

УНИВЕРЗИТЕТ У БЕОГРАДУ
МАТЕМАТИЧКИ ФАКУЛТЕТ

Лазар М. Мркела

МЕТАХЕУРИСТИЧКЕ МЕТОДЕ
ВИШЕКРИТЕРИЈУМСКЕ ОПТИМИЗАЦИЈЕ
И ПРИМЕНЕ НА ДИСКРЕТНЕ
ЛОКАЦИЈСКЕ ПРОБЛЕМЕ

докторска дисертација

Београд, 2024.

UNIVERSITY OF BELGRADE
FACULTY OF MATHEMATICS

Lazar M. Mrkela

METAHEURISTIC METHODS FOR
MULTI-OBJECTIVE OPTIMIZATION AND
APPLICATIONS TO DISCRETE LOCATION
PROBLEMS

Doctoral Dissertation

Belgrade, 2024.

Ментор:

др Зорица СТАНИМИРОВИЋ, редовни професор
Универзитет у Београду - Математички факултет

Чланови комисије:

др Мирослав МАРИЋ, редовни професор
Универзитет у Београду - Математички факултет

др Нина РАДОЛИЧИЋ МАТИЋ, доцент
Универзитет у Београду - Математички факултет

др Татјана ДАВИДОВИЋ, научни саветник
Математички институт САНУ, Београд

Датум одбране:

Мојим родитељима

Захвалнице

Захваљујем се свом ментору проф. др Зорици Станимировић на стручној помоћи, саветима и одличној сарадњи. Захвалан сам и члановима комисије проф. др Мирославу Марићу, доц. др Нини Радојичић Матић и др Татјани Давидовић који су својим коментарима и сугестијама значајно допринели квалитету ове дисертације. Захвалност дугујем и проф. др Милану Туби који ме је заинтересовао за науку писањем првог рада за научну конференцију. Посебно сам захвалан својој породици, тј. родитељима и сестри на неизмерној подршци и разумевању. Такође, захвалан сам и мојим пријатељима који су били уз мене.

Наслов дисертације: Метахеуристичке методе вишекритеријумске оптимизације и примене на дискретне локацијске проблеме

Резиме: У овој дисертацији разматрана су два дискретна локацијска проблема и њихове двокритеријумске варијанте. Разматран је проблем максималног покривања локација са преференцијама корисника и ограничењима буџета за отварање објеката. Ова варијанта проблема максималног покривања није до сада разматрана у литератури. За разлику од класичног проблема максималног покривања, проблем разматран у овој дисертацији укључује преференције корисника ка локацијама, при чему се сваки корисник додељује локацији коју највише преферира, а која има отворен објекат. Поред тога, различите локације имају различите цене постављања објеката, а расположиви буџет је ограничен. Овај проблем је решаван методом променљивих околина, а добијени резултати су упоређени са резултатима егзактног решавача на модификованим инстанцама из литературе. Додатно, решавана је и постојећа варијанта проблема максималног покривања која уместо ограниченог буџета укључује ограничења на број објеката које треба отворити.

Разматран је и проблем постављања регенератора у оптичким мрежама. Код оптичких мрежа квалитет сигнала опада са растојањем, те је потребно поставити скупе уређаје који ће опоравити сигнал. У овој дисертацији разматран је постојећи модел, где је скуп локација за постављање регенератора и скуп корисничких чворова различит и проблем се назива уопштеним. Уопштени проблем постављања регенератора у оптичким мрежама је такође решаван методом променљивих околина, а резултати су упоређени са најбољим доступним из литературе.

Дефинисане су и двокритеријумске варијанте наведених проблема. Код проблема максималног покривања локација, преференције корисника су укључене као тежински фактори у укупној покривеној потражњи, што чини прву функцију циља. Друга функција представља број непокривених корисника и тежи да оствари правичност у моделу. Код проблема постављања регенератора у оптичким мрежама, претпоставка је да, услед ограниченог буџета, није могуће обезбедити несметану комуникацију између свих парова корисничких чворова. Сваки пар има додељену тежину, а сума тежина повезаних парова чини прву функцију циља. Друга функција циља је цена постављања регенератора. Двокритеријумске варијанте су решаване прилагођеном вишекритеријумском ва-

ријантом методе променљивих околина, а приказани су резултати поређења са уопштеним еволутивним алгоритмима.

Кључне речи: вишекритеријумска оптимизација, проблем максималног покривања локација, проблем оптималног постављања регенератора, метахеуристике, метода променљивих околина, еволутивни алгоритми

Научна област: Рачунарство и информатика

Ужа научна област: Рачунарска интелигенција

Dissertation title: Metaheuristic methods for multi-objective optimization and applications to discrete location problems

Abstract: This dissertation examines two discrete location problems and their bi-objective variants. The first problem under consideration is the maximal covering location problem with user preferences and budget constraints imposed on facility opening. This variant of the maximal covering problem has not been previously studied in the literature. Unlike the classical maximal covering problem, the variant proposed in this dissertation includes user preferences for locations, where users are assigned to the location with opened facility that they prefer the most. Additionally, different locations have different costs for establishing facilities, and the available budget for opening facilities is limited. This problem is solved using the Variable Neighborhood Search (VNS) method, and the results were compared with the ones obtained by an exact solver on modified instances from the literature. Furthermore, an existing variant of the maximal covering problem is also addressed, which imposes the limit on the number of opened facilities instead of limiting the budget for opening facilities.

The second problem examined is the regenerator placement in optical networks. In optical networks, signal quality degrades with distance, necessitating the placement of costly devices to restore the signal. This dissertation studies an existing model where the set of possible regenerator locations and the set of user nodes are different, defining the problem as generalized. The generalized regenerator placement problem in optical networks is also solved using the Variable Neighborhood Search method, with results compared to the best available solutions from the literature.

Bi-objective variants of these problems are defined as well. For the maximal covering location problem, user preferences are included as weighted factors in the total covered demand, forming the first objective function. The second objective function represents the number of uncovered users and aims to ensure fairness in the model. In the regenerator placement problem for optical networks, it is assumed that, due to budget constraints, uninterrupted communication between all pairs of user nodes may not be feasible. Each pair is assigned a weight, and the sum of the weights of connected pairs constitutes the first objective function, while the second objective function represents the cost of placing regenerators. These bi-objective variants are solved using an adapted multi-objective version of the Variable Neigh-

neighborhood Search method, and the results are compared with general evolutionary algorithms.

Keywords: multi-objective optimisation, maximal covering location problem, re-generator location problem, metaheuristics, variable neighborhood search, evolutionary algorithms

Research area: Computer Science

Research sub-area: Computational Intelligence

Садржај

1	Увод	1
1.1	Математичка оптимизација	4
1.2	Локацијски проблеми	8
1.3	Егзактне методе оптимизације	12
1.4	Метахеуристичке методе	15
2	Проблем максималног покривања локација са ограниченим буџетом и преференцијама корисника	37
2.1	Опис проблема и преглед релевантне литературе	37
2.2	Математичка формулација проблема	42
2.3	Метода променљивих околина за решавање проблема	44
2.4	Експериментални резултати	52
3	Двокритеријумски проблем максималног покривања локација са преференцијама корисника	73
3.1	Опис проблема и преглед релевантне литературе	73
3.2	Математичка формулација проблема	77
3.3	Вишекритеријумска метода променљивих околина	80
3.4	Експериментални резултати	86
4	Уопштени проблем постављања регенератора у оптичким мрежама	95
4.1	Опис проблема и преглед релевантне литературе	95
4.2	Формулација проблема на графу	97
4.3	Унапређена метода променљивих околина за WGRLP	99
4.4	Експериментални резултати	112

5	Двокритеријумски уопштени проблем постављања регенера- тора у оптичким мрежама	119
5.1	Опис проблема и преглед релевантне литературе	119
5.2	Вишекритеријумска метода променљивих околина	121
5.3	Експериментални резултати	122
	Закључак	137
	Библиографија	139

Глава 1

Увод

У многим реалним ситуацијама потребно је оптимизовати више (потенцијално супростављених) функција циља истовремено. Како није могуће добити једно оптимално решење, циљ је пронаћи скуп решења која представљају најбољи компромис између свих функција циља и таква решења се називају Парето оптимална. Код Парето оптималних решења није могуће побољшати једну од функција циља, а да се не погорша нека друга. Скуп свих Парето оптималних решења неког проблема назива се Парето скуп (ако посматрамо простор решења) или Парето фронт (ако посматрамо простор функција циља) [100].

Једноставни начин за решавање проблема вишекритеријумске оптимизације представља свођење на једнокритеријумску оптимизацију. На пример, може се оптимизовати линеарна комбинација функција циља са тежинским коефицијентима. Код ове методе корисник треба да одреди које ће тежине поставити. Променом тежина могу се добити различита компромисна решења, али овакав начин може пропустити нека решења из Парето скупа. Још једна могућност је и употреба хијерархијског приступа, где корисник мора да одреди приоритет функцијама циља. Већина постојећих егзактних решавача нуди овакве начине решавања (GUROBI, CPLEX). Код оба наведена начина, корисник мора унапред да дефинише која му је функција приоритетна или да рангира функције циља по приоритетима, а да му још нису познате све опције, тј. сва компромисна решења. Како би особа која доноси одлуке имала потпуне информације о могућим компромисним решењима потребно је пронаћи скуп свих Парето оптималних решења. Тада се може изабрати једно или више компромисних решења у зависности од примене и приоритета особе која одлучује.

Проналажење скупа свих Парето оптималних решења може бити временски

веома захтевно. У случају НП-тешких проблема и инстанци великих димензија, егзактна решења се често не могу добити у прихватљивом времену. Метахеуристички алгоритми могу ефикасно пронаћи апроксимативни скуп решења који у реалним ситуацијама може да се користи уместо правог Парето скупа. У литератури се често користе еволутивни алгоритми за решавање вишекритеријумских проблема, јер популација решења таквих алгоритама природно одговара апроксимативном скупу решења [23, 175, 186]. Ови алгоритми су постали и део оквира за развој софвера и библиотека за различите програмске језике [20, 80]. Међутим, у новијим радовима се јављају и вишекритеријумске варијанте алгоритама који су засновани на локалној претрази, као што су метода променљивих околина (енгл. Variable Neighborhood Search, VNS) и насумично похлепна адаптивна претрага (енгл. Greedy Randomized Adaptive Search Procedure, GRASP). Ове методе дају добре резултате на једнокритеријумским варијантама многих дискретних локацијских проблема, па њихова примена на вишекритеријумске варијанте може потенцијално побољшати решења постојећих еволутивних алгоритама.

У оквиру ове дисертације разматрана су два локацијска проблема и њихове вишекритеријумске варијанте: Проблем максималног покривања локација са преференцијама корисника (енгл. Maximal Covering Location Problem with Customer Preferences, MCLP) и уопштени проблем постављања регенератора у оптичким мрежама (енгл. Generalized Regenerator Location Problem, GRLP).

MCLP представља локацијски проблем чији је циљ наћи оптималне локације p објеката, тако да се максимизује покривена потражња корисника [37]. Корисници су покривени ако се налазе на растојању мањем од унапред задатог $r > 0$ који се назива полупречник покривања. У варијанти проблема са преференцијама корисника, сваки корисник има преференцију ка одређеној локацији. Функција циља је сума покривене потражње корисника коју треба максимизовати, с тим што је претпоставка да већ постоје постављени објекти конкуренције, а корисници бирају објекат на локацији коју више преферирају. У овој дисертацији разматрана је варијанта MCLP која боље одсликава реалне ситуације. Уместо фиксног броја p нових објеката који се постављају нова варијанта укључује ограничени буџет и различите цене постављања на различитим локацијама.

У оквиру ове дисертације, дефинисана је и нова вишекритеријумска варијанта проблема. Мотивација је налажење компромиса између покривене по-

тражње корисника са тежинским коефицијентима пропорционалним преференцијама корисника и броја непокривених корисника. Идеја је да се што више задовоље преференције корисника са великом потражњом и оствари што већа укупна задовољена потражња, а да се води рачуна и о покривености осталих корисника, тј. да се узме у обзир и правичност при одлучивању.

Проблем оптималног постављања регенератора (RLP) представља локацијски проблем који потиче из дизајна оптичких мрежа [35]. У оптичким мрежама, после неког растојања, квалитет сигнала нагло опада, те је потребно инсталирати скупе уређаје - регенераторе, чији је задатак да опораве квалитет сигнала. Код основне варијанте проблема (RLP) скуп потенцијалних локација за регенераторе и скуп локација крајњих корисника су исти, док су код уопштене варијанте (GRLP) различити [34]. Код обе варијанте (RLP и GRLP), циљ је смањити број регенератора, а при томе очувати несметану комуникацију између свих задатих парова крајњих корисника мреже. Тежинска варијанта проблема (WGRLP) укључује различите цене инсталације регенератора у зависности од локације што више одговара реалним ситуацијама.

У случају ограниченог буџета, није увек могуће остварити несметану комуникацију између свих крајњих корисника путем оптичке мреже. Стога је у овој дисертацији дефинисан двокритеријумски модел који узима у обзир и трошкове инсталације (тј. трошкове постављања регенератора) и могућност комуникације парова крајњих корисника. Како сви парови у пракси не морају да имају исти приоритет, за другу функцију циља узима се збир тежинских коефицијената парова.

Наведени проблеми су НП-тешки, а у реалним ситуацијама се јавља велики број потенцијалних локација, тако да егзактни решавачи не могу у прихватљивом времену да добију оптимална решења или не могу дати чак ни допустиво решење усред меморијског ограничења. Поред тога, проблеми вишекритеријумске оптимизације су још захтевнији јер треба пронаћи скуп свих Парето оптималних решења. Стога, хеуристичке и метахеуристичке методе представљају ефикасне начине решавања ових проблема у пракси.

Остатак дисертације је организован на следећи начин. У оквиру уводне главе дати су основни појмови и дефиниције у математичкој оптимизацији, затим је дат кратак опис и преглед локацијских проблема, као и метода оптимизације за њихово решавање. Глава 2 се бави проблемом максималног покривања локација са ограниченим буџетом и преференцијама корисника, док се у глави 3

разматра двокритеријумска варијанта овог проблема. У глави 4 је предложена метода за решавање уопштеног проблема постављања регенератора у оптичким мрежама, док се у глави 5 дефинише и решава двокритеријумска варијанта овог проблема. Закључак ове дисертације приказан је у оквиру главе 6.

