦИЈАМА

---

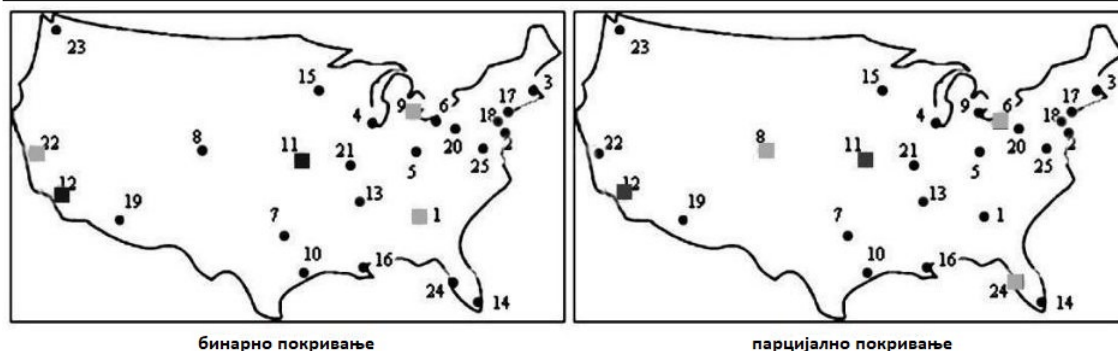
Последња врста табеле, означена са „просек“, садржи просечне вредности представљених резултата по одговарајућим колонама. Вредности функције циља које одговарају оптималним или најбољим познатим решењима су истакнуте.

На основу резултата презентованих у Табели 2.3, може се закључити да формулације USApHMCP наведене у Одељку 2.2 омогућавају CPLEX решавачу налажење свих оптималних решења за САВ инстанце, у оба случаја покривања. Користећи формулацију Рекега и Каре из [121], CPLEX налази оптимална решења за креће време у односу на Општу формулацију предложену у овој дисертацији. У просеку, у случају бинарног покривања, користећи формулацију из [121], CPLEX достиже оптимална решења за 9.20 секунди, а користећи Општу формулацију, за 16.61 секунду.

Анализирајући вредности функције циља оптималних решења, може се закључити да увођење парцијалног покривања води ка повећавању вредности функције циља оптималног решења у односу на бинарно покривање. То се може илустровати на примеру САВ инстанце са  $n = 25$  чворова,  $p = 5$  хабова и параметром  $\alpha = 0.8$  (видети Сliku 2.2). На левој страни Сlike 2.2, оптималне локације хабова у случају бинарног покривања (1, 9, 11, 12, 22) су обележене квадратићима, док су на десној страни квадратићима представљене оптималне локације хабова у случају парцијалног покривања (6, 8, 11, 12, 24). Сиви квадратићи означавају хаб локације које се разликују у бинарном и парцијалном случају, док су исте хаб локације означене црним квадратићима. Када се упореде проценти максималног покривања у оптималним решењима представљеним на Сlici 2.2, може се закључити да је вредност оптималног решења у случају парцијалног покривања (89.29%) већа него у случају бинарног покривања (86.23%). Један од разлога за ово повећање покривености је промена хаб локација и алокација, а други разлог је чињеница да неки О–Д парови који нису покривени у бинарном случају могу постати делимично покривени када се посматра случај парцијалног покривања (чак и када је одабран исти скуп хабова).

Резултати приказани у Табели 2.4 показују да у случају парцијалног покривања, за AP инстанце са  $n = 25$  чворова, CPLEX достиже оптимална решења за 72.84 секунде (у просеку) када користи формулацију из [121] и 166.08 секунди (у просеку) када користи Општу формулацију. Имајући у виду да формулација приказана у [121] има мање променљивих и услова у односу на Општу формулацију, такав резултат је очекиван. Када се анализирају резултати пре-

ГЛАВА 2. ПРОБЛЕМ  $p$ -ХАБ МАКСИМАЛНОГ ПОКРИВАЊА  
НЕОГРАНИЧЕНИХ КАПАЦИТЕТА СА ЈЕДНОСТРУКИМ АЛОКАЦИЈАМА



Слика 2.2: Оптималне хаб локације  $USA_{p=5}HMCSP$  за САВ инстанцу са  $p = 5$  хабова,  $\alpha = 0.8$ , бинарним покривањем (лево) и парцијалним покривањем (десно)

дложене GVNS хеуристике за САВ скуп података, може се приметити да GVNS достиже сва оптимална решења у оба случаја покривања за само 0.03 секунде (у просеку).

У Табели 2.4 су приказани резултати за AP инстанце до 50 чворова за случајеве бинарног и парцијалног покривања. И на овим инстанцама, формулација презентована у [121] даје боље резултате у поређењу са Општом формулацијом. У случају бинарног покривања, CPLEX користећи формулацију из [121] проналази сва оптимална решења, док Општа формулација није погодна за решавање AP инстанци са више од 40 чворова. Сличан закључак се може извести и за случај парцијалног покривања. Користећи Општу формулацију, CPLEX није пронашао чак ни допустива решења при решавању инстанци са више од 25 чворова. Са друге стране, користећи формулацију из [121], CPLEX успео да реши све AP инстанце мањих димензија до оптималности, док су у случају 8 AP инстанци добијена допустива решења за које CPLEX није успео да докаже оптималност у датом времену извршавања. Из Табеле 2.4 се може видети да предложена GVNS хеуристика достиже сва позната оптимална решења добијена CPLEX-ом и даје нова најбоља решења за 7 од 8 тестираних инстанци, за које оптималност није доказана. У случају инстанце  $n = 50$ ,  $p = 10$  и концепта парцијалног покривања, допустиво решење добијено CPLEX решавачем (оптималност није доказана) се поклапа са најбољим решењем добијеним GVNS методом. Максимално повећање вредности функције циља решења добијеног GVNS методом у односу на допустиво решење добијено CPLEX решавачем је 1.25%. У просеку, време извршавања GVNS хеуристике је 1.47 секунде за би-

*ГЛАВА 2. ПРОБЛЕМ  $p$ -ХАБ МАКСИМАЛНОГ ПОКРИВАЊА  
НЕОГРАНИЧЕНИХ КАПАЦИТЕТА СА ЈЕДНОСТРУКИМ АЛОКАЦИЈАМА*

---

нарно и 1.42 секунде за парцијално покривање.

Табела 2.3: Експериментални резултати за USArNmSP на САВ инстанцама

		Бинарно покривање				Парцијално покривање												
$n, p$	$\alpha$	Optimal	Pecker & Kara	Општа форм.	GVNS	Optimal	Pecker & Kara	Општа форм.	GVNS									
		value(%)	dev(%)	time(s)	dev(%)	time(s)	dev(%)	time(s)	dev(%)	time(s)								
25.2	0.2	<b>92.66</b>	0.00	17.46	0.00	31.39	0.00	0.00	0.00	25.2	0.2	<b>96.19</b>	0.00	24.25	0.00	160.38	0.00	0.01
25.2	0.4	<b>94.01</b>	0.00	11.22	0.00	23.66	0.00	0.01	0.00	25.2	0.4	<b>96.54</b>	0.00	33.87	0.00	96.47	0.00	0.01
25.2	0.6	<b>90.01</b>	0.00	5.60	0.00	14.71	0.00	0.01	0.00	25.2	0.6	<b>93.47</b>	0.00	40.00	0.00	243.35	0.00	0.01
25.2	0.8	<b>87.79</b>	0.00	3.05	0.00	9.77	0.00	0.00	0.00	25.2	0.8	<b>91.83</b>	0.00	31.60	0.00	74.00	0.00	0.00
25.3	0.2	<b>95.97</b>	0.00	17.52	0.00	24.24	0.00	0.01	0.00	25.3	0.2	<b>97.76</b>	0.00	35.50	0.00	149.29	0.00	0.02
25.3	0.4	<b>95.40</b>	0.00	8.61	0.00	18.41	0.00	0.01	0.00	25.3	0.4	<b>96.72</b>	0.00	30.48	0.00	276.44	0.00	0.01
25.3	0.6	<b>91.91</b>	0.00	8.93	0.00	10.16	0.00	0.01	0.00	25.3	0.6	<b>94.02</b>	0.00	57.76	0.00	200.02	0.00	0.01
25.3	0.8	<b>87.35</b>	0.00	2.78	0.00	11.42	0.00	0.02	0.00	25.3	0.8	<b>90.86</b>	0.00	68.62	0.00	204.80	0.00	0.02
25.4	0.2	<b>95.67</b>	0.00	20.02	0.00	28.94	0.00	0.03	0.00	25.4	0.2	<b>97.07</b>	0.00	56.22	0.00	150.03	0.00	0.03
25.4	0.4	<b>94.39</b>	0.00	10.82	0.00	12.10	0.00	0.02	0.00	25.4	0.4	<b>96.49</b>	0.00	54.98	0.00	138.91	0.00	0.03
25.4	0.6	<b>91.52</b>	0.00	7.23	0.00	9.55	0.00	0.02	0.00	25.4	0.6	<b>94.62</b>	0.00	45.96	0.00	131.87	0.00	0.10
25.4	0.8	<b>87.30</b>	0.00	3.13	0.00	13.89	0.00	0.08	0.00	25.4	0.8	<b>90.87</b>	0.00	149.88	0.00	305.03	0.00	0.08
25.5	0.2	<b>92.38</b>	0.00	13.37	0.00	19.40	0.00	0.05	0.00	25.5	0.2	<b>94.68</b>	0.00	139.48	0.00	143.95	0.00	0.07
25.5	0.4	<b>89.20</b>	0.00	8.84	0.00	13.20	0.00	0.04	0.00	25.5	0.4	<b>93.40</b>	0.00	99.16	0.00	94.03	0.00	0.02
25.5	0.6	<b>88.38</b>	0.00	5.57	0.00	10.71	0.00	0.05	0.00	25.5	0.6	<b>93.00</b>	0.00	98.55	0.00	132.03	0.00	0.03
25.5	0.8	<b>86.23</b>	0.00	3.01	0.00	14.19	0.00	0.04	0.00	25.5	0.8	<b>89.29</b>	0.00	122.83	0.00	240.81	0.00	0.04
<b>Просек:</b>		<b>91.26</b>	0.00	9.20	0.00	16.61	0.00	<b>0.03</b>	<b>Просек:</b>	<b>94.18</b>	0.00	68.07	0.00	171.34	0.00	<b>0.03</b>		

Табела 2.4: Експериментални резултати за USArNmSP на АР инстанцама малих димензија

		Бинарно покривање				Парцијално покривање										
$n, p$	Optimal	Pecker & Kara	Општа форм.	GVNS	Optimal/Best	Pecker & Kara	Општа форм.	GVNS								
		value(%)	dev(%)	time(s)	dev(%)	time(s)	dev(%)	time(s)								
25.2	<b>93.91</b>	0.00	5.02	0.00	13.09	0.00	0.00	0.00								
25.3	<b>93.19</b>	0.00	8.13	0.00	7.47	0.00	0.02	25.3								
25.4	<b>94.53</b>	0.00	8.14	0.00	18.11	0.00	0.03	25.4								
25.5	<b>96.31</b>	0.00	3.12	0.00	12.51	0.00	0.08	25.5								
25.10	<b>99.70</b>	0.00	0.28	0.00	2.11	0.00	0.07	25.10								
40.2	<b>97.14</b>	0.00	148.93	0.00	945.36	0.00	0.03	40.2								
40.3	<b>96.63</b>	0.00	246.14	0.00	555.76	0.00	0.05	40.3								
40.4	<b>95.20</b>	0.00	616.45	0.00	580.97	0.00	1.54	40.4								
40.5	<b>96.00</b>	0.00	199.70	0.00	355.78	0.00	1.72	40.5								
40.10	<b>99.08</b>	0.00	4.28	0.00	83.06	0.00	3.78	40.10								
50.2	<b>97.32</b>	0.00	627.68	-	-	0.00	0.14	50.2								
50.3	<b>97.28</b>	0.00	1419.52	-	-	0.00	0.18	50.3								
50.4	<b>97.01</b>	0.00	1346.77	-	-	0.00	1.09	50.4								
50.5	<b>96.96</b>	0.00	1348.08	-	-	0.00	4.49	50.5								
50.10	<b>99.21</b>	0.00	49.08	-	-	0.00	8.88	50.10								
<b>Просек:</b>		<b>96.63</b>	0.00	402.09	-	-	0.00	<b>1.47</b>	<b>Просек:</b>	<b>98.06</b>	0.00	2020.22	-	-	0.00	<b>1.42</b>

ГЛАВА 2. ПРОБЛЕМ  $p$ -ХАБ МАКСИМАЛНОГ ПОКРИВАЊА  
НЕОГРАНИЧЕНИХ КАПАЦИТЕТА СА ЈЕДНОСТРУКИМ АЛОКАЦИЈАМА

За AP инстанце већих димензија са 100 и 200 чворова и оба случаја покривања, CPLEX због ограничене меморије, не даје чак ни допустива решења ни за једну од наведених формулација. Из тог разлога, у Табели 2.5 су приказани само резултати предложене GVNS методе. У просеку, GVNS методи је потребно 475.37 секунди да достигне најбоља решења за AP инстанце већих димензија у случају бинарног покривања и 359.51 секунда за случај парцијалног покривања. Просечна вредност функције циља најбољих GVNS решења на AP инстанцама са 100 и 200 чворова је 98.46% у случају бинарног и 99.26% за случај парцијалног покривања.

У Табели 2.6 су приказани резултати тестираних URAND инстанци великих димензија добијених GVNS методом, за оба концепта покривања. Ове инстанце такође није могуће решити коришћењем егзактног CPLEX решавача. Предложена GVNS метода се показала веома ефикасном при решавању URAND инстанци, које представљају изазов за истраживаче, имајући у виду велики број чворова које укључују. У просеку, GVNS обезбеђује најбоља решења за 665.31 секунду у случају бинарног и 687.58 секунди у случају парцијалног покривања. Просечна вредност функције циља најбољих GVNS решења на URAND инстанцама са 100, 200, 300, 400 и 1000 чворова је 86.79% у случају бинарног и 93.52% за случај парцијалног покривања. URAND инстанце нису до сада коришћене за тестирање метода за решавање USApHMCP. У овој дисертацији, резултати тестирања GVNS методе на URAND инстанцама су приказани по први пут у литератури.

Табела 2.5: Резултати предложене GVNS методе за USApHMCP на AP инстанцама великих димензија

n,p	Бинарно покривање		Парцијално покривање	
	Best value(%)	time(s)	Best value(%)	time(s)
100.2	97.56	0.84	98.72	0.65
100.3	97.16	2.73	98.55	8.36
100.4	96.76	6.81	98.34	45.49
100.5	97.48	27.56	98.68	28.17
100.10	98.99	29.59	99.59	87.27
100.15	99.42	378.63	99.75	76.87
100.20	99.58	309.26	99.82	142.28
200.2	98.32	22.39	99.20	40.79
200.3	98.40	73.55	99.30	116.66
200.4	98.08	116.96	99.11	199.59
200.5	98.12	282.40	99.06	323.33
200.10	99.27	1920.18	99.71	955.18
200.15	99.59	1999.61	99.85	1259.24
200.20	99.76	1484.65	99.89	1749.30
<b>Просек:</b>	98.46	475.37	99.26	359.51

ГЛАВА 2. ПРОБЛЕМ  $p$ -ХАБ МАКСИМАЛНОГ ПОКРИВАЊА  
НЕОГРАНИЧЕНИХ КАПАЦИТЕТА СА ЈЕДНОСТРУКИМ АЛОКАЦИЈАМА

Табела 2.6: Резултати предложене GVNS методе за USApHMCP на URAND инстанцама великих димензија

n,p	Бинарно покривање		Парцијално покривање	
	Best value(%)	time(s)	Best value(%)	time(s)
100.02	84.45	0.67	92.56	0.95
100.03	79.92	7.95	89.41	6.85
100.04	78.74	10.09	88.58	15.69
100.05	81.65	9.94	90.14	69.38
100.10	87.01	81.46	93.24	103.82
100.15	89.91	200.02	94.98	144.10
100.20	91.87	367.34	96.00	390.84
200.02	87.45	146.65	94.76	21.85
200.03	82.77	119.31	90.77	132.49
200.04	82.37	108.20	90.91	461.88
200.05	81.04	488.87	89.66	518.14
200.10	88.01	600.03	93.64	600.12
200.15	88.31	600.04	94.05	600.37
200.20	90.00	600.06	94.98	600.53
300.02	89.30	75.29	95.27	131.44
300.03	86.09	230.51	92.98	400.04
300.04	84.32	542.94	91.90	600.63
300.05	83.85	600.30	91.16	600.58
300.10	87.58	600.05	94.13	602.46
300.15	89.58	600.02	94.87	601.98
300.20	90.11	600.04	95.17	601.60
400.02	89.32	479.07	95.54	600.44
400.03	84.97	600.52	92.92	602.88
400.04	84.69	600.76	92.68	601.82
400.05	83.60	600.55	91.55	602.52
400.10	88.04	600.08	94.04	605.97
400.15	89.92	600.07	95.14	606.85
400.20	90.33	600.26	95.34	606.58
1000.02	91.23	1811.74	96.67	1806.32
1000.03	87.87	1801.17	94.85	1802.79
1000.04	87.65	1800.19	94.57	1801.16
1000.05	85.28	1800.18	93.51	1807.89
1000.10	89.94	1801.04	95.45	1807.01
1000.15	90.02	1800.13	95.68	1802.68
1000.20	90.33	1800.47	96.18	1804.61
<b>Просек:</b>	86.79	665.31	93.52	687.58

У литератури је предложено неколико хеуристика за решавање USApHMCP са бинарним покривањем, а најбољи резултати до сада су добијени хеуристиком табу претраживања (TS) из рада [133]. TS хеуристика је тестирана на САВ и АР инстанцама димензије до 200 чворова. Резултати добијени TS хеуристиком су упоређени са резултатима GVNS методе за решавање USApHMCP у случају бинарног покривања и са истим вредностима параметара  $\chi$ ,  $\delta$ ,  $\alpha$  и  $\beta$ , као у раду [133]. GVNS параметар  $k_{max}$  је постављен на  $p$ , док је вредност параметра  $t_{max}$  прилагођена димензијама решаваних инстанци –  $t_{max} = n$  секунди за инстанце до 50 чворова и  $t_{max} = 2000$  за велике инстанце са 100 и 200 чворова. Према подацима из рада [133], хеуристика табу претраживања је тестирана на



ГЛАВА 2. ПРОБЛЕМ  $p$ -ХАБ МАКСИМАЛНОГ ПОКРИВАЊА  
НЕОГРАНИЧЕНИХ КАПАЦИТЕТА СА ЈЕДНОСТРУКИМ АЛОКАЦИЈАМА

рачунару високих перформанси - коришћен је кластер са 64 сервера, од којих сваки има 10 језгара са 512 GB RAM меморије и процесором Intel(R) Xeon(R) CPU E7-2870 са 2.40 GHz. Са друге стране, GVNS је тестиран на рачунару знатно слабијих перформанси - коришћен је процесор Intel(R) Core(TM) I7-5500U са 2.8 GHz CPU и RAM меморијом од 8 GB. У циљу коректног поређења резултата TS и GVNS метода, извршено је скалирање времена извршавања TS методе приказаних у [133], коришћењем приступа описаног у раду [42] и података са сајта <http://www.cpubenchmark.net/>. Скалирано време TS алгоритма, у ознаци  $n\text{time}(s)$ , се рачуна на основу времена  $\text{time}(s)$  из [133] на следећи начин:

$$n\text{time}(s) = \text{time}(s) \cdot \frac{PCPUS(Intel(R)Xeon(R)CPUE7 - 2870)}{PCPUS(Intel(R)Core(TM)I7 - 5500U)}. \quad (2.38)$$

PCPUS представља одговарајући Passmark CPU Score за сваку од коришћених конфигурација, добијен са наведеног сајта.

Табела 2.7: Резултати TS хеуристике из [133] и предложене GVNS методе за USApHMCP на САВ инстанцама - случај бинарног покривања

n.p	$\alpha$	Optimal value	TS			GVNS	
			Best value	time(s)	ntime(s)	Best value	time(s)
25.3	0.2	<b>8206780</b>	<b>8206780</b>	1.50	66.93	<b>8206780</b>	0.03
25.3	0.4	<b>8146860</b>	<b>8146860</b>	1.53	68.00	<b>8146860</b>	0.01
25.3	0.6	<b>7848986</b>	<b>7848986</b>	1.69	75.12	<b>7848986</b>	0.05
25.3	0.8	<b>7459456</b>	<b>7459456</b>	1.70	75.47	<b>7459456</b>	0.05
25.4	0.2	<b>8170172</b>	<b>8170172</b>	3.37	149.83	<b>8170172</b>	0.06
25.4	0.4	<b>8062774</b>	<b>8062774</b>	2.95	131.28	<b>8062774</b>	0.03
25.4	0.6	<b>7815464</b>	<b>7815464</b>	2.95	131.05	<b>7815464</b>	0.05
25.4	0.8	<b>7433730</b>	<b>7433730</b>	3.10	137.99	<b>7433730</b>	0.19
25.5	0.2	<b>7889428</b>	<b>7889428</b>	5.09	226.42	<b>7889428</b>	0.13
25.5	0.4	<b>7624346</b>	<b>7624346</b>	5.07	225.48	<b>7624346</b>	0.13
25.5	0.6	<b>7552580</b>	<b>7552580</b>	5.04	224.37	<b>7552580</b>	0.06
25.5	0.8	<b>7473474</b>	<b>7473474</b>	4.95	220.41	<b>7473474</b>	0.30
<b>Average:</b>		<b>7807004.17</b>	<b>7807004.17</b>	3.24	144.36	<b>7807004.17</b>	<b>0.09</b>

ГЛАВА 2. ПРОБЛЕМ  $p$ -ХАБ МАКСИМАЛНОГ ПОКРИВАЊА  
НЕОГРАНИЧЕНИХ КАПАЦИТЕТА СА ЈЕДНОСТРУКИМ АЛОКАЦИЈАМА

Табела 2.8: Резултати TS хеуристике из [133] и предложене GVNS методе за USApHMCP на AP инстанцама - случај бинарног покривања

n.p	Optimal/Best value	TS			GVNS	
		Best value	time(s)	ntime(s)	Best value	time(s)
25.2	<b>3787.08</b>	<b>3787.08</b>	0.56	24.96	<b>3787.08</b>	0.01
25.3	<b>3718.46</b>	<b>3718.46</b>	1.69	75.21	<b>3718.46</b>	0.03
25.4	<b>3761.13</b>	<b>3761.13</b>	3.42	152.37	<b>3761.13</b>	0.06
25.5	<b>3832.13</b>	<b>3832.13</b>	5.68	252.58	<b>3832.13</b>	0.14
40.2	<b>3870.69</b>	<b>3870.69</b>	3.55	157.89	<b>3870.69</b>	0.08
40.3	<b>3876.29</b>	<b>3876.29</b>	10.73	477.40	<b>3876.29</b>	0.14
40.4	<b>3831.77</b>	<b>3831.77</b>	21.60	961.33	<b>3831.77</b>	0.56
40.5	<b>3819.57</b>	<b>3819.57</b>	36.44	1621.71	<b>3819.57</b>	4.08
50.2	<b>3921.67</b>	<b>3921.67</b>	8.45	375.89	<b>3921.67</b>	0.09
50.3	<b>3912.26</b>	<b>3912.26</b>	25.66	1142.05	<b>3912.26</b>	0.94
50.4	<b>3859.75</b>	<b>3859.75</b>	52.50	2336.38	<b>3859.75</b>	2.67
50.5	<b>3858.03</b>	<b>3858.03</b>	83.33	3708.14	<b>3858.03</b>	10.88
100.2	<b>3909.69</b>	3909.32	126.78	5641.49	<b>3909.69</b>	10.33
100.3	<b>3898.43</b>	<b>3898.43</b>	383.02	17044.48	<b>3898.43</b>	14.14
100.4	<b>3874.87</b>	<b>3874.87</b>	783.47	34864.42	<b>3874.87</b>	42.00
100.5	<b>3882.63</b>	3882.24	1344.83	59844.76	<b>3882.63</b>	45.66
200.2	<b>3920.44</b>	3919.57	2749.37	122346.97	<b>3920.44</b>	153.38
200.3	<b>3926.45</b>	3926.30	8531.63	379657.54	<b>3926.45</b>	265.17
200.4	<b>3910.02</b>	3909.65	16899.61	752032.65	<b>3910.02</b>	266.89
200.5	<b>3918.60</b>	3910.93	29887.01	1329971.95	<b>3918.60</b>	318.76
<b>Просек:</b>	<b>3864.50</b>	3864.01	3047.97	135634.51	<b>3864.50</b>	<b>56.80</b>

У Табелама 2.7 и 2.8 су приказани резултати TS метахеуристике предложене у [133] и GVNS метахеуристике предложене у овој дисертацији на САВ и AP инстанцама, у случају бинарног покривања. Колона *Optimal/Best value* садржи вредности функције циља које одговарају оптималним решењима добијеним CPLEX решавачем, односно вредности функције циља које одговарају најбољим познатим решењима добијеним TS или GVNS алгоритмом. Из резултата приказаних у Табели 2.7 може се видети да су оба алгоритма достигла оптимална решења (претходно добијена CPLEX решавачем) на свим САВ и AP инстанцама до 50 чворова. Са друге стране, GVNS метода је на овим инстанцама доста бржа у односу на TS хеустику. Скалирано просечно време потребно TS алгоритму за добијање свих оптималних решења на САВ инстанцама је 144.36 секунди, док је просечно време потребно GVNS алгоритму 0.09 секунди. На AP инстанцама већих димензија (100 и 200 чворова), GVNS алгоритам достиже најбоља позната решења за 6 од 8 тестираних инстанци, а на преостале две GVNS и TS методе дају иста решења. У просеку, за све тестиране AP инстанце, скалирано време извршавања TS алгоритма је 135634.51 секунда, а време извршавања GVNS алгоритма је 56.80 секунди.