1.1 Математичка оптимизација

У многим областима, доносиоци одлука имају задатак да нађу оптимално решење неког проблема из датог скупа допустивих решења. Мотивација је доношење одлука које ће побољшати ефикасност система, смањити трошкове или унапредити перформансе. Математичка оптимизација је област примењене математике која се бави проналажењем најбољег решења (једног или више) из скупа допустивих решења [166]. Следећи појмови су кључни за дефиницију проблема математичке оптимизације:

- **Функција циља:** Функција $f(\mathbf{x})$ коју треба оптимизовати (било минимизовати или максимизовати).
- **Променљиве одлучивања:** Променљиве, представљене вектором \mathbf{x} , у зависности од којих се мења вредност функције циља.
- **Ограничења:** Ограничења променљивих одлучивања која дефинишу простор допустивих решења \mathbf{S} .

На основу дефинисаних појмова општа форма проблема математичке оптимизације је:

$$\text{Минимизовати (или максимизовати)} \quad f(\mathbf{x}) \quad (1.1)$$

$$\text{при ограничењима:} \quad g_i(\mathbf{x}) \leq b_i, \quad i = 1, \dots, m \quad (1.2)$$

$$h_j(\mathbf{x}) = d_j, \quad j = 1, \dots, p \quad (1.3)$$

У наведеној форми, $f(\mathbf{x})$ представља функцију циља коју оптимизујемо. Вектор \mathbf{x} је вектор променљивих одлучивања, чије вредности можемо изабрати. Простор допустивих решења је дефинисан ограничењима (1.2) и (1.3), где су $g_i(\mathbf{x})$ ограничења неједнакости, док су $h_j(\mathbf{x})$ ограничења једнакости. Решење

проблема је представљено одређеним избором вредности за променљиве одлучивања. Решење из допустивог скупа решења које даје најмању (или највећу) вредност функције циља представља оптимално решење проблема.

Проблеми оптимизације се могу класификовати у зависности од особина функције циља и ограничења. Функција која се оптимизује може бити конвексна или неконвексна. Затим, променљиве одлучивања могу узимати реалне вредности (континуална оптимизација) или вредности из пребројивог скупа (дискретна оптимизација). У случају да су ограничења и функција циља линеарне комбинације променљивих одлучивања имамо проблем линеарног програмирања, док у случају квадратне функције циља проблем се назива квадратно програмирање. Проблеми решавања у овој дисертацији су проблеми дискретне, тј. комбинаторне оптимизације, тако да ће ова класа проблема, као и методе њиховог решавања бити детаљније размотрени у наредним одељцима.

За формалну дефиницију комплексности решавања проблема комбинаторне оптимизације користе се следећи појмови. У класу НП проблема спадају они проблеми за чије се задато решење може утврдити у полиномијалном времену да ли је то решење допустиво. Проблем А се може свести на проблем Б у полиномијалном времену ако постоји алгоритам који трансформише било коју инстанцу проблема А у инстанцу проблема Б, тако да решење проблема Б решава и проблем А. Проблем А је НП-тежак ако за сваки проблем Б из НП класе постоји свођење проблема Б на проблем А у полиномијалном времену. Проблем је НП-комплетан ако је и НП и НП-тежак проблем.

Вишекритеријумска оптимизација

Проблеми из праксе често укључују више циљева (критеријума), који могу бити супростављени. На пример, побољшање вредности једне функције циља може довести до погоршања вредности неке друге функције циља. Доносиоци одлука тада траже решења која представљају компромисе између различитих функција циља. Овакав вид оптимизације се назива вишекритеријумска оптимизација. У случају да постоје тачно два супростављена циља, проблем се назива двокритеријумским. Слично као и код проблема са једном функцијом циља, у зависности од могућих вредности за променљиве одлучивања, можемо имати проблеме комбинаторне (дискретне) и непрекидне (континуалне) оптимизације [41].

Методe за решавање проблема вишекритеријумске оптимизације могу да се поделе у три групе на основу тога да ли се и како информација о преференцијама доносиоца одлука укључује у процес оптимизације: априори, апостериори и интерактивне методе. Код априори метода, као резултат методе добијамо једно решење које одговара преференцијама доносиоца одлука. На пример, функцијама циља се могу доделити тежине и онда оптимизовати тежинска сума. Код апостериори метода, циљ је пронаћи сва компромисна решења и понудити их доносиоцу одлука који онда на основу неке друге методе (на пример методама из теорије одлучивања) бира решење које му највише одговара. Интерактивне методе укључују доносиоца одлука у сам процес оптимизације и претраге за компромисним решењима [23]. Даље, као и за оптимизацију са једном функцијом циља, могу се користити егзактне методе или (мета)хеуристичке. У овој дисертацији, фокус је на двокритеријумским проблемима комбинаторне оптимизације, а за њихово решавање користе се апостериори метахеуристичке методе.

Како би се формално дефинисао појам компромисног решења, уводи се Парето¹ оптималност (ефикасност) као фундаментални концепт вишекритеријумске оптимизације. Решење је Парето оптимално ако побољшање једне функције циља доводи до погоршања друге функције циља, што значи да не можемо више побољшати једну или више функција циља, а да не покваримо неку другу функцију. Другим речима, не постоји решење које је боље од тог решења по свим функцијама циља истовремено.

Формално, за вектор функција циља $\mathbf{f} = (f_1, f_2, \dots, f_M)$ и њихову минимизацију, допустиво решење \mathbf{x} је Парето оптимално ако важи:

$$\nexists \mathbf{x}' \text{ тако да } \forall i : f_i(\mathbf{x}') \leq f_i(\mathbf{x}) \text{ и } \exists j : f_j(\mathbf{x}') < f_j(\mathbf{x}),$$

где $f_i(\mathbf{x})$ представља вредности функције циља f_i које одговарају решењу \mathbf{x} .

У циљу описивања метода решавања вишекритеријумских проблема оптимизације уводи се појам Парето доминантности. Прецизније, у случају да важи $\forall i : f_i(\mathbf{x}) \leq f_i(\mathbf{x}')$ и $\exists j : f_j(\mathbf{x}) < f_j(\mathbf{x}')$, онда се са $\mathbf{x}' \prec \mathbf{x}$ значава да решење x доминира над решењем x' .

За разлику од оптимизације са једном функцијом циља, као решење вишекритеријумског проблема се добија скуп, а не само једно решење. Парето

¹Вилфредо Парето (1848-1923) је био италијански инжењер, социолог, економиста и филозоф.

оптималан скуп је скуп свих Парето оптималних решења неког проблема. Овакав скуп се још назива и Парето фронт. Решење \mathbf{x}^* је Парето оптимално у односу на скуп S ако не постоји неко друго решење $\mathbf{x} \in S$ тако да \mathbf{x} Парето доминира над \mathbf{x}^* . Формално, Парето оптималан скуп P се дефинише као:

$$P = \{\mathbf{x}^* \in S \mid \nexists \mathbf{x} \in S \text{ тако да } \mathbf{x} \prec \mathbf{x}^*\}$$

Проналажење Парето фронта или његове апроксимације је основни задатак вишекритеријумске оптимизације.

Разматра се вишекритеријумски проблем са M функција циља, где се врши минимизација функција и постоји n променљивих одлучивања, а допустиви регион простора решења је дефинисан скупом ограничења. У општем случају проблем вишекритеријумске оптимизације може бити дефинисан на следећи начин:

$$\begin{aligned} & \text{Минимизовати} && \mathbf{f}(\mathbf{x}) \\ & \text{при ограничењима} && \mathbf{g}(\mathbf{x}) \leq \mathbf{0} \\ & && \mathbf{h}(\mathbf{x}) = \mathbf{0}, \end{aligned}$$

где су:

- $\mathbf{x} = (x_1, x_2, \dots, x_n)$ вектор променљивих одлучивања.
- $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_M(\mathbf{x}))$ вектор функција циља које треба оптимизовати.
- $\mathbf{g}(\mathbf{x}) \leq \mathbf{0}$ ограничења типа неједнакости, $\mathbf{g}(x) = (g_1(x), \dots, g_k(x))$.
- $\mathbf{h}(\mathbf{x}) = \mathbf{0}$ ограничења типа једнакости, $\mathbf{h}(x) = (h_1(x), \dots, h_p(x))$.

У литератури се могу наћи различите априори методе. Тежинске методе су оне где се унапред одређују тежине које се користе за комбиновање више функција циља у једну, где је најпознатија метода тежинске суме (енгл. *weighted-sum*) [61]. Лексикографске методе дају приоритет функцијама циља, тако што се прво оптимизују оне са вишим приоритетом [118]. Методе засноване на циљевима, као што је метода циљаног програмирања (енгл. *goal programming*), настоје да смање нежељено одступање од унапред задатих циљних вредности за сваку од функција циља [174]. Методе засноване на референтним тачкама (енгл.

reference-point) усмеравају претрагу ка одређеном делу Парето фронта [100]. Методе засноване на функцији корисности (енгл. utility function) користе ову функцију како би се представиле преференције доносиоца одлука [183]. Фази методе укључују фази логику како би се урачунала и непоузданост у преференцијама доносиоца одлука [100].

Слично, у литератури се могу наћи и различити апостериори приступи. Једна од познатих метода је метода епсилон ограничења (енгл. epsilon-constraint), где се једна функција циља оптимизује, а остале претварају у ограничења [118]. Нормализована варијанта методе нормалних ограничења (енгл. Normalized normal constraint method) укључује нормализовање функција циља како би се избегли проблеми са различитим величинама и примену нормалних ограничења. Изменама нормалних ограничења могу се генерисати Парето решења која су равномерно распоређена на фронту [128]. У литератури се могу наћи и вишекритеријумске варијанте класичних егзактних метода оптимизације, као што су линеарно програмирање [111] и метода гранања и ограничавања [146]. Поред тога, примењују се и алгоритми засновани на динамичком програмирању [74]. Метода нормалног пресека граница (енгл. Normal boundary intersection) представља алтернативу методи епсилон ограничења где се применом скаларизације генерише равномерно распоређен фронт [164]. Код сукцесивне Парето оптимизације (енгл. successive Pareto optimization), решења се генеришу корак по корак на основу претходних решења [141].

У овој дисертацији коришћене су апостериори метахеуристичке методе које проналазе довољно добру апроксимацију правог Парето фронта у разумном времену. Ово омогућава доносиоцу одлука да ефикасно добије разноврстан скуп компромисних решења од којих може изабрати оно које му највише одговара у датој ситуацији. Развој и примене оваквих метода представљају активну област истраживања, а у релевантној литератури су посебно заступљени еволутивни алгоритми [110].

1.2 Локацијски проблеми

Донносиоци одлука користе различите доступне информације како би донели одлуке у складу са неким циљем. У случају да су информације при одлучивању просторне (географске), добијамо локацијске проблеме, где је потребно одредити оптималне локације за неке објекте. Различити појмови се односе на

ову област, као што су локацијски проблеми, локацијски модели и локацијска анализа или теорија. Овакви проблеми се често јављају у области операционих истраживања и науке о управљању. Како овакви проблеми у суштини представљају математичке проблеме, њихова дефиниција, анализа и решавање спадају и у област примењене математике, математичког моделирања и оптимизације. Поред тога, решавање оваквих проблема у реалним ситуацијама је комплексно и временски захтевно, тако да развој различитих алгоритама за ефикасно решавање оваквих проблема представља активну област рачунарства.

Локацијски проблеми су и пре више векова били део истраживања из разних области, као што су географија, економија и транспорт. Међутим, као први проблем локацијске анализе као посебне области наводи се Веберов проблем који је дефинисао Алфред Вебер 1909. године. У овом проблему треба наћи оптималну локацију једног објекта, тако да тежинска сума удаљености од свих корисника до тог објекта буде минимална [57].

Основни елементи локацијских проблема су локације, објекти и корисници (потрошачи). У погледу топографије простора, локацијске проблеме можемо поделити на континуалне, дискретне и мрежне. Код континуалних простор где се врши лоцирање неких објеката је описан континуалним променљивим, тј. координатама. Код оваквих проблема не постоји листа коначног броја локација кандидата, већ је саме кандидате потребно пронаћи, али тако да они буду део допустивог региона простора. Код дискретних, циљ је изабрати локације за нове објекте из коначног скупа потенцијалних локација. Мрежни локацијски проблеми представљају посебну класу проблема где је позната мрежа, тј. потенцијалне локације за постављање објеката и локације корисника су чворови графа. Овакви проблеми могу да укључе дискретне променљиве које одговарају чворовима графа, као и континуалне које одговарају ивицама графа.

Код локацијских проблема кључно је дефинисати метрику растојања, често је то једноставно Еуклидско растојање. Функције циља у том случају укључују растојања било између постављених објеката или између корисника и објеката. На пример, циљ може бити минимизација укупних растојања или максималног растојања између објеката и корисника, у циљу ефикасног снабдевања. Са друге стране, циљ може бити максимизација растојања између различитих објеката, ово може бити захтев у случају одређивања локација опасних постројења или складишта опасних материја (на пример хемијских или нуклеарних) који не смеју бити близу.

Даље, на основу броја објеката које треба успоставити као параметра, проблеми се могу поделити на ендogene и екзогене. Код ендогених се овај параметар одређује у самом процесу оптимизације, док је код екзогених он унапред задат. Проблеми могу бити неограничених капацитета или са ограничењима капацитета. На пример, може се задати максимални капацитет објеката или гране којом се транспортује проток између објеката и корисника. Проблеми могу укључити трошкове успостављања објеката и трошкове везе између објекта и корисника. Трошкови успостављања објеката могу бити фиксни, који не зависе од тога колико се објекат користи, или променљиви, који зависе од нивоа коришћења објекта. Слично томе, трошкови везе између објекта и корисника могу бити фиксни, који се јављају једнократно приликом успостављања везе, или променљиви, који зависе од нивоа употребе везе, на пример количине протока.

Међу првим применама локацијских проблема у операционим истраживањима су распоређивање објеката јавног или приватног сектора, на пример школа, болница или продавница. Последњих година, овакви проблеми налазе још различитих примена које се односе на реаговање у ванредним ситуацијама, природним непогодама, акцидентима, као и примене фокусиране на очување животне средине, управљање отпадом, уштедом енергије и генерално смањивањем загађења.

Неки од најпознатијих и највише изучаваних локацијских проблема из литературе су:

- Локацијски проблем без капацитета (енгл. Uncapacitated facility location problem - UFLP): Циљ је наћи оптималне локације фиксног броја објеката тако да се минимизују трошкови транспорта до потрошача.
- Проблем p -центра (енгл. p -center problem): Поставити p објеката и притом минимизовати максималну удаљеност потрошача од најближег објекта.
- Проблем p -медијане (енгл. p -median problem): Поставити p објеката и притом минимизовати укупно тежинско растојање између сваког потрошача и његовог најближег објекта (тј. објекта којем је он додељен). Тежине одговарају потражњи потрошача.
- Хаб-локацијски проблем (енгл. Hub-location problem): Постоје специјални објекти између којих је транспорт јефтинији. Циљ је успостављањем хабова на оптималним локацијама смањити укупне трошкове транспорта.

- Проблем покривања скупа (енгл. Set covering problem): Потребно је пронаћи минималан број објеката који ће покривати све потрошаче.
- Проблем максималног покривања локација (енгл. Maximal covering location problem): Поставити највише p објеката тако да се максимизује укупна покривена потражања потрошача.

Преглед локацијских проблема и метода за њихово решавање може се наћи у радовима [14, 30, 66, 153, 165].

Локацијски проблеми са више функција циља

У применама из реалног света, доносиоци одлука желе да оптимизују више функција циља како би ефикасно искористили ресурсе и постигли различите циљеве. Како неки циљеви могу бити супростављени добијамо вишекритеријумске варијанте локацијских проблема. Често се као супростављени циљеви јављају трошкови постављања објеката на локације и трошкови функционисања тих објеката после постављања, а са друге стране профит који они доносе, на пример количина покривене потражње корисника. Ово углавном важи за случај где су доносиоци одлука приватне компаније, док јавни сектор тежи да оптимизује заједничке интересе свих корисника, где се јављају и друге функције циља поред профита, на пример равноправно пружање услуга [99].

У радовима који се баве локацијским проблемима са више функција циља, могу се наћи функције циља које укључују неке од следећих вредности: укупну цену постављања, цену годишњег рада или одржавања, удаљеност од постојећих објеката, број објеката, време одзива објекта, покривену потражњу корисника, удаљеност корисника објекта (просечна, максимална). У последње време, јављају се и циљеви који се односе на очување животне средине, као што су смањење потрошње енергије и загађења. Наравно, за све ове циљеве, тј. критеријуме, основни проблем је како их измерити [68].

Један од првих радова који се бави вишекритеријумским локацијским проблемима је [143], где су дефинисане вишекритеријумске варијанте оригиналног Веберовог проблема. У литератури постоје различите вишекритеријумске варијанте познатих локацијских проблема, а неке од њих су:

- Двокритеријумски проблем p -медијане са опасним објектима: Треба поставити тачно p објеката што даље од корисника, а у исто време и што

даље једне од других [42, 159].

- Проблем k -балансираног центра: Задатак је отворити тачно k објеката са циљем да се минимизује максимално растојање између било којег корисника и његовог најближег објекта, а да се при томе оствари максимални баланс између објеката [49].
- Вишекритеријумски локацијски проблем без капацитета: Прва функција циља је укупна или просечна удаљеност корисника од објеката, друга функција је укупан број покривених корисника, док трећа представља максимално растојање између корисника и објеката. Прву и трећу функцију циља треба минимизовати, док другу треба максимизовати. Проблем је погодан за примене код лоцирања објеката хитних служби [99].
- Вишекритеријумски проблем покривања скупа: Свака функција циља има посебне коефицијенте придружене локацијама. Све функције циља укључују укупну суму коефицијената локација са отвореним објектима коју треба минимизовати [72, 95].

Вишекритеријумски локацијски проблеми налазе бројне примене у решавању проблема реалног света, на пример: распоређивање привремених локација за складиштење отпада у случају катастрофа и ванредних ситуација [145], распоређивање возила хитне помоћи [7], постављање објеката уз смањење утицаја на животну средину [88], постављање различитих медицинских ресурса у току епидемије [64]. Преглед вишекритеријумских локацијских проблема и метода за њихово решавање може се наћи у раду [68].

1.3 Егзактне методе оптимизације

Теоријски, егзактне методе оптимизације се реализују кроз коначан број итерација (корака) и дају решење чија је оптималност загарантована. Међутим, у пракси се може десити да егзактна метода не пронађе оптимално решење услед ограниченог времена извршавања или недостатка меморијских ресурса. Технике решавања, односно кораци егзактне методе, зависе од типа проблема оптимизације који се разматра.

Развијање нових и унапређење постојећих егзактних метода је активна област. Егзактне методе и алгоритми описани у овом одељку, као они примењени за

решавање проблема ове дисертације могу се користити у оквиру комерцијалних решавача, као што су IBM ILOG CPLEX, LINGO, FICO Xpress, Gurobi и други. Овакви софтвери пружају могућност ефикасног решавања различитих проблема комбинаторне и континуалне оптимизације и често се користе у литератури за испитивање перформанси и квалитета решења апроксимативних алгоритама.

Симплекс метод представља основну технику за решавање проблема линеарног програмирања у случају континуалне (глобалне) оптимизације [163]. Основна идеја је да се кроз више итерација истражују ивице простора решења и да се у сваком кораку побољшава вредност функције циља. Међутим, ова техника не може да се примени на проблеме комбинаторне оптимизације, где је потребно претражити дискретан простор решења.

Метода гранања и ограничавања (енгл. branch-and-bound) се користи за решавање проблема целобројног програмирања. Основна идеја ове методе је подела проблема на мање потпроблеме који су једноставнији за решавање. Простор решења се претражује систематски, где се гране које не могу довести до побољшања решења елиминишу. Ограничења се рачунају на основу решавања релаксираног проблема, тј. проблема где се дозвољава да вредности променљивих не буду целобројне [182].

Метода гранања и одсецања (енгл. branch-and-cut) представља побољшање методе гранања и ограничавања, тако што се на основу особина допустивог простора решења уводе равни одсецања које представљају нова ограничења [182]. Идеја је да се сузи простор решења релаксираног проблема целобројног линеарног програмирања. Ова метода се често користи у оквиру комерцијалних егзактних решавача.

Метода гранања и оцењивања (енгл. branch-and-price) представља варијанту методе гранања и ограничавања где се у сваком чвору линеарна релаксација решава методом генерисања колона (енгл. column generation) [13]. Ова метода је погодна за решавање комплексних проблема чија структура се може искористити за декомпозицију проблема на мање потпроблеме. У случају да се код ове методе користе и равни одсецања, добија се метода гранања оцењивања и одсецања (енгл. branch-price-and-cut).

Егзактне методе за вишекритеријумску оптимизацију

Егзактно решавање проблема оптимизације са више функција циља је још комплексније у односу у односу на случај проблема једнокритеријумске оптимизације. За овакве вишекритеријумске проблеме постоје различите егзактне методе предложене у литератури, а развој нових метода и унапређивање постојећих је активна област истраживања [46, 122]. Егзактне методе за вишекритеријумску оптимизацију најчешће користе егзактне методе за оптимизацију проблема са једном функцијом циља како би решили проблем са више функција циља [100]. Две најпознатије технике су метода тежинске суме и метода ε -ограничења.

Метода суме са тежинама додељује свакој функцији циља одређену тежину и затим сабира вредности свих функција циља помножених одговарајућим тежинама. При томе сви циљеви треба да буду истог типа, тј. минимизација или максимизација. На тај начин се проблем вишекритеријумске оптимизације своди на проблем са једном функцијом циља. Тежине се могу бирати на основу преференци доносиоца одлука, а могуће је и мењати тежине у више покретања алгорита и тако генерисати различита Парето решења. На пример, ако су f_i функције циља које треба минимизовати, а w_i тежине, тада је нова функција коју треба минимизовати:

$$F(\mathbf{x}) = \sum_{i=1}^M w_i f_i(\mathbf{x}),$$

где је M број функција циља, w_i тежина додељена функцији циља f_i , а $f_i(\mathbf{x})$ вредност функције циља i .

Метода ε -ограничења претвара изабране функције циља у ограничења, док се једна од функција циља оптимизује [121]. Овим се проблем своди на проблем са једном функцијом циља. Општи облик проблема који се решава овом методом:

$$\begin{array}{ll} \text{минимизовати} & f_1(\mathbf{x}) \\ \text{при ограничењима} & f_i(\mathbf{x}) \leq \varepsilon_i, \quad i = 2, 3, \dots, M, \end{array}$$

Вредности ε_i се могу мењати и на тај начин добити различита Парето оптимална решења. Основни проблеми код примене ове методе су како изабрати која функција се оптимизује и како изабрати на које вредности ограничити

остале функције циља. Преглед и детаљнији опис егзактних метода за више-критеријумску оптимизацију може се наћи у књизи [41].

1.4 Метахеуристичке методе

Овакве методе не могу да гарантују оптималност, тако да се могу сврстати у групу апроксимативних метода. Међутим, за разлику од апроксимативних алгоритама који гарантују да добијено решење неће одступати од оптималног за неки фактор, метахеуристике не дају никакву гаранцију за квалитет решења. У односу на хеуристике чији су кораци креирани према карактеристикама једног конкретног проблема који се решава, метахеуристике представљају опште оквири за решавање проблема оптимизације. Метахеуристике могу укључити хеуристике као помоћне кораке у решавању проблема. Основна идеја метахеуристике је добити довољно добро решење за прихватљиво време. Које решење је довољно добро и колико рачунарског времена је прихватљиво за решавање зависи од конкретних проблема и примена.

За успешну имплементацију метахеуристике, неопходно је обезбедити баланс између експлорације и експлоатације простора решења, као и механизам за избегавање локалних оптимума. Први радови у којима је коришћен концепт метахеуристике укључују истраживања из стохастичке оптимизације [155], еволутивних процеса [150] и случајне претраге [149]. Један од кључних момената за развој и популаризацију ове области јесте увођење генетских алгоритама [90].

Метахеуристичке методе се примењују на комплексне проблеме оптимизације у разним доменама, посебно када су проблеми НП-тешки, где су егзактне методе рачунарски веома захтевне или када саме функције циља немају погодан облик за примену егзактних метода [22].

Метахеуристике су нашле своје примене у разним областима, на пример: решавање основних проблема комбинаторне оптимизације као што је проблем трговачког путника [63], проблема операционих истраживања, разних инжењерских проблема [184], а у скорије време и проблема из области вештачке интелигенције, као што је подешавање хиперпараметара модела машинског учења или селекција атрибута или самих модела [144].

До сада су у литератури предложене бројне метахеуристике у областима комбинаторне и глобалне оптимизације. Метахеуристике могу бити класификоване у групе према различитим карактеристикама. Основна подела метахеу-

ристика је на методе са једним решењем (енгл. single-solution based) и методе са популацијом решења (енгл. population based) [173]. Метакхеуристике се такође могу класификовати и на основу других критеријума, на пример: детерминистичке или стохастичке, биолошки инспирисане или математички засноване, итеративне или конструктивне, са употребом меморије или без памћења/учења.

Метакхеуристике засновне на побољшању једног решења се називају и методе са путањом, тј. трајекторијом. Овакве методе креирају једно почетно решење које се онда итеративно побољшава док се не дође до оптималног или приближног решења које је прихватљиво или док се не достигне неки други критеријум заустављања (максимално време извршавања, максималан број итерација, максималан број итерација без побољшања најбољег решења итд). Итеративни процес се обично заснива на истраживању суседних решења и преласку у та решења у случају побољшања функције циља. [22]. Најпознатије метакхеуристике из ове групе су:

- **Итеративна локална претрага** (енгл. Iterated Local Search - ILS) [113]: Ова метода представља општи оквир за комбиновање фазе локалне претраге и фазе стохастичке промене решења. Почетно решење се генерише на случајан начин или неком хеуристиком, а затим се примењује локална претрага. Овако добијено решење, тј. локални минимум се мења стохастичком процедуром, тј. врши се пертурбација решења након чега се поново покреће локална претрага из измењеног решења. Ново решење се може прихватити или одбацити, у зависности од дефинисаног критеријума. Овај поступак се понавља док се не испуни неки од критеријума заустављања. Основни проблем је како изабрати добру пертурбацију решења, тако да се претрага не заглави у локалном оптимуму, а да се промена решења не сведе на скроз ново насумично решење.
- **Симулирано каљење** (енгл. Simulated Annealing - SA) [103]: Ова метода је инспирисана процесом у металургији, где се метали загревају, а затим постепено хладе како би се постигле жељене карактеристике метала. Алгоритам са одређеном вероватноћом прихвата и лошија решења, где се та вероватноћа временом смањује. Вероватноћа зависи од степена деградације лошијег решења у односу на текуће, као и од тренутне температуре, тј. броја итерација. На пример, ако је циљ минимизација функције $f(x)$ и ако са T означимо тренутну температуру, а са $\Delta E = f(x') - f(x)$ ра-

злику вредности функција циља новог x' и текућег решења x , онда се вероватноћа прихватања рачуна на следећи начин:

$$P(\Delta E, T) = \begin{cases} 1 & \text{ако је } \Delta E < 0 \\ e^{-\Delta E/T} & \text{ако је } \Delta E \geq 0 \end{cases}$$

- **Табу претрага** (енгл. Tabu Search - TS) [76]: Основна идеја ове методе је чување решења (или његових елемената, тј. атрибута) у табу листи, како би се избегло понављање, тј. циклуси и истражили нови делови простора решења. Решења или њихови елементи се чувају у листи одређени број итерација, што представља један од параметара методе. Дужина табу листе може бити још један параметар методе или се може мењати динамички током претраге. Може се дефинисати и аспирацијска функција, које дефинише критеријум на основу којег се прихватају и табу решења, на пример ако је решење боље од тренутно најбољег, оно се прихвата. Поред тога, могу се користити и различите стратегије за усмеравање претраге, које користе средњорочну или дугорочну меморију. На пример, чување елемената решења који се често појављују током претраге или чување елемената решења високог квалитета.
- **Похлепна стохастичко-адаптивна метода претраге** (енгл. Greedy Randomized Adaptive Search Procedure - GRASP) [70]: Свака итерација GRASP методе састоји се од две фазе: фазе конструкције решења и фазе локалног претраживања. Основна идеја прве фазе је комбиновање случајности и похлепности код конструкције решења. Уместо да се у сваком кораку конструкције решења додаје најбољи елемент, тј. елемент који највише доприноси функцији циља, елемент се бира на случајан начин из редуковане листе кандидата. Ова листа садржи подскуп кандидата (елементе решења) који су изабрани у односу на допринос функцији циља. Случајним избором елемента решења из редуковане листе кандидата, обезбеђује се баланс између похлепности и насумичности конструкције решења. У другој фази, решење конструисано у првој фази се даље побољшава процедуром заснованом на локалном претраживању. Излаз из ове фазе је решење једне итерације GRASP-а, а најбоље решење добијено током свих итерација је резултат GRASP методе.