На основу изложених експерименталних резултата, може се закључити да предложена GVNS метахеуристика успешно решава USApHMCP за оба кон-

*ГЛАВА 2. ПРОБЛЕМ  $p$ -ХАБ МАКСИМАЛНОГ ПОКРИВАЊА  
НЕОГРАНИЧЕНИХ КАПАЦИТЕТА СА ЈЕДНОСТРУКИМ АЛОКАЦИЈАМА*

---

цента покривања.

## Глава 3

# Проблем $p$ -хаб максималног покривања неограничених капацитета са вишеструким алокацијама

### 3.1 Опис проблема и преглед релевантне литературе

Циљ проблема  $p$ -хаб максималног покривања неограничених капацитета са вишеструким алокацијама (UMArHMCP) је одређивање оптималних локација за успостављање тачно  $p$  хабова, тако да се максимизује укупан проток по свим покривеним паровима снабдевач-корисник, уз претпоставку бинарног или парцијалног покривања. Шема вишеструке алокације допушта сваком не-хаб чвору  $i \in N$  придруживање једног или више хабова  $h_i \in \bar{H}$ . Природно, сваки хаб чвор је придружен самом себи. Не-хаб чворови у мрежи не могу бити директно повезани, већ се проток од чвора снабдевача до чвора корисника реализује преко најмање једног, а највише два хаба.

Шема вишеструке алокације омогућава већу флексибилност модела у поређењу са шемом једноструке алокације. У реалним ситуацијама, за дати скуп хабова, при рутирању протока између снабдевача и корисника, могуће је избрати пут са најмањим трошковима транспорта, независно од осталих чворова. На пример, при испоруци пошиљки од пошиљаоца до примаоца, пошиљка може

бити дистрибуирана преко различитих међустаница (терминала), што може значајно умањити укупне трошкове. Слично, у авио-саобраћају при налажењу што повољнијег плана путовања од почетног до крајњег одредишта, путници могу изабрати лет који укључује једну или више потенцијалних међустаница (аеродрома-хабова).

Прву математичку формулацију UMAPHMSP са концептом бинарног покривања, у виду мешовитог целобројног програма, предложио је Campbell у [25]. Ова формулација има  $O(n^4)$  променљивих и  $O(n^4)$  ограничења. Касније су Weng и сар. у [148] предложили нову, целобројну формулацију истог проблема, са  $O(n^2)$  променљивих и  $O(n^2)$  ограничења. Користећи ову формулацију, егзактни решавач LINGO је на тестираним AP инстанцама до 25 чворова успео да достигне сва оптимална решења. У раду [148] су такође предложене две хеуристике за решавање UMAPHMSP у случају бинарног покривања, засноване на табу претраживању (TS) и генетском алгоритму (GA). Хеуристике су тестиране на AP инстанцама до 25 чворова и инстанцама већих димензија насталих на основу података о авио-саобраћају између 82 града у Кини. Peker и Kara су у свом раду [121] из 2015. године предложиле нову математичку формулацију за UMAPHMSP, која се може применити на оба концепта покривања, бинарном и парцијалном. Ова формулација је представљена као мешовити целобројни модел са  $O(n^3)$  променљивих и  $O(n^3)$  ограничења. У [121] је такође показано да се формулација UMAPHMSP предложена у [25] може лако адаптирати и за случај са парцијалним покривањем, што није могуће урадити са формулацијом из [148]. Из тог разлога су у [121] изложени само експериментални резултати добијени коришћењем предложених формулација из [25] и [121] за оба случаја покривања. Тестирања су вршена на САВ инстанцама са 25 чворова и TR (енгл. *Turkish network*) инстанцама са 81 чвором (градом), од којих су 22 чвора (индустријски најјачи градови у Турској) потенцијалне локације хабова [150]. Показано је да формулација из [121] даје боље резултате у оба случаја покривања, што је и очекивано, с обзиром да укључује мањи број променљивих и ограничења. У [121] је такође доказано да UMAPHMSP припада класи NP-тешких проблема.

## 3.2 Математичке формулације проблема

Математичка формулација УМАрНМСР, која је предложена у [121], користи нотацију наведену у претходној Одељку 2.2 и следеће скупове променљивих:

- Бинарна променљива  $x_{ijk}$  добија вредност 1 уколико је  $k$  први хаб на успостављеном путу од снабдевача  $i$  до корисника  $j$ , а 0 иначе. Бинарна променљива  $y_{ijm}$  добија вредност 1 уколико је  $m$  други хаб на успостављеном путу од снабдевача  $i$  до корисника  $j$ , а 0 иначе. Другим речима, ако је  $x_{ijk} = 1$  и  $y_{ijm} = 1$ , тада је успостављен пут  $i \rightarrow k \rightarrow m \rightarrow j$ ;
- Бинарна променљива  $H_k$  добија вредност 1 уколико је чвор  $k$  изабран за хаб, а 0 иначе.

Користећи изложену нотацију, УМАрНМСР са концептом бинарног покривања може бити формулисан као проблем мешовитог целобројног линеарног програмирања, као што је предложено у раду [121]:

$$\max \sum_{i \in N} \sum_{j \in N} T_{ij} Z_{ij} \quad (3.1)$$

при условима:

$$\sum_{k \in H} H_k = p \quad (3.2)$$

$$Z_{ij} \leq \sum_{k \in H} q_{ij}^{km} x_{ijk} + \lambda_{ij}(1 - y_{ijm}), \quad \forall i, j \in N, m \in H \quad (3.3)$$

$$Z_{ij} \leq \sum_{m \in H} q_{ij}^{km} y_{ijm} + \lambda_{ij}(1 - x_{ijk}), \quad \forall i, j \in N, k \in H \quad (3.4)$$

$$\sum_{k \in H} x_{ijk} = 1, \quad \forall i, j \in N \quad (3.5)$$

$$\sum_{m \in H} y_{ijm} = 1, \quad \forall i, j \in N \quad (3.6)$$

$$x_{ijk} \leq H_k, \quad \forall i, j \in N, k \in H \quad (3.7)$$

$$y_{ijm} \leq H_m, \quad \forall i, j \in N, m \in H \quad (3.8)$$

$$x_{ijk} \in \{0, 1\}, \quad \forall i, j \in N, k \in H \quad (3.9)$$

$$y_{ijm} \in \{0, 1\}, \quad \forall i, j \in N, m \in H \quad (3.10)$$

$$H_k \in \{0, 1\}, \quad \forall k \in H \quad (3.11)$$

$$Z_{ij} \geq 0, \quad \forall i, j \in N. \quad (3.12)$$

Функција циља (3.1) дефинисана је као укупан покривен проток између свих О–Д парова  $i - j$ , чију вредност треба максимизовати. Ограничење (3.2) обезбеђује избор локација за тачно  $p$  хабова. Услови (3.3) и (3.4) дефинишу део протока за покривени О–Д пар  $i - j$ . Услови (3.5) обезбеђују да на сваком путу од чвора  $i$  до чвора  $j$  само један хаб може бити изабран као први хаб, док услови (3.6) гарантују да на истом путу само један хаб може бити проглашен за други хаб. Условима (3.7) и (3.8) је дефинисано да успостављени пут  $i - j$  може садржати чворове  $k$  и  $m$ , респективно, само ако су  $k$  и  $m$  изабрани за хабова. Типови променљивих  $x_{ijk}$ ,  $y_{ijm}$  и  $H_k$  су одређени редом условима (3.9)–(3.11), док услови (3.12) означавају да реалне променљиве  $Z_{ij}$  морају бити ненегативне. Изложени модел (3.1)–(3.12) има  $O(n^3)$  променљивих и  $O(n^3)$  ограничења. Може се приметити да се овај модел лако може адаптирати и на случај парцијалног покривања, простом заменом параметра  $q_{ij}^{km}$  параметром  $r_{ij}^{km}$ .

Општи модели за USApHMCSP и оба концепта покривања, предложени у Одељку 2.2, могу се малим модификацијама лако трансформисати у одговарајуће моделе за UMAPHMCSP. У ту сврху, неопходно је заменити променљиве  $x_{ik}$  са  $H_k$ . Нови модел за UMAPHMCSP са концептом бинарног покривања може бити представљен као следећи мешовити целобројни линеарни програм:

$$\max \sum_{i \in N} \sum_{j \in N} T_{ij} Z_{ij} \quad (3.13)$$

при условима:

$$\sum_{k \in H} H_k = p \quad (3.14)$$

$$w_{ijk} \leq \sum_{m \in G_{ij}^k} H_m, \quad \forall i, j \in N, k \in H \quad (3.15)$$

$$w_{ijk} \leq H_k, \quad \forall i, j \in N, k \in H \quad (3.16)$$

$$\sum_{k \in H} w_{ijk} \leq 1, \quad \forall i, j \in N \quad (3.17)$$

$$Z_{ij} \leq \sum_{k \in H} w_{ijk}, \quad \forall i, j \in N \quad (3.18)$$

$$H_k \in \{0, 1\}, \quad \forall k \in H \quad (3.19)$$

$$Z_{ij} \geq 0, \quad \forall i, j \in N \quad (3.20)$$

$$w_{ijk} \in \{0, 1\}, \quad \forall i, j \in N, k \in H. \quad (3.21)$$

Функција циља (3.13) и услови (3.14)–(3.21) имају исто значење као у Општем моделу за USApHMCSP, представљеном у Одељку 2.2. Може се приметити да модел (3.13)–(3.21) такође има  $O(n^3)$  променљивих и  $O(n^3)$  ограничења.

Аналогно процедури описаној у Одељку 2.2, формулација (3.13)–(3.21) се лако прилагођава и случају са парцијалним покривањем. Користећи нотацију из Одељка 2.2, након дефинисања скупа  $G_{ij}^{kl} = \{m \in H : r_{ij}^{km} = v_{ijkl}\}$  и увођења бинарне променљиве  $w_{ijkl}$ , која одговара скупу  $G_{ij}^{kl}$ , UMApHMCSP са концептом парцијалног покривања може се формулисати као следећи мешовити целобројни линеарни програм:

$$\max \sum_{i \in N} \sum_{j \in N} T_{ij} Z_{ij} \quad (3.22)$$

при условима:

$$\sum_{k \in H} H_k = p \quad (3.23)$$

$$w_{ijkl} \leq \sum_{m \in G_{ij}^{kl}} H_m, \quad \forall i, j \in N, k, l \in L_{ij}^k \quad (3.24)$$

$$w_{ijkl} \leq H_k, \quad \forall i, j \in N, k, l \in L_{ij}^k \quad (3.25)$$





Итерације се понављају све док се не достигне постављено максимално време извршавања алгоритама  $t_{max}$ .

---

**Алгоритам 4 Основна метода променљивих околина за UMApHMSP**

---

```

1: function BVNS( $k_{max}, t_{max}$ );
2:    $S \leftarrow$  Initial solution;
3:   repeat
4:      $k \leftarrow 1$ ;
5:     while  $k \leq k_{max}$  do
6:        $S' \leftarrow$  Shake( $S, k$ );
7:        $S'' \leftarrow$  Local search( $S'$ );
8:        $k \leftarrow k + 1$ ;
9:       if  $S''$  is better than  $S$  then
10:         $S \leftarrow S''$ ;
11:         $k \leftarrow 1$ ;
12:      end while
13:    end repeat
14:    $t \leftarrow$  CpuTime();
15:   until  $t > t_{max}$ ;
16: return  $S$ ;

```

---

Како UMApHMSP подразумева шему вишеструке алокације која допушта сваком не-хаб чвору да шаље и прима проток преко више од једног хаба, довољно је да **репрезентација решења** садржи само информацију о изабраним хабовима. Односно, решење проблема  $S$  је представљено као скуп од  $p$  изабраних хабова. Након одређивања  $p$  локација за хабове, најкраћи путеви између О-Д парова могу се одредити коришћењем модификације добро познатог Floyd-Warshall алгорита за налажење најкраћих путева у највише  $O(n^2p)$  операција (видети [3, 53]).

**Иницијално решење** за UMApHMSP се конструише применом похлепне процедуре, тако што се чворови са  $p$  најмањих вредности функције  $g(h)$  (видети (2.35)) бирају за иницијални скуп који садржи тачно  $p$  хабова.

Предложена BVNS метода за UMApHMSP претражује простор решења користећи једну **структуру околина**, *Interchange\_hub*( $S$ ), која је детаљно описана у Одељку 2.3. Ова структура околина се истражује у оквиру процедуре размрдавања *Shake* и локалне претраге. Процедура размрдавања се користи у циљу избегавања конвергенције ка локалном оптимуму. BVNS алгоритам за UMApHMSP користи исту процедуру размрдавања као и GVNS за USApHMSP. Као што је приказано Алгоритмом 2 у Одељку 2.3, у свакој од  $k$  итерација процедуре размрдавања, решење  $S$  се замењује неким случајно изабраним решењем

из околине  $Interchange\_hub(S)$ . Решење  $S'$  које се добија у  $k$ -тој итерацији је излаз из процедуре размрдавања. Даље се локалном претрагом покушава побољшати решење  $S'$ , при чему се као најбоље пронађено решење из локалне претраге добија ново решење  $S''$ . Након тога се ред околине  $k$  повећава за 1. Уколико је  $f(S'') > f(S)$ , где је са  $f(S)$  означена вредност функције циља текућег решења  $S$ , побољшано решење  $S''$  постаје текуће, а ред околине  $k$  се поставља поново на 1. Алгоритам се завршава када се достигне постављено максимално време извршавања  $t_{max}$ .

### 3.4 Експериментални резултати

BVNS метахеуристика за решавање UMAPHMCP имплементирана је у програмском језику C++. Математичке формулације изложене у Одељку 3.2 су упоређене коришћењем егзактног решавача CPLEX 12.6, а све формулације су такође кодирани у програмском језику C++. Сва тестирања су извршена под Windows 10 оперативним системом са процесором Intel Core I7 са 2.8 GHz CPU и RAM меморијом од 8 GB.

Експериментални резултати добијени су коришћењем CAB, AP и URAND скупова података, описаних у Табели 2.1 из Одељка 2.4. Како су тест инстанце великих димензија недостижне за егзактни решавач, нове формулације за UMAPHMCP и оба концепта покривања су упоређене са одговарајућим формулацијама предложеним у [121] само на CAB и AP инстанцама до 50 чворова. Развијена BVNS метахеуристика за решавање UMAPHMCP у оба случаја покривања је тестирана на свим инстанцама описаним у Табели 2.1, малих, средњих и великих димензија. Треба нагласити да AP инстанце са више од 25 чворова и URAND инстанце нису коришћене у ранијој литератури за тестирање UMAPHMCP.

При тестирању BVNS методе за решавање UMAPHMCP, коришћене су исте вредности параметара  $\chi$ ,  $\alpha$ ,  $\delta$ ,  $q_{ij}^{km}$ ,  $r_{ij}^{km}$  и  $\beta$  као при тестирању GVNS методе за решавање USAPHMCP, изложене у Одељку 2.4.

BVNS параметар  $k_{max}$  је постављен на  $p$  при решавању свих тест инстанци, док је вредност параметра  $t_{max}$  прилагођена димензијама решаваних инстанци, као што је приказано у Табели 3.1.

Табела 3.1: Вредности параметра  $t_{max}$  за тестирање BVNS алгоритма

Скуп података	Број чворова	$t_{max}$ (у секундама)
САВ	$n = 25$	1
АР	$10 \leq n \leq 50$	1
АР	$n = 100$	120
АР	$n = 200$	180
URAND	$100 \leq n \leq 400$	600
URAND	$n = 1000$	1800

Квалитет резултата добијених BVNS метахеурстиком је оцењен упоређивањем са оптималним и субоптималним решењима добијеним егзактним решавачем CPLEX који користи формулације проблема представљене у Одељку 3.2. Време извршавања CPLEX решавача је ограничено на 1h (3600s), за сваку тестирану инстанцу.

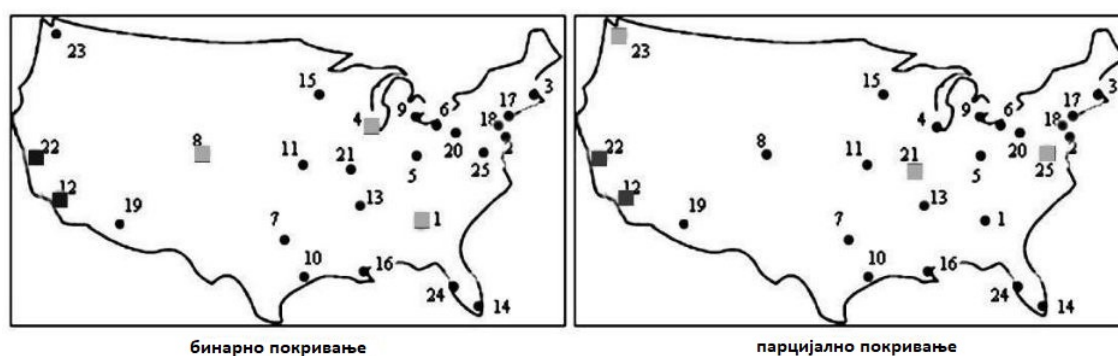
У Табелама 3.2 и 3.3 су приказани резултати добијени коришћењем CPLEX решавача за математичке формулације UMAPHMCP наведене у Одељку 3.2. Ове табеле такође садрже резултате предложене BVNS метахеуристике за решавање UMAPHMCP и оба концепта покривања, добијене на САВ и АР инстанцама мањих димензија. Како CPLEX решавач на инстанцама већих димензија не даје чак ни допустива решења, у Табелама 3.4 и 3.5 су приказани само BVNS резултати за АР инстанце већих и URAND инстанце великих димензија. Заглавља колона Табела 3.2–3.5 су дефинисана на исти начин као у Одељку 2.4.

Из резултата представљених у Табели 3.2, може се закључити да Опште формулације UMAPHMCP предложене у овој дисертацији омогућавају CPLEX решавачу достизање бољих решења у поређењу са формулацијама које су увели Рекер и Кага у [121]. Користећи Општу формулацију, CPLEX достиже оптимална решења на свим САВ инстанцама за оба концепта покривања. Са друге стране, у случају парцијалног покривања, користећи формулације презентоване у [121], CPLEX не успева да пронађе оптимална решења за четири САВ инстанце. Такође, време извршавања CPLEX решавача је знатно краће при коришћењу Опште формулације у односу на формулацију из [121]. У просеку, у случају бинарног покривања, користећи формулацију из [121], CPLEX достиже оптимална решења за 163.72 секунде, а користећи Општу формулацију, за 8.82 секунде. У случају парцијалног покривања, користећи формулацију из [121], CPLEX налази решења (не сва оптимална) за 2727.37 секунди, а користећи Општу формулацију, достиже оптимална решења за 66.05 секунди (у просеку).

ГЛАВА 3. ПРОБЛЕМ  $p$ -ХАБ МАКСИМАЛНОГ ПОКРИВАЊА  
НЕОГРАНИЧЕНИХ КАПАЦИТЕТА СА ВИШЕСТРУКИМ АЛОКАЦИЈАМА

Када се анализирају резултати предложене BVNS хеуристике за САВ скуп података, може се приметити да BVNS у кратком времену извршавања достиже сва позната оптимална решења, у оба случаја покривања. У просеку, време извршавања BVNS хеуристике је скоро тренутно (мање од 0.005 секунди) за бинарно и 0.01 секунда за парцијално покривање.

Слично као у Одељку 2.4, може се закључити да увођење парцијалног покривања води ка повећавању вредности функције циља оптималног решења UМАрНМСР, у односу на бинарно покривање. То се може илустровати на примеру САВ инстанце са  $n = 25$  чворова,  $p = 5$  хабова и параметром  $\alpha = 0.8$  (видети Сliku 3.1). У случају бинарног покривања, оптималне локације хабова (1, 4, 8, 12, 22) су обележене квадратићима на левој страни Сlike 3.1, док су у случају парцијалног покривања, оптималне локације хабова (12, 21, 22, 23, 25) обележене квадратићима на десној страни Сlike 3.1. Сиви квадратићи означавају хаб локације које се разликују у бинарном и парцијалном случају, док су исте хаб локације означене црним квадратићима. Упоређујући проценте максималног покривања у оптималним решењима представљеним на Сlici 3.1, може се закључити да је вредност оптималног решења у случају бинарног покривања (89.05%) мања него у случају парцијалног покривања (91.51%). Такође се може приметити да су, због различитих алокацијских шема, вредности функције циља оптималних решења UМАрНМСР у оба случаја покривања веће од одговарајућих вредности функције циља оптималних решења USAрНМСР за исти пример (анализиран у Одељку 2.4).



Слика 3.1: Оптималне локације хабова UМАрНМСР за САВ инстанцу са  $p = 5$ ,  $\alpha = 0.8$ , бинарним покривањем (лево) и парцијалним покривањем (десно)

Резултати добијени на AP инстанцама малих димензија, до 50 чворова, за

оба случаја покривања, приказани су у Табели 3.3. Поново се може закључити да у смислу квалитета решења и времена извршавања CPLEX решавача, Општа формулација даје боље резултате од формулације презентоване у [121]. У случају бинарног покривања, CPLEX користећи Општу формулацију достиже сва оптимална решења. Користећи формулацију из [121], CPLEX у случају 8 AP инстанци добија само допустива решења, за која не успева да докаже оптималност у задатом максималном времену извршавања. Такође, просечно време потребно CPLEX решавачу за решавање проблема AP инстанци мањих димензија је више од 8 пута краће користећи Општу формулацију, у односу на формулацију из [121]. Слична је ситуација и у случају парцијалног покривања. Користећи Општу формулацију, CPLEX није пронашао оптимална решења само за 4 AP инстанце са  $n = 50$  чворова, док са формулацијом из [121], CPLEX не успева да пронађе оптимална решења за 11 AP инстанци мањих димензија (иако су пронађена допустива решења). Ако се посматрају резултати предложене BVNS хеуристике за UMAPHMCP на AP инстанцама мањих димензија, може се закључити да хеуристика достиже сва позната оптимална решења претходно добијена CPLEX решавачем, за оба концепта покривања. На осталим тестираним инстанцама, BVNS успоставља нова најбоља решења. Просечно време извршавања BVNS хеуристике у оба случаја покривања је 0.08 секунди, што је много краће у поређењу са временом потребним CPLEX решавачу.

Табела 3.2: Експериментални резултати за УМАрНМСР на САВ инстанцама

Бинарно покривање				Парцијално покривање			
$n, p$	$\alpha$	Peker & Kara		Општа форм.		BVNS	
		dev(%)	time(s)	dev(%)	time(s)	dev(%)	time(s)
25.2	0.2	0.00	334.39	0.00	8.85	0.00	0.00
25.2	0.4	0.00	255.59	0.00	3.44	0.00	0.00
25.2	0.6	0.00	142.27	0.00	2.48	0.00	0.00
25.2	0.8	0.00	88.27	0.00	2.62	0.00	0.00
25.3	0.2	0.00	303.31	0.00	20.68	0.00	0.00
25.3	0.4	0.00	172.90	0.00	16.35	0.00	0.00
25.3	0.6	0.00	147.65	0.00	7.08	0.00	0.00
25.3	0.8	0.00	72.90	0.00	2.92	0.00	0.00
25.4	0.2	0.00	321.43	0.00	17.94	0.00	0.01
25.4	0.4	0.00	195.22	0.00	19.77	0.00	0.01
25.4	0.6	0.00	94.32	0.00	4.51	0.00	0.01
25.4	0.8	0.00	87.00	0.00	2.63	0.00	0.01
25.5	0.2	0.00	183.67	0.00	18.54	0.00	0.01
25.5	0.4	0.00	99.00	0.00	7.20	0.00	0.01
25.5	0.6	0.00	80.81	0.00	4.63	0.00	0.01
25.5	0.8	0.00	40.77	0.00	1.45	0.00	0.01
<b>Просек:</b>		<b>0.00</b>	<b>163.72</b>	<b>0.00</b>	<b>8.82</b>	<b>0.00</b>	<b>0.00</b>

$n, p$	$\alpha$	Peker & Kara		Општа форм.		BVNS	
		dev(%)	time(s)	dev(%)	time(s)	dev(%)	time(s)
25.2	0.2	0.00	1822.50	0.00	110.59	0.00	0.01
25.2	0.4	0.00	2068.67	0.00	55.72	0.00	0.00
25.2	0.6	0.00	1571.23	0.00	74.29	0.00	0.00
25.2	0.8	0.00	2644.80	0.00	25.76	0.00	0.01
25.3	0.2	0.00	3600.07	0.00	125.43	0.00	0.00
25.3	0.4	0.00	3516.98	0.00	78.13	0.00	0.00
25.3	0.6	0.00	2324.90	0.00	65.21	0.00	0.00
25.3	0.8	0.00	2251.87	0.00	43.58	0.00	0.01
25.4	0.2	0.02	3600.16	0.00	134.05	0.00	0.01
25.4	0.4	0.00	3367.36	0.00	63.32	0.00	0.01
25.4	0.6	0.00	3583.47	0.00	50.37	0.00	0.01
25.4	0.8	0.00	1941.94	0.00	26.21	0.00	0.02
25.5	0.2	0.38	3600.10	0.00	91.49	0.00	0.01
25.5	0.4	0.00	3600.39	0.00	56.06	0.00	0.01
25.5	0.6	0.00	2467.35	0.00	33.95	0.00	0.02
25.5	0.8	0.00	1676.17	0.00	22.62	0.00	0.03
<b>Просек:</b>		<b>0.00</b>	<b>2727.37</b>	<b>0.00</b>	<b>66.05</b>	<b>0.00</b>	<b>0.01</b>

Табела 3.3: Експериментални резултати за УМАрНМСР на АР инстанцама малих димензија

Бинарно покривање				Парцијално покривање			
$n, p$	Optimal value(%)	Peker & Kara		Општа форм.		BVNS	
		dev(%)	time(s)	dev(%)	time(s)	dev(%)	time(s)
25.2	95.88	0.00	150.2	0.00	6.07	0.00	0.00
25.3	95.71	0.00	115.78	0.00	2.52	0.00	0.00
25.4	97.16	0.00	142.86	0.00	5.13	0.01	0.00
25.5	98.35	0.00	100.83	0.00	4.83	0.01	0.01
25.10	99.70	0.00	10.98	0.00	0.60	0.01	0.02
40.2	97.83	0.01	3600.19	0.00	149.12	0.00	0.00
40.3	97.96	0.01	3600.26	0.00	152.29	0.00	0.01
40.4	97.38	15.63	3600.31	0.00	152.29	0.00	0.01
40.5	97.80	0.05	3600.25	0.00	97.20	0.00	0.03
40.10	99.31	0.00	226.52	0.00	14.24	0.00	0.20
50.2	97.89	96.16	3600.83	0.00	692.30	0.00	0.01
50.3	98.45	96.88	3600.44	0.00	819.42	0.00	0.01
50.4	98.23	82.26	3600.37	0.00	861.33	0.00	0.03
50.5	98.22	97.61	3600.64	0.00	666.69	0.00	0.20
50.10	99.51	0.00	1186.92	0.00	106.84	0.00	0.72
<b>Просек:</b>		<b>25.91</b>	<b>2049.16</b>	<b>0.00</b>	<b>248.72</b>	<b>0.00</b>	<b>0.08</b>

$n, p$	Optimal/Best value(%)	Peker & Kara		Општа форм.		BVNS	
		dev(%)	time(s)	dev(%)	time(s)	dev(%)	time(s)
25.2	97.56	0.00	2349.16	0.00	90.71	0.00	0.00
25.3	97.71	0.00	3600.11	0.00	51.91	0.00	0.00
25.4	98.70	0.00	2965.27	0.00	50.45	0.00	0.00
25.5	99.23	0.00	1986.48	0.00	30.44	0.00	0.01
25.10	99.82	0.00	1578.69	0.00	3.77	0.00	0.02
40.2	99.00	1.30	3600.74	0.00	2583.47	0.00	0.00
40.3	99.13	95.43	3600.29	0.00	3034.43	0.00	0.01
40.4	98.49	96.89	3600.29	0.00	3268.78	0.00	0.20
40.5	98.90	97.35	3600.24	0.00	1839.55	0.00	0.06
40.10	99.71	9.50	3600.27	0.00	75.92	0.00	0.32
50.2	99.06	94.74	3600.94	0.02	3600.75	0.00	0.02
50.3	99.31	95.86	3600.87	13.84	3600.66	0.00	0.01
50.4	99.02	96.69	3600.59	16.61	3601.07	0.00	0.07
50.5	99.06	97.21	3600.45	2.63	3600.45	0.00	0.25
50.10	99.77	14.84	3600.43	0.00	1174.92	0.00	0.19
<b>Просек:</b>		<b>46.65</b>	<b>3232.32</b>	<b>2.21</b>	<b>1773.82</b>	<b>0.00</b>	<b>0.08</b>

ГЛАВА 3. ПРОБЛЕМ  $p$ -ХАБ МАКСИМАЛНОГ ПОКРИВАЊА  
НЕОГРАНИЧЕНИХ КАПАЦИТЕТА СА ВИШЕСТРУКИМ АЛОКАЦИЈАМА

За AP инстанце већих димензија, URAND инстанце великих димензија и оба случаја покривања, CPLEX решавач услед ограничења меморије не успева да пронађе чак ни допустива решења ни за једну од наведених формулација. Из тог разлога, у Табелама 3.4 и 3.5 су представљени редом само резултати добијени предложеном BVNS методом за AP инстанце са 100 и 200 чворова и URAND инстанце до 1000 чворова. Из резултата приказаних у Табели 3.4 може се видети да је BVNS методи, у просеку, потребно 21.68 секунди за достизање најбољих решења за AP инстанце већих димензија у случају бинарног и 27.16 секунди за случај парцијалног покривања. Просечна вредност функције циља најбољих BVNS решења на овим инстанцама је 99.07% за концепт бинарног и 99.58% за концепт парцијалног покривања. Из приказаних резултата у Табели 3.5 за URAND инстанце великих димензија, може се закључити да се BVNS метода показала веома ефикасном у случају оба концепта покривања. У просеку, BVNS обезбеђује најбоља решења за 319.99 секунди у случају бинарног и за 332.97 секунди за случај парцијалног покривања. Узимајући у обзир димензије разматраних проблема, може се закључити да су ово релативно кратка времена извршавања. Просечна вредност функције циља најбољих BVNS решења на URAND инстанцама са 100, 200, 300 и 1000 чворова је 91.71% у случају бинарног и 96.02% за случај парцијалног покривања.

Табела 3.4: Резултати предложене BVNS методе за UMAPHMCР на AP инстанцама великих димензија

n.p	Бинарно покривање		Парцијално покривање	
	Best value(%)	time(s)	Best value(%)	time(s)
100.2	98.32	0.11	99.24	0.11
100.3	98.47	0.80	99.31	0.48
100.4	98.20	0.55	99.06	0.80
100.5	98.65	0.14	99.39	0.18
100.10	99.24	4.39	99.69	10.14
100.15	99.51	30.66	99.79	48.77
100.20	99.63	24.72	99.84	110.39
200.2	98.81	4.32	99.52	0.23
200.3	99.25	0.49	99.67	0.67
200.4	98.94	4.20	99.54	2.20
200.5	98.93	9.42	99.53	2.33
200.10	99.49	40.19	99.80	23.24
200.15	99.72	107.79	99.89	15.56
200.20	99.80	75.70	99.91	165.16
<b>Просек:</b>	99.07	21.68	99.58	27.16



ГЛАВА 3. ПРОБЛЕМ  $p$ -ХАБ МАКСИМАЛНОГ ПОКРИВАЊА  
НЕОГРАНИЧЕНИХ КАПАЦИТЕТА СА ВИШЕСТРУКИМ АЛОКАЦИЈАМА

Табела 3.5: Резултати предложене BVNS методе за UМАрНМСП на URAND инстанцама великих димензија

n.p	Бинарно покривање		Парцијално покривање	
	Best value(%)	time(s)	Best value(%)	time(s)
100.02	89.48	0.15	95.05	0.61
100.03	89.33	0.17	94.76	0.22
100.04	87.33	0.28	93.36	0.28
100.05	88.97	0.24	94.20	0.28
100.10	89.75	1.88	94.80	4.25
100.15	91.27	11.67	95.54	16.49
100.20	92.72	72.72	96.40	172.08
200.02	93.14	0.53	97.08	1.32
200.03	90.70	0.69	95.46	0.58
200.04	89.06	11.91	94.60	11.78
200.05	88.03	1.75	93.76	42.62
200.10	90.48	396.07	95.42	33.65
200.15	90.30	528.19	95.18	177.32
200.20	91.76	164.47	95.96	304.68
300.02	93.89	3.95	97.24	4.11
300.03	92.68	5.87	96.60	17.05
300.04	90.30	11.52	95.17	6.12
300.05	89.91	33.03	94.97	14.36
300.10	91.50	186.94	95.92	525.26
300.15	92.13	516.34	96.20	616.25
300.20	92.22	336.42	96.20	542.46
400.02	94.11	5.79	97.45	2.34
400.03	92.93	26.37	96.74	45.13
400.04	90.75	14.20	95.49	125.22
400.05	90.36	34.94	95.15	163.97
400.10	92.08	157.38	96.32	337.27
400.15	92.72	568.23	96.57	178.18
400.20	92.80	510.43	96.58	712.75
1000.02	95.91	50.45	98.40	431.39
1000.03	94.69	155.07	97.78	1087.62
1000.04	93.25	191.20	97.11	238.35
1000.05	92.35	1808.24	96.52	421.93
1000.10	93.85	1792.43	97.29	1806.70
1000.15	94.07	1798.52	97.41	1805.23
1000.20	94.97	1801.77	97.86	1806.13
<b>Просек:</b>	91.71	319.99	96.02	332.97

## Глава 4

# Проблем $p$ -хаб максималног покривања неограничених капацитета са $r$ -алокацијском шемом

### 4.1 Опис проблема и преглед релевантне литературе

Проблем  $p$ -хаб максималног покривања неограничених капацитета са  $r$ -алокацијском шемом ( $UrArHMCP$ ), при чему је  $1 \leq r \leq p$ , представља генерализацију проблема  $p$ -хаб максималног покривања неограничених капацитета са једноструким и вишеструким алокацијама. Прецизније, за  $r = 1$ ,  $UrArHMCP$  се своди на  $USArHMCP$ , а за  $r = p$  на  $UMArHMCP$ . При разматрању  $UrArHMCP$  важе све претпоставке које су наведене у Одељку 2.1 за  $USArHMCP$ . Једина разлика је у алокацијској шеми, јер је код  $UrArHMCP$  сваком не-хаб чвору  $i \in N$  дозвољено да комуницира са највише  $r$  хабова. Мотивација за коришћење  $r$ -алокацијске шеме произилази из чињенице да је претпоставка једноструке алокације превише рестриктивна, док концепт вишеструке алокације доводи до компликованих мрежа протока и високих фиксних трошкова. Концепт  $r$ -алокације је 2011. године увела Yaman у раду [149] за проблем  $p$ -хаб медијане. Овај тип алокације коришћен је у литератури за различите хаб локацијске проблеме (видети [119, 120, 143]). Проблем  $p$ -хаб максималног покривања

са  $r$ -алокацијском шемом је по први пут разматран у овој дисертацији. Циљ UrApHMCP је одредити оптималне локације тачно  $p$  хабова, тако да се максимизује укупан проток по свим покривеним паровима снабдевач-корисник, уз претпоставку бинарног или парцијалног покривања. UrApHMCP припада класи NP-тешких проблема, као генерализација USApHMCP и UMApHMCP за које је доказано да су NP-тешки (видети [7, 25]).

## 4.2 Математичке формулације проблема

У овом одељку су предложене три математичке формулације UrApHMCP (Четвороиндексна, Троиндексна и Општа), које се могу применити на оба концепта покривања - бинарно и парцијално.

Четвороиндексна математичка формулација UrApHMCP користи нотацију наведену у Одељку 2.1 и четвороиндексне бинарне променљиве  $y_{ijkm}$  које узимају вредност 1 уколико је чвор снабдевач  $i \in N$  придружен хабу  $k \in H$  и чвор корисник  $j \in N$  хабу  $m \in H$ , а 0 иначе.

UrApHMCP са концептом бинарног покривања може бити формулисан као проблем целобројног линеарног програмирања (енгл. *Integer Linear Programming* - *ILP*) са четвороиндексним променљивим на следећи начин:

$$\max \sum_{i \in N} \sum_{j \in N} \sum_{k \in H} \sum_{m \in H} q_{ij}^{km} T_{ij} y_{ijkm} \quad (4.1)$$

при условима:

$$x_{ik} \leq x_{kk} \quad \forall i \in N, k \in H \quad (4.2)$$

$$\sum_{k \in H} x_{ik} \leq r \quad \forall i \in N \quad (4.3)$$

$$\sum_{k \in H} x_{kk} = p \quad (4.4)$$

$$\sum_{k \in H} \sum_{m \in H} y_{ijkm} = 1 \quad \forall i, j \in N \quad (4.5)$$

$$y_{ijkm} \leq x_{ik} \quad \forall i, j \in N, k, m \in H \quad (4.6)$$

$$y_{ijkm} \leq x_{jm} \quad \forall i, j \in N, k, m \in H \quad (4.7)$$

$$x_{ik} \in \{0, 1\} \quad \forall i \in N, k \in H \quad (4.8)$$

$$y_{ijkm} \in \{0, 1\} \quad \forall i, j \in N, k, m \in H. \quad (4.9)$$

Функција циља (4.1) дефинисана је као укупан покривен проток између свих О–Д парова  $i - j$ , чију вредност треба максимизовати. Услови (4.2) и (4.3) обезбеђују да сваки чвор комуницира са највише  $r$  хабова, док услов (4.4) указује да је успостављено тачно  $p$  хабова. За сваки О–Д пар, проток од почетног до крајњег чвора се спроводи преко највише два успостављена хаба, што је одређено условима (4.5). Ако је проток усмерен од почетног чвора  $i$  до крајњег чвора  $j$  редом преко успостављених хабова  $k$  и  $m$ , тада чвор  $i$  мора бити придружен хабу  $k$ , а чвор  $j$  хабу  $m$ , што је обезбеђено условима (4.6) и (4.7). Типови променљивих  $x_{ik}$  и  $y_{ijkm}$  су дефинисани редом условима (4.8) и (4.9). Овај модел укључује  $O(n^4)$  променљивих и  $O(n^4)$  ограничења.

Предложена Четвороиндексна формулација (4.1)–(4.9) се лако може прилагодити случају са парцијалним покривањем, и то једноставном заменом параметра  $q_{ij}^{km}$  параметром  $r_{ij}^{km}$ . Може се приметити да се у случајевима  $r = 1$  и  $r = p$ , формулација (4.1)–(4.9) за УрАрНМСР лако трансформише у одговарајуће формулације предложене у раду [80] редом за УСАрНМСР и УМАрНМСР.

Користећи нотацију наведену у Одељку 3.2, УрАрНМСР се у случају бинарног покривања може формулисати и као проблем мешовитог целобројног линеарног програмирања, који користи највише троиндексне променљиве, на следећи начин:

$$\max \sum_{i \in N} \sum_{j \in N} T_{ij} Z_{ij} \quad (4.10)$$

при условима: (4.2)–(4.4), (4.8) и

$$Z_{ij} \leq \sum_{k \in H} q_{ij}^{km} x_{ijk} + \lambda_{ij}(1 - y_{ijm}) \quad \forall i, j \in N, m \in H \quad (4.11)$$

$$Z_{ij} \leq \sum_{m \in H} q_{ij}^{km} y_{ijm} + \lambda_{ij}(1 - x_{ijk}) \quad \forall i, j \in N, k \in H \quad (4.12)$$

$$\sum_{k \in H} x_{ijk} = 1 \quad \forall i, j \in N \quad (4.13)$$

$$\sum_{m \in H} y_{ijm} = 1 \quad \forall i, j \in N \quad (4.14)$$

$$x_{ijk} \leq x_{ik} \quad \forall i, j \in N, k \in H \quad (4.15)$$

$$y_{ijm} \leq x_{jm} \quad \forall i, j \in N, m \in H \quad (4.16)$$

$$x_{ijk} \in \{0, 1\} \quad \forall i, j \in N, k \in H \quad (4.17)$$

$$y_{ijm} \in \{0, 1\} \quad \forall i, j \in N, m \in H \quad (4.18)$$

$$Z_{ij} \geq 0 \quad \forall i, j \in N. \quad (4.19)$$

Значење функције циља (4.10) и услова (4.2)–(4.4), (4.8) је исто као у претходно изложеној Четвороиндексној формулацији за UrArHMCP. Услови (4.11) и (4.12) дефинишу део протока за О–Д пар  $(i, j)$  који је покривен. Услови (4.13) и (4.14) редом обезбеђују да на сваком путу од почетног чвора  $i$  до крајњег чвора  $j$  само један хаб може бити одређен за колекцију и само један хаб за дистрибуцију протока. Пут од чвора  $i$  до чвора  $j$  може садржати чворове  $k$  и  $m$ , само уколико су  $k$  и  $m$  успостављени хабови, што је дефинисано условима (4.15) и (4.16). Коначно, услови (4.17) и (4.18) означавају да су променљиве  $x_{ijk}$  и  $y_{ijm}$  бинарне, док услови (4.19) означавају да су реалне променљиве  $Z_{ij}$  ненегативне. Овај Троиндексни модел има  $O(n^3)$  променљивих и  $O(n^3)$  ограничења.

Предложени Троиндексни модел (4.10), (4.2)–(4.4), (4.8), (4.11)–(4.19) се може применити и на случај парцијалног покривања, заменом параметара  $q_{ij}^{km}$  параметром  $r_{ij}^{km}$ . Може се приметити да Троиндексни модели за UrArHMCP представљају генерализацију Троиндексних модела за USArHMCP и UMArHMCP, предложених у [121].

Општи модели за USArHMCP и оба концепта покривања, изложени у Одељку 2.2, се малим модификацијама трансформишу у одговарајуће моделе UrArHMCP. Користећи претходно уведеног нотацију, UrArHMCP са концептом бинарног покривања се може формулисати као проблем мешовитог целобројног линеарног програмирања, на следећи начин:

$$\max \sum_{i \in N} \sum_{j \in N} T_{ij} Z_{ij} \quad (4.20)$$

при условима:

$$x_{ik} \leq x_{kk} \quad \forall i \in N, k \in H \quad (4.21)$$

$$\sum_{k \in H} x_{ik} \leq r \quad \forall i \in N \quad (4.22)$$

$$\sum_{k \in H} x_{kk} = p \quad (4.23)$$

$$w_{ijk} \leq \sum_{m \in G_{ij}^k} x_{jm} \quad \forall i, j \in N, k \in H \quad (4.24)$$

$$w_{ijk} \leq x_{ik} \quad \forall i, j \in N, k \in H \quad (4.25)$$

$$\sum_{k \in H} w_{ijk} \leq 1 \quad \forall i, j \in N \quad (4.26)$$

$$Z_{ij} \leq \sum_{k \in H} w_{ijk} \quad \forall i, j \in N \quad (4.27)$$

$$w_{ijk} \in \{0, 1\} \quad \forall i, j \in N, k \in H \quad (4.28)$$

$$x_{ik} \in \{0, 1\} \quad \forall i \in N, k \in H \quad (4.29)$$

$$Z_{ij} \geq 0 \quad \forall i, j \in N. \quad (4.30)$$

Општа формулација (4.20)–(4.30) се лако може прилагодити случају UrApH-MCP са парцијалним покривањем. У том случају, користећи нотацију и поступак описан у Одељку 2.2, услови (4.24)–(4.27) који важе за случај бинарног покривања се трансформишу у следеће услове:

$$w_{ijkl} \leq \sum_{m \in G_{ij}^{kl}} x_{jm} \quad \forall i, j \in N, k \in H, l \in L_{ij}^k \quad (4.31)$$

$$w_{ijkl} \leq x_{ik} \quad \forall i, j \in N, k \in H, l \in L_{ij}^k \quad (4.32)$$

$$\sum_{k \in H} \sum_{l \in L_{ij}^k} w_{ijkl} \leq 1 \quad \forall i, j \in N \quad (4.33)$$

$$Z_{ij} \leq \sum_{k \in H} \sum_{l \in L_{ij}^k} v_{ijkl} w_{ijkl} \quad \forall i, j \in N. \quad (4.34)$$

UrApHMCP до сада није разматран у литератури, тако да не постоје предложене методе за његово решавање. У овој дисертацији су дизајниране и имплементирани три метахеуристике за решавање UrApHMCP: општа метода променљивих околина (GVNS-R), похлепна стохастичко-адаптивна процедура претраге са методом променљивог спуста (GRASP-VND) и генетски алгоритам (GA).

### 4.3 Општа метода променљивих околина за решавање проблема

GVNS-R метода за решавање UrApHMCP представља модификацију GVNS методе за решавање USApHMCP, описане у Одељку 2.3. Обе методе претражују простор решења користећи структуре околина дефинисане на исти начин, а разликују се у конструкцији иницијалног решења и процедурама побољшања решења, којима се истражују околине. Основни кораци GVNS-R методе, која се може применити на оба концепта покривања, приказани су Алгоритмом 5.

---

#### Алгоритам 5 Општа метода променљивих околина за UrApHMCP

---

```

1: function GVNS( $k_{max}$ ,  $t_{max}$ );
2:    $S \leftarrow$  Initial solution;
3:   repeat
4:      $k \leftarrow 1$ ;
5:     while  $k \leq k_{max}$  do
6:        $S' \leftarrow$  Shake( $S$ ,  $k$ );
7:        $S'' \leftarrow$  SeqVND( $S'$ );
8:        $k \leftarrow k + 1$ ;
9:       if  $S''$  is better than  $S$  then
10:         $S \leftarrow S''$ ;
11:         $k \leftarrow 1$ ;
12:      end while
13:    end repeat
14:    $t \leftarrow$  CpuTime();
15:   until  $t > t_{max}$ ;
16:   return  $S$ ;

```

---

---

**Алгоритам 6 SeqVND**

---

```

1: function SeqVND( $S$ );
2:    $k \leftarrow 1$ ;
3:   while  $k \leq 2$  do
4:     if  $k = 1$  then
5:       Select  $S'$  as the best solution in Interchange_hub( $S$ );
6:     else
7:       Select  $S'$  as the best solution in ReAllocate_node( $S$ );
8:     end
9:      $k \leftarrow k + 1$ ;
10:    if  $S'$  is better than  $S$  then
11:       $S \leftarrow S'$ ;
12:       $k \leftarrow 1$ ;
13:    end
14:  end
15: return  $S$ ;

```

---

Предложена GVNS-R метода има два параметра:  $t_{max}$  који представља максимално дозвољено време извршавања и  $k_{max}$  који представља максималан број итерација унутар процедуре размрдавања. GVNS-R метахеуристика започиње рад похлепном процедуром генеришући почетно решење  $S$ , које се након постављања реда околине  $k$  на 1, у оквиру петље са променама околина, прослеђује процедури размрдавања (*Shake*), а резултујуће решење се даље прослеђује процедури побољшања решења SeqVND. Ове две процедуре се, заједно са фазом помераја (у којој се текуће решење замењује бољим), извршавају наизменично док се не достигне постављено максимално време извршавања алгоритма.

Како је потребно да **репрезентација решења** садржи информације о успостављеним хабовима и алокацијама, природно је да решење буде представљено уређеним паром  $S = (\bar{H}, A)$ , где је  $\bar{H} = \{h_1, h_2, \dots, h_p\}$  скуп  $p$  одабраних хабова, док је  $A = (A_k)$   $k = 1, \dots, p$  матрица алокације димензије  $p \times n$ . Скуп чворова придружених хабу  $h_k$  је означен са  $A_k$ ,  $h_k \in \bar{H}$ . Када су познате локације хабова и алокације не-хаб чворова, лако је пронаћи најкраће путеве за сваки пар  $i - j$  и израчунати вредност функције циља. Израчунавање свих О-Д најкраћих путева захтева  $O(n^2r^2)$  операција у најгорем случају. У случају када је  $r = p$ , коришћењем модификације добро познатог Floyd-Warshall алгоритма за налажење најкраћих путева, израчунавање се реализује кроз највише  $O(n^2r)$  операција.

За конструкцију **иницијалног решења** је примењена процедура *Initial solution* која користи функције  $g(h)$  и  $alloc(i, h)$ ,  $i \in N$ ,  $h \in H$  дефинисане



на следећи начин. Функција

$$g(h) = \max\{c_{ih} : i \in N, i \neq h\} \quad (4.35)$$

представља максималну удаљеност чвора  $h$  од свих осталих чворова, док функција

$$alloc(i, h) = \chi c_{ih} + \max\{\alpha c_{hh'} : h' \in H\} \quad (4.36)$$

рачуна најдужи пут од од чвора  $i$  до неког хаба, преко хаба  $h$ . Чворови са  $p$  најмањих вредности функције  $g(h)$  се бирају за иницијални скуп  $\bar{H}$  који садржи тачно  $p$  хабова. Након тога, сваком чвору  $i$  се придружује  $r$  хабова са најмањим вредностима функције  $alloc(i, h)$ . Оваква стратегија избора локација хабова и алокација не-хаб чворова повећава шансу успостављања што краћих путева између О–Д парова.

**Структуре околина** које се истражују у оквиру процедура размрдавања и побољшања решења предложене GVNS-R методе за решавање UrArHMCP дефинисане су на исти начин као структуре околина које се истражују у оквиру GVNS методе за решавање USArHMCP, описане у Одељку 2.3. Основни кораци примењене **процедуре размрдавања** *Shake* су такође описани у Одељку 2.3.

Као почетно решење **процедуре побољшања** користи се решење добијено као излаз из процедуре размрдавања. Предложена GVNS-R метода користи процедуру побољшања решења засновану на секвенцијалној методи променљивог спуста *SeqVND*. Ова метода истражује околине *Interchange\_hub(S)* и *ReAllocate\_node(S)*, једну за другом. Када се пронађе локални оптимум у односу на околину *Interchange\_hub(S)*, претрага се наставља у околини *ReAllocate\_node(S)*. Уколико нема побољшања у овој околини, односно уколико је текуће решење локални оптимум у односу на обе околине, процедура *SeqVND* се завршава. У супротном, *SeqVND* наставља претрагу у околини *Interchange\_hub(S)*. Кораци примењене *SeqVND* процедуре представљени су Алгоритмом 6.