- **Метода променљивих околина** (енгл. Variable Neighborhood Search - VNS) [131]: Ова метода систематски мења околине решења приликом претраге. Дефинисање околина, њихових величина, као и редоследа њихове употребе представља први корак пре извршавања алгоритма. Током извршавања, итеративно се комбинују две фазе, фаза размрдавања и фаза локалне претраге заједно са кораком одлучивања о промени околине. Прва фаза случајним избором решења из околине текућег решења има улогу диверсификације, чиме се смањује могућност да алгоритам заврши у локалном оптимуму. Почевши од тако изабраног решења, примењује се друга фаза која покушава да побољша тренутно решење. Уколико је дошло до побољшања решења, тренутно решење се ажурира новим, а претрага се наставља у првој околини. У супротном, претрага се наставља у следећој околини. Како је метода променљивих околина коришћена у овој дисертацији, детаљнији опис ове методе и њених варијанти дат је у наставку.

Метахеуристике засноване на популацији решења раде над скупом тренутних решења који се назива популација. Ове метахеуристике су обично инспирисане понашањем јединки у популацији, процесима у природи, начином функционисања живих организама и слично. Свака јединка у популацији одговара решењу у простору претраге и њен квалитет се мери коришћењем функције прилагођености (то може бити функција циља или нека друга функција која адекватно одсликава квалитет решења). Овакав приступ решавању проблема може довести до повећане експлорације простора решења и повећати степен диверсификације претраге што повећава шансе за избегавање локалних оптимума [173].

У наставку је наведен кратак преглед често коришћених популационих метахеуристика из литературе.

- **Генетски алгоритми** (енгл. Genetic Algorithms - GA): Алгоритам је инспирисан процесом биолошке еволуције. Јединка се представља генетским кодом који одговара једном решењу. Скуп јединки се назива популација. У једној итерацији, применом оператора селекције, мутације и укрштања на јединке тренутне популације, добија се нова популација (генерација) јединки. Оператором селекције се бирају јединке из тренутне популације које ће учествовати у креирању јединки нове популације. Селекција

је стохастичка, где је већа вероватноћа да буду изабране боље јединке. Мутација представља мале промене генетског кода јединке са неком вероватноћом. Оператор укрштања врши комбиновање генетског кода две јединке како би се добила нова јединка. Укрштање је инспирисано природним комбиновањем генетског материјала родитеља. Основна примена је за дискретну оптимизацију, али се може прилагодити и непрекидној, уз различите начине кодирања решења [89]. Како су неки од алгоритама коришћених у овој дисертацији засновани на генетском алгоритму, овај алгоритам ће бити детаљније описан у наставку.

- **Оптимизација ројем честица** (енгл. Particle Swarm Optimization - PSO): Метода је инспирисана понашањем ројева јединки као што су птице или инсекти. Свака јединка је идентификована као једна честица у роју (популацији) и носи следеће информације: положај честице (који одговара решењу проблема), најбољи положај честице током процеса претраге (компонента сопственог искуства) и најбољи положај свих честица или неке околине дате честице (социјална компонента). У току претраге, честице се крећу кроз простор решења, где свака честица мења своју локацију на основу сопственог или искуства целог роја, као и инерције (правца кретања под претпоставком да нема других утицаја). Кретање честице представља промену тог решења. Метода укључује различите параметре (три основна су тежина инерције, когнитивни коефицијент и социјални коефицијент) чије вредности могу значајно да утичу на ефикасност претраге и квалитет решења. Оригинално, ова метода је предложена за проблеме континуалне оптимизације, али се може прилагодити и за решавање дискретних проблема [101].
- **Еволутивне стратегије** (енгл. Evolutionary Strategies - ES): Још један алгоритам инспирисан еволуцијом, где је фокус на мутацији, док је укрштање опционо. Параметар јачине мутације је адаптиван и мења се током извршавања алгоритма. Селекција је детерминистичка и елитистичка где најбоља решења пролазе у следећу итерацију. Основна примена је за непрекидну оптимизацију. [18]
- **Диференцијална еволуција** (енгл. Differential Evolution - DE): Алгоритам инспиран природним процесом еволуције јединки у популацији, где се за креирање нове популације јединки користе тежинске разлике вектора

решења тренутне популације у свакој итерацији. Највише се користи за континуалну оптимизацију [171].

- **Оптимизација мрављим колонијама** (енгл. Ant Colony Optimization - АСО): Метода инспирисана понашањем мрава у потрази за храном. Мрави током потраге остављају трагове феромона на својим путањама које онда остале јединке могу да прате. Метода је погодна за дискретне проблеме, посебно за проблеме на графовима. Решења се граде инкрементално додавањем нових елемената у решење, где на избор утиче искуство мрава (феромони) у грађењу претходних решења. У случају да је изграђено решење добро, појачава се феромон за избор његових елемената, тј. вероватноћа да ће се ти елементи изабрати приликом изградње новог решења. [55]
- **Оптимизација колонијом пчела** (енгл. Bee Colony Optimization - ВСО): Метода је инспирисана понашањем колоније пчела током потраге за храном. Слично као код оптимизације мрављим колонијама, решења се граде инкрементално, а искуство одређених пчела може утицати на избор елемената приликом конструкције решења осталих пчела. Након конструкције (парцијалних) решења и њихове евалуације, свака пчела одлучује да ли ће остати лојална, тј. задржати (парцијално) решење које је конструисала или не. Пчеле које задржавају своје решење постају регруттери и њихово решење могу да преузму неопредељене пчеле, тј. оне које нису остале лојалне свом решењу. Бољи квалитет (парцијалног) решења значи већу вероватноћу да ће пчела остати лојална свом решењу. За разлику од основне верзије где се решења граде инкрементално, код варијанте са побољшањем (ВСОi) јединке на почетку добијају комплетна решења која побољшавају кроз итерације [48].

У последње време приметан је пораст радова који уводе нове метахеуристике засноване на понашању организама, на пример китова, лисица, сова, којота, али и људи, њихових послова и односа и других појава и процеса у природи. Овакви радови су такође критиковани од стране неких аутора због непотребне употребе метафора из реалног света, као и понављања истих идеја само описаних на различите начине или уз минималне измене [6].

Метода променљивих околина и њене варијанте

Метода променљивих околина је заснована на следеће три чињенице [131]:

- Локални оптимум у односу на једну структуру околине није нужно локални оптимум за неку другу структуру околине.
- Глобални оптимум је локални оптимум у односу на све структуре околина.
- За многе проблеме, локални оптимуми у односу на једну или више околина су релативно близу једни другима.

Стога, променом околина током претраге могу се избећи локални оптимуми и евентуално достићи глобални. Ова метода је нашла различите примене због своје једноставности и ефикасности [24, 84].

Алгоритам почиње од почетног генерисаног решења и затим комбинацијом размрдавања и локалне претраге у различитим околинама покушава да побољша тренутно најбоље решење. Основни корак код примене ове методе на конкретан проблем јесте дефинисање околина $N_k, k = 1, \dots, k_{max}$. Основна варијанта методе може се представити Алгоритмом 1.

Алгоритам 1 Основна метода променљивих околина

```
1: Input:  $x_0, k_{max}$ 
2: Output:  $x$ 
3:  $x \leftarrow x_0$ 
4: while stopping criteria not met do
5:    $k \leftarrow 1$ 
6:   while  $k < k_{max}$  do
7:      $x' \leftarrow Shake(x, k)$ 
8:      $x'' \leftarrow LocalSearch(x')$ 
9:     if  $f(x'') < f(x)$  then
10:       $x \leftarrow x''$ 
11:       $k \leftarrow 1$ 
12:     else
13:       $k \leftarrow k + 1$ 
```

Почетно решење се може генерисати на случајан начин или применом неке похлепне хеуристике у складу са карактеристикама решаваног проблема. Фаза размрдавања (означена са Shake) представља случајан избор новог решења из околине тренутног решења $N_k(x)$. Локална претрага (означена са LocalSearch)

1.4. МЕТАХЕУРИСТИЧКЕ МЕТОДЕ

се може вршити на различите начине, на пример стратегијом првог или најбољег побољшања. Локална претрага може да се врши у произвољној околини $N(x)$, независно од околине за фазу размрдавања. Локална претрага почиње од решења добијеног из фазе размрдавања. Корак локалне претраге се може имплементирати као Алгоритам 2, где се користи стратегија првог побољшања решења или као Алгоритам 3, где се користи стратегија најбољег побољшања.

Алгоритам 2 Локална претрага са првим побољшањем

```
1: procedure LOCALSEARCH( $x$ )
2:    $improvement \leftarrow \text{true}$ 
3:   while  $improvement$  do
4:      $improvement \leftarrow \text{false}$ 
5:     for  $x' \in N(x)$  do
6:       if  $f(x') < f(x)$  then
7:          $x \leftarrow x'$ 
8:          $improvement \leftarrow \text{true}$ 
9:       break
10:  return  $x$ 
```

Алгоритам 3 Локална претрага са најбољим побољшањем

```
1: procedure LOCALSEARCH( $x$ )
2:    $improvement \leftarrow \text{true}$ 
3:   while  $improvement$  do
4:      $improvement \leftarrow \text{false}$ 
5:      $best\_neighbor \leftarrow x$ 
6:     for  $x' \in N(x)$  do
7:       if  $f(x') < f(best\_neighbor)$  then
8:          $best\_neighbor \leftarrow x'$ 
9:          $improvement \leftarrow \text{true}$ 
10:     $x \leftarrow best\_neighbor$ 
11:  return  $x$ 
```

Поред основне, постоје и друге варијанте методе променљивих околине. Редукована варијанта (енгл. Reduced VNS) се добија избацавањем локалне претраге из основне (брисање линије 6 из Алгоритма 1). Код адаптивне варијанте (енгл. Skewed VNS) врши се прихватање и лошијег решења чија је вредност функције циља близу вредности функције циља тренутног решења, ако је то решење далеко од тренутног у простору решења. Ова варијанта је приказана као Алгоритам 4, где је $d(x, x'')$ растојање између решења x и x'' , а α параметар методе који одређује колико то растојање утиче на функцију циља.

Алгоритам 4 Адаптивна метода променљивих околина

```
1: Input:  $x_0, k_{max}, \alpha$ 
2: Output:  $x_{best}$ 
3:  $x \leftarrow x_0$ 
4:  $x_{best} \leftarrow x_0$ 
5: while stopping criteria not met do
6:    $k \leftarrow 1$ 
7:   while  $k < k_{max}$  do
8:      $x' \leftarrow Shake(x, k)$ 
9:      $x'' \leftarrow LocalSearch(x')$ 
10:    if  $f(x'') < f(x_{best})$  then
11:       $x_{best} \leftarrow x''$ 
12:    if  $f(x'') - \alpha \cdot d(x, x'') < f(x)$  then
13:       $x \leftarrow x''$ 
14:       $k \leftarrow 1$ 
15:    else
16:       $k \leftarrow k + 1$ 
17: return  $x_{best}$ 
```

Локална претрага која користи различите околине током извршавања назива се метода променљивог спуста (енгл. Variable Neighborhood Descend) и приказана је као Алгоритам 5. Уопштена варијанта методе променљивих околина (енгл. General VNS) има исту структуру као и Алгоритам 1, али користи методу променљивог спуста уместо обичне локалне претраге (линија 6). За имплементацију ове варијанте потребно је дефинисати околине за методу променљивог спуста $N_l, l = 1, \dots, l_{max}$, које могу бити различите од околина дефинисаних за фазу размрдавања.

Алгоритам 5 Метода променљивог спуста

```
1: Input:  $l_{max}, x$ 
2: Output:  $x$ 
3:  $l \leftarrow 1$ 
4: while  $l < l_{max}$  do
5:    $x' \leftarrow$  best solution from  $N_l(x)$ 
6:   if  $f(x') < f(x)$  then
7:      $x \leftarrow x'$ 
8:      $l \leftarrow 1$ 
9:   else
10:     $l \leftarrow l + 1$ 
11: return  $x$ 
```

Методе засноване на VNS-у су успешно примењене на различите локацијске проблеме [43, 53, 94, 115, 161], укључујући и проблеме сличне MCLP [44, 47, 129]. Детаљан опис различитих варијанти методе променљивих околина, као и њихове примене могу се наћи у радовима [84] и [86].

Генетски алгоритам

Решење проблема се кодира на начин који зависи од проблема који се решава (на пример као бинарни низ), а код решења се назива јединка (хромозом). Функција прилагођености одређује колико је нека јединка добра и обично одговара функцији циља или њеној модификацији. Алгоритам почиње од иницијалне популације (скупа јединки) која се обично генерише стохастички, затим се кроз имитацију процеса еволуције, операторима селекције, укрштања и мутације добијају нове популације у нади да ће се међу њима наћи све боље јединке, тј. решења која ће довести до оптималних. Основни кораци ове методе су приказани као Алгоритам 6 [78].

Алгоритам 6 Генетски алгоритам

- 1: **Input:** initial population of individuals
 - 2: **Output:** best fitted individual
 - 3: **while** stopping criteria not met **do**
 - 4: Select parents based on their fitness
 - 5: Perform crossover
 - 6: Perform mutation
 - 7: Evaluate the population
 - 8: Select individuals for the next iteration (replacement)
-

Оператор селекције бира јединке које ће учествовати у креирању нове популације, односно пренети свој генетски материјал у наредну итерацију. Над селектованим јединкама се примењују остали оператори и тако добијају решења која учествују у формирању нове популације. У литератури су предложени различити оператори селекције, међу којима се најчешће користе рулетска и турнирска селекција.

Код рулетске селекције вероватноћа избора јединке је пропорционална функцији прилагођености те јединке. Нека је $P = \{p_1, p_2, \dots, p_n\}$ популација n јединки. Ако је функција прилагођености јединке i једнака $f(p_i)$, онда се веро-

ватноћа $Pr(p_i)$ избора јединке i може израчунати на следећи начин:

$$Pr(p_i) = \frac{f(p_i)}{\sum_{j=1}^n f(p_j)}$$

Код турнирске селекције се на случајан начин бира k јединки из популације, а најбоља од њих у односу на функцију прилагођености пролази даље. Параметром k се на овај начин може одређивати јачина, тј. притисак селекције који утиче на ефикасност решавања. Повећањем величине турнира се смањује шанса да слабија решења учествују у процесу репродукције, тј. генерисања нове популације. Фино градирана варијанта турнирске селекције укључује извршавање турнира различитих величина током једне селекције. Уводи се нови параметар уместо величине турнира који представља жељену просечну величину турнира и који узима реалне вредности [65].