## 4.4 Похлепна стохастичко-адаптивна процедура претраге са методом променљивог спуста за решавање проблема

Похлепна стохастичко-адаптивна процедура претраге за решавање проблема (GRASP) је метахеуристика заснована на комбинацији конструктивне хеури-

стике и локалног претраживања, коју су увели Feo и Bard [55], а касније популаризовали Feo и Resende [56]. GRASP је итеративна метода у којој се свака итерација састоји из две фазе: конструкције решења и локалне претраге. У првој фази се генерише допустиво решење за разматрани проблем применом похлепно стохастичке процедуре за конструкцију решења (енгл. *Greedy Randomized Construction procedure - GRC*), док се у другој фази примењује локална претрага над решењем конструисаним у GRC фази, у циљу налажења локалног оптимума у односу на специфичне околине. Итерације се смењују све док се не задовољи неки критеријум заустављања. У литератури се могу наћи успешне примене ове методе на решавање различитих локацијских проблема (видети [40, 41, 95, 120]).

У овој дисертацији је предложена хибридна метахеуристика за решавање  $U_rArHMCP$ , заснована на комбинацији GRASP методе и секвенцијалне методе променљивог спуста (SeqVND). Martins и сар. су у [108] први комбиновали GRASP и VND хеуристике за решавање проблема Штајнеровог стабла у графу. У литератури постоје примери примене овакве хибридизације и на друге проблеме оптимизације, као што су: проблем максималног одсецања у графу (енгл. *MAX-CUT Problem in Graphs*) [57], варијанта проблема минималног разаципућег стабла (енгл. *Degree-constrained Minimum Spanning Tree Problem*) [130], проблем филогеније (енгл. *Phylogeny Problem*) [131], проблем трговачког путника са доставом и преузимањем једног производа (енгл. *One-commodity Pickup-and-Delivery Traveling Salesman Problem*) [78], проблем рутирања камиона са приколицом уз коришћење пратећих депоа (енгл. *Single Truck and Trailer Routing Problem With Satellite Depots*) [146], проблем дводимензионалног и тродимензионалног паковања у контејнере (енгл. *Two-and Three-dimensional Bin Packing Problem*) [118], проблем минималног кашњења (енгл. *Traveling Repairman Problem*) [132] итд.

Основни кораци предложене хибридизације GRASP-VND за  $U_rArHMCP$  су приказани Алгоритмом 7. Репрезентација решења је иста као код GVNS-R методе изложене у Одељку 4.3, тј. решење  $S = (\bar{H}, A)$  је представљено скупом  $\bar{H}$  од  $p$  одабраних хабова и матрицом алокације  $A$ . Свака итерација GRASP-VND методе се састоји из конструкције иницијалног решења GRC процедуром, примене SeqVND процедуре уместо стандарде локалне претраге и помераја..

Улога похлепно стохастичке процедуре за конструкцију решења јесте генерисање иницијалног решења које се прослеђује процедури побољшања. GRC

---

**Алгоритам 7** Хибридна GRASP-VND метода за UrApHMCP

---

```

1: function GRASP-VND( $t_{max}, \varphi$ );
2:   Read_input();
3:   repeat
4:      $S \leftarrow \text{Greedy\_Randomized\_Construction}()$ ;
5:      $S' \leftarrow \text{SeqVND}(S)$ ;
6:     if  $S'$  is better than  $S$  then
7:        $S \leftarrow S'$ ;
8:     end
9:      $t \leftarrow \text{CpuTime}()$ ;
10:  until  $t > t_{max}$ ;
11: return  $S$ ;

```

---

**Алгоритам 8** GRC фаза

---

```

1: function GREEDY_RANDOMIZED_CONSTRUCTION()
2:    $\bar{H} \leftarrow \emptyset$ ;
3:   Create candidate list  $C$  of elements that can be added to  $\bar{H}$ ;
4:   while  $|\bar{H}| < p$  do
5:     Build the restricted candidate list (RCL( $C$ ));
6:     Select an element  $h$  from the RCL( $C$ ) at random;
7:      $\bar{H} \leftarrow \bar{H} \cup \{h\}$ ;
8:     Update  $C$ ;
9:   end
10:  Create allocation matrix  $A$ ;
11: return  $S = (\bar{H}, A)$ ;

```

---

почиње са празним скупом хабова  $\bar{H}$  и итеративно додаје један по један елемент (хаб), све док се не комплетира скуп  $\bar{H}$  од  $p$  хабова. Нека је  $C$  листа кандидата (енгл. *Candidate List*) састављена од свих елемената који могу бити додати у скуп  $\bar{H}$  (иницијално,  $C = H$ ). У GRC фази предложене GRASP-VND методе, листа кандидата се формира на основу вредности похлепне функције  $g(h) = \max\{c_{ih} : i \in N, i \neq h\}$ , која одражава промене у вредности функције циља које би настале укључивањем елемента  $h \in C$  у парцијални скуп  $\bar{H}$ . Следећи корак је конструкција ограничене листе кандидата (енгл. *Restricted Candidate List - RCL*) помоћу израчунатих вредности похлепне функције  $g(h)$ ,  $h \in C$ . RCL се састоји од свих кандидата  $h \in C$  за које важи  $g(h) \leq g_{min} + \varphi(g_{max} - g_{min})$ , где је  $\varphi \in [0, 1]$ ,  $g_{min} = \min_{h \in C} g(h)$  и  $g_{max} = \max_{h \in C} g(h)$ . Елемент  $h$  (не нужно најбољи) се на случајан начин бира из RCL и придружује парцијалном скупу  $\bar{H}$ . Уколико након придруживања извесног броја хабова ограничена листа кандидата буде празна, GRC фаза се понавља. Након

креирања скупа  $\bar{H}$  који садржи  $p$  хабова, алокације не-хаб чворова се одређују на исти начин као у предложеној GVNS-R методи за решавање UrApHMCP, тј. сваки чвор  $i$  је повезан са  $r$  хабова са најмањим вредностима функције  $alloc(i, h) = \chi c_{ih} + \max\{\alpha c_{hh'} : h' \in H\}$ . Након тога, лако је одредити најкраће путеве за сваки О-Д пар  $i - j$  и израчунати вредности функције циља, као што је описано у Одељку 4.3.

Решење генерисано у GRC фази постаје текуће најбоље решење и прослеђује се процедури побољшања решења. У предложеној GRASP-VND методи, као процедура побољшања коришћена је SeqVND хеуристика из GVNS-R методе која је детаљно описана у Одељку 4.3. Процедура SeqVND враћа као излаз пронађени локални оптимум. Предност при коришћењу SeqVND методе у односу на обичну локалну претрагу је у брзини извршавања процедуре, што побољшава укупну ефикасност алгорита.

Након примене SeqVND следи фаза помераја, у којој се текуће најбоље решење упоређује са решењем добијеним из фазе побољшања. Ако је решење из SeqVND фазе боље од тренутног решења, проглашава се за ново текуће најбоље решење. Итерације се смењују док се не задовољи критеријум заустављања.

Предложена GRASP-VND метода има два параметра -  $t_{max}$  и  $\varphi$ . Параметар  $t_{max}$  представља максимално дозвољено време извршавања алгорита. Избор елемената за креирање RCL зависи од параметра  $\varphi \in [0, 1]$ . Код основне варијанте GRASP хеуристике, која је имплементирана у оквиру хибридне методе,  $\varphi$  има фиксну вредност у свим итерацијама. Може се приметити да се у случају  $\varphi = 0$ , GRC фаза своди на класичан похлепни алгорита, док је случај  $\varphi = 1$  еквивалентан конструкцији решења на потпуно случајан начин.

## 4.5 Генетски алгорита за решавање проблема

Генетски алгорита (GA) припада групи стохастичких метахеуристика заснованих на популацији. Показао се врло ефикасним при решавању различитих проблема оптимизације (видети [27, 36, 69, 110, 126]). У литератури постоје бројни примери успешне примене ове методе на решавање хаб локацијских проблема, као што су: варијанте хаб локацијских проблема неограничених капацитета са једноструким алокацијама [1, 16, 35, 106, 144], хаб локацијских проблема ограничених капацитета са једноструким алокацијама [22, 137], проблем  $p$ -хаб медијане неограничених капацитета са једноструким алокацијама [98], проблем

$p$ -хаб медијане ограничених капацитета са једноструким алокацијама [135], проблем  $p$ -хаб центра неограничених капацитета са вишеструким алокацијама [97] итд. У монографији [136] су предложене варијанте GA за решавање различитих хаб локацијских проблема:  $p$ -хаб медијане и  $p$ -хаб центра неограничених капацитета са једноструким или вишеструким алокацијама, хаб локацијских проблема неограничених капацитета са једноструким или вишеструким алокацијама,  $p$ -хаб медијане и  $p$ -хаб центра ограничених капацитета са једноструким алокацијама и хаб локацијских проблема ограничених капацитета са једноструким или вишеструким алокацијама. Ова истраживања су била мотивација за дизајнирање различитих варијанти GA за решавање UrApHMCP, при чему ће у овој дисертацији бити приказана најуспешнија.

Основни кораци предложеног GA за решавање UrApHMCP су приказани Алгоритмом 9. Предложени GA користи бинарну **репрезентацију јединки**. Генетски код јединке садржи  $n$  гена, од којих сваки одговара једном чвору мреже. Уколико ген има вредност 1, успостављен је хаб на датој локацији, док 0 означава да хаб није успостављен. **Почетна популација** се састоји од  $N_{ind}$  јединки генерисаних на псеудослучајан начин. Тачно  $p$  случајно изабраних гена добија вредност 1, док остали узимају вредност 0. На овај начин се креира коректна јединка која одговара решењу са тачно  $p$  успостављених хабова. Дакле, генерисана почетна популација се састоји само од коректних јединки, док примењени генетски оператори чувају њихову коректност током GA генерација.

Након генерисања почетне популације, потребно је одредити квалитет сваке јединке. У ту сврху се свакој јединки придружује тзв. *функција прилагођености* (енгл. *fitness function*). Обично већа вредност функције прилагођености означава бољи квалитет јединке. Како је UrApHMCP проблем типа максимизације, у овој GA имплементацији, функција прилагођености је једнака одговарајућој вредности функције циља и рачуна се похлепном процедуром на следећи начин.

Нека је  $H = \{h_1, h_2, \dots, h_p\}$  скуп индекса који представљају тачно  $p$  хаб чворова, добијених из генетског кода јединке. Сваки хаб је придружен самом себи, док је за не-хаб чворове неопходно пронаћи адекватне алокације, користећи функцију  $alloc(i, h)$ , дефинисану на следећи начин:

$$alloc(i, h) = \chi_{c_{ih}} + \sum_{j \in N} \delta_{c_{hj}}, \forall i \in N, \forall h \in \{h_1, \dots, h_p\}. \quad (4.37)$$

Први сабирак у (4.37) представља трошкове од почетног чвора  $i \in N$  до хаба  $h \in H$ , док други сабирак представља збир трошкова на гранама од хаба  $h \in H$  до крајњих чворова. Након израчунавања вредности  $alloc(i, h), \forall i \in N, \forall h \in \{h_1, \dots, h_p\}$ , сваком не-хаб чвору  $i$  додељује се тачно  $r$  хабова са најнижим вредностима функције  $alloc(i, h)$ .

---

#### Алгоритам 9 Генетски алгоритам за UrApHMSP

---

- 1: Function GA( $max_1, max_2, N_{ind}, N_{el}, N_{nonel}, F_{tour}, p_{mut}$ );
  - 2: Generate initial population  $P$ ;
  - 3: Calculate fitness values for all individuals in population;
  - 4: **while** (no. iter. without impr.  $< max_1$  && no. iter  $< max_2$ ) **do**
  - 5:     Set the next population  $P_1 = \emptyset$ ;
  - 6:     Choose  $N_{el}$  elite individuals;
  - 7:     Add  $N_{el}$  elite individuals to  $P_1$ ;
  - 8:      $P_2 \leftarrow$  FGTS (non-elite individuals,  $F_{tour}$ );
  - 9:     **for**  $i = 1$  to  $N_{nonel}$  **do**
  - 10:         Select two parent-individuals  $p_1$  and  $p_2$  from  $P_2$ ;
  - 11:          $o_1 \leftarrow$  Crossover ( $p_1, p_2$ );
  - 12:          $o_1 \leftarrow$  Mutate ( $o_1, p_{mut}$ );
  - 13:         Calculate fitness value of  $o_1$ ;
  - 14:         Add  $o_1$  to  $P_1$ ;
  - 15:     **end**
  - 15:      $P \leftarrow P_1$ ;
  - 16: **end**
  - 16: Return the best individual from  $P$  and its fitness value;
- 

Нека су  $K$  и  $M$  скупови од  $r$  хабова повезаних редом са чворовима  $i$  и  $j$ . За сваки О–Д пар  $i - j$ , хабови  $k \in K$  и  $m \in M$ , преко којих се врши транспорт од почетног чвора  $i$  до крајњег чвора  $j$ , се бирају тако да трошкови транспорта  $d_{ij}^{km} = \chi c_{ik} + \alpha c_{km} + \delta c_{mj}$  буду минимални, тј.  $(k, m) = argmin\{d_{ij} | k \in K, m \in M\}$ . Природно, неки од чворова  $i, j, k$ , и  $m$  се могу поклопити. Такође, чак и ако скупови  $K$  и  $M$  имају заједнички хаб, саобраћај од  $i$  до  $j$  не мора бити обавезно усмерен преко тог хаба. При налажењу најповољнијих хабова  $k$  и  $m$  за реализацију транспорта од чвора  $i$  до чвора  $j$ , неопходно је испитати свих  $r^2$  могућности, што значи да рачунање свих О–Д најкраћих путева захтева  $O(n^2 r^2)$  операција. У најгорем случају када је  $r = p$ , најкраћи путеви се могу одредити користећи модификовани Floyd-Warshall алгоритам за налажење најкраћих путева. Након тога, вредност функције циља је лако израчунати једноставним сумирањем потражњи на покривеним путевима. Коначно, вредност функције прилагођености дате јединке је једнака добијеној вредности функције

циља, срачунате описаном похлепном процедуром. Уколико се деси да неком хабу није придружен ниједан не-хаб чвор, јединка се сматра некоректном и вредност њене функције прилагођености се поставља на нулу.

Предложени ГА користи *елитистичку стратегију* која има за циљ да сачува високо прилагођене јединке популације (видети [106, 135, 137]). У свакој итерацији,  $N_{el}$  најбољих јединки из тренутне популације директно прелазе у наредну. Ове јединке се називају *елитним јединкама*. Како за њих није потребно поново рачунати функције прилагођености, долази до значајне уштеде процесорског времена. Остатак популације је генерисан применом генетских оператора селекције, укрштања и мутације на скуп неелитних јединки са  $N_{nonel} = N_{ind} - N_{el}$  чланова.

Оператор **селекције** бира јединке из скупа неелитних јединки које учествују у процесу креирања нове генерације, према вредностима функције прилагођености. При томе, у скуп изабраних јединки од  $N_{nonel}$  елемената долази до понављања јединки бољег квалитета. Креиран ГА за УгАрНМСР користи *фино-градирану турнирску селекцију* (енгл. *fine-grained tournament selection - FGTS*), предложена у [58], која представља унапређење стандардне турнирске селекције. Овај оператор користи реални параметар  $F_{tour}$ , који представља жељену средњу величину турнира. Претпоставља се постојање два типа турнира. Први се спроводи  $k_1$  пута и његова величина је  $\lfloor F_{tour} \rfloor$ , док се други реализује  $k_2$  пута са  $\lceil F_{tour} \rceil$  јединки, учесника турнира. При томе је  $\lfloor F_{tour} \rfloor = x$ ,  $\lceil F_{tour} \rceil = y \Leftrightarrow x \leq F_{tour} \leq y$  ( $x, y \in \mathbb{Z}$ ,  $F_{tour} \in \mathbb{R}$ ) и  $F_{tour} \approx (k_1 * \lfloor F_{tour} \rfloor + k_2 * \lceil F_{tour} \rceil) / (k_1 + k_2)$ . Време извршавања FGTS оператора је  $O(N_{nonel} F_{tour})$ . У пракси се параметар  $F_{tour}$  сматра константним, тако да је временска сложеност  $O(N_{nonel})$ .

Оператор **укрштања** се примењује на пар изабраних јединки-родитеља стварајући јединку-потомка за наредну генерацију јединки. Стандардни оператор једнопозиционог укрштања размењује делове генетског кода родитеља почевши од случајно изабране тачке укрштања. Овај оператор није адекватан за УгАрНМСР, јер број јединица (хабова) у потомцима може постати различит од  $p$ , што значи да би добијене јединке-потомци биле некоректне. Зато се у предложеној ГА имплементацији користи модификован оператор укрштања који чува коректност јединки на следећи начин. Пар јединки родитеља изабраних на случајан начин се укршта и производи једну јединку потомка. Скуп успостављених хабова у потомку настаје као унија хабова из јединки родитеља. Очигледно је да потомак има више од  $p$  хабова, па је неопходно вредност неких

гена поставити на 0. У ту сврху се конструише матрица  $E$  са  $n$  врста и  $q$  колона, где је  $q$  број хабова у потомку. Елемент  $E(i, j)$  означава трошкове транспорта од чвора  $i$  до хаба  $h_j$ :

$$E(i, j) = \chi c_{ij}. \quad (4.38)$$

Коришћењем матрице  $E$ , хабови се итеративно уклањају следећом процедуром. Нека је  $m(E)$  сума најмањих вредности  $E(i, j)$  по свим врстама, тј.

$$m(E) = \sum_{i=1}^n \min_{j \in \{1, 2, \dots, q\}} E(i, j) \quad (4.39)$$

и нека је  $E_j$  матрица добијена из  $E$  брисањем колоне  $j$ . Како уклањање хаба  $h_j$  одговара брисању  $j$ -те колоне матрице  $E$ , хаб  $h_j$  за уклањање је изабран као хаб који одговара  $j$ -тој колони за коју је разлика између  $m(E_j)$  и  $m(E)$  најмања. Након брисања хаба  $h_j$ , матрица  $E$  се замењује матрицом  $E_j$ . Описани процес се понавља до испуњења услова да је број хабова у потомку једнак  $p$ .

Свака јединка-потомак подлеже оператору **мутације**. У предложеној GA имплементацији, коришћена је проста мутација. На случајан начин се бира ген у генетском коду јединке који се затим инвертује са вероватноћом  $p_{mut}$ . Након мутације потомка, могуће је да број јединица у његовом генетском коду постане већи или мањи од  $p$ . У овим случајевима, потомак се поправља додавањем или одузимањем неопходног броја хабова, изабраних на случајан начин. Улога примењеног оператора мутације је да доприноси повећавању разноврсности генетског материјала и спречавању преране конвергенције ка локалном оптимуму.

Описани процес се итеративно понавља до задовољења бар једног од критеријума заустављања – максималног броја GA итерација без побољшања решења ( $max_1$ ) или укупног броја итерација ( $max_2$ ). Након завршеног процеса, алгоритам пријављује решење које одговара најбоље прилагођеној јединки из текуће популације, као и одговарајућу вредност функције циља. Квалитет решења добијених генетским алгоритмом у великој мери зависи од избора вредности улазних параметара  $N_{ind}$ ,  $N_{el}$ ,  $F_{tour}$  и  $p_{mut}$ , који се најчешће одређују експерименталним путем.



## 4.6 Експериментални резултати

GVNS-R, GRASP-VND и GA метахеуристике за решавање UrApHMCP имплементирани су у програмском језику C++. Математичке формулације разматраног проблема, изложене у Одељку 4.2, су упоређене коришћењем егзактног решавача CPLEX 12.6, а све формулације су такође кодирани у програмском језику C++. Сва тестирања су извршена на рачунару са процесором Intel Core I7 са 2.8 GHz CPU и RAM меморијом од 8 GB под Windows 10 оперативним системом.

Експериментални резултати добијени су тестирањем стандардних CAB и AP инстанци, али и скупа података USA423, предложеног у [120]. USA423 садржи датотеку реалних података о растојањима и протоку путника за 423 града у Сједињеним Америчким Државама, евидентираним на кварталном нивоу. Ове инстанце до сада нису коришћене за тестирање  $p$ -хаб проблема максималног покривања. За CAB инстанце, параметар  $\alpha$  узима вредности од 0.2 до 0.8, а параметри  $\chi$  и  $\delta$  су постављени на 1. Број хабова  $p \in \{3, 4, 5\}$ , док је  $r \in \{2, \dots, p-1\}$ . За AP инстанце, параметар  $\alpha = 0.75$ , а параметри  $\chi$  и  $\delta$  су такође постављени на 1. Број хабова  $p \in \{3, 4, 5, 10\}$  и  $r \in \{2, \dots, p-1\}$ . За USA423 инстанце,  $p \in \{3, \dots, 7\}$  и  $r \in \{2, \dots, p-1\}$ . За сваку комбинацију параметара  $p$  и  $r$ , коришћена су два различита скупа параметара  $\chi, \alpha, \delta$ : 0.1, 0.07, 0.09 и 0.09, 0.075, 0.08.

Параметри  $q_{ij}^{km}$  и  $r_{ij}^{km}$  коришћени редом за бинарно и парцијално покривање, су дефинисани као у Одељку 2.4. Вредности параметра  $\beta$  су такође наведене у Одељку 2.4.

### Резултати GVNS-R и GRASP-VND метода за решавање UrApHMCP

У циљу коректног поређења резултата предложених GVNS-R и GRASP-VND метода за решавање UrApHMCP, коришћене су исте вредности параметра  $t_{max}$ , прилагођене величинама тестираних инстанци. За CAB и AP инстанце до 200 чворова,  $t_{max} = n$  секунди, док је за USA423 инстанце са 423 чвора,  $t_{max} = 1000$  секунди, за обе хеуристике. За сва тестирања, GVNS-R параметар  $k_{max}$  је постављен на  $p$ . Вредност параметра  $\varphi$  из GRC фазе GRASP-VND методе је 0.8. Ове вредности параметара су одређене на основу низа прелиминарних тестирања предложених метода на инстанцама проблема свих диманзија.

За инстанце малих димензија, квалитет резултата добијених предложеним методама је оцењен упоређивањем са резултатима добијеним егзактним решавачем CPLEX 12.6. Максимално време дозвољено CPLEX решавачу је постављено на 1h (3600s), за сваку тестирану инстанцу. У Табелама 4.1–4.4 су приказани резултати добијени CPLEX решавачем, за све три предложене UrApHMCP формулације, као и резултати добијени тестирањем дизајнираних GVNS-R и GRASP-VND метахеуристика, на инстанцама мањих димензија ( $n = 25, 40, 50$ ). У случајевима када CPLEX 12.6 није успео да пронађе чак ни допустиво решење, због ограничене меморије или времена, уписана је „–“ у одговарајућем пољу. Табеле 4.5–4.7 садрже резултате добијене предложеним GVNS и GRASP-VND методама тестирањем на инстанцама проблема великих димензија ( $n = 100, 200, 423$ ). Ове инстанце су такође биле недостижне за CPLEX 12.6 решавач. Заглавља колона Табела 4.1–4.7 су дефинисана на исти начин као у Одељку 2.4.

Из резултата представљених у Табели 4.1 може се закључити да на САВ инстанцама, у случају бинарног покривања, Општа формулација даје боље резултате у поређењу са Четвороиндексном и Троиндексном формулацијом. CPLEX решавач користећи Општу формулацију достиже сва оптимална решења, за 37.82 секунде (у просеку). Користећи Четвороиндексну формулацију, CPLEX за ограничено време од 1h не достиже оптимална решења за 3 од 24 САВ инстанце. Користећи Троиндексну формулацију, CPLEX је успео да реши 12 САВ инстанци до оптималности, док су за преостале инстанце пронађена само допустива решења за које CPLEX није успео да докаже оптималност у датом времену извршавања. Међутим, може се приметити да се вредности функције циља за 8 од добијених 12 допустивих решења поклапају са оптималним вредностима добијеним CPLEX-ом користећи Општу формулацију. Из резултата приказаних у последње четири колоне Табеле 4.1 може се приметити да предложене GVNS-R и GRASP-VND методе достижу сва оптимална решења за кратко време извршавања. У просеку, време извршавања GVNS-R методе је 0.07 секунди, а GRASP-VND методе 1.57 секунди.

Резултати добијени на САВ инстанцама у случају парцијалног покривања су приказани у Табели 4.2. Може се приметити да 4-индексна и Општа формулација омогућавају CPLEX решавачу налажење свих оптималних решења. Међутим, време извршавања CPLEX решавача је око 2.8 пута краће при коришћењу Опште формулације. Показало се да је Троиндексна формулација

ГЛАВА 4. ПРОБЛЕМ  $p$ -ХАБ МАКСИМАЛНОГ ПОКРИВАЊА  
НЕОГРАНИЧЕНИХ КАПАЦИТЕТА СА  $r$ -АЛОКАЦИЈСКОМ ШЕМОМ

Табела 4.1: Експериментални резултати и поређења решења UrApHMSP на САВ инстанцама - случај бинарног покривања

Бинарно покривање													
$n.p$	$r$	$\alpha$	Optimal	4-индексна ф.		3-индексна ф.		Општа ф.		GVNS-R		GRASP-VND	
			value(%)	dev(%)	time(s)	dev(%)	time(s)	dev(%)	time(s)	dev(%)	time(s)	dev(%)	time(s)
25.3	2	0.2	<b>96.58</b>	<b>0.00</b>	1805.86	<b>0.00</b>	3600.08	<b>0.00</b>	103.35	<b>0.00</b>	0.01	<b>0.00</b>	0.02
25.4	2	0.2	<b>95.71</b>	<b>0.00</b>	1541.47	2.23	3600.17	<b>0.00</b>	93.44	<b>0.00</b>	0.04	<b>0.00</b>	0.12
25.4	3	0.2	<b>95.71</b>	<b>0.00</b>	2100.11	1.81	3600.18	<b>0.00</b>	82.94	<b>0.00</b>	0.03	<b>0.00</b>	0.01
25.5	2	0.2	<b>92.70</b>	<b>0.00</b>	2087.15	<b>0.00</b>	3600.07	<b>0.00</b>	43.78	<b>0.00</b>	0.08	<b>0.00</b>	0.37
25.5	3	0.2	<b>92.70</b>	<b>0.00</b>	2849.29	<b>0.00</b>	3600.15	<b>0.00</b>	38.32	<b>0.00</b>	0.06	<b>0.00</b>	0.03
25.5	4	0.2	<b>92.70</b>	<b>0.00</b>	2431.20	0.34	3600.37	<b>0.00</b>	35.17	<b>0.00</b>	0.02	<b>0.00</b>	0.03
25.3	2	0.4	<b>96.24</b>	<b>0.00</b>	2305.22	<b>0.00</b>	2570.21	<b>0.00</b>	50.68	<b>0.00</b>	0.01	<b>0.00</b>	0.02
25.4	2	0.4	<b>95.01</b>	1.29	3600.09	0.56	3600.15	<b>0.00</b>	54.62	<b>0.00</b>	0.02	<b>0.00</b>	0.07
25.4	3	0.4	<b>95.01</b>	<b>0.00</b>	2049.70	<b>0.00</b>	3600.06	<b>0.00</b>	44.16	<b>0.00</b>	0.04	<b>0.00</b>	0.11
25.5	2	0.4	<b>91.45</b>	<b>0.00</b>	1267.85	<b>0.00</b>	3600.05	<b>0.00</b>	25.44	<b>0.00</b>	0.17	<b>0.00</b>	1.67
25.5	3	0.4	<b>91.84</b>	<b>0.00</b>	1105.92	<b>0.00</b>	3600.08	<b>0.00</b>	26.85	<b>0.00</b>	0.11	<b>0.00</b>	0.13
25.5	4	0.4	<b>91.84</b>	<b>0.00</b>	1110.36	<b>0.00</b>	2918.78	<b>0.00</b>	25.91	<b>0.00</b>	0.06	<b>0.00</b>	0.08
25.3	2	0.6	<b>93.17</b>	<b>0.00</b>	354.23	<b>0.00</b>	874.69	<b>0.00</b>	22.37	<b>0.00</b>	0.02	<b>0.00</b>	0.03
25.4	2	0.6	<b>93.62</b>	<b>0.00</b>	1055.33	<b>0.00</b>	965.70	<b>0.00</b>	25.60	<b>0.00</b>	0.04	<b>0.00</b>	1.89
25.4	3	0.6	<b>93.63</b>	<b>0.00</b>	831.95	<b>0.00</b>	961.41	<b>0.00</b>	24.02	<b>0.00</b>	0.29	<b>0.00</b>	0.07
25.5	2	0.6	<b>90.12</b>	0.13	3600.05	<b>0.00</b>	3600.16	<b>0.00</b>	41.44	<b>0.00</b>	0.07	<b>0.00</b>	16.74
25.5	3	0.6	<b>90.19</b>	<b>0.00</b>	2727.55	<b>0.00</b>	3600.06	<b>0.00</b>	27.82	<b>0.00</b>	0.12	<b>0.00</b>	4.37
25.5	4	0.6	<b>90.19</b>	<b>0.00</b>	3531.83	<b>0.00</b>	2935.68	<b>0.00</b>	25.17	<b>0.00</b>	0.14	<b>0.00</b>	0.02
25.3	2	0.8	<b>90.08</b>	<b>0.00</b>	682.41	<b>0.00</b>	486.60	<b>0.00</b>	25.16	<b>0.00</b>	0.01	<b>0.00</b>	0.02
25.4	2	0.8	<b>89.60</b>	0.65	3600.07	<b>0.00</b>	1141.17	<b>0.00</b>	26.19	<b>0.00</b>	0.09	<b>0.00</b>	0.33
25.4	3	0.8	<b>89.61</b>	<b>0.00</b>	1829.71	<b>0.00</b>	1477.23	<b>0.00</b>	23.66	<b>0.00</b>	0.04	<b>0.00</b>	0.02
25.5	2	0.8	<b>89.03</b>	<b>0.00</b>	728.15	<b>0.00</b>	427.32	<b>0.00</b>	14.68	<b>0.00</b>	0.17	<b>0.00</b>	8.45
25.5	3	0.8	<b>89.05</b>	<b>0.00</b>	605.09	<b>0.00</b>	1077.34	<b>0.00</b>	18.27	<b>0.00</b>	0.04	<b>0.00</b>	3.02
25.5	4	0.8	<b>89.05</b>	<b>0.00</b>	730.11	<b>0.00</b>	513.68	<b>0.00</b>	8.52	<b>0.00</b>	0.03	<b>0.00</b>	0.05
<b>Просек:</b>			<b>92.28</b>	0.09	1855.45	0.21	2481.31	<b>0.00</b>	37.82	<b>0.00</b>	<b>0.07</b>	<b>0.00</b>	1.57

најмање погодна за решавање САВ инстанци у случају паријалног покривања. Користећи ову формулацију, CPLEX решавач је за све тестиране инстанце пронашао само допустива решења, за које оптималност није доказана. Просечно одступање од оптималних решења је 19.50%. Предложена GVNS-R метахеуристика достиже сва позната оптимална решења добијена CPLEX решавачем за 0.79 секунди (у просеку), док GRASP-VND метода не налази оптимално решење само за једну од 24 тестираних САВ инстанци. Просечно време извршавања GRASP-VND методе (0.74 секунди) је мало краће у односу на просечно време извршавања GVNS-R методе (0.79 секунди).

У Табели 4.3 су приказани резултати тестирања на AP инстанцама до 50 чворова, у случају бинарног покривања, на основу којих се може закључити да Општа формулација надмашује Четвороиндексну и Троиндексну формулацију у смислу квалитета решења и просечног времена извршавања CPLEX решавача. Користећи Општу формулацију, CPLEX достиже оптимална решења за 25 од 42 тестиране AP инстанце, док за преосталих 17 инстанци даје допустива решења. Користећи Троиндексну формулацију, CPLEX налази оптимална ре-

ГЛАВА 4. ПРОБЛЕМ  $p$ -ХАБ МАКСИМАЛНОГ ПОКРИВАЊА  
НЕОГРАНИЧЕНИХ КАПАЦИТЕТА СА  $r$ -АЛОКАЦИЈСКОМ ШЕМОМ

Табела 4.2: Експериментални резултати и поређења решења UrApHMSP на САВ инстанцама - случај парцијалног покривања

Парцијално покривање													
$n.p$	$r$	$\alpha$	Optimal value(%)	4-индексна ф.		3-индексна ф.		Општа ф.		GVNS-R		GRASP-VND	
				dev(%)	time(s)	dev(%)	time(s)	dev(%)	time(s)	dev(%)	time(s)	dev(%)	time(s)
25.3	2	0.2	<b>98.26</b>	<b>0.00</b>	1474.38	6.33	3600.10	<b>0.00</b>	622.34	<b>0.00</b>	0.01	<b>0.00</b>	0.02
25.4	2	0.2	<b>97.14</b>	<b>0.00</b>	1242.43	56.31	3600.06	<b>0.00</b>	416.52	<b>0.00</b>	0.05	<b>0.00</b>	0.1
25.4	3	0.2	<b>97.14</b>	<b>0.00</b>	1275.13	56.87	3600.06	<b>0.00</b>	438.32	<b>0.00</b>	0.09	<b>0.00</b>	0.01
25.5	2	0.2	<b>94.91</b>	<b>0.00</b>	1100.67	16.01	3600.21	<b>0.00</b>	272.19	<b>0.00</b>	0.25	<b>0.00</b>	0.39
25.5	3	0.2	<b>94.91</b>	<b>0.00</b>	906.86	26.15	3600.16	<b>0.00</b>	219.08	<b>0.00</b>	0.05	<b>0.00</b>	0.03
25.5	4	0.2	<b>94.91</b>	<b>0.00</b>	689.01	34.73	3600.08	<b>0.00</b>	228.76	<b>0.00</b>	0.05	<b>0.00</b>	0.04
25.3	2	0.4	<b>97.23</b>	<b>0.00</b>	578.59	1.98	3600.08	<b>0.00</b>	279.55	<b>0.00</b>	0.01	<b>0.00</b>	0.02
25.4	2	0.4	<b>96.90</b>	<b>0.00</b>	448.57	44.23	3600.12	<b>0.00</b>	282.18	<b>0.00</b>	0.49	<b>0.00</b>	0.02
25.4	3	0.4	<b>96.90</b>	<b>0.00</b>	819.38	48.46	3600.09	<b>0.00</b>	175.96	<b>0.00</b>	0.22	<b>0.00</b>	0.02
25.5	2	0.4	<b>95.04</b>	<b>0.00</b>	429.15	6.32	3600.04	<b>0.00</b>	174.47	<b>0.00</b>	0.51	<b>0.00</b>	1.27
25.5	3	0.4	<b>95.12</b>	<b>0.00</b>	265.16	6.70	3600.22	<b>0.00</b>	116.47	<b>0.00</b>	0.05	<b>0.00</b>	0.03
25.5	4	0.4	<b>95.12</b>	<b>0.00</b>	295.22	2.77	3600.16	<b>0.00</b>	185.97	<b>0.00</b>	0.10	<b>0.00</b>	0.04
25.3	2	0.6	<b>95.75</b>	<b>0.00</b>	457.27	0.73	3600.22	<b>0.00</b>	330.25	<b>0.00</b>	0.11	<b>0.00</b>	0.02
25.4	2	0.6	<b>95.99</b>	<b>0.00</b>	484.10	33.03	3600.13	<b>0.00</b>	391.46	<b>0.00</b>	0.08	<b>0.00</b>	2.3
25.4	3	0.6	<b>96.12</b>	<b>0.00</b>	292.23	3.84	3600.19	<b>0.00</b>	283.04	<b>0.00</b>	0.05	<b>0.00</b>	0.04
25.5	2	0.6	<b>94.56</b>	<b>0.00</b>	770.64	41.93	3600.16	<b>0.00</b>	204.69	<b>0.00</b>	0.48	<b>0.00</b>	1.23
25.5	3	0.6	<b>94.61</b>	<b>0.00</b>	706.50	22.92	3600.10	<b>0.00</b>	150.15	<b>0.00</b>	0.05	<b>0.00</b>	0.02
25.5	4	0.6	<b>94.61</b>	<b>0.00</b>	481.97	9.21	3600.24	<b>0.00</b>	115.27	<b>0.00</b>	0.13	<b>0.00</b>	0.03
25.3	2	0.8	<b>93.66</b>	<b>0.00</b>	320.51	2.04	3600.17	<b>0.00</b>	191.57	<b>0.00</b>	0.11	<b>0.00</b>	0.02
25.4	2	0.8	<b>93.58</b>	<b>0.00</b>	74.11	14.55	3600.21	<b>0.00</b>	196.08	<b>0.00</b>	0.54	<b>0.00</b>	0.42
25.4	3	0.8	<b>93.58</b>	<b>0.00</b>	68.41	3.15	3600.29	<b>0.00</b>	157.57	<b>0.00</b>	0.11	<b>0.00</b>	0.07
25.5	2	0.8	<b>91.40</b>	<b>0.00</b>	2054.33	7.41	3600.09	<b>0.00</b>	194.40	<b>0.00</b>	14.73	<b>0.22</b>	3.81
25.5	3	0.8	<b>91.51</b>	<b>0.00</b>	508.65	5.88	3600.13	<b>0.00</b>	186.28	<b>0.00</b>	0.12	<b>0.00</b>	7.81
25.5	4	0.8	<b>91.51</b>	<b>0.00</b>	698.70	16.37	3600.17	<b>0.00</b>	138.91	<b>0.00</b>	0.63	<b>0.00</b>	0.09
<b>Просек:</b>			<b>95.02</b>	<b>0.00</b>	685.08	19.50	3600.15	<b>0.00</b>	247.98	<b>0.00</b>	0.79	<b>0.01</b>	<b>0.74</b>

шења за 12 од 42 тестиране AP инстанце и даје допуштена решења за преостале. Четвороиндексна формулација је показала најлошије перформансе, јер се испоставило да није погодна за решавање AP инстанци са више од 40 чворова. Може се закључити да особине ових формулација долазе до изражаја са повећавањем димензије проблема, што може бити смерница у будућим детаљнијим анализама. Из колоне означене са  $time(s)$ , може се видети да је време извршавања CPLEX решавача значајно краће са Општом формулацијом у односу на Четвороиндексну и Троиндексну формулацију. Предложена GVNS-R метода достиже сва позната оптимална решења добијена CPLEX решавачем и даје нова најбоља решења за инстанце на којима оптималност није доказана. У случају инстанце  $n = 40, p = 3, r = 2$ , допуштено решење добијено CPLEX решавачем (оптималност није доказана) се поклапа са најбољим решењем добијеним GVNS методом. Та чињеница води ка претпоставци да је на поменутој инстанци ипак достигнуто оптимално решење (иако није доказано). Хибридна GRASP-VND метода само у случају једне тестиране инстанце ( $n = 40, p = 10, r = 2$ ) не достиже оптимално решење. Такође, за четири AP инстанце, GRASP-VND

ГЛАВА 4. ПРОБЛЕМ  $p$ -ХАБ МАКСИМАЛНОГ ПОКРИВАЊА  
НЕОГРАНИЧЕНИХ КАПАЦИТЕТА СА  $r$ -АЛОКАЦИЈСКОМ ШЕМОМ

---

обезбеђује лошија решења у поређењу са GVNS-R методом. Како су просечна времена извршавања GVNS-R и GVNS-VND метода редом 7.19 и 33.61 секунда, може се закључити да је при тестирању AP инстанци до 50 чворова, у случају бинарног покривања, предложена GVNS-R метода у просеку око 4.7 пута бржа од GVNS-VND методе.

Експериментални резултати добијени на AP инстанцама до 50 чворова, у случају парцијалног покривања, приказани су у Табели 4.4. Поново се може закључити да је Општа формулација показала боље перформансе у поређењу са Четвороиндексном и Троиндексном, у смислу квалитета решења и просечног времена извршавања CPLEX решавача. Користећи Општу формулацију, CPLEX решавач проналази оптимална решења за 20 од 42 тестиране AP инстанце, док користећи Троиндексну формулацију, CPLEX не успева да пронађе оптимално решење ни за једну AP инстанцу са више од 50 чворова. Слично као у случају бинарног покривања, експериментални резултати показују да Четвороиндексна формулација није погодна за решавање AP инстанци са више од 40 чворова. Предложена GVNS-R метахеуристика не проналази позната оптимална решења за три тестиране AP инстанце ( $n = 40, p = 10, r = 2; n = 40, p = 10, r = 3; n = 40, p = 10, r = 6$ ). GRASP-VND метода не достиже оптимално решење само за једну тестирану AP инстанцу ( $n = 40, p = 10, r = 2$ ). Иако су просечна одступања вредности функције циља најбољих GVNS-R и GRASP-VND решења од оптималних или најбољих познатих 0.0019%, односно 0.0043%, на 5 инстанци са допустивим решењима (за које CPLEX није успео да докаже оптималност), GRASP-VND добија боља решења у поређењу са GVNS-R методом. У просеку, GVNS-R обезбеђује најбоља решења за 14.82 секунде, а GRASP-VND метода за 46.37 секунди, што значи да је на овим инстанцама GVNS-R метода у просеку око 3.13 пута бржа од GRASP-VND методе.

За инстанце већих димензија ( $n = 100, 200, 423$ ), CPLEX због ограничене меморије не обезбеђује чак ни допустива решења, за оба концепта покривања. Из тог разлога, у Табелама 4.5–4.7 су приказани само резултати добијени предложеним GVNS-R и GRASP-VND метахеуристикима.

У Табели 4.5 су приказани резултати тестираних AP инстанци са 100 и 200 чворова, за оба концепта покривања. Иако обе предложене метахеуристике проналазе квалитетна решења UрArHMCP, може се приметити да за релативно кратко време извршавања, GVNS-R даје решења бољег квалитета у односу на GRASP-VND. У случају бинарног покривања, просечне вредности функције

ГЛАВА 4. ПРОБЛЕМ  $p$ -ХАБ МАКСИМАЛНОГ ПОКРИВАЊА  
НЕОГРАНИЧЕНИХ КАПАЦИТЕТА СА  $r$ -АЛОКАЦИЈСКОМ ШЕМОМ

Табела 4.3: Експериментални резултати и поређења решења UrArHMCP на AP инстанцама до 50 чворова - случај бинарног покривања

Бинарно покривање												
$n.p$	$r$	Optimal/Best value(%)	4-индексна ф.		3-индексна ф.		Општа ф.		GVNS-R		GRASP-VND	
			dev(%)	time(s)	dev(%)	time(s)	dev(%)	time(s)	dev(%)	time(s)	dev(%)	time(s)
25.3	2	<b>95.1324</b> <sup>opt</sup>	<b>0.0000</b>	765.58	<b>0.0000</b>	2312.74	<b>0.0000</b>	39.15	<b>0.0000</b>	0.10	<b>0.0000</b>	0.14
25.4	2	<b>96.9396</b> <sup>opt</sup>	<b>0.0000</b>	3000.50	0.0391	3600.13	<b>0.0000</b>	43.70	<b>0.0000</b>	1.99	<b>0.0000</b>	2.84
25.4	3	<b>97.1639</b> <sup>opt</sup>	<b>0.0000</b>	1068.12	<b>0.0000</b>	2364.42	<b>0.0000</b>	37.51	<b>0.0000</b>	0.07	<b>0.0000</b>	0.23
25.5	2	<b>97.8875</b> <sup>opt</sup>	0.4243	3600.07	0.1257	3600.19	<b>0.0000</b>	47.65	<b>0.0000</b>	3.43	<b>0.0000</b>	6.06
25.5	3	<b>98.3509</b> <sup>opt</sup>	0.3270	3600.10	<b>0.0000</b>	2408.74	<b>0.0000</b>	48.05	<b>0.0000</b>	1.31	<b>0.0000</b>	0.66
25.5	4	<b>98.3509</b> <sup>opt</sup>	0.2959	3600.09	<b>0.0000</b>	2196.76	<b>0.0000</b>	29.38	<b>0.0000</b>	0.12	<b>0.0000</b>	0.81
25.10	2	<b>99.6980</b> <sup>opt</sup>	<b>0.0000</b>	121.69	<b>0.0000</b>	1082.08	<b>0.0000</b>	4.86	<b>0.0000</b>	0.05	<b>0.0000</b>	1.09
25.10	3	<b>99.6980</b> <sup>opt</sup>	<b>0.0000</b>	112.54	<b>0.0000</b>	117.03	<b>0.0000</b>	14.51	<b>0.0000</b>	0.10	<b>0.0000</b>	1.89
25.10	4	<b>99.6980</b> <sup>opt</sup>	<b>0.0000</b>	330.98	<b>0.0000</b>	926.81	<b>0.0000</b>	7.50	<b>0.0000</b>	0.63	<b>0.0000</b>	0.92
25.10	5	<b>99.6980</b> <sup>opt</sup>	<b>0.0000</b>	121.49	<b>0.0000</b>	685.08	<b>0.0000</b>	3.47	<b>0.0000</b>	0.9	<b>0.0000</b>	1.09
25.10	6	<b>99.6980</b> <sup>opt</sup>	<b>0.0000</b>	113.81	<b>0.0000</b>	264.89	<b>0.0000</b>	2.48	<b>0.0000</b>	0.76	<b>0.0000</b>	0.05
25.10	7	<b>99.6980</b> <sup>opt</sup>	<b>0.0000</b>	117.42	<b>0.0000</b>	76.33	<b>0.0000</b>	3.31	<b>0.0000</b>	0.07	<b>0.0000</b>	0.05
25.10	8	<b>99.6980</b> <sup>opt</sup>	<b>0.0000</b>	106.90	<b>0.0000</b>	39.06	<b>0.0000</b>	2.97	<b>0.0000</b>	0.11	<b>0.0000</b>	0.06
25.10	9	<b>99.6980</b> <sup>opt</sup>	<b>0.0000</b>	76.00	<b>0.0000</b>	191.07	<b>0.0000</b>	3.11	<b>0.0000</b>	0.05	<b>0.0000</b>	0.06
40.3	2	<b>97.8088</b>	16.0967	3600.36	23.1551	3604.80	<b>0.0000</b>	3600.18	<b>0.0000</b>	1.79	<b>0.0000</b>	1.83
40.4	2	<b>97.0439</b>	260.5502	3600.35	41.7530	3617.50	0.4127	3600.17	<b>0.0000</b>	0.22	<b>0.0000</b>	16.47
40.4	3	<b>97.2930</b> <sup>opt</sup>	239.6639	3600.36	3809.0998	3602.17	<b>0.0000</b>	3061.63	<b>0.0000</b>	0.78	<b>0.0000</b>	0.18
40.5	2	<b>97.4376</b>	154.4156	3600.36	47.7431	3602.57	0.2275	3600.2	<b>0.0000</b>	2.87	<b>0.0000</b>	21.95
40.5	3	<b>97.7540</b> <sup>opt</sup>	156.8911	3600.45	11102.3488	3603.45	<b>0.0000</b>	2955.14	<b>0.0000</b>	2.30	<b>0.0000</b>	22.04
40.5	4	<b>97.7996</b> <sup>opt</sup>	171.3475	3600.35	11107.5741	3602.63	<b>0.0000</b>	2305.26	<b>0.0000</b>	2.39	<b>0.0000</b>	0.48
40.10	2	<b>99.3047</b> <sup>opt</sup>	220.7537	3600.31	50.5739	3605.21	<b>0.0000</b>	517.00	<b>0.0000</b>	50.87	0.0582	744.63
40.10	3	<b>99.3101</b> <sup>opt</sup>	5.6377	3600.37	11280.6624	3605.70	<b>0.0000</b>	774.49	<b>0.0000</b>	16.85	<b>0.0000</b>	190.04
40.10	4	<b>99.3101</b> <sup>opt</sup>	232.9079	3600.33	11280.6478	3616.11	<b>0.0000</b>	651.68	<b>0.0000</b>	6.62	<b>0.0000</b>	90.37
40.10	5	<b>99.3101</b> <sup>opt</sup>	0.0901	3600.40	13.9121	3600.24	<b>0.0000</b>	277.49	<b>0.0000</b>	0.47	<b>0.0000</b>	42.02
40.10	6	<b>99.3101</b> <sup>opt</sup>	<b>0.0000</b>	3600.61	12.5676	3600.38	<b>0.0000</b>	123.79	<b>0.0000</b>	2.39	<b>0.0000</b>	28.76
40.10	7	<b>99.3101</b> <sup>opt</sup>	0.0866	3600.37	16.2000	3600.41	<b>0.0000</b>	154.54	<b>0.0000</b>	1.37	<b>0.0000</b>	22.98
40.10	8	<b>99.3101</b> <sup>opt</sup>	0.0054	3600.59	11.9239	3600.41	<b>0.0000</b>	386.17	<b>0.0000</b>	1.53	<b>0.0000</b>	5.59
40.10	9	<b>99.3101</b> <sup>opt</sup>	0.0164	3600.58	11280.6645	3605.04	<b>0.0000</b>	308.45	<b>0.0000</b>	2.48	<b>0.0000</b>	12.02
50.3	2	<b>98.3191</b>	-	-	4192.4832	3601.41	38977.5437	3600.32	<b>0.0000</b>	1.59	<b>0.0000</b>	5.42
50.4	2	<b>97.9180</b>	-	-	40.6715	3601.28	177.0694	3600.36	<b>0.0000</b>	47.06	0.0246	41.06
50.4	3	<b>98.2281</b>	-	-	7002.4764	3601.35	600.6184	3600.31	<b>0.0000</b>	0.94	<b>0.0000</b>	4.05
50.5	2	<b>97.9350</b>	-	-	45.3059	3601.20	32.3729	3600.32	<b>0.0000</b>	34.41	<b>0.0000</b>	34.40
50.5	3	<b>98.2186</b>	-	-	8251.8046	3601.20	109.5518	3600.32	<b>0.0000</b>	12.94	<b>0.0000</b>	7.39
50.5	4	<b>98.2186</b>	-	-	8251.8046	3601.18	117.4691	3600.44	<b>0.0000</b>	0.75	<b>0.0000</b>	1.19
50.10	2	<b>99.4682</b>	-	-	47.5806	3601.15	1.9744	3600.29	<b>0.0000</b>	55.96	0.1454	27.84
50.10	3	<b>99.5099</b>	-	-	8361.6081	3601.14	36.4881	3600.18	<b>0.0000</b>	14.83	0.0209	0.80
50.10	4	<b>99.5099</b>	-	-	8361.6081	3601.14	0.8187	3600.84	<b>0.0000</b>	4.96	0.0209	0.66
50.10	5	<b>99.5099</b>	-	-	8361.6081	3601.21	73.1486	3600.26	<b>0.0000</b>	1.95	<b>0.0000</b>	5.54
50.10	6	<b>99.5099</b>	-	-	8361.6081	3601.16	83.3199	3600.36	<b>0.0000</b>	11.57	<b>0.0000</b>	5.39
50.10	7	<b>99.5099</b>	-	-	8361.6081	3601.12	77.1411	3600.19	<b>0.0000</b>	6.07	<b>0.0000</b>	50.90
50.10	8	<b>99.5099</b>	-	-	8361.6081	3601.14	82.0785	3600.37	<b>0.0000</b>	2.24	<b>0.0000</b>	5.88
50.10	9	<b>99.5099</b>	-	-	8361.6081	3601.15	75.8256	3600.23	<b>0.0000</b>	4.18	<b>0.0000</b>	5.70
Просек:		<b>98.7118</b>	-	-	3486.7232	2874.97	963.0016	1738.30	<b>0.0000</b>	<b>7.19</b>	0.0064	33.61