Код селекције засноване на рангу, јединкама се додељује ранг (тј. јединке се сортирају) на основу функције прилагођености. Јединке се бирају на основу додељеног ранга, а не директно на основу вредности функције прилагођености, што доводи до боље диверсификације претраге јер даје више шансе и слабијим јединкама.

Оператор укрштања комбинује два решења у ново решење. Имплементација овог оператора зависи од начина кодирања решења. У литератури, као и код проблема ове дисертације, често је то бинарни низ па се користи једнопозиционо укрштање. Ако су дата два вектора који представљају родитеље r_1 и r_2 , док се тачка укрштања k бира случајно, онда се два потомка o_1 and o_2 креирају на следећи начин:

$$o_1 = \{r_{1,1}, r_{1,2}, \dots, r_{1,k}, r_{2,k+1}, \dots, r_{2,L}\},$$

$$o_2 = \{r_{2,1}, r_{2,2}, \dots, r_{2,k}, r_{1,k+1}, \dots, r_{1,L}\}$$

Могуће је користити и укрштање са две или више позиција, као и друге начине, на пример униформно укрштање где се за сваки бит одређује од којег родитеља се узима.

Оператор мутације има за циљ да доведе до мањих измена новог решења на случајан начин. Ако је кодирање бинарно, где је решење, тј. хромозом $c = \{g_1, g_2, \dots, g_L\}$, мутира се сваки ген, тј. бит g_i са вероватноћом p_m . Код бинарног кодирања мутирање гена представља инвертовање вредности. На

овај начин се добија ново решење $c' = \{g'_1, g'_2, \dots, g'_L\}$, где је:

$$g'_i = \begin{cases} \neg g_i, & \text{са вероватноћом } p_m \\ g_i, & \text{иначе} \end{cases}$$

Још један елемент и битан корак генетских алгоритама представља замену старих решења новогенерисаним. На овај начин се бирају јединке које чине наредну генерацију, односно нову популацију јединки. Политика замене генерација може значајно утицати на ефикасност алгорита, тј. његову конвергенцију. Постоје различити начини за имплементацију овог корака, на пример замена целе генерације новим решењима, замена само неколико најслабијих решења, примена елитизма, где се одређени број решења преноси у следећу генерацију без промена, замена најсличнијих (у односу на неку метрику) решења уместо најслабијих, вероватноћа замене пропорционална функцији прилагођености и слично [130]. Посебан проблем представља одржавање допустивости решења, једна од начина је поправка новог решења до допустивог.

Метахеуристичке методе за вишекритеријумску оптимизацију

Ове методе настоје да у једном покретању дају као резултат добру апроксимацију Парето оптималног скупа решења. Циљ је што више приближити решења правом Парето оптималном скупу и одржати разноликост скупа решења, тј. пронаћи што више недоминираних решења која су равномерно распоређена по фронту [39]. Овакве методе спадају у апостериори технике за решавање проблема вишекритеријумске оптимизације, јер пружају различита компромисна решења, где се онда неком другом техником бира решење које највише одговара доносиоцу одлука, тј. конкретној ситуацији.

Један од првих радова који се бави решавањем проблема вишекритеријумске оптимизације применом метахеуристичких је докторска дисертација из 1984. године [160], где је генетски алгоритама проширен и измењен тако да ради са више функција циља. Тренутно најпопуларније метахеуристичке методе за решавање проблема вишекритеријумске оптимизације у литератури су вишекритеријумски еволутивни алгоритми (енгл. Multi-objective evolutionary algorithms - MOEAs), [2, 110, 175, 179, 186].

Генерално, еволутивни алгоритми за решавање проблема вишекритеријумске оптимизације имају следеће основне кораке:

1. иницијализација,
2. селекција,
3. укрштање и мутација,
4. евалуација,
5. ажурирање популације (и архиве).

Кораци 2-5 се понављају све док се не испуни критеријум заустављања и као резултат се враћа популација недоминираних решења. Основна разлика између ових метода је у начину избора решења за следећу итерацију (популацију решења) и начину рачунања прилагођености решења. У наставку ове главе описани су неки од еволутивних алгоритама познатих у литератури који су у оквиру ове дисертације прилагођени за решавање разматраних проблема и упоређени са решењима добијеним развијеном методом променљивих околина.

NSGA-II [51] - Представља унапређену верзију NSGA (Non-dominated Sorting Genetic Algorithm) алгорита [168] и користи елитизам као и ефикасно недоминирајуће сортирање. NSGA-II превазилази проблеме оригиналног алгорита као што су временска комплексност и употреба дељеног параметра. Код дељеног параметра рачуна се растојање између два решења и уколико је оно мање од задате вредности параметра, решења се означавају као слична. Таквим решењима се смањује функција прилагођености. Већи дељени параметар значи да се тежи да се решења боље распореде по фронту, да не буду збијена. Решења популације се ефикасно сортирају по недоминирајућим фронтовима и додељује им се ранг у зависности којем фронту припадају. Сортирање се врши тако што се у први ранг убаце сва недоминирана решења из популације. Затим се та решења бришу и сва недоминирана решења нове популације прелазе у други ранг. Овај поступак се понавља све док има решења која су доминирана. Приликом селекције, решења се пореде по рангу и метрици гомилања (енгл. *crowding distance*). Циљ коришћења ове метрике је очување разноврсности решења, тј. њихово равномерно распоређивање по фронту.

За дато решење i , метрика (растојање) гомилања d_i се рачуна на следећи начин:

$$d_i = \sum_{m=1}^M \left(\frac{f_m(i+1) - f_m(i-1)}{f_m^{max} - f_m^{min}} \right),$$

где је:

- M број функција циља.
- $f_m(i)$ вредност функције циља f_m за решење i .
- $f_m(i + 1)$ и $f_m(i - 1)$ су вредности функције циља f_m за решења испред и иза решења i у листи решења сортираној на основу функције циља f_m .
- f_m^{max} и f_m^{min} су максималне и минималне вредности функције циља f_m на целој популацији.

Решења за које нека од функција циља има минималну или максималну вредност се називају граничним решењима. За ова решења, вредност метрике гомилања се поставља на бесконачно како би увек била изабрана у следећу генерацију.

SPEA-II [187] - У односу на оригинални SPEA (Strength Pareto Evolutionary Algorithm) алгоритам [188], у SPEA-II укључују се додатни елементи: фина стратегија за рачунање функције прилагођености, техника за естимацију густине решења, као и побољшана верзија методе за одбацавање решења из архиве. Решење се оцењује на основу тога над колико других решења доминира и колико других решења доминира над њим. Поред тога, користи се и информација о густини суседних решења како би се одржала разноврсност решења. Густина се процењује на основу удаљености од k -тог најближег решења, где је k параметар методе, и рачуна се на следећи начин:

$$D(i) = \frac{1}{\sigma_{i,k}^2 + 2},$$

где је $\sigma_{i,k}$ Еуклидово растојање од решења i до његовог k -ог најближег суседа у простору функција циља. Димензије овог простора одговарају функцијама циља, док је свако решење представљено као тачка чије су координате вредности функција циља тог решења.

SMS-EMOA (S-Metric Selection Evolutionary Multiobjective Optimization Algorithm) [17] - Код овог алгоритма се директно оптимизује хиперзапремина решења (енгл. *hypervolume* - HV). Оператор селекције комбинује HV и концепт недоминирајућег сортирања. Приликом селекције користи се допринос јединке вредности HV , који се дефинише као разлика између вредности HV са и без тог решења у скупу. Такође, SMS-EMOA користи стратегију стабилног стања (енгл. *steady-state*), где се једна јединка брише из популације, а једна нова додаје, како би се смањио број рачунања вредности HV . Прво се додаје

нова јединка у популацију, креирана применом оператора укрштања и мутације, а затим се све јединке сортирају по не-доминирајућим фронтovima као и код NSGA-II и на крају се уклања јединка из последњег фронта која најмање доприноси вредности HV. Проверава се да ли је вредност HV нове популације већа или једнака претходној. На овакав начин се обезбеђује да вредност HV никад не опада у новој генерацији.

ε -МОЕА [50] - Метода је заснована на концепту ε -доминантности из рада [105]. Ова метода одржава ε -архиву решења која се користи заједно са тренутном популацијом за примену оператора селекције и укрштања. У архиви се чувају недоминирана решења. Прво се бира једно решење из тренутне популације и једно решење из архиве, а затим се креира ново решење за које се одређује да ли ће бити укључено у популацију и архиву. Посебне процедуре се примењују за селектовање и укључивање како у популацију, тако и у архиву. Приликом селекције користи се бинарна турнирска селекција прилагођена за више функција циља. Случајно се бирају две јединке и ако једна доминира над другом, јединка која доминира пролази даље. У супротном, врши се случајан избор између посматране две јединке. Јединка из архиве се бира на случајан начин. Приликом укључивања решења x у архиву, ново решење се пореди са сваким решењем архиве по критеријуму ε -доминантности. Сваком решењу x се придружује вектор решења \mathbf{A} чије су координате вредности функције циља за посматрано решење, тј. $\mathbf{A} = (f_1(x), f_2(x), \dots, f_M(x))$, где је M број функција циља. Затим се вектор решења \mathbf{A} мења новим $\mathbf{B} = (B_1, B_2, \dots, B_M)^T$, где је сваки елемент вектора \mathbf{B} дефинисан на следећи начин:

$$B_j(f) = \begin{cases} \lfloor (f_j - f_j^{\min})/\varepsilon_j \rfloor, & \text{за минимизацију } f_j, \\ \lceil (f_j - f_j^{\min})/\varepsilon_j \rceil, & \text{за максимизацију } f_j, \end{cases}$$

где је f_j^{\min} најмања вредност функције циља j , а ε_j је дозвољена толеранција за вредност функције циља j .

Овако добијени нови вектори пореде се на основу принципа обичне доминантности. Ако ново решење ε -доминира над неким решењем из архиве, онда се решење из архиве брише, тј. замењује новим решењем. Уколико неко од решења из архиве доминира над новим решењем, онда се ново решење не прихвата. Овакав начин поређења решења омогућава да се постигне добра раширеност решења архиве, јер се не дозвољава да решења буду укључена у архиву уколико су близу постојећих.

Описани еволутивни алгоритми су такође успешно примењивани и на локацијске проблеме у литератури. На пример, Atta и сар. [10] су користили NSGA-II и генетски алгоритам са тежинском сумом (WSGA) за решавање вишекритеријумског локацијског проблема без капацитета и са преференцијама корисника. Karasakal и Silav [98] су предложили измењени SPEA-II алгоритам (mSPEA-II) за двокритеријумски локацијски проблем са парцијалним покривањем. Код предложеног алгоритма, функција прилагођености је измењена и користи се метрика гомилања из NSGA-II. Doerner и сар. [54] су применили NSGA-II за решавање вишекритеријумског проблема покривања локација са применом на постављање јавних објеката у областима које су под ризиком од удара цунамија. Двокритеријумски локацијски проблем разматран је у раду [19] и решаван помоћу NSGA-II. Villegas и сар. [178] су разматрали двокритеријумски локацијски проблема који максимизује покривену потражњу корисника и минимизује цену повезивања корисника са објектима, као и фиксну цену отварања објеката. У раду [178] коришћена је хибридна метода, настала комбиновањем MOEA алгоритма и метода математичког програмирања. Medaglia и сар. [124] су користили еволутивне алгоритме за решавање двокритеријумског локацијског проблема са применама у управљању медицинским отпадом. Још неки примери примене еволутивних алгоритама на вишекритеријумске локацијске проблеме могу се наћи у раду [69].

Како су се метахеуристике са једним решењем и оне засноване на локалној претрази добро показале у применама за решавање локацијских проблема, исте идеје се могу пренети и на више функција циља. У наставку су укратко приказане неке метахеуристике за вишекритеријумску оптимизацију из литературе које користе ове идеје, док је вишекритеријумска варијанта методе променљивих околина описана детаљније, јер је коришћена за решавање проблема у оквиру ове дисертације.

MOSA алгоритам (енгл. Multi-Objective Simulated Annealing) је вишекритеријумска варијанта методе симулираног каљења. Идеја првог вишекритеријумског SA алгоритма из рада [162] јесте да током итерација недоминирана решења имају већу стационарну вероватноћу, тј. више шансе да буду изабрана као тренутно решење. Поред тога, могуће је да два недоминирана решења имају различите вероватноће у зависности од преференци доносиоца одлука. У раду [162] разматрана су различита правила за прихватање решења, како би се претрага фокусирала на недоминирана решења. Касније су настале нове

варијанте MOSA, као што је варијанта представљена у раду [62], а касније проширена у раду [172]. Код алгоритма из рада [62], вредности функција циља се комбинују у једну на основу следеће формуле:

$$G = \sum_{i=1}^M \ln f_i,$$

где су f_1, \dots, f_M посматране функције циља које треба минимизовати. Варијанта из рада [62] користи и архиву недоминирајућих решења, где се решења обрађивана током стандардног процеса симулираног каљења убацују уколико нису доминирана постојећим члановима архиве. За разлику од алгоритма из рада [62], MOSA предложен у [172] не комбинује функције циља у једну нову функцију, већ се промене у вредностима сваке од функција циља пореде директно. MOSA алгоритам предложен у раду [142] се заснива на одржавању скупа недоминираних решења кроз итерације. Избор тренутног решења и критеријум прихватања решења су дефинисани на следећи начин:

- Ако кандидат доминира над неким решењима, она се избацују, кандидат постаје тренутно решење и убацује се у скуп недоминираних решења;
- Ако неко решење доминира над кандидатом, кандидат постаје тренутно решење са одређеном вероватноћом која зависи од температуре;
- Ако није испуњен ниједан од претходна два услова, кандидат постаје тренутно решење и убацује се у скуп решења.

Више о MOSA и његовим варијантама може се наћи у прегледном раду [5].

У постојећој литератури предложене су различите варијанте вишекритеријумске **табу претраге**. На пример, у раду [82] се користи тежински просек свих функција циља како би се нашло најбоље решење у околини сваког тренутног решења. Тежине се затим динамички ажурирају током претраге како би се истражили сви региони Парето фронта, што представља механизам диверсификације претраге. Аутори рада [73] су предложили сличан алгоритам, који такође користи адаптивне тежине. Приликом избора решења из околине узима се оно које максимизује тежинску функцију свих функција циља, а да при томе није у табу листи. Током претраге се одржава скуп недоминираних решења, где се додавањем нових решења бришу она која постају доминирана. Другачији приступ избору решења из околине је предложен у раду [15], где се решење бира

на случајан начин из листе кандидата која садржи сва недоминирана решења у односу на тренутна решења околине и тренутни скуп недоминираних решења. Како би се избегли проблеми са избором тежина и скалирањем функција циља, аутори рада [104] предлажу употребу мултиномне расподеле вероватноће за избор активне функције циља у свакој итерацији алгорита. Затим се приликом избора најбољег решења из околине, бира оно које има најбољу вредност те активне функције, а које притом није означено као табу. Додатно, могуће је изабрати и табу решење уколико оно доминира над неким од тренутних недоминираних решења. Овај алгоритам користи рестарт процедуре као механизам диверсификације, где се на случајан начин бира једно недоминирано решење које постаје тренутно, а табу листа се празни.