ГЛАВА 4. ПРОБЛЕМ  $p$ -ХАБ МАКСИМАЛНОГ ПОКРИВАЊА  
НЕОГРАНИЧЕНИХ КАПАЦИТЕТА СА  $r$ -АЛОКАЦИЈСКОМ ШЕМОМ

Табела 4.4: Експериментални резултати и поређења решења UгАрНМСР на АР  
инстанцама до 50 чворова - случај парцијалног покривања

		Парцијално покривање										
$n.p$	$r$	Optimal/Best value(%)	4-индексна ф.		3-индексна ф.		Општа ф.		GVNS-R		GRASP-VND	
			dev(%)	time(s)	dev(%)	time(s)	dev(%)	time(s)	dev(%)	time(s)	dev(%)	time(s)
25.3	2	<b>97.2157<sup>opt</sup></b>	<b>0.0000</b>	207.94	316.8106	3600.07	<b>0.0000</b>	269.64	<b>0.0000</b>	0.10	<b>0.0000</b>	0.16
25.4	2	<b>98.3824<sup>opt</sup></b>	<b>0.0000</b>	1393.83	23.0021	3600.07	<b>0.0000</b>	503.97	<b>0.0000</b>	16.63	<b>0.0000</b>	1.41
25.4	3	<b>98.6952<sup>opt</sup></b>	<b>0.0000</b>	766.83	30.5850	3600.07	<b>0.0000</b>	432.92	<b>0.0000</b>	0.52	<b>0.0000</b>	0.33
25.5	2	<b>98.9925<sup>opt</sup></b>	<b>0.0000</b>	3600.28	24.5929	3600.08	<b>0.0000</b>	540.51	<b>0.0000</b>	3.98	<b>0.0000</b>	0.05
25.5	3	<b>99.2295<sup>opt</sup></b>	<b>0.0000</b>	1134.58	46.8021	3600.05	<b>0.0000</b>	355.65	<b>0.0000</b>	0.83	<b>0.0000</b>	0.52
25.5	4	<b>99.2295<sup>opt</sup></b>	<b>0.0000</b>	1244.12	29.0224	3600.11	<b>0.0000</b>	331.90	<b>0.0000</b>	0.15	<b>0.0000</b>	0.71
25.10	2	<b>99.8213<sup>opt</sup></b>	<b>0.0000</b>	81.48	1.8883	3600.07	<b>0.0000</b>	52.90	<b>0.0000</b>	1.24	<b>0.0000</b>	10.28
25.10	3	<b>99.8213<sup>opt</sup></b>	<b>0.0000</b>	74.95	1.3930	3600.07	<b>0.0000</b>	51.87	<b>0.0000</b>	0.60	<b>0.0000</b>	2.11
25.10	4	<b>99.8213<sup>opt</sup></b>	<b>0.0000</b>	68.67	1.8556	3600.07	<b>0.0000</b>	46.92	<b>0.0000</b>	0.89	<b>0.0000</b>	1.11
25.10	5	<b>99.8213<sup>opt</sup></b>	<b>0.0000</b>	82.71	2.4474	3600.04	<b>0.0000</b>	50.28	<b>0.0000</b>	0.25	<b>0.0000</b>	0.97
25.10	6	<b>99.8213<sup>opt</sup></b>	<b>0.0000</b>	51.45	0.6285	3600.07	<b>0.0000</b>	48.95	<b>0.0000</b>	2.31	<b>0.0000</b>	1.01
25.10	7	<b>99.8213<sup>opt</sup></b>	<b>0.0000</b>	68.13	0.6418	3600.07	<b>0.0000</b>	39.59	<b>0.0000</b>	0.86	<b>0.0000</b>	1.42
25.10	8	<b>99.8213<sup>opt</sup></b>	<b>0.0000</b>	67.36	3.8815	3600.07	<b>0.0000</b>	39.83	<b>0.0000</b>	0.57	<b>0.0000</b>	0.22
25.10	9	<b>99.8213<sup>opt</sup></b>	<b>0.0000</b>	67.80	1.5376	3600.09	<b>0.0000</b>	39.00	<b>0.0000</b>	0.82	<b>0.0000</b>	0.17
40.3	2	<b>98.9775</b>	155.5027	3601.28	23.2809	3600.69	23.2044	3600.25	<b>0.0000</b>	3.19	<b>0.0000</b>	1.16
40.4	2	<b>98.2117</b>	79.9733	3600.92	31.0271	3600.00	28.6620	3600.31	<b>0.0000</b>	11.36	<b>0.0000</b>	5.73
40.4	3	<b>98.4510</b>	99.5513	3601.21	-	-	5091.4680	3600.31	<b>0.0000</b>	1.92	<b>0.0000</b>	2.84
40.5	2	<b>98.5227</b>	67.4288	3601.17	35.2688	3600.05	32.4634	3600.42	0.0539	38.12	<b>0.0000</b>	191.00
40.5	3	<b>98.7692</b>	117.2032	3601.24	96.6668	3600.02	3435.1731	3600.27	0.0107	15.41	<b>0.0000</b>	1.50
40.5	4	<b>98.8951</b>	112.1224	3601.19	60.9357	3609.22	10.7983	3600.31	<b>0.0000</b>	8.01	<b>0.0000</b>	1.56
40.10	2	<b>99.7134<sup>opt</sup></b>	<b>0.0000</b>	1918.16	36.9037	3611.11	<b>0.0000</b>	1896.59	0.0021	27.01	0.0136	259.48
40.10	3	<b>99.7134<sup>opt</sup></b>	92.4535	3601.82	62.4595	3619.06	<b>0.0000</b>	2750.40	0.0021	31.56	<b>0.0000</b>	765.36
40.10	4	<b>99.7134<sup>opt</sup></b>	<b>0.0000</b>	2312.54	75.5911	3604.26	<b>0.0000</b>	2559.14	<b>0.0000</b>	1.79	<b>0.0000</b>	48.47
40.10	5	<b>99.7134<sup>opt</sup></b>	<b>0.0000</b>	1717.49	59.6209	3611.94	0.0379	3608.91	<b>0.0000</b>	19.40	<b>0.0000</b>	164.31
40.10	6	<b>99.7134<sup>opt</sup></b>	0.0146	3604.76	86.9021	3613.28	<b>0.0000</b>	2558.72	0.0019	7.78	<b>0.0000</b>	104.90
40.10	7	<b>99.7134<sup>opt</sup></b>	<b>0.0000</b>	1955.23	70.2306	3605.49	<b>0.0000</b>	3181.47	<b>0.0000</b>	19.30	<b>0.0000</b>	4.70
40.10	8	<b>99.7134<sup>opt</sup></b>	<b>0.0000</b>	2350.45	74.7710	3611.49	<b>0.0000</b>	3007.06	<b>0.0000</b>	32.85	<b>0.0000</b>	65.46
40.10	9	<b>99.7134<sup>opt</sup></b>	<b>0.0000</b>	1987.10	74.1504	3619.41	1480.7451	3600.45	<b>0.0000</b>	6.27	<b>0.0000</b>	19.60
50.3	2	<b>99.2268</b>	-	-	23.3816	3601.38	23.3592	3600.62	<b>0.0000</b>	6.44	<b>0.0000</b>	4.59
50.4	2	<b>98.8445</b>	-	-	29.9091	3601.24	29.2232	3600.74	0.0070	37.64	<b>0.0000</b>	23.56
50.4	3	<b>99.0232</b>	-	-	-	-	8246.5273	3600.67	<b>0.0000</b>	11.25	<b>0.0000</b>	37.59
50.5	2	<b>98.7802</b>	-	-	34.4870	3601.22	32.8922	3600.63	<b>0.0000</b>	35.58	0.0328	34.18
50.5	3	<b>99.0145</b>	-	-	-	-	28.8169	3600.70	<b>0.0000</b>	41.07	0.0357	22.47
50.5	4	<b>99.0552</b>	-	-	-	-	5123.3284	3600.54	<b>0.0000</b>	9.55	<b>0.0000</b>	11.61
50.10	2	<b>99.7642</b>	-	-	35.8266	3601.20	26.2824	3600.60	<b>0.0000</b>	48.62	0.0401	15.20
50.10	3	<b>99.7575</b>	-	-	-	-	24.2623	3600.56	<b>0.0000</b>	48.23	<b>0.0000</b>	41.52
50.10	4	<b>99.7696</b>	-	-	-	-	22.9394	3600.56	<b>0.0000</b>	9.09	0.0157	12.38
50.10	5	<b>99.7696</b>	-	-	-	-	23.1838	3600.63	<b>0.0000</b>	4.83	<b>0.0000</b>	7.02
50.10	6	<b>99.7630</b>	-	-	-	-	2150.2594	3600.74	<b>0.0000</b>	47.09	0.0077	23.41
50.10	7	<b>99.7696</b>	-	-	240.2561	3601.20	2150.4083	3600.64	<b>0.0000</b>	47.70	0.0121	4.93
50.10	8	<b>99.7696</b>	-	-	151.9115	3601.22	2150.4083	3600.75	<b>0.0000</b>	13.22	0.0110	27.37
50.10	9	<b>99.7696</b>	-	-	139.3694	3600.95	2150.4083	3600.72	<b>0.0000</b>	8.61	0.0110	25.03
<b>Процек:</b>		<b>99.3316</b>	-	-	-	3602.97	768.6869	2332.80	<b>0.0019</b>	<b>14.82</b>	0.0043	46.37

ГЛАВА 4. ПРОБЛЕМ  $p$ -ХАБ МАКСИМАЛНОГ ПОКРИВАЊА  
НЕОГРАНИЧЕНИХ КАПАЦИТЕТА СА  $r$ -АЛОКАЦИЈСКОМ ШЕМОМ

Табела 4.5: GVNS и GRASP-VND резултати за UrApHMCP на AP инстанцама великих димензија - бинарно и парцијално покривање

$n, p$	$r$	Бинарно покривање				Парцијално покривање			
		GVNS-R		GRASP-VND		GVNS-R		GRASP-VND	
		Best value(%)	time(s)	Best value(%)	time(s)	Best value(%)	time(s)	Best value(%)	time(s)
100.3	2	<b>98.2054</b>	45.42	98.1204	94.55	<b>99.1643</b>	101.38	99.1264	31.67
100.4	2	<b>97.7337</b>	16.72	97.7148	89.14	<b>98.8726</b>	72.03	98.8068	86.29
100.4	3	<b>98.1171</b>	100.87	98.1085	56.83	<b>99.0608</b>	50.14	<b>99.0608</b>	45.84
100.5	2	<b>98.1498</b>	69.55	97.9909	113.67	<b>99.1346</b>	74.71	98.8851	32.91
100.5	3	<b>98.5551</b>	68.50	98.4629	89.01	<b>99.2711</b>	68.55	99.2624	75.90
100.5	4	<b>98.6464</b>	9.20	98.6039	29.87	<b>99.3913</b>	46.88	99.3893	54.67
200.3	2	<b>99.0192</b>	98.73	98.9908	200.41	<b>99.5715</b>	120.47	99.5094	93.03
200.4	2	<b>98.7340</b>	142.85	98.6059	116.66	99.3247	88.20	<b>99.3576</b>	93.61
200.4	3	<b>98.8756</b>	167.23	98.8588	69.84	99.4631	185.64	<b>99.5081</b>	41.95
200.5	2	<b>98.6068</b>	199.33	98.5789	369.43	<b>99.2931</b>	280.81	99.2680	217.54
200.5	3	98.8101	174.03	<b>98.8223</b>	214.52	99.4547	218.76	<b>99.4736</b>	171.89
200.5	4	<b>98.8727</b>	191.16	98.8583	243.95	<b>99.4888</b>	108.80	99.4555	148.94
<b>Просек:</b>		<b>98.5272</b>	<b>106.97</b>	98.4764	140.66	<b>99.2909</b>	118.03	99.2586	<b>91.19</b>

циља најбољих решења су 98.5272% за GVNS-R и 98.4764% за GRASP-VND. Слично, у случају парцијалног покривања, просечне вредности функције циља најбољих решења су 99.2909% за GVNS-R и 99.2586% за GRASP-VND. GRASP-VND методом се добија боље решење у односу на GVNS-R методу за једну инстанцу у случају бинарног покривања ( $n = 200, p = 5, r = 3$ ) и за три инстанце у случају парцијалног покривања ( $n = 200, p = 4, r = 2; n = 200, p = 4, r = 3; n = 200, p = 5, r = 3$ ). Када се упореде просечна времена извршавања алгоритама на овим инстанцама, може се закључити да је GVNS-R бржи од GRASP-VND у случају бинарног покривања: просечна времена извршавања су 106.97 секунди за GVNS и 140.66 секунди за GRASP-VND. У случају парцијалног покривања, GRASP-VND је ефикаснији од GVNS-R, јер су просечна времена извршавања 91.19 секунди за GRASP-VND и 118.03 секунди за GVNS.

Коначно, у Табелама 4.6-4.7 су приказани резултати тестираних USA423 инстанци, које до сада нису разматране у литератури за  $p$ -хаб проблеме максималног покривања. Генерално, може се приметити да за оба концепта покривања GVNS-R даје решења бољег квалитета у поређењу са GRASP-VND методом, док GRASP-VND показује значајну предност над GVNS-R у погледу времена извршавања. Међутим, за USA423 инстанце I типа, GRASP-VND успева да побољша најбоља решења добијена GVNS-R методом на 3 инстанце у случају бинарног покривања и 3 инстанце, у случају парцијалног покривања. За овај скуп тест инстанци и случај бинарног покривања, просечне вредности функције циља најбољих GVNS-R и GRASP-VND решења су 99.1434%, односно 98.0659%.



ГЛАВА 4. ПРОБЛЕМ  $p$ -ХАБ МАКСИМАЛНОГ ПОКРИВАЊА  
НЕОГРАНИЧЕНИХ КАПАЦИТЕТА СА  $r$ -АЛОКАЦИЈСКОМ ШЕМОМ

Табела 4.6: GVNS и GRASP-VND резултати за УгАрНМСП на USA423 инстанцама I типа - бинарно и парцијално покривање

$p$	$r$	Бинарно покривање				Парцијално покривање			
		GVNS-R		GRASP-VND		GVNS-R		GRASP-VND	
		Best value(%)	time(s)	Best value(%)	time(s)	Best value(%)	time(s)	Best value(%)	time(s)
3	2	<b>99.7942</b>	249.96	99.1625	2046.06	<b>99.8060</b>	796.25	99.6293	931.16
4	2	<b>98.7876</b>	1134.80	98.4588	3684.18	<b>99.4009</b>	1214.04	91.1560	3609.47
4	3	<b>99.7143</b>	2099.97	99.5684	2778.94	<b>99.4528</b>	998.58	98.8994	2247.97
5	2	<b>99.3089</b>	2894.58	98.2507	5193.65	<b>99.7796</b>	2983.60	99.5722	5082.38
5	3	<b>98.0659</b>	4635.24	97.7419	1170.50	<b>99.6290</b>	4910.82	99.0448	1110.80
5	4	<b>99.6738</b>	1207.65	99.4765	2132.17	99.7806	1652.36	<b>99.7887</b>	1390.90
6	2	<b>99.5969</b>	7695.62	95.1557	5850.34	<b>99.7495</b>	8236.95	97.0201	7800.70
6	3	<b>99.3557</b>	3787.20	98.0516	2560.35	<b>99.6835</b>	3942.97	95.7067	3086.64
6	4	<b>99.2433</b>	5186.58	99.2294	7930.44	<b>99.7184</b>	4005.20	99.6525	8196.86
6	5	97.0808	3219.88	<b>98.9667</b>	1949.29	98.9875	1735.80	<b>99.4767</b>	2179.00
7	2	<b>99.6943</b>	7338.16	92.0994	2289.57	<b>99.2332</b>	3812.72	96.1341	2599.94
7	3	<b>98.5678</b>	7165.71	96.8236	1860.81	<b>99.7398</b>	8249.81	98.3082	1253.25
7	4	<b>99.6038</b>	6681.60	98.7418	2165.39	<b>99.7600</b>	6603.31	99.1558	4900.32
7	5	99.4726	3528.60	<b>99.7600</b>	804.93	<b>99.7942</b>	5058.27	99.7925	1112.63
7	6	99.1904	5259.29	<b>99.5012</b>	747.55	99.5879	1902.61	<b>99.6873</b>	2389.31
<b>Просек:</b>		<b>99.1434</b>	4138.99	98.0659	<b>2877.61</b>	<b>99.6069</b>	3740.22	98.2016	<b>3192.76</b>

Табела 4.7: GVNS и GRASP-VND резултати за УгАрНМСП на USA423 инстанцама II типа - бинарно и парцијално покривање

$p$	$r$	Бинарно покривање				Парцијано покривање			
		GVNS-R		GRASP-VND		GVNS-R		GRASP-VND	
		Best value(%)	time(s)	Best value(%)	time(s)	Best value(%)	time(s)	Best value(%)	time(s)
3	2	<b>99.7982</b>	179.13	99.2223	1711.34	<b>99.8060</b>	723.89	99.8032	508.84
4	2	<b>99.8053</b>	485.87	98.3746	4074.08	<b>99.7765</b>	617.41	98.8739	1463.35
4	3	99.4100	1184.06	<b>99.7357</b>	1523.58	99.7803	1428.63	<b>99.7877</b>	2329.77
5	2	<b>99.8058</b>	1174.77	99.7465	4276.56	<b>99.8060</b>	1259.29	99.8046	4293.10
5	3	98.3045	4634.92	<b>99.3379</b>	4538.44	<b>99.7180</b>	4224.87	99.6701	1248.29
5	4	<b>99.5565</b>	1586.81	98.8268	693.60	<b>99.7378</b>	1771.23	99.7331	1314.93
6	2	<b>99.7953</b>	3936.68	99.7941	1639.27	99.7122	3777.99	<b>99.8954</b>	5573.56
6	3	<b>98.8793</b>	2644.66	95.2078	3736.52	<b>99.7860</b>	2687.98	99.6969	8014.76
6	4	99.1245	8464.6	<b>99.4527</b>	3109.99	99.4981	4995.40	<b>99.6950</b>	3270.70
6	5	99.0598	2298.57	<b>99.4493</b>	1654.89	99.5551	1409.90	<b>99.6906</b>	1656.22
7	2	<b>99.7816</b>	5879.37	95.3338	4653.76	99.7434	7341.69	<b>99.7962</b>	8789.88
7	3	<b>99.7381</b>	6765.22	99.3185	2787.41	<b>99.8011</b>	5228.07	99.6399	9550.07
7	4	<b>99.3703</b>	7493.38	96.5145	5490.29	<b>99.6927</b>	9453.83	99.4870	1666.76
7	5	<b>99.7913</b>	9920.24	99.7643	2375.68	<b>99.8020</b>	10138.11	99.7935	2414.50
7	6	99.5091	2784.81	<b>99.7600</b>	1184.79	<b>99.7926</b>	2611.60	99.7759	1985.68
<b>Просек:</b>		<b>99.4486</b>	3962.21	98.6559	<b>2896.68</b>	<b>99.7339</b>	3844.66	99.6762	<b>3605.36</b>

У случају парцијалног покривања, просечне вредности функције циља најбољих GVNS-R и GRASP-VND решења су редом 99.6069% и 98.2016%.

За USA423 инстанце II типа, GRASP-VND је побољшао решења добијена GVNS-R методом на 5 инстанци у случају бинарног покривања и 5 инстанци у случају парцијалног покривања. За овај скуп тест инстанци и случај бинарног покривања, просечне вредности функције циља најбољих GVNS-R и GRASP-VND решења су 99.4486%, односно 98.6559%. У случају парцијалног покривања, просечне вредности функције циља најбољих GVNS-R и GRASP-VND решења су редом 99.7339% и 99.6762%. На USA423 инстанцама I типа, GRASP-VND метода је у просеку око 1.4 пута бржа у поређењу са GVNS-R методом у случају бинарног покривања и око 1.2 пута бржа у случају парцијалног покривања. На USA423 инстанцама II типа, у просеку, GRASP-VND обезбеђује своја најбоља решења за око 1.4 пута краће време у односу на GVNS-R методу, у случају бинарног покривања и за око 1.1 пута краће време, у случају парцијалног покривања.

## Резултати генетског алгорита за решавање UrApHMCP

Предложени GA није показао добре перформансе за решавање UrApHMCP у случају парцијалног покривања, тако да су у овом одељку приказани само добијени GA резултати на САВ и АР инстанцама до 200 чворова, у случају бинарног покривања.

Број јединки у популацији је  $N_{ind} = 150$ . У свакој итерацији, 30% најбољих јединки из тренутне популације директно прелазе у наредну. Односно,  $N_{el} = 45$ . Остатак популације је генерисан применом генетских оператора на скуп неелитних јединки са  $N_{nonel} = 105$  чланова. При решавању већине хаб локацијских проблема показано је да FGTS даје најбоље резултате за  $F_{tour} = 5.4$  (видети [98, 135, 137]). Ова вредност средње величине турнира је коришћена и при тестирању предложеног GA. FGTS оператор је примењен на 105 неелитних јединки, са  $k_1 = 63$  одржаних турнира величине 5 и  $k_2 = 42$  одржана турнира величине 6. На селектовани пар јединки-родитеља се примењује оператор укрштања, стварајући једну јединку-потомка, која подлеже оператору мутације. Ген јединке-потомка се мутира са вероватноћом  $p_{mut} = 0.03$ . Максималан број GA итерација без побољшања решења је постављен на 300, док је укупан број итерација 1000. GA је извршен 10 пута на свакој тестираној инстанци.

У Табелама 4.8 – 4.9 су приказани резултати тестиране ГА методе на инстанцама малих димензија ( $n = 25, 40$ ), а у Табелама 4.10 – 4.11 су приказани резултати добијени на инстанцама средњих и великих димензија ( $n = 50, 100, 200$ ). Прве три колоне Табела 4.8 – 4.9 садрже број чворова, број лоцираних хабова ( $n.p$ ) и вредност параметара  $r$  и  $\alpha$ . Следећа колона садржи вредности функције циља које одговарају оптималним решењима добијеним CPLEX решавачем са Четвороиндексном формулацијом. Вредности функције циља у колони *Opt.value* су дате као проценат укупног покривеног протока у оптималном решењу или најбољем решењу, уколико CPLEX није нашао оптимална решења у максималном временском ограничењу од 1h. У другом случају, ознака \* се налази уз одговарајућу вредност функције циља допустивог решења у датој колони. Одговарајуће време изражено у секундама, за које CPLEX даје оптимално или допустиво решење, је приказано у колони  $t(s)$ . Колона *Best GA value* садржи вредности функције циља које одговарају најбољим решењима добијеним генетским алгоритмом ГА, а изражене као проценат укупног покривеног протока. У наредној колони, означеној са  $t_{tot}$ , приказана су просечна укупна времена извршавања ГА у 10 покретања. Колона означена са *gen.avg* садржи просечан број ГА генерација. Квалитет решења у свих 10 покретања алгоритма је процењен помоћу вредности  $agap = \frac{1}{10} \sum_{i=1}^{10} gap_i$ , где је  $gap_i$  процентуално одступање вредности функције циља решења од вредности која одговара оптималном или најбољем познатом решењу. У последњој колони су приказане стандардне девијације просечног одступања, у ознаци  $\sigma$ .

Из резултата приказаних у Табели 4.8, може се закључити да CPLEX решавач за 3 од 24 тестиране САВ инстанце (САВ25.4 за  $r = 2, \alpha = 0.4$ , САВ25.5 за  $r = 2, \alpha = 0.6$  и САВ25.4 за  $r = 2, \alpha = 0.8$ ) добија допустива решења, за које оптималност није доказана у датом времену извршавања од 1h. Предложени ГА достиже сва позната оптимална решења и побољшава доње границе добијене CPLEX решавачем, за три инстанце које нису решене до оптималности. Просечно време извршавања CPLEX решавача на инстанцама које су решене до оптималности је 1606.21 секунди, док је одговарајуће просечно време извршавања генетског алгоритма 0.21 секунда. За преостале 3 инстанце, предложени ГА налази најбоља решења за 0.12 секунде, у просеку.

ГЛАВА 4. ПРОБЛЕМ  $p$ -ХАБ МАКСИМАЛНОГ ПОКРИВАЊА  
НЕОГРАНИЧЕНИХ КАПАЦИТЕТА СА  $r$ -АЛОКАЦИЈСКОМ ШЕМОМ

Табела 4.8: Експериментални резултати и поређења решења UrApНМСР на САВ инстанцама - бинарно покривање

n.p	r	$\alpha$	CPLEX		GA				
			Opt. value(%)	t(s)	Best GA value(%)	$t_{tot}(s)$	gen.avg	agap(%)	$\sigma$ (%)
25.3	2	0.2	96.58	1805.86	96.58	0.08	303.10	0.00	0.00
25.4	2	0.2	95.71	1541.47	95.71	0.13	330.20	0.00	0.00
25.4	3	0.2	95.71	2100.11	95.71	0.16	331.50	0.00	0.00
25.5	2	0.2	92.70	2087.15	92.70	0.18	420.20	0.00	0.00
25.5	3	0.2	92.70	2849.29	92.70	0.25	385.20	0.00	0.00
25.5	4	0.2	92.70	2431.20	92.70	0.31	409.20	0.00	0.00
25.3	2	0.4	96.24	2305.22	96.24	0.76	300.80	0.00	0.00
25.4	2	0.4	93.80*	3600.09	<b>95.01</b>	0.10	302.70	0.00	0.00
25.4	3	0.4	95.01	2049.70	95.01	0.14	303.50	0.00	0.00
25.5	2	0.4	91.45	1267.85	91.45	0.15	340.60	0.25	0.25
25.5	3	0.4	91.84	1105.92	91.84	0.25	367.50	0.00	0.00
25.5	4	0.4	91.84	1110.36	91.84	0.28	351.50	0.00	0.00
25.3	2	0.6	93.17	354.23	93.17	0.08	300.70	0.00	0.00
25.4	2	0.6	93.62	1055.33	93.62	0.11	318.80	0.00	0.00
25.4	3	0.6	93.63	831.95	93.63	0.13	320.50	0.01	0.01
25.5	2	0.6	90.00*	3600.05	<b>90.12</b>	0.14	314.10	0.00	0.00
25.5	3	0.6	90.19	2727.55	90.19	0.23	342.30	0.05	0.04
25.5	4	0.6	90.19	3531.83	90.19	0.26	353.50	0.00	0.00
25.3	2	0.8	90.08	682.41	90.08	0.08	303.30	0.00	0.00
25.4	2	0.8	89.02*	3600.07	<b>89.60</b>	0.11	311.00	0.00	0.00
25.4	3	0.8	89.61	1829.71	89.61	0.15	322.20	0.00	0.01
25.5	2	0.8	89.03	728.15	89.03	0.16	329.00	0.29	0.28
25.5	3	0.8	89.05	605.09	89.05	0.24	350.90	0.00	0.00
25.5	4	0.8	89.05	730.11	89.05	0.27	345.30	0.00	0.00
<b>Просек:</b>			92.20	1855.45	<b>92.22</b>	0.20	335.73	0.03	0.02

Резултати предложеног GA на AP инстанцама са 25 чворова и AP инстанцама са 40 чворова су приказани у Табели 4.9. CPLEX решавач је за 4 од 6 тестираних AP инстанци са 25 чворова и за све тестиране AP инстанце са 40 чворова успео да добије само допустива решења, за која оптималност није доказана. Са друге стране, за AP инстанце са 25 чворова, GA достиже сва позната оптимална решења и побољшава доње границе добијене CPLEX решавачем на инстанцама које нису решене до оптималности. Просечно време извршавања CPLEX решавача на инстанцама које су решене до оптималности је 1611.4 секунде, док је одговарајуће просечно време извршавања GA 0.19 секунди, у просеку. На свим AP инстанцама са 40 чворова, GA побољшава доње границе добијене CPLEX решавачем за просечно време извршавања од 0.85 секунди.

ГЛАВА 4. ПРОБЛЕМ  $p$ -ХАБ МАКСИМАЛНОГ ПОКРИВАЊА  
НЕОГРАНИЧЕНИХ КАПАЦИТЕТА СА  $r$ -АЛОКАЦИЈСКОМ ШЕМОМ

Табела 4.9: Експериментални резултати и поређења решења UrArНМСР на AP инстанцама са 25 и 40 чворова - бинарно покривање

$n.p$	$r$	$\alpha$	CPLEX		GA				
			Opt. value(%)	t(s)	Best GA value(%)	$t_{tot}(s)$	gen.avg	agap(%)	$\sigma$ (%)
25.3	2	0.75	95.13	765.58	95.13	0.08	302.00	0.15	0.15
25.4	2	0.75	96.94*	3000.50	96.94	0.12	306.50	0.00	0.00
25.4	3	0.75	97.16	1068.12	97.16	0.16	305.50	0.05	0.05
25.5	2	0.75	97.47*	3600.07	<b>97.87</b>	0.17	338.50	0.00	0.00
25.5	3	0.75	98.03*	3600.10	98.03	0.28	394.10	0.00	0.00
25.5	4	0.75	98.06*	3600.09	98.06	0.35	389.80	0.12	0.20
40.3	2	0.75	84.25*	3600.36	<b>97.77</b>	0.22	309.10	0.00	0.00
40.4	2	0.75	26.92*	3600.35	<b>96.85</b>	0.42	356.00	0.03	0.04
40.4	3	0.75	28.64*	3600.36	<b>97.22</b>	0.70	392.60	0.00	0.00
40.5	2	0.75	38.30*	3600.36	<b>97.04</b>	0.60	376.60	0.00	0.00
40.5	3	0.75	38.05*	3600.45	<b>97.25</b>	1.31	483.70	0.16	0.09
40.5	4	0.75	36.04*	3600.35	<b>97.44</b>	1.84	516.90	0.10	0.05
<b>Просек:</b>			69.58	3103.06	<b>97.23</b>	0.52	372.61	0.05	0.05

AP инстанце већих димензија са 50, 100 и 200 чворова превазилазе могућности CPLEX решавача при решавању разматраног UrArНМСР. Због ограничења меморије, CPLEX није у могућности да пронађе чак ни допуштена решења за максимално време од 1h. Из тог разлога, четврта колона Табела 4.10 – 4.11, означена са *Best-known value*, садржи вредности функције циља које одговарају најбољим познатим решењима, изражене као проценат укупног покривеног протока. У колони означеној са *GAP* су приказана процентуална одступања вредности функције циља GA решења од вредности које одговарају најбољим познатим решењима. Може се приметити да GA није успео да достигне најбоље познато решење ни на једној од тестираних инстанци већих димензија. У просеку, за достизање најбољих решења у 10 покретања на AP инстанцама са 50 чворова, GA је потрошио 20.26 секунди, а на AP инстанцама великих димензија са 100 и 200 чворова, 70.36 секунди.

ГЛАВА 4. ПРОБЛЕМ  $p$ -ХАБ МАКСИМАЛНОГ ПОКРИВАЊА  
НЕОГРАНИЧЕНИХ КАПАЦИТЕТА СА  $r$ -АЛОКАЦИЈСКОМ ШЕМОМ

Табела 4.10: GA резултати за UrArHMCP на AP инстанцама са 50 чворова - бинарно покривање

n.p	r	$\alpha$	Best-known value (%)	GA					
				Best GA value (%)	$t_{tot}(s)$	gen.avg	agap(%)	$\sigma(\%)$	GAP (%)
50.3	2	0.75	<b>98.32</b>	97.86	0.44	333.60	0.14	0.05	0.47
50.4	2	0.75	<b>97.92</b>	97.53	0.80	333.90	0.07	0.07	0.40
50.4	3	0.75	<b>98.23</b>	98.05	1.56	446.00	0.02	0.01	0.18
50.5	2	0.75	<b>97.94</b>	97.66	1.40	514.50	0.06	0.02	0.29
50.5	3	0.75	<b>98.22</b>	97.89	2.87	528.70	0.06	0.05	0.34
50.5	4	0.75	<b>98.22</b>	97.90	5.32	488.50	0.03	0.04	0.33
50.10	2	0.75	<b>99.47</b>	99.20	3.60	740.10	0.02	0.03	0.27
50.10	3	0.75	<b>99.51</b>	99.27	9.73	802.20	0.04	0.04	0.24
50.10	4	0.75	<b>99.51</b>	99.29	18.76	778.80	0.02	0.01	0.22
50.10	5	0.75	<b>99.51</b>	99.32	30.47	787.70	0.03	0.02	0.19
50.10	6	0.75	<b>99.51</b>	91.11	43.41	773.80	0.04	0.02	8.44
50.10	7	0.75	<b>99.51</b>	91.08	80.06	831.70	0.02	0.01	8.47
50.10	8	0.75	<b>99.51</b>	99.35	64.92	809.80	0.02	0.02	0.16
<b>Просек:</b>			<b>98.88</b>	97.35	20.26	628.41	0.05	0.03	1.54

Табела 4.11: GA резултати за UrArHMCP на AP инстанцама великих димензија - бинарно покривање

n.p	r	$\alpha$	Best-known value (%)	GA					
				Best GA value (%)	$t_{tot}(s)$	gen.avg	agap(%)	$\sigma(\%)$	GAP (%)
100.3	2	0.75	<b>98.21</b>	97.74	3.76	365.33	0.02	0.03	0.48
100.4	2	0.75	<b>97.73</b>	97.39	10.54	456.67	0.05	0.04	0.35
100.4	3	0.75	<b>98.12</b>	97.85	19.71	410.67	0.03	0.05	0.28
100.5	2	0.75	<b>98.15</b>	98.03	15.94	610.67	0.00	0.00	0.12
100.5	3	0.75	<b>98.56</b>	98.35	36.49	720.00	0.00	0.00	0.21
100.5	4	0.75	<b>98.65</b>	98.15	51.45	494.00	0.09	0.09	0.51
200.3	2	0.75	<b>99.02</b>	98.83	56.50	603.50	0.01	0.01	0.19
200.4	2	0.75	<b>98.73</b>	98.62	97.47	703.00	0.00	0.00	0.11
200.4	3	0.75	<b>98.88</b>	98.81	207.09	631.50	0.01	0.01	0.07
200.5	2	0.75	<b>98.61</b>	98.54	204.69	825.00	0.05	0.05	0.07
200.5	3	0.75	<b>98.82</b>	98.67	458.91	501.00	0.02	0.02	0.15
<b>Просек:</b>			<b>98.50</b>	98.23	70.36	574.67	0.03	0.03	0.23

У циљу лакшег поређења резултата предложених GVNS-R, GRASP-VND и GA метода за решавање UrArHMCP у случају бинарног покривања, у Табели 4.12 су приказане просечне вредности функције циља које одговарају најбољим познатим вредностима добијеним предложеним методама на AP инстанцама са 50, 100 и 200 чворова. Како су GVNS-R и GRASP-VND методе извршаване само једном на свакој тестираној инстанци, а GA десет пута, да би поређење резултата било коректно, у колонама  $t_{max}(s)$  су наведена максимална дозвољена времена извршавања GVNS-R и GRASP-VND метода, док су у колони  $time(s)$  приказане просечне вредности за укупна времена извршавања GA, у свих 10 покретања (изражене у секундама).

ГЛАВА 4. ПРОБЛЕМ  $p$ -ХАБ МАКСИМАЛНОГ ПОКРИВАЊА  
НЕОГРАНИЧЕНИХ КАПАЦИТЕТА СА  $r$ -АЛОКАЦИЈСКОМ ШЕМОМ

Може се приметити да у погледу квалитета решења, GVNS-R представља најбољи, а GA најлошији приступ при решавању UrApHMCP у случају бинарног покривања. Поред тога, GA је утрошио највише времена за налажење решења најлошијег квалитета. Прецизније, за достизање најбољих вредности функције циља на AP инстанцама са 50, 100 и 200 чворова, GVNS-R и GRASP-VND методама је потребно највише редом 50, 100 и 200 секунди, док је GA утрошио редом 202.6, 229.8 и 2049.3 секунде.

Табела 4.12: Поређења резултата предложених GVNS-R, GRASP и GA метода за UrApHMCP на AP инстанцама - случај бинарног покривања

$n$	GVNS-R		GRASP-VND		GA	
	Av.value(%)	$t_{max}(s)$	Av.value(%)	$t_{max}(s)$	Av.value(%)	time(s)
50	<b>98.8742</b>	50	98.8581	50	97.3500	202.6
100	<b>98.2346</b>	100	98.1669	100	97.9196	229.8
200	<b>98.8091</b>	200	98.7713	200	98.6938	2049.3

## Глава 5

### Закључак

У овој докторској дисертацији су разматране три варијанте проблема хаба максималног покривања: проблем  $p$ -хаб максималног покривања неограничених капацитета са једноструким алокацијама (USApHMCP), проблем  $p$ -хаб максималног покривања неограничених капацитета са вишеструким алокацијама (UMApHMCP) и проблем  $p$ -хаб максималног покривања неограничених капацитета са  $r$ -алокацијском шемом (UrApHMCP). За сва три проблема анализирана су оба концепта покривања - бинарно и парцијално. UrApHMCP није разматран у досадашњој литератури.

Постигнута су два главна научна циља. Први је развој нових математичких модела за USApHMCP и UMapHMCP са бинарним и парцијалним покривањем. Такође, први пут у литератури је формулисан UrApHMCP, који представља генерализацију USApHMCP и UMapHMCP. За овај проблем су предложена три математичка модела са бинарним и парцијалним покривањем. Нови математички модели за USApHMCP и UMapHMCP су упоређени са моделима Рекега и Каге из [121], који су у досадашњој литератури показали најбоље перформансе, а могу се применити на оба концепта покривања. Модели су упоређени у смислу ефикасности коришћењем егзактног решавача CPLEX 12.6. Екпериментални резултати показују да су нови модели за UMapHMCP у оба случаја покривања бољи од модела предложених у [121], тј. омогућавају егзактном решавачу добијање већег броја оптималних решења на тестираним инстанцама, за краће просечно време извршавања. Са друге стране, нови модели за USApHMCP и оба концепта покривања показују лошије перформансе у поређењу са моделима из [121]. Међутим, главна предност нових модела проблема  $p$ -хаб максималног покривања са бинарним и парцијалним покривањем и једноструким алокаци-



јама се огледа у чињеници да нови модели обједињују моделе за проблеме  $p$ -хаб максималног покривања са различитим алокацијским шемама. Нови модели се малим модификацијама лако могу трансформисати у одговарајуће моделе за сва три разматрана проблема. Из тог разлога, у дисертацији су предложени модели за USApHMCP, UMapHMCP и UrapHMCP, у случају бинарног и парцијалног покривања, названи *Општим моделима*. У случају UrapHMCP са оба концепта покривања, поред Општег модела, предложена су још два модела, Четвороиндексни и Троиндексни. Сва три модела за UrapHMCP су упоређена у смислу ефикасности коришћењем егзактног решавача CPLEX 12.6. Добијени резултати показују да је Општи модел ефикаснији у односу на остала два у оба случаја покривања, јер даје решења бољег квалитета за краће време извршавања.

Сва три разматрана проблема хаб максималног покривања су NP-тешки проблеми комбинаторне оптимизације, тако да егзактне методе дају оптимална решења само за тест примере малих димензија. Из тог разлога, развијене су и имплементирани метахеуристичке методе за решавање разматраних проблема, што представља други научни циљ дисертације. Предложене су три метахеуристике засноване на методи променљивих околина (VNS): општа VNS метода (GVNS) за решавање проблема са једноструким алокацијама, основна VNS метода (BVNS) за решавање проблема са вишеструким алокацијама и нова варијанта опште VNS методе (GVNS-R) за решавање проблема са  $r$ -алокацијском шемом. У случају UrapHMCP, дизајниране су још две метахеуристике: похлепна стохастичко-адаптивна процедура претраге са методом променљивог спуста (GRASP-VND) и генетски алгоритам (GA). Конструктивни елементи и параметри метахеуристичких метода су прилагођени карактеристикама разматраних проблема, тако да предложене метахеуристике достижу оптимална или субоптимална решења (уколико оптимална нису позната), у кратком времену извршавања.

Развијене GVNS и BVNS метахеуристике за решавање USApHMCP и UMapHMCP су тестиране на стандардним CAB и AP хаб инстанцама до 200 чворова, али и на URAND инстанцама до 1000 чворова, које представљају прави изазов за истраживаче. Скупови AP и URAND инстанци су у овој дисертацији први пут тестирани за USApHMCP и UMapHMCP са бинарним и парцијалним покривањем. Изложени експериментални резултати показују да су обе хеуристике успешне при решавању разматраних проблема, тј. успевају да про-

нају сва позната оптимална решења или добију нова најбоља решења. Чак и у случајевима решавања инстанци проблема великих димензија, предложене хеуристике достижу најбоља решења за кратко време извршавања.

GVNS-R и GRASP-VND метахеуристике за решавање U $r$ ArHMCP на CAB и AP инстанцама до 50 чворова достижу позната оптимална решења претходно добијена егзактним решавачем или успостављају нова најбоља позната решења, иначе (осим у случају неколико инстанци проблема при решавању GRASP-VND методом). На AP инстанцама великих димензија са 100 и 200 чворова, GVNS-R и GRASP-VND методе показују сличне перформансе, јер обе дају решења високог квалитета за кратко време извршавања. У просеку, GVNS-R метода се показала успешнијом у поређењу са GRASP-VND методом када је у питању квалитет решења добијених на великим AP инстанцама (у оба случаја покривања), док GRASP-VND метода има значајно краће време извршавања у случају парцијалног покривања. На USA423 инстанцама, које до сада нису разматране у литератури за  $p$ -хаб проблеме максималног покривања, GVNS-R даје решења бољег квалитета у поређењу са GRASP-VND методом, али GRASP-VND надмашује GVNS-R у погледу времена извршавања. Иако генерално GVNS-R даје решења бољег квалитета од GRASP-VND методе за све разматране скупове тест инстанци, GRASP-VND успева да побољша најбоља решења добијена GVNS-R методом на неколико инстанци из сваког скупа. Предложени GA за решавање U $r$ ArHMCP није показао добре перформансе у случају парцијалног покривања, тако да су у овој дисертацији приказани само добијени GA резултати у случају бинарног покривања. На CAB и AP инстанцама до 200 чворова, предложени GA је показао практичност и ефикасност у погледу квалитета решења, али и времена извршавања.

Научни допринос ове дисертације чине следећи резултати:

- Формулисање нових математичких модела за проблеме  $p$ -хаб максималног покривања неограничених капацитета са једноструким и вишеструким алокацијама (U $r$ ArHMCP и U $M$ ArHMCP) и оба концепта покривања - бинарно и парцијално;
- Формулисање проблема  $p$ -хаб максималног покривања неограничених капацитета са  $r$ -алокацијском шемом (U $r$ ArHMCP), који до сада није разматран у литератури;

- Креирање математичких модела за  $U_rArHMSP$  за оба типа покривања - бинарно и парцијално;
- Упоредивање познатих и нових формулација за  $USArHMSP$  и  $UMArHMSP$  у смислу ефикасности;
- Упоредивање предложених формулација за  $U_rArHMSP$  у смислу ефикасности;
- Испитивање домета егзактних алгоритама имплементираних у оквиру решавача CPLEX 12.6 за добијање оптималних решења или њихових горњих граница на инстанцама проблема различитих димензија;
- Имплементација више варијанти методе променљивих околина, хибридне GRASP методе и генетског алгорита, прилагођених разматраним проблемима и примењених на оригиналан начин;
- Анализа резултата различитих варијанти VNS, GRASP-VND и GA метахеуристика имплементираних на хаб инстанцама из литературе, малих, средњих и великих димензија;
- Поређење добијених резултата различитих варијанти VNS, GRASP-VND и GA метода на тестираним инстанцама, као и поређење са оптималним и субоптималним решењима добијеним егзактним решавачем CPLEX 12.6.

Неки од резултата приказаних у овој докторској дисертацији публиковани су у радовима [83–85] и представљају научни допринос областима оптимизације, математичког моделирања, хаб локацијских проблема и метахеуристичких метода.

Анализирајући изложене резултате, може се закључити да предложене метахеуристике успешно решавају проблеме  $p$ -хаб максималног покривања са различитим алокацијским шемама. Будућа истраживања могу укључити паралелизацију и хибридизацију предложених метода са другим хеуристикама за решавање ових или сличних хаб локацијских проблема. Планирана су и проширења модела проблема  $p$ -хаб максималног покривања која укључују капацитете хабова или грана, фиксне трошкове за успостављање хабова и друге карактеристике, у циљу бољег одсликавања ситуација из праксе.

# Библиографија

- [1] S. Abdinnour-Helm and M.A. Venkataramanan. Solution approaches to hub location problems. *Annals of Operations Research*, 78:31–50, 1998.
- [2] B. Adenso-Diaz and F. Rodriguez. A simple search heuristic for the MCLP: Application to the location of ambulance bases in a rural region. *Omega*, 25(2):181–187, 1997.
- [3] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network flows: theory, algorithms, and applications*. Prentice Hall, New Jersey, USA, 1993.
- [4] İ. Akgün and B.Ç. Tansel. p-hub median problem for non-complete networks. *Computers and Operations Research*, 95:56–72, 2018.
- [5] E. Alba, G. Luque, and S. Nesmachnow. Parallel metaheuristics: recent advances and new trends. *International Transactions in Operational Research*, 20(1):1–48, 2013.
- [6] S. Aljančić. *Uvod u realnu i funkcionalnu analizu*. Gradjevinka knjiga, Beograd, 1968.
- [7] S. Alumur and B. Y. Kara. Network hub location problems: The state of the art. *European Journal of Operational Research*, 190(1):1–21, 2008.
- [8] S. Alumur and B.Y. Kara. A hub covering network design problem for cargo applications in Turkey. *Journal of the Operational Research Society*, 60(10):1349–1359, 2009.
- [9] K. Apt. *Principles of constraint programming*. Princeton University Press, NJ, USA, 2003.
- [10] M. Avazbeigi. An overview of complexity theory. In F.R. Zanjirani and M. Hekmatfar, editors, *Facility Location*, pages 19–36. Springer, 2009.

- [11] Т. Aykin. Lagrangian relaxation based approaches to capacitated hub-and-spoke network design problem. *European Journal of Operational Research*, 79(3):501–523, 1994.
- [12] P. Bachmann. *Die analytische zahlentheorie*. Teubner, Leipzig, 1894.
- [13] T. Back. *Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms*. Oxford University Press, Oxford, 1996.
- [14] J. E. Beasley. Obtaining test problems via internet. *Journal of Global Optimization*, 8(4):429–433, 1996.
- [15] R. Bellman. *Dynamic Programming*. Princeton University Press, Princeton, New Jersey, 1957.
- [16] A. Benaini, A. Berrajaa, J. Boukachour, and M. Oudani. Parallel genetic algorithm for the uncapacitated single allocation hub location problem on GPU. In *Proceedings of the 13th International Conference of Computer Systems and Applications (AICCSA), Agadir, Morocco, 29 Nov.-2 Dec.*, pages 1–8. IEEE, 2016.
- [17] O. Berman and D. Krass. The generalized maximal covering location problem. *Computers and Operations Research*, 29(6):563–581, 2002.
- [18] O. Berman, D. Krass, and Z. Drezner. The gradual covering decay location problem on a network. *European Journal of Operational Research*, 151(3):474–480, 2003.
- [19] C. Blum, J. Puchinger, G. R Raidl, and A. Roli. Hybrid metaheuristics in combinatorial optimization: a survey. *Applied Soft Computing*, 11(6):4135–4151, 2011.
- [20] C. Blum and A. Roli. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Computing Surveys (CSUR)*, 35(3):268–308, 2003.
- [21] N. Boland, M. Krishnamoorthy, A.T. Ernst, and J. Ebery. Preprocessing and cutting for multiple allocation hub location problems. *European Journal of Operational Research*, 155(3):638–653, 2004.

- [22] N. Bostel, P. Dejax, and M. Zhang. A model and a metaheuristic method for the hub location routing problem and application to postal services. In *Proceedings of the International Conference on Industrial Engineering and Systems Management (IESM 2015), Seville, Spain, 21-23 October*, pages 1383–1389. IEEE, 2015.
- [23] J. Brimberg, N. Mladenović, R. Todosijević, and D. Urošević. A basic variable neighborhood search heuristic for the uncapacitated multiple allocation p-hub center problem. *Optimization Letters*, 11(2):313–327, 2017.
- [24] J. Brimberg, N. Mladenović, R. Todosijević, and D. Urošević. General variable neighborhood search for the uncapacitated single allocation p-hub center problem. *Optimization Letters*, 11(2):377–388, 2017.
- [25] J. F. Campbell. Integer programming formulations of discrete hub location problems. *European Journal of Operational Research*, 72(2):387–405, 1994.
- [26] G. Caporossi, P. Hansen, and N. Mladenović. Variable neighborhood search. In Siarry P., editor, *Metaheuristics*, pages 77–98. Springer, 2016.
- [27] L.D. Chambers. *The practical handbook of genetic algorithms: applications*. Chapman and Hall/CRC, New York, 2000.
- [28] R.L. Church and K.L. Roberts. Generalized coverage models and public facility location. *Papers in Regional Science*, 53(1):117–135, 1983.
- [29] I. Contreras and E. Fernández. General network design: A unified view of combined location and network design problems. *European Journal of Operational Research*, 219(3):680–697, 2012.
- [30] I. Contreras, E. Fernández, and A. Marín. Tight bounds from a path based formulation for the tree of hub location problem. *Computers and Operations Research*, 36(12):3117–3127, 2009.
- [31] S. A. Cook. The complexity of theorem-proving procedures. In *Proceedings of the Third Annual ACM Symposium on Theory of Computing, Shaker Heights, Ohio, USA, 03 - 05 May*, pages 151–158, 1971.
- [32] D. Corne, M. Dorigo, and F. Glover. *New ideas in optimization*. McGraw-Hill, New York, 1999.