Прва вишекритеријумска варијанта **GRASP** алгоритма је заснована на тежинској комбинацији функција циља и примењена је на решавање вишекритеријумске варијанте проблема ранца [177]. Предложени алгоритам у свакој итерацији користи друге тежине како би се добила различита недоминирана решења. Касније се у литератури јављају и друге варијанте вишекритеријумске GRASP методе, које не свде проблем на једну функцију циља и одржавају скуп недоминираних решења током претраге. Приликом конструкције решења може се користити комбинована функција циља или изабрати једна од функција циља. Ако користимо једну од функција циља, она се може за сваку конструкцију изабрати на случајан начин или неким утврђеним редоследом. Код комбиновања функција циља, може се користити тежинска функција или да се функције бирају за сваки корак конструкције. Након конструкције решења извршава се нека варијанта вишекритеријумске локалне претраге која одговара кораку конструкције. На пример, ако је извршена комбинована конструкција, код локалне претраге се за избор најбољег решења из околине могу користити различите функције циља. За свако решење пронађено приликом претраге се проверава да ли се укључује у скуп недоминираних решења. Детаљни преглед и евалуација оваквих варијанти дати су у раду [119].

Метода променљивих околина је дала одличне резултате за проблеме са једном функцијом циља разматране у овој дисертацији, што је била мотивација за примену њене вишекритеријумске варијанте на двокритеријумске варијанте ових проблема. Једна од првих примена методе променљивих околина на проблеме вишекритеријумске оптимизације налази се у раду [75], за решавање вишекритеријумског проблема распоређивања послова. Затим је објављено још

неколико радова који на различите начине прилагођавају методу променљивих околина конкретним проблемима вишекритеријумске оптимизације. Метода је примењена за решавање различитих проблема алокације ресурса [107]. Проблем распоређивања на једној машини са различитим временским прозорима је решаван у раду [9]. Аутори рада [108] су решавали проблем алокације редундантности компоненти система. Динамички проблем распоређивања послова решаван је у раду [1].

Општи оквир за примену методе променљивих околина је предложен у раду [59], где је промењена дефиниција тренутног решења и појма унапређивања решења. Тренутно решење је тренутни скуп недоминираних решења (тачака) који се ажурира током итерација. На крају извршавања алгоритама, тренутно решење се враћа као апроксимативни Парето скуп, тј. скуп свих недоминираних решења пронађених током претраге. Решење E' представља унапређење решења E уколико постоји елемент e решења E' који није доминаран тачкама из решења E . Другим речима, унапређење решења постоји ако постоји нови недоминирани елемент који се додаје у тренутно решење. Поступак провере побољшања решења приказан је Алгоритмом 7.

Алгоритам 7 Провера побољшања решења за вишекритеријумски проблем

```
1: procedure MO-IMPROVEMENT( $E, E'$ )
2:   for all  $e \in E'$  do
3:     if  $e \notin E \wedge \neg \text{Dominated}(e, E)$  then
4:       return True
5:   return False
```

Користећи наведене дефиниције претходних појмова, може се представити и вишекритеријумска промена околина која је приказана Алгоритмом 8. Основна идеја је иста као код корака промене околина за проблеме са једном функцијом циља, уколико је дошло до побољшања онда се враћа на прву околину, иначе се иде на следећу. Процедура UPDATE ажурира тренутно решење E тако што додаје нове недоминиране елементе из E' , а избацује оне који су постали доминирани.

Код вишекритеријумске фазе размрдавања, за сваку тачку тренутног решења се врши размрдавање у околини k . Нови генерисани елементи се чувају у новом решењу E' . Овај поступак је приказан Алгоритмом 9.

Комбинацијом ова два елемента, фазе размрдавања и фазе промене околина, добијамо редуковану варијанту вишекритеријумске методе променљивих око-

Алгоритам 8 Вишекритеријумска промена околине

```
1: procedure MO-NEIGHBORHOODCHANGE( $E, E', k$ )
2:   if MO-IMPROVEMENT( $E, E'$ ) then
3:      $k \leftarrow 1$ 
4:      $E \leftarrow \text{UPDATE}(E, E')$ 
5:   else
6:      $k \leftarrow k + 1$ 
```

Алгоритам 9 Вишекритеријумска фаза размрдавања

```
1: procedure MO-SHAKE( $E, k$ )
2:    $E' \leftarrow \emptyset$ 
3:   for all  $e \in E$  do
4:      $e' \leftarrow \text{SHAKE}(e, k)$ 
5:      $E' \leftarrow E' \cup \{e'\}$ 
6:   return  $E'$ 
```

лина, која је приказана Алгоритмом 10. Улазни параметри су почетно решење E , максималан број околина k_{\max} , максимално дозвољено процесорско време извршавања t_{\max} или параметар за неки други критеријум заустављања (на пример, максималан број итерација без побољшања решења). Резултат извршавања алгоритма је скуп недоминираних решења E' . Протекло процесорско време извршавања алгоритма означено је са CPU TIME.

Алгоритам 10 Вишекритеријумска редукована метода променљивих околина

```
1: procedure MO-RVNS( $E, k_{\max}, t_{\max}$ )
2:    $E' \leftarrow \emptyset$ 
3:   repeat
4:      $k \leftarrow 1$ 
5:     repeat
6:        $E' \leftarrow \text{MO-SHAKE}(E, k)$ 
7:        $E \leftarrow \text{MO-NEIGHBORHOODCHANGE}(E, E', k)$ 
8:     until  $k = k_{\max}$ 
9:      $t \leftarrow \text{CPU TIME}$ 
10:  until  $t > t_{\max}$ 
11:  return  $E'$ 
```

Код вишекритеријумске варијанте променљивог спуста врши се променљиви спуст за тренутна решења по свакој функцији циља појединачно. За разлику од варијанте са једном функцијом циља, приликом локалне претраге сваки елемент који се добије из околине треба проверити за укључивање у тренутни скуп

решења. Тако је повратна вредност променљивог спуста (или локалне претраге) скуп решења, а не једно решење. Алгоритам 11 приказује основну структуру методе променљивог спуста за вишекритеријумску оптимизацију. Улазни параметри су почетно решење E , максималан број околина k'_{\max} и укупан број функција циља M . Функције циља се смењују током претраге све док има побољшања. Променљиви спуст у односу на функцију циља f_i означен је са VNDi. Идеја је да се свака тачка тренутног решења претражи у свакој околини у односу на сваку функцију циља. За сваку функцију циља се одржава и скуп решења S_i . Процедура SELECT бира на случајан начин елемент који већ није обрађен у односу на дату функцију циља. Процедура MO-OBJECTIVECHANGE ради по истом принципу као и MO-NEIGHBORHOODCHANGE процедура, где се у случају побољшања враћа на прву функцију циља и ажурира тренутно решење пронађеним тачкама.

Алгоритам 11 Вишекритеријумски променљиви спуст

```

1: procedure MO-VND( $E, k'_{\max}, M$ )
2:    $S_1 \leftarrow \emptyset, S_2 \leftarrow \emptyset, \dots, S_r \leftarrow \emptyset$ 
3:    $i \leftarrow 1$ 
4:   repeat
5:     repeat
6:        $e' \leftarrow \text{SELECT}(E \setminus S_i)$ 
7:        $E_i \leftarrow \text{VNDi}(e', k'_{\max})$ 
8:        $S_i \leftarrow S_i \cup E_i$ 
9:     until  $E \setminus S_i = \emptyset$ 
10:    MO-OBJECTIVECHANGE( $E, S_i, i$ )
11:  until  $i > r$ 
12:  return  $E$ 

```

Комбиновањем процедуре променљивог спуста и фазе размрдавања добија се вишекритеријумска општа метода променљивих околина, која је приказана Алгоритмом 12. У оквиру унутрашње петље, процедуре MO-SNAKE, MO-VND и MO-NEIGHBORHOODCHANGE се секвенцијално извршавају све док се не истраже све околине, тј. док се не достигне вредност параметра k_{\max} . Спољашња петља се извршава све док не истекне дозвољено време за извршавање, тј. док се не прекорачи вредност параметра t_{\max} . У случају да се уместо променљивог спуста VNDi за сваку функцију циља f_i користи обична локална претрага у односу на ту функцију циља f_i , добија се вишекритеријумска основна метода променљивих околина.

Алгоритам 12 Вишекритеријумска општа метода променљивих околина

```
1: procedure MO-GVNS( $E, k_{\max}, M, k'_{\max} t_{\max}$ )
2:    $E' \leftarrow \emptyset$ 
3:   repeat
4:      $k \leftarrow 1$ 
5:     repeat
6:        $E' \leftarrow \text{MO-SHAKE}(E, k)$ 
7:        $E'' \leftarrow \text{MO-VND}(E', k'_{\max}, M)$ 
8:        $E \leftarrow \text{MO-NEIGHBORHOODCHANGE}(E, E'', k)$ 
9:     until  $k = k_{\max}$ 
10:     $t \leftarrow \text{CPU TIME}$ 
11:  until  $t > t_{\max}$ 
12:  return  $E'$ 
```

Детаљније о примени метахеуристика за вишекритеријумску оптимизацију које су засноване на побољшању једног решења, а посебно оних заснованих на локалној претрази, може се наћи у прегледном раду [21].

Глава 2

Проблем максималног покривања локација са ограниченим буџетом и преференцијама корисника

2.1 Опис проблема и преглед релевантне литературе

Проблем максималног покривања локација

Проблем максималног покривања локација (енгл. Maximal Covering Location Problem - MCLP) је један од најпроучаванијих локацијских модела у литератури, због широког спектра његових примена (планирање услужних и објеката за хитне службе, класификација, апстракција података, управљање ланцем снабдевања). У основној варијанти MCLP, коју су увели Church и ReVelle у раду [37], познати су скуп корисника са задатом потражњом и скуп потенцијалних локација за успостављање објеката. Претпоставља се да корисника могу услужити само отворени објекти чије растојање од датог корисника није веће од критичне удаљености (радијус покривања). Циљ MCLP-а је одредити локације за успостављање задатог броја објеката и придружити сваког корисника тачно једном од успостављених објеката, тако да се максимизује количина покривене потражње. Потражња корисника се сматра покривеном уколико се корисник снабдева од објекта који је од њега удаљен не више од задатог радијуса покривања. У раду [125] доказано је да је MCLP НП-тежак.

2.1. ОПИС ПРОБЛЕМА И ПРЕГЛЕД РЕЛЕВАНТНЕ ЛИТЕРАТУРЕ

У циљу представљања математичке формулације проблема користе се следеће ознаке за улазне параметре:

- J - Скуп корисника;
- I - Скуп потенцијалних локација објеката;
- S_j - Скуп локација које могу покрити корисника $j \in J$;
- a_j - Потражња корисника $j \in J$;
- p - Број објеката које треба успоставити.

Математичка формулација користи два скупа бинарних променљивих:

- x_i - бинарна променљива која узима вредност 1 ако је објекат отворен на локацији $i \in I$, иначе има вредност 0.
- y_j - бинарна променљива која узима вредност 1 ако је потрошач $j \in J$ покривен неким објектом, иначе има вредност 0.

Користећи наведену нотацију, проблем се може записати у виду целобројног линеарног програма (енгл. Integer Linear Program - ILP) [37]:

$$\max \sum_{j \in J} a_j y_j \quad (2.1)$$

$$\text{при условима} \quad \sum_{i \in I} x_i = p, \quad (2.2)$$

$$y_j \leq \sum_{i \in S_j} x_i, \quad \forall j \in J \quad (2.3)$$

$$x_i \in \{0, 1\}, \quad \forall i \in I \quad (2.4)$$

$$y_j \in \{0, 1\}, \quad \forall j \in J \quad (2.5)$$

Функција циља (2.1) једнака је суми покривене потражње корисника, а циљ проблема је њена максимизација. Услов (2.2) обезбеђује да се отвара тачно p објеката. Услов (2.3) обезбеђује да се потрошач i може означити покривеним само ако је отворен бар један објекат j који се налази у његовом радијусу покривања. Променљиве x_i и y_j су ограничене на бинарне вредности условима (2.4) и (2.5).

Након рада [37] у којем је уведен основни MCLP, појавиле су се бројне студије које уводе различите варијанте овог проблема и предлажу методе за њихово решавање. Нове варијанте проблема MCLP су често добијене укључивањем ограничења и претпоставки из различитих реалних ситуација. У раду [181] разматрана је варијанта MCLP-а са једнаким захтевима корисника са циљем да се максимизује број покривених корисника. Планарна варијанта MCLP-а изучавана је у радовима [38] и [180], док су ReVelle и Hogan у раду [151] предложили MCLP са вероватноћама за отварање објекта за сваку локацију из датог скупа. ReVelle је у раду [152] разматрао условну варијанту MCLP-а (енгл. conditional MCLP) са циљем да се максимизује број објеката покривених другим објектима (секундарна подршка), док је примарна покривеност објектима обезбеђена за све кориснике. Вишекритеријумска условна варијанта, такође предложена у [152], даје компромис између укупне покривене потражње са примарним покрићем и броја објеката са секундарним покрићем. Verma и Krass су у раду [16] формулисали генерализацију MCLP-а која дозвољава делимичну покривеност корисника, са степеном покривености који се даје нерастућом степенастом функцијом удаљености до најближег отвореног објекта. Још једна генерализација MCLP-а, предложена у раду [44], укључује различите типове објекта који служе корисницима уз ограничење да се само одређени број различитих типова објеката може активирати на истој локацији. У раду [169] предложена је варијанта која је настала на основу реалног проблема дизајнирања мрежа хитне службе. Варијанта из [169] укључује више типова услуга и више типова услужних центара, као и претпоставке о хијерархијској уређености типова центара и ограничењу броја отворених центара за сваки тип. У раду [148] предложен је вишепериодни MCLP (енгл. multi-period MCLP) за проналажење оптималне локације сигурносних камера на раскрсници на мрежи градског саобраћаја како би се минимизовао број саобраћајних несрећа. Zarandi и сар. у раду [185] уводе динамички MCLP (енгл. dynamic MCLP) узимајући у обзир више од једног временског периода у моделу. Робусна варијанта динамичког MCLP-а је формулисана у раду [129], где је укључена непоузданост у потражњи корисника. Фази (енгл. fuzzy) MCLP предложен у раду [56] претпоставља две врсте фази бројева за описивање два главна параметра MCLP-а: радијус покривања и растојања између локација. MCLP је и даље веома актуелан, с обзиром на његов практични значај и различите области примене. Метода променљивих околина примењена је за решавање оригиналног MCLP у

раду [138]. Детаљан опис MCLP-а, његових варијанти и примена може се наћи у [67] и [165].