- [33] P. Crescenzi and V. Kann. Approximation on the web: A compendium of NP optimization problems. In J. Rolim, editor, *Randomization and Approximation Techniques in Computer Science*. Springer, Berlin, Heidelberg, 1997.
- [34] D. Cvetković, M. Čangalović, Đ. Dugošija, V. Kovačević-Vujčić, S. Simić, and J. Vuleta. *Kombinatorna optimizacija*. Društvo operacionih istraživača Jugoslavije, Beograd, 1996.
- [35] H. Damgacioglu, D. Dinler, N.E. Ozdemirel, and C. Iyigun. A genetic algorithm for the uncapacitated single allocation planar hub location problem. *Computers and Operations Research*, 62:224–236, 2015.
- [36] D. Dasgupta and Z. Michalewicz. *Evolutionary algorithms in engineering applications*. Springer Science & Business Media, New York, 2013.
- [37] T. Davidović, D. Ramljak, M. Šelmić, and D. Teodorović. Bee colony optimization for the p-center problem. *Computers and Operations Research*, 38(10):1367–1376, 2011.
- [38] T. Davidović, D. Teodorović, and M. Šelmić. Bee colony optimization Part I: The algorithm overview. *Yugoslav Journal of Operations Research*, 25(1):33–56, 2015.
- [39] L. Davis. *Handbook of genetic algorithms*. CumInCAD, New York, 1991.
- [40] H. Delmaire, J. A. Díaz, E. Fernández, and M. Ortega. Reactive GRASP and tabu search based heuristics for the single source capacitated plant location problem. *INFOR: Information Systems and Operational Research*, 37(3):194–225, 1999.
- [41] J. A. Díaz, D. E. Luna, J. Camacho-Vallejo, and M. Casas-Ramírez. GRASP and hybrid GRASP-Tabu heuristics to solve a maximal covering location problem with customer preference ordering. *Expert Systems with Applications*, 82:67–76, 2017.
- [42] J.J. Dongarra. *Performance of various computers using standard linear equations software*. University of Tennessee, Knoxville, 1993.
- [43] M. Dorigo and M. Birattari. Ant colony optimization. In C. Sammut and G.I. Webb, editors, *Encyclopedia of machine learning*, pages 36–39. Springer, 2011.

- [44] M. Dorigo, G.D. Caro, and L.M. Gambardella. Ant algorithms for discrete optimization. *Artificial Life*, 5(2):137–172, 1999.
- [45] M. Dorigo and L.M. Gambardella. Ant colony system: a cooperative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation*, 1(1):53–66, 1997.
- [46] M. Dorigo and T. Stützle. The ant colony optimization metaheuristic: algorithms, applications, and advances. In F. Glover and G.A. Kochenberger, editors, *Handbook of metaheuristics*, pages 250–285. Springer, 2003.
- [47] Z. Drezner, J. Brimberg, N. Mladenović, and S. Salhi. New heuristic algorithms for solving the planar p-median problem. *Computers and Operations Research*, 62:296–304, 2015.
- [48] R.C. Eberhart, Y. Shi, and J. Kennedy. *Swarm Intelligence*. Morgan Kaufmann Publishers, Burlington, 2001.
- [49] J. Ebery. Solving large single allocation p-hub problems with two or three hubs. *European Journal of Operational Research*, 128(2):447–458, 2001.
- [50] A. T. Ernst, H. Jiang, M. Krishnamoorthy, and H. Baatar. Reformulations and computational results for uncapacitated single and multiple allocation hub covering problems. *Working Paper Series*, 1:1–18, 2011.
- [51] A.T. Ernst, H. Hamacher, H. Jiang, M. Krishnamoorthy, and G. Woeginger. Uncapacitated single and multiple allocation p-hub center problems. *Computers and Operations Research*, 36(7):2230–2241, 2009.
- [52] A.T. Ernst and M. Krishnamoorthy. Efficient algorithms for the uncapacitated single allocation p-hub median problem. *Location Science*, 4(3):139–154, 1996.
- [53] A.T. Ernst and M. Krishnamoorthy. Exact and heuristic algorithms for the uncapacitated multiple allocation p-hub median problem. *European Journal of Operational Research*, 104(1):100–112, 1998.
- [54] R. Z. Farahani, M. Hekmatfar, A. B. Arabani, and E. Nikbakhsh. Hub location problems: a review of models, classification, solution techniques, and applications. *Computers and Industrial Engineering*, 64(4):1096–1109, 2013.



- [55] T.A. Feo and J.F. Bard. Flight scheduling and maintenance base planning. *Management Science*, 35(12):1415–1432, 1989.
- [56] T.A. Feo and M.G.C. Resende. Greedy randomized adaptive search procedures. *Journal of Global Optimization*, 6(2):109–133, 1995.
- [57] P. Festa, P.M. Pardalos, M.G.C. Resende, and C.C. Ribeiro. Randomized heuristics for the MAX-CUT problem. *Optimization Methods and Software*, 17(6):1033–1058, 2002.
- [58] V. Filipović. Fine-grained tournament selection operator in genetic algorithms. *Computing and Informatics*, 22(2):143–161, 2012.
- [59] G. Fishman. *Monte Carlo: concepts, algorithms, and applications*. Springer Science & Business Media, New York, 2013.
- [60] R.D. Galvão and C. ReVelle. A lagrangean heuristic for the maximal covering location problem. *European Journal of Operational Research*, 88(1):114–123, 1996.
- [61] S. García, M. Landete, and A. Marín. New formulation and a branch-and-cut algorithm for the multiple allocation p-hub median problem. *European Journal of Operational Research*, 220(1):48–57, 2012.
- [62] M.R. Garey and D.S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Company, New York, 1979.
- [63] M. Gendreau and J.-Y. Potvin. Metaheuristics in combinatorial optimization. *Annals of Operations Research*, 140(1):189–213, 2005.
- [64] F. Glover. Future paths for integer programming and links to artificial intelligence. *Computers and Operations Research*, 13(5):533–549, 1986.
- [65] F. Glover. Tabu search – part I. *ORSA Journal on Computing*, 1(3):190–206, 1989.
- [66] F. Glover. Tabu search and adaptive memory programming—advances, applications and challenges. In R.S. Barr, R.V. Helgason, and J.L. Kennington, editors, *Interfaces in Computer Science and Operations Research*, pages 1–75. Springer, 1997.

- [67] Fred Glover. Tabu search – part II. *ORSA Journal on Computing*, 2(1):4–32, 1990.
- [68] F.W. Glover and G.A. Kochenberger. *Handbook of Metaheuristics*, volume 57. Springer-Verlag, New York, 2003.
- [69] D.E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, Boston, 1989.
- [70] A.J. Goldman. Optimal locations for centers in a network. *Transportation Science*, 3(4):352–360, 1969.
- [71] W.J. Gutjahr. Convergence analysis of metaheuristics. In Maniezzo V., Stützle T., and Voß S., editors, *Metaheuristics*, pages 159–187. Springer, 2009.
- [72] P. Hansen and N. Mladenović. An introduction to variable neighborhood search. In S. Voß, S. Martello, I.H. Osman, and C. Roucairol, editors, *Meta-Heuristics*, pages 433–458. Springer, Boston, 1999.
- [73] P. Hansen and N. Mladenović. Variable neighborhood search. In E. K. Burke and R. D. Graham, editors, *Search methodologies: Introductory Tutorials in Optimization and Decision Support Techniques*, pages 313–337. Springer-Verlag, New York, 2014.
- [74] P. Hansen, N. Mladenović, J. Brimberg, and J.A.M. Pérez. Variable neighborhood search. In M. Gendreau and J.Y. Potvin, editors, *Handbook of metaheuristics*, pages 61–86. Springer, Boston, 2010.
- [75] P. Hansen, N. Mladenović, and J.A. Moreno Pérez. Variable neighbourhood search: methods and applications. *4OR*, 6(4):319–360, 2008.
- [76] P. Hansen, N. Mladenović, and J.A. Moreno Pérez. Variable neighbourhood search: methods and applications. *Annals of Operations Research*, 175(1):367–407, 2010.
- [77] P. Hansen, N. Mladenović, R. Todosijević, and S. Hanafi. Variable neighborhood search: basics and variants. *EURO Journal on Computational Optimization*, 5(3):423–454, 2017.
- [78] H. Hernández-Pérez, I. Rodríguez-Martín, and J. J. Salazar-González. A hybrid GRASP/VND heuristic for the one-commodity pickup-and-delivery

- traveling salesman problem. *Computers and Operations Research*, 36(5):1639–1645, 2009.
- [79] J.H. Holland. *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. The University of Michigan Press, Ann Arbor, 1975.
- [80] Y.H. Hwang and Young Hoon Lee. Uncapacitated single allocation p-hub maximal covering problem. *Computers and Industrial Engineering*, 63(2):382–389, 2012.
- [81] A. Ilić, D. Urošević, J. Brimberg, and N. Mladenović. A general variable neighborhood search for solving the uncapacitated single allocation p-hub median problem. *European Journal of Operational Research*, 206(2):289–300, 2010.
- [82] L. Ingber. Simulated annealing: Practice versus theory. *Mathematical and Computer Modelling*, 18(11):29–57, 1993.
- [83] O. Janković. An efficient genetic algorithm for the uncapacitated r-allocation p-hub maximal covering problem. *Yugoslav Journal of Operations Research*, 28(2):201–218, 2018.
- [84] O. Janković, S. Mišković, Z. Stanimirović, and R. Todosijević. Novel formulations and VNS-based heuristics for single and multiple allocation p-hub maximal covering problems. *Annals of Operations Research*, 259(1-2):191–216, 2017.
- [85] O. Janković and Z. Stanimirović. A general variable neighborhood search for solving the uncapacitated r-allocation p-hub maximal covering problem. *Electronic Notes in Discrete Mathematics*, 58:23–30, 2017.
- [86] M. Jünger, G. Reinelt, and S. Thienel. Practical problem solving with cutting plane algorithms in combinatorial optimization. In W. Cook, L. Lovász, and P. Seymour, editors, *Series in Discrete Mathematics and Theoretical Computer Science*, volume 20, pages 111–152. 1995.
- [87] B.Y. Kara and B.Ç. Tansel. On the single-assignment p-hub center problem. *European Journal of Operational Research*, 125(3):648–655, 2000.

- [88] B.Y. Kara and B.Ç. Tansel. The single-assignment hub covering problem: Models and linearizations. *Journal of the Operational Research Society*, 54(1):59–64, 2003.
- [89] O. Karasakal and E.K. Karasakal. A maximal covering location model in the presence of partial coverage. *Computers and Operations Research*, 31(9):1515–1526, 2004.
- [90] H. Karimi and M. Bashiri. Hub covering location problems with different coverage types. *Scientia Iranica*, 18(6):1571–1578, 2011.
- [91] R.M. Karp. Reducibility among combinatorial problems. In R. E. Miller and J. W. Thatcher, editors, *Complexity of Computer Computations*, pages 85–103. Springer, 1972.
- [92] J. Kennedy and R. Eberhart. Particle swarm optimization. In *Proceedings of International Conference on Neural Networks (ICNN'95), Perth, Australia, 27 Nov.-1 Dec.*, volume 4, pages 1942–1948. IEEE, 1995.
- [93] A. Kimms. Economies of scale in hub & spoke network design models: We have it all wrong. In M. Morlock, C. Schwindt, N. Trautmann, and J. Zimmermann, editors, *Perspectives on Operations Research*, pages 293–317. Springer, 2006.
- [94] S. Kirkpatrick, C.D. Gelatt, and M.P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.
- [95] J.G. Klincewicz. Avoiding local optima in the p-hub location problem using tabu search and GRASP. *Annals of Operations Research*, 40(1):283–302, 1992.
- [96] R.E. Korf. Depth-first iterative-deepening: An optimal admissible tree search. *Artificial Intelligence*, 27(1):97–109, 1985.
- [97] J. Kratica and Z. Stanimirović. Solving the uncapacitated multiple allocation p-hub center problem by genetic algorithm. *Asia-Pacific Journal of Operational Research*, 23(04):425–437, 2006.
- [98] J. Kratica, Z. Stanimirović, D. Tošić, and V. Filipović. Two genetic algorithms for solving the uncapacitated single allocation p-hub median problem. *European Journal of Operational Research*, 182(1):15–28, 2007.

- [99] M. Labbé and H. Yaman. Solving the hub location problem in a star–star network. *Networks: An International Journal*, 51(1):19–33, 2008.
- [100] E. Landau. *Handbuch der Lehre von der Verteilung der Primzahlen*. Teubner, Leipzig, 1909.
- [101] E.L. Lawler and D.E. Wood. Branch-and-bound methods: a survey. *Operations Research*, 14(4):699–719, 1966.
- [102] H.R. Lourenço, O.C. Martin, and T. Stützle. Iterated local search. In Glover F. and Kochenberger G.A., editors, *Handbook of metaheuristics*, pages 320–353. Springer, 2003.
- [103] R.F. Love, J.G. Morris, and G.O. Wesolowski. *Facility Location: Models and Methods, vol. 7*. Publications in Operations Research, New York, 1996.
- [104] P. Lučić and D. Teodorović. Bee system: modeling combinatorial optimization transportation engineering problems by swarm intelligence. In *Preprints of the Triennial Symposium on Transportation Analysis (TRISTAN IV), Sao Miguel, Azores Islands, 13-19 June*, pages 441–445, 2001.
- [105] P. Lučić and D. Teodorović. Computing with bees: attacking complex transportation engineering problems. *International Journal on Artificial Intelligence Tools*, 12(03):375–394, 2003.
- [106] M. Marić, Z. Stanimirović, and P. Stanojević. An efficient memetic algorithm for the uncapacitated single allocation hub location problem. *Soft Computing*, 17(3):445–466, 2013.
- [107] R. Martí and G. Reinelt. *The linear ordering problem: exact and heuristic methods in combinatorial optimization*, volume 175. Springer-Verlag, Berlin, Heidelberg, 2011.
- [108] S.L. Martins, M.G.C. Resende, C.C. Ribeiro, and P.M. Pardalos. A parallel GRASP for the steiner tree problem in graphs using a hybrid local search strategy. *Journal of Global Optimization*, 17(1):267–283, 2000.
- [109] T. Meyer, A.T. Ernst, and M. Krishnamoorthy. A 2-phase algorithm for solving the single allocation p-hub center problem. *Computers and Operations Research*, 36(12):3143–3151, 2009.

- [110] Z. Michalewicz. Evolution strategies and other methods. In *Genetic Algorithms + Data Structures = Evolution Programs*, pages 159–177. Springer, Berlin, Heidelberg, 1996.
- [111] N. Mladenović and P. Hansen. Variable neighborhood search. *Computers and Operations Research*, 24(11):1097–1100, 1997.
- [112] Y.N. Moschovakis. What is an algorithm? In B. Engquist and W. Schmid, editors, *Mathematics Unlimited—2001 and Beyond*, pages 919–936. Springer, Berlin, Heidelberg, 2001.
- [113] Z. Ognjanović and N. Krdžavac. *Uvod u teorijsko racunarstvo*. Fakultet organizacionih nauka, Beograd, 2004.
- [114] M.E. O’Kelly. A quadratic integer program for the location of interacting hub facilities. *European Journal of Operational Research*, 32(3):393–404, 1987.
- [115] M.E. O’Kelly and D.L. Bryan. Hub location with flow economies of scale. *Transportation Research Part B: Methodological*, 32(8):605–616, 1998.
- [116] I.H. Osman and G. Laporte. Metaheuristics: a bibliography. *Annals of Operations Research*, 63:513–623, 1996.
- [117] C.H. Papadimitriou and K. Steiglitz. *Combinatorial optimization: algorithms and complexity*. Dover Publications, New York, 1998.
- [118] F. Parreño, R. Alvarez-Valdés, J.F. Oliveira, and J.M. Tamarit. A hybrid GRASP/VND algorithm for two-and three-dimensional bin packing. *Annals of Operations Research*, 179(1):203–220, 2010.
- [119] J. Peiró, Á. Corberán, M. Laguna, and R. Martí. Models and solution methods for the uncapacitated r-allocation p-hub equitable center problem. *International Transactions in Operational Research*, 2017.
- [120] J. Peiró, Á. Corberán, and R. Martí. GRASP for the uncapacitated r-allocation p-hub median problem. *Computers and Operations Research*, 43(1):50–60, 2014.
- [121] M. Peker and B.Y. Kara. The p-hub maximal covering problem and extensions for gradual decay functions. *Omega*, 54:158–172, 2015.

- [122] M. Pirlot. General local search methods. *European journal of operational research*, 92(3):493–511, 1996.
- [123] R. Poli. Analysis of the publications on the applications of particle swarm optimisation. *Journal of Artificial Evolution and Applications*, 2008, 2008.
- [124] M. Prais and C.C. Ribeiro. Reactive GRASP: An application to a matrix decomposition problem in TDMA traffic assignment. *INFORMS Journal on Computing*, 12(3):164–176, 2000.
- [125] C. Reeves. Genetic algorithms. In F. Glover and G.A. Kochenberger, editors, *Handbook of metaheuristics*, pages 55–82. Springer, 2003.
- [126] C.R. Reeves. *Modern heuristic techniques for combinatorial problems*. John Wiley & Sons, New York, 1993.
- [127] M.G.C. Resende and C.C. Ribeiro. Greedy randomized adaptive search procedures: advances, hybridizations, and applications. In M. Gendreau and J.Y. Potvin, editors, *Handbook of Metaheuristics*, pages 283–319. Springer, 2010.
- [128] M.G.C. Resende and C.C. Ribeiro. GRASP: Greedy randomized adaptive search procedures. In E.K. Burke and G. Kendall, editors, *Search Methodologies - Introductory tutorials in optimization and decision support systems*, pages 287–312. Springer, 2014.
- [129] C.C. Ribeiro and P. Hansen. *Essays and surveys in metaheuristics*, volume 15. Springer Science & Business Media, Berlin, Heidelberg, 2012.
- [130] C.C. Ribeiro, E. Uchoa, and R.F. Werneck. A hybrid GRASP with perturbations for the steiner problem in graphs. *INFORMS Journal on Computing*, 14(3):228–246, 2002.
- [131] C.C. Ribeiro and D.S. Vianna. A GRASP/VND heuristic for the phylogeny problem using a new neighborhood structure. *International Transactions in Operational Research*, 12(3):325–338, 2005.
- [132] A. Salehipour, K. Sörensen, P. Goos, and O. Bräysy. Efficient GRASP+VND and GRASP+VNS metaheuristics for the traveling repairman problem. *Quarterly Journal of Operations Research*, 9(2):189–209, 2011.

- [133] M.R. Silva and C.B. Cunha. A tabu search heuristic for the uncapacitated single allocation p-hub maximal covering problem. *European Journal of Operational Research*, 262(3):954–965, 2017.
- [134] D. Skorin-Kapov, J. Skorin-Kapov, and M. O’Kelly. Tight linear programming relaxations of uncapacitated p-hub median problems. *European Journal of Operational Research*, 94(3):582–593, 1996.
- [135] Z. Stanimirović. A genetic algorithm approach for the capacitated single allocation p-hub median problem. *Computing and Informatics*, 29(1):117–132, 2012.
- [136] Z. Stanimirović, Kratica J., Filipović V., and D Tošić. *Evolutivni pristup za rešavanje hab lokacijskih problema*. Zavod za udžbenike i nastavna sredstva, Beograd, 2011.
- [137] P. Stanojević, M. Marić, and Z. Stanimirović. A hybridization of an evolutionary algorithm and a parallel branch and bound for solving the capacitated single allocation hub location problem. *Applied Soft Computing*, 33:24–36, 2015.
- [138] B. Suman and P. Kumar. A survey of simulated annealing as a tool for single and multiobjective optimization. *Journal of the Operational Research Society*, 57(10):1143–1160, 2006.
- [139] E.G. Talbi. A taxonomy of hybrid metaheuristics. *Journal of Heuristics*, 8(5):541–564, 2002.
- [140] E.G. Talbi. *Metaheuristics: from design to implementation*, volume 74. John Wiley & Sons, New York, 2009.
- [141] D. Teodorović. Bee colony optimization (BCO). In C.P. Lim, L.C. Jain, and S. Dehuri, editors, *Innovations in swarm intelligence*, pages 39–60. Springer, 2009.
- [142] D. Teodorović, M. Šelmić, and T. Davidović. Bee Colony Optimization-part II: The application survey. *Yugoslav Journal of Operations Research*, 25(2):185–219, 2015.



- [143] R. Todosijević, D. Urošević, N. Mladenović, and S. Hanafi. A general variable neighborhood search for solving the uncapacitated r-allocation p-hub median problem. *Optimization Letters*, 11(6):1109–1121, 2017.
- [144] H. Topcuoglu, F. Corut, M. Ermis, and G. Yilmaz. Solving the uncapacitated hub location problem using genetic algorithms. *Computers and Operations Research*, 32(4):967–984, 2005.
- [145] V.V. Vazirani. *Approximation algorithms*. Springer Science & Business Media, Berlin, 2013.
- [146] J.G. Villegas, C. Prins, C. Prodhon, A.L. Medaglia, and N. Velasco. GRASP/VND and multi-start evolutionary local search for the single truck and trailer routing problem with satellite depots. *Engineering Applications of Artificial Intelligence*, 23(5):780–794, 2010.
- [147] B. Wagner. Model formulations for hub covering problems. *Journal of the Operational Research Society*, 59(7):932–938, 2008.
- [148] K. Weng, C. Yang, and Y. Ma. Two artificial intelligence heuristics in solving multiple allocation hub maximal covering problem. *Intelligent Computing*, pages 737–744, 2006.
- [149] H. Yaman. Allocation strategies in hub networks. *European Journal of Operational Research*, 211(3):442–451, 2011.
- [150] H. Yaman, O.E. Karasan, and B.Y. Kara. Release time scheduling and hub location for next-day delivery. *Operations Research*, 60(4):906–917, 2012.
- [151] M. Živković. *Algoritmi*. Matematički fakultet, Beograd, 2000.

# Биографија аутора

Оливера Станчић (Јанковић) је рођена 14.04.1987. године у Крушевцу, где је завршила основну школу и Гимназију са одличним успехом.

Математички факултет у Београду, смер Нумеричка математика и оптимизација, уписала је 2006. године. Дипломирала је 2010. године, са просечном оценом 9.50. Исте године је уписала мастер студије на Математичком факултету у Београду, смер Нумеричка математика и оптимизација. Мастер рад под насловом „*Методе решавања транспортног проблема*” одбранила је 11.10.2011. године, под менторством проф. др Ђорђа Дугошије, чиме је стекла звање Дипломирани математичар-мастер. Докторске студије на Математичком факултету у Београду, смер Нумеричка математика и оптимизација, уписала је 2011. године.

Запослена је на Економском факултету Универзитета у Крагујевцу где је од марта 2011. до марта 2013. године радила као сарадник у настави, а од марта 2013. године је изабрана у звање асистента на Катедри за информатику и квантитативне методе. Од 2011. године до данас изводи рачунске вежбе на предмету *Математика у економији* на основним студијама на Економском факултету у Крагујевцу. Од 2011. године ангажована је на пројекту Министарства просвете, науке и технолошког развоја под називом „*Апроксимација интегралних и диференцијалних оператора и примене*”.

Прилог 1.

## Изјава о ауторству

Потписани-а Оливера Станчић

број индекса 2018/2011

### Изјављујем

да је докторска дисертација под насловом

„Проблеми хаб максималног покривања - нови математички модели и методе решавања“

- резултат сопственог истраживачког рада,
- да предложена дисертација у целини ни у деловима није била предложена за добијање било које дипломе према студијским програмима других високошколских установа,
- да су резултати коректно наведени и
- да нисам кршио/ла ауторска права и користио интелектуалну својину других лица.

**Потпис докторанда**

У Београду, 16.10. 2018.

\_\_\_\_\_

Прилог 2.

## Изјава о истоветности штампане и електронске верзије докторског рада

Име и презиме аутора Оливера Станчић

Број индекса 2018/2011

Студијски програм Математика

Наслов рада „Проблеми хаб максималног покривања - нови математички модели и методе решавања“

Ментор проф. др Зорица Станимировић

Потписани/а Оливера Станчић

Изјављујем да је штампана верзија мог докторског рада истоветна електронској верзији коју сам предао/ла за објављивање на порталу **Дигиталног репозиторијума Универзитета у Београду**.

Дозвољавам да се објаве моји лични подаци везани за добијање академског звања доктора наука, као што су име и презиме, година и место рођења и датум одбране рада.

Ови лични подаци могу се објавити на мрежним страницама дигиталне библиотеке, у електронском каталогу и у публикацијама Универзитета у Београду.

**Потпис докторанда**

У Београду, 16.10. 2018.

\_\_\_\_\_

### Прилог 3.

## Изјава о коришћењу

Овлашћујем Универзитетску библиотеку „Светозар Марковић“ да у Дигитални репозиторијум Универзитета у Београду унесе моју докторску дисертацију под насловом:

„Проблеми хаб максималног покривања - нови математички модели и методе решавања“

---

која је моје ауторско дело.

Дисертацију са свим прилозима предао/ла сам у електронском формату погодном за трајно архивирање.

Моју докторску дисертацију похрањену у Дигитални репозиторијум Универзитета у Београду могу да користе сви који поштују одредбе садржане у одабраном типу лиценце Креативне заједнице (Creative Commons) за коју сам се одлучио/ла.

1. Ауторство

2. Ауторство - некомерцијално

3. Ауторство – некомерцијално – без прераде

4. Ауторство – некомерцијално – делити под истим условима

5. Ауторство – без прераде

6. Ауторство – делити под истим условима

(Молимо да заокружите само једну од шест понуђених лиценци, кратак опис лиценци дат је на полеђини листа).

**Потпис докторанда**

У Београду, 16.10.2018.

---