Проблем максималног покривања локација са преференцијама корисника

Када се корисници распоређују на отворене објекте, стандардни приступ је да се сваки корисник додели најближем отвореном објекту. Реалнија претпоставка је да корисници бирају успостављени објекат на основу својих преференција. Преференције корисника су засноване на различитим факторима, као што су: старост корисника, квалитет услуге, приходи корисника, удаљеност, начин превоза, итд. Први рад у литератури о локацијским проблемима који укључује и преференције корисника у претпоставке проблема је [81]. Аутори су проширили добро познати локацијски проблем без капацитета (енгл. Uncapacitated Facility Location Problem - UFLP) узимајући у обзир сортирање објеката по преференцијама корисника и предложили алгоритам за решавање новог проблема заснован на похлепној хеуристици у комбинацији са методом гранања и ограничавања. Неколико реформулација UFLP-а са сортирањем објеката по преференцијама корисника, анализом сложености, полиедарским моделима и доњим границама представљено је у радовима [26, 79, 83, 92, 176]. Hansen и сар. су у раду [83] реформулисали проблем као двостепени линеарни програм (енгл. Bilevel Linear Program - BLUFLP) и предложена је метода решавања заснован на хибридизацији шеме Стакелбергове равнотеже и еволутивног алгоритма. Две нове формулације двостепеног модела BLUFLP-а у мешовити целобројни једностепени програм су представили Camacho-Vallejo и сар. у раду [25], где је предложен и еволутивни алгоритам заснован на равнотежи у Стакелберговој игри за решавање двостепеног модела. У раду [117] изложене су три метахеуристичке методе за решавање BLUFLP-а: алгоритам оптимизације ројем честица, метода симулираног каљења и варијанта методе променљивих околина. Варијанта BLUFLP-а са фиксним бројем објеката које треба отворити, означена као проблем p -медијане са преференцијама клијената разматрали су Alekseva и Kochetov у раду [4], а у истом раду имплементиран је и генетски алгоритам за решавање разматраног проблема. Исти проблем је проучаван у раду [27], где су предложене две једностепене реформулације, као и хибридни хеуристички алгоритам заснован на методи расуте претраге (енгл.

scatter search) и GRASP алгоритму.

У раду [93] предложена је фази варијанта UFLP-а са преференцијама корисника. Разматрана су два критеријума оптимизације: проналажење локације објекта која максимизује минимални степен задовољства међу свим тачкама потражње и максимизација преференција корисника локације. Локацијски проблем са ограничењима покривања и преференцијама корисника је проучаван у раду [106]. Аутори су предложили формулацију мешовитог целобројног програмирања засновану на ограничењима корисника и хеуристици Лагранжеве релаксације.

Недавне студије о MCLP-у такође укључују претпоставку да корисници бирају успостављени објекат на основу својих преференција. На пример, Casas-Ramirez и сар. су у раду [28] увели двостепени MCLP (енгл. bi-level MCLP), где се разматра случај фирме која жели да пронађе локације за отварање p објеката са циљем максимизације покривене потражње. Претпоставља се да скуп објеката који припадају конкуренцији већ постоји на тржишту и корисницима је дозвољено да се распореде на отворене објекте на основу својих преференција. Аутори рада [28] су реформулисали двостепени MCLP модел у једностепени и предложили генетски алгоритам побољшан локалном претрагом као приступ решавању разматраног проблема. Diaz и сар. су у раду [52] разматрали исту ситуацију на тржишту и предложена је нова једностепена формулација конкурентног двостепеног MCLP-а. Аутори су применили GRASP алгоритам и хибридную методу GRASP алгоритма и табу претраге у циљу добијања решења за инстанце великих димензија. Мркела и Станимировић су у раду [133] разматрали варијанту са преференцијама корисника из рада [52] и предложили два ефикасна метахеуристичка приступа за исти проблем: Метода променљивих околина и хибридизација редуковане методе променљивих околина, табу претраге и основне методе променљивих околина, означене као RVNS-TS-VNS. Експериментални резултати представљени у раду [133] показују потенцијал и VNS и RVNS-TS-VNS приступа решавању разматране варијанте MCLP-а.

У оквиру ове дисертације, представљена је нова варијанта MCLP-а, различита од оне варијанте предложене у раду [52] и касније разматране у раду [133]. Полази се од истих претпоставки проблема као у [52], али уместо фиксног броја објеката које треба успоставити, уводи се претпоставка да компанија која излази на тржиште има ограничен буџет за успостављање објеката. Свакој потенцијалној локацији додељују се фиксни трошкови за отварање објекта који одражавају

цене некретнина и друге трошкове везане за ову локацију. Ово је реалнија претпоставка, имајући у виду да се трошкови некретнина могу значајно разликовати у урбаним и руралним срединама, у центру града и његовим предграђима, али чак и на различитим локацијама истог тржног или пословног центра, итд. Идеју о ограничавању буџета уместо фиксирања броја објеката у класичном MCLP-у увели су Khuller и сар. у раду [102], а касније је коришћена и у студијама о различитим локацијским проблемима, као што су: [3], [33], [40], [157], итд. Описана нова варијанта MCLP, означена је као MCLP са ограниченим буџетом (енгл. budget-constrained MCLP) и преференцијама корисника. У наредном одељку представљена је целобројна линеарна математичка формулација полазећи од формулације варијанте MCLP из рада [52]. За сваку од потенцијалних локација објеката укључују се фиксни трошкови за успостављање објекта и додаје се ограничење буџета за отварање свих објеката. Након тога формулисан је одговарајући линеарни математички модел. Математичка формулација проблема MCLP са ограниченим буџетом и преференцијама корисника је први пут публикована у раду [137].

2.2 Математичка формулација проблема

Нека је I_1 скуп потенцијалних локација за отварање објеката компаније која излази на тржиште и I_2 скуп локација отворених објеката које припадају конкурентима, док је $I = I_1 \cup I_2$. Скуп корисника је $J = J_1 \cup J_2$, где J_1 представља скуп корисника који нису покривени постојећим отвореним објектима који припадају конкурентима, а J_2 означава скуп корисника покривених отвореним објектима конкурента. Корисник $j \in J$ се сматра покривеним (енгл. covered) ако постоји отворен објекат чије растојање до корисника j није веће од датог полупречника покривања R , а непокривеним у супротном. Потражња D_j је додељена сваком кориснику $j \in J$, док $I(j) \subseteq I$ представља подскуп локација објеката који могу покривати корисника j . За сваку потенцијалну локацију објекта $i \in I$, цена успостављања објекта c_i је дата и скуп корисника који могу бити покривени од стране i је означен као $J(i)$. Преференција корисника $j \in J$ према објекту у $i \in I$ је задата као вредност $g_{ij} > 0$. Ако важи $g_{i_1j} > g_{i_2j}$, то значи да корисник j преферира да буде додељен објекту на локацији i_1 у односу на објекат на локацији i_2 . Компанија која планира да изађе на тржиште има ограничен буџет B за отварање објеката на некој од локација из скупа I_1 .

2.2. МАТЕМАТИЧКА ФОРМУЛАЦИЈА ПРОБЛЕМА

У циљу представљања математичке формулације проблема, уведе се два скупа бинарних променљивих. Променљива y_i узима вредност 1 ако се објекат налази на локацији $i \in I$, а 0 у супротном. Променљива x_{ij} се поставља на 1 ако је корисник $j \in J$ додељен објекту који је успостављен на локацији $i \in I$, а на 0 у супротном.

Користећи горњу нотацију, MCLP ограниченог буџета са преференцијама корисника може се формулисати као целобројни линеарни програм на следећи начин:

$$\max \sum_{i \in I_1} \sum_{j \in J(i)} D_j x_{ij} \quad (2.6)$$

при условима

$$\sum_{i \in I_1} c_i y_i \leq B, \quad (2.7)$$

$$\sum_{i \in I(j)} x_{ij} = 1 \quad \forall j \in J_2, \quad (2.8)$$

$$\sum_{i \in I(j)} x_{ij} \leq 1 \quad \forall j \in J_1, \quad (2.9)$$

$$x_{ij} \leq y_i \quad \forall i \in I, j \in J(i), \quad (2.10)$$

$$y_i = 1 \quad \forall i \in I_2, \quad (2.11)$$

$$\sum_{s=k+1}^{|I(j)|} x_{is,j} + y_{i_k} \leq 1 \quad \forall j \in J, k \in 1, \dots, |I(j)| - 1, \quad (2.12)$$

$$y_i \in \{0, 1\} \quad \forall i \in I, \quad (2.13)$$

$$x_{ij} \in \{0, 1\} \quad \forall i \in I, j \in J. \quad (2.14)$$

Функција циља (2.6) је једнака укупној потражњи корисника покривених новим успостављеним објектима из I_1 , а циљ проблема је њена максимизација. Трошкови отварања објеката не смеју да пређу дати буџет B , што је обезбеђено ограничењем (2.7). Кориснике покривене постојећим објектом из I_2 услужује тачно један објекат (2.8). Корисници који нису покривени постојећим објектима могу бити додељени највише једном новом објекту из I_1 (2.9). Ограничења (2.10) обезбеђују да су корисници додељени само отвореним објектима, док ограничења (2.11) обезбеђују да постојећи објекти који припадају конкурентима остају отворени. Сваки корисник је придружен својој најпожељнијој локацији (у смислу преференција) која може покрити датог корисника (2.12).

Ограничења (2.13)–(2.14) дефинишу тип променљивих одлучивања y_i и x_{ij} . Разматрана варијанта MCLP-а је НП-тежак проблем оптимизације, као генерализација класичног MCLP-а, за који је доказано да је НП-тежак у раду [125].

2.3 Метода променљивих околина за решавање проблема

Методе засноване на VNS-у успешно су примењене на различите проблеме покривања локација [43, 94, 115, 161], као и на различите варијанте MCLP, као што су: варијанта MCLP са фази полупречником покривања [47], варијанта MCLP са различитим типовима објеката [44] и робусна варијанта динамичког MCLP [129]. У овом одељку представљени су детаљи VNS имплементације за MCLP ограниченог буџета са преференцијама корисника. VNS методе предложене у [133] показале су се успешним при решавању сличне варијанте MCLP-а која такође укључује преференције корисника, али претпоставља унапред задат број објеката које треба отворити. Због сличности ове две MCLP варијанте, VNS предложен у овој дисертацији дели неке заједничке елементе са VNS хеуристиком из рада [133]. Предложена имплементација за решавање разматране MCLP варијанте детаљно је описана у раду [137], у којем су приказани и резултати на модификованим MCLP инстанцама из литературе.

Представљање решења и рачунање функције циља

Решење проблема је представљено као бинарни низ sol дужине $|I_1|$, где I_1 означава скуп потенцијалних локација објеката. Сваки бит решења sol одговара једној локацији кандидата за отварање објекта. Ако је $sol(i) = 1$, $i \in I_1$, то значи да је објекат успостављен на локацији i , док $sol(i) = 0$ означава да није. Решење се сматра допустивим ако збир трошкова c_i отворених објеката не прелази дати буџет B .

Вредност функције циља допустивог решења sol израчунава се на следећи начин. Прво, индекси успостављених објеката се добијају из решења sol . Затим се корисници додељују отвореним објектима на основу преференција корисника и полупречника покривања. Корисник j се додељује свом најпожељнијем успостављеном објекту на локацији i само ако $i \in I(j)$. Коначно, вредност функције

циља се добија сумирањем потражње корисника који су додељени на нове успостављене објекте.

Генерисање почетног решења

Почетно решење за VNS је конструисано похлепном процедуром приказаном Алгоритмом 13. Процедура почиње „празним” решењем sol које нема успостављених објеката (sol је низ нула дужине $|I_1|$). Процедура затим итеративно отвара један по један објекат док се не добије допустиво решење. Скуп потенцијалних локација за успостављање објеката, означен као $candidates$, у почетку је једнак I_1 . За сваку локацију i из $candidates$, процедура израчунава додатну потражњу корисника која би могла бити покривена након отварања објекта на i . Израчуната вредност је означена као потенцијални $gain$ (добит) локације кандидата i . Објекат се успоставља на локацији i_{max} која има максималан однос добити и трошкова $gain/c_i$, где c_i представља трошкове отварања објекта на i . Индекс локације i_{max} са новоотвореним објектом се брише из скупа $candidates$. Описани кораци се понављају све док укупни трошкови успостављања објеката не прелазе расположиви буџет B .

Алгоритам 13 Похлепна конструкција решења

```
1: Input:  $\emptyset$ 
2: Output:  $sol$ 
3:  $sol \leftarrow 0$ ;
4:  $candidates \leftarrow I_1$ ;
5: while solution  $sol$  is feasible do
6:   for  $i$  in  $candidates$  do
7:      $gain_i \leftarrow$  the demand that would be covered
8:     if a facility is opened at location  $i$ ;
9:    $i_{max} \leftarrow$  location with maximal gain to cost ratio;
10:   $sol(i_{max}) \leftarrow 1$ ;
11:  remove location  $i_{max}$  from  $candidates$ ;
12: return  $sol$ 
```

ОкоLINE решења

Четири окоLINE решења се истражују у оквиру предложеног VNS-а. ОкоLINE су дефинисане операцијама које мењају локације објеката у тренутном решењу:

- Операција *замени* (енгл. Swap) се врши затварањем отвореног објекта i у решењу sol и успостављањем објекта на другој локацији j , $i, j \in I_1$. Околина $N_k^1(sol)$ садржи сва решења добијена из sol извођењем операције *замени* k пута, $k \in N$.
- Операција *додај* (енгл. Add) отвара објекат на новој локацији $i \in I_1$ у решењу sol . Сва решења добијена из sol понављањем операције *додај* k пута припадају околини $N_k^2(sol)$, $k \in N$.
- Операција *обриши* (енгл. Delete) затвара претходно успостављени објекат на локацији $i \in I_1$ у решењу sol . Околина $N_k^3(sol)$, $k \in N$ састоји се од свих решења добијених затварањем k објеката у sol .
- *Обриши и додај* (енгл. Delete&Add) операција се врши затварањем одређеног броја објеката у тренутном решењу sol и отварањем објеката на новим локацијама док се не достигне дато ограничење буџета. Локације за отварање објеката бирају се на исти начин као и у процедури похлепне конструкције решења (Алгоритам 13). Ако је број објеката које треба затворити k , добијена решења припадају околини $N_k^4(sol)$.

Као што се види из дефиниције околина решења, операције отварања и затварања објеката се изводе велики број пута током извршавања VNS-а. Да би се самњило укупно време извршавања, користи се стратегија из [133] за ажурирање вредности функције циља након отварања и затварања објеката. Као и у [133], за сваку локацију $i \in I$, разматрају се само корисници j из листе $J(i)$. За сваког корисника $j \in J$, листа локација $i \in I(j)$ је сортирана у опадајућем редоследу на основу преференција корисника j и чува се индекс отвореног објекта који тренутно услужује корисника j .

Приликом затварања објекта на локацији $i \in I_1$, укупна покривена потражња се смањује ако а) нема отвореног објекта из I_1 који може покрити кориснике које су претходно били покривени објектом на локацији i , или б) корисници претходно додељени локацији i преферирају објекат који су отворили конкуренти у односу на било који други успостављени објекат из I_1 који може покрити посматране кориснике. Стога, након затварања објекта на локацији i , довољно је размотрити само кориснике који су додељени овом објекту. За сваког корисника j из овог скупа, процедура претражује сортирану листу локација кандидата из $I(j)$ и корисник j се додељује првом отвореном објекту k

из сортиране листе. Ако нема отвореног објекта k у листи или је пронађени објекат k у власништву конкурената, функција циља се смањује за потражњу корисника j .

Приликом отварања објекта на локацији $i \in I_1$, вредност функције циља се повећава ако *a*) новоотворени објекат може покрити кориснике који раније нису били покривени, или *b*) постоје корисници додељени конкурентним објектима који преферирају новоотворени објекат i у односу на објекте којима су тренутно додељени. Зато се само листа корисника $j \in J(i)$ обрађује после успостављања објекта на локацију i . Ако је корисник $j \in J(i)$ био покривен неким другим објектом из I_1 који је био мање пожељан од i , корисник j се додељује новом објекту на локацији i , али не долази до промене функције циља. Ако корисник j није био покривен ниједним отвореним објектом, или је био покривен мање пожељним објектом конкуренције, корисник j се додељује новом објекту на локацији i , а вредност функције циља се повећава за вредност потражње корисника j .

Структура предложене методе

Псеудокод предложеног VNS-а за MCLP ограниченог буџета са преференцијама корисника представљен је Алгоритмом 15. Први корак је процедура похлепне конструкције решења која враћа почетно решење *sol*. Конструисано почетно решење постаје тренутно, које даље улази у процедуру *Размрдавања*.

Процедура *Размрдавања* користи може да користи све четири дефинисане околине следећим редоследом. Прво се истражују околине N_k^1 , $k = 0, \dots, k_1$ дефинисане операцијом *замена*, затим околине N_k^4 , $k = 0, \dots, k_4$ добијене извођењем операције *обриши и додај*. Размрдавање се даље наставља у околинама N_k^2 , $k = 0, \dots, k_2$ понављањем операције *додај* и на крају се истражују околине N_k^3 , $k = 0, \dots, k_3$ дефинисане операцијом *обриши*. Овај редослед истраживања околине добијен је кроз извођење прелиминарних експеримената (видети одељак 2.4). Ради једноставности, околине решења *sol* које се користе у фази *размрдавања* су означене као $N_k(sol)$, $k = 1, \dots, k_{max}$ где је $k_{max} = k_1 + k_2 + k_3 + k_4$. Параметри $k_1, k_2, k_3, k_4 \in \{0, 1, \dots, k_{max}\}$ се одређују експерименталним путем. Уколико неки од ових параметара k_i једнак 0, то значи да се околине N_k^i и не истражују приликом процедуре размрдавања. На пример, уколико је $k_{max} = 3$ и притом је $k_1 = 1, k_2 = 0, k_3 = 2$ и $k_4 = 0$, то значи да се најпре истражује околина N_1^1 , а затим околине N_1^3 и N_2^3 , док се околине N_k^2 и N_k^4 не истражују.

Треба приметити да операције *обриши* и *обриши и додај* увек дају допустиво решење, док операције *замени* и *додај* могу нарушити допустивост решења у вези са датим ограничењем буџета. Стога, може се десити да решење sol' добијено из фазе *размрдавања* буде недопустиво. У том случају, решење се прослеђује процедури за похлепно поправљање решења које је описана Алгоритмом 14.

Процедура почиње од недопустивог решења sol и креира листу свих парова локација (i, j) , где $i, j \in I_1$, тако да је $sol(i) = 1$, $sol(j) = 0$ и $c_j < c_i$. Креирана листа је означена као $pairs$ (парови) и чине је кандидати за операцију замене (енгл. swap) - затварање објекта на локацији i и отварање објекта на j , али тако да су трошкови отварања објекта на локацији j мањи у односу на трошкове отварања на локацији i . За сваки пар $(i, j) \in pairs$, процедура израчунава однос потражње и трошкова $demand/cost$, где је $demand$ укупна покривена потражња, а $cost$ укупни трошкови у решењу који би се добили након извршења ове замене. Пар (i', j') са максималним односом потражње и трошкова се бира из $pairs$ и врши се одговарајућа замена. У случају да је листа парова кандидата празна (нема операције замене која доводи до смањења укупне цене), процедура испитује све отворене објекте у sol и затвара објекат i за који се добија решење са највећим односом потражње и трошкова након затварања i . Описани кораци се понављају док се не добије допустиво решење.

Алгоритам 14 Похлепно поправљање решења

Input: sol

Output: feasible solution sol

- 1: **while** solution sol is infeasible **do**
 - 2: $pairs \leftarrow$ the list of all location pairs (i, j) : $sol(i) = 1$, $sol(j) = 0$, $c_j < c_i$;
 - 3: **if** $pairs$ is not empty **then**
 - 4: **for** (i, j) in $pairs$ **do**
 - 5: $demand_{(i,j)} \leftarrow$ total demand of sol if swap is performed on i and j ;
 - 6: $cost_{(i,j)} \leftarrow$ total cost of sol if swap is performed on i and j ;
 - 7: $(i', j') \leftarrow$ pair with maximal demand to cost ratio;
 - 8: swap pair (i', j') ;
 - 9: **else**
 - 10: $i_{max} \leftarrow$ location i : $sol(i) = 1$, closing i results in maximal demand to cost ratio;
 - 11: $sol(i_{max}) \leftarrow 0$;
 - 12: **return** sol
-

Извршени прелиминарни експерименти су показали да се проценат појављивања недопустивог решења током фазе *размрдавања* креће између 8 % и 45 % (у просеку) за разматране инстанце проблема. На основу тога може се закључити да је стратегија поправљања недопустивог решења након фазе *размрдавања* неопходна како би се обезбедила боља диверсификација претраге. Ако се дозволе само допустиве операције у оквиру разматраних околина, повећава се шанса да се алгоритам заглави у локалном оптимуму.

Фаза *Локалне претраге* се примењује након фазе *Размрдавања*, покушавајући да побољша sol' истраживањем његове околине $N_1^1(sol')$. Када се пронађе прво допустиво решење sol'' боље од sol' , претрага унутар $N_1^1(sol')$ се зауставља и sol' се замењује са sol'' . *Локална претрага* се такође зауставља ако се не пронађе побољшање у околини $N_1^1(sol')$.

Алгоритам 15 Предложени VNS

```

1: Input:  $I_{max}, k_{max}$ 
2: Output:  $sol$ 
3:  $sol \leftarrow$  Greedy Solution Construction
4:  $I_{count} \leftarrow 0$ 
5: while  $I_{count} \leq I_{max}$  do
6:    $k \leftarrow 1$ 
7:   while  $k \leq k_{max}$  do
8:      $sol' \leftarrow$  randomly generated solution from  $N_k(sol)$            ▷ Shaking step
9:     if  $sol'$  is infeasible then
10:       $sol' \leftarrow$  Greedy Solution Repair ( $sol'$ )
11:     while true do                                           ▷ Local Search step
12:        $sol'' \leftarrow$  the first feasible solution from  $N_1^1(sol')$  better than  $sol'$ 
13:       if  $sol''$  exists then
14:          $sol' \leftarrow sol''$ 
15:       else
16:         break
17:       if  $obj(sol') > obj(sol)$  then                             ▷ Move or Not step
18:          $sol \leftarrow sol'$ 
19:          $k \leftarrow 1$ 
20:          $I_{count} \leftarrow 0$ 
21:       else
22:          $k \leftarrow k + 1$ 
23:      $I_{count} \leftarrow I_{count} + 1$ 
24: return  $sol$ 

```

Ефикасност ажурирања функције циља је додатно побољшана ефикасном имплементацијом операције *замени*. Пре извршавања операције *замени*, прво

се уклања претходно успостављени објекат i , а затим се испитује да ли постоји локација j без успостављеног објекта тако да ће отварање објекта на j довести до побољшања вредности функције циља. Само у случају таква локација постоји, врши се операција *замене* за локације i и j . На овај начин се избегавају *замене* операције које не доносе побољшања.

Након што је *Локална претрага* завршена, одлучује се да ли се прелази у ново решење (корак Move or not). Алгоритам ће прећи на решење sol' само ако је боље од тренутно најбољег решења sol . У овом случају, k се поставља на 1, а процедура *Размрдавања* почиње од $N_1(sol)$. У супротном, k се повећава за 1 и *Размрдавање* се наставља у $N_k(sol)$. Главна VNS петља се понавља све док се не испуни услов заустављања, а то је I_{max} узастопних итерација без побољшања тренутног најбољег решења.

Адаптивна метода променљивих околина

У циљу испитивања ефекта прихватања локалног оптимума који је лошији од текућег решења, имплементиран је адаптивни VNS (енгл. Skewed VNS - SVNS) за разматрани MCLP ограниченог буџета са преференцијама корисника. SVNS је варијанта VNS метахеуристике која омогућава истраживање и решења са лошијом функцијом циља у односу на функцију циља тренутног решења уколико се репрезентација новог решења довољно разликује од тренутног [85, 87]. Структура имплементираниог SVNS-а је слична структури VNS-а представљеној Алгоритмом 15, а разлика је у правилу за прихватање решења. Прецизније, SVNS дозвољава прихватање локалног оптимума sol'' добијеног у кораку локалне претраге, чак и ако је добијени локални оптимум лошији од постојећег решења sol .

Алгоритам 16 Адаптивни VNS

```

1: Input:  $I_{max}, k_{max}, \alpha$ 
2: Output:  $best$ 
3:  $sol \leftarrow$  Greedy Solution Construction
4:  $best \leftarrow sol$ 
5:  $I_{count} \leftarrow 0$ 
6: while  $I_{count} \leq I_{max}$  do
7:    $k \leftarrow 1$ 
8:   while  $k \leq k_{max}$  do
9:      $sol' \leftarrow$  Shaking in  $N_k(sol)$ 
10:     $sol'' \leftarrow$  Local Search in  $N_1^1(sol')$ 
11:    if  $obj(sol'') \times (1 + \alpha \times d(sol, sol'')) > obj(sol)$  then            $\triangleright$  Move or Not step
12:       $sol \leftarrow sol''$ 
13:       $k \leftarrow 1$ 
14:      if  $obj(sol'') > obj(best)$  then
15:         $best \leftarrow sol''$ 
16:         $I_{count} \leftarrow 0$ 
17:      else
18:         $k \leftarrow k + 1$ 
19:       $I_{count} \leftarrow I_{count} + 1$ 
20: return  $best$ 

```

Локални оптимум sol'' се прихвата ако важи:

$$obj(sol'') \cdot (1 + \alpha \cdot d(sol, sol'')) > obj(sol).$$

Овде $d(sol, sol'')$ означава Хамингово растојање између кодова решења, подељено са бројем потенцијалних локација m [116]. Параметар α који регулише критеријум прихватања је подешен експериментално (видети одељак 2.4). Кораци предложеног SVNS-а за разматрани MCLP ограниченог буџета са преференцијама корисника су представљени Алгоритмом 16.

Метода са стратешком осцилацијом

За разматрани проблем, имплементирана је још једна варијанта VNS-а која омогућава и истраживање околине недопустивог решења. Идеја стратешке осцилације (енгл. Strategic Oscillation, SO) састоји се у томе да се кретањем напред-назад између скупа допустивих и скупа недопустивих решења дође до глобалног оптимума [77]. Процедура стратешке осцилације је коришћена у оквиру VNS имплементације у раду [60].

Структура VNS-а са стратешком осцилацијом (VNS-SO) за MCLP са ограниченим буџетом и преференцијама корисника је иста као структура VNS-а

2.4. ЕКСПЕРИМЕНТАЛНИ РЕЗУЛТАТИ

представљена у Алгоритму 15, са изузетком процедуре локалне претраге. Улаз у процедуру локалне претраге у VNS-SO је решење sol из корака размрдавања, као и параметар β који контролише степен недопустивости. Прецизније, β представља процентуално повећање од почетне вредности буџета B . Процедура локалне претраге истражује околинау $N_1^1(sol)$ и добија се потенцијално побољшано решење sol . Буџет B се повећава за β %, а процедура покушава да пронађе побољшање у $N_1^1(sol)$ са повећаним буџетом и као резултат добија се решење sol' . Затим, буџет се враћа на оригиналну вредност и процедура похлепног поправљања решења се примењује на добијени локални оптимум sol' како би се обезбедила његова допустивост. Коначно, околина N_1^1 поправљеног допустивог решења се истражује, а резултујући локални оптимум sol' замењује sol ако је $obj(sol') > obj(sol)$. Описани кораци се понављају све док постоји побољшање решења sol (погледати Алгоритам 17).

Алгоритам 17 Локална претрага са стратешком осцилацијом

Input: sol, β

Output: improved solution sol

- 1: $sol \leftarrow$ Local Search in $N_1^1(sol)$
 - 2: **repeat**
 - 3: increase budget by $\beta\%$
 - 4: $sol' \leftarrow$ Local Search in $N_1^1(sol)$
 - 5: restore original budget value
 - 6: $sol' \leftarrow$ Greedy Solution Repair (sol')
 - 7: $sol' \leftarrow$ Local Search in $N_1^1(sol')$
 - 8: **if** $obj(sol') > obj(sol)$ **then**
 - 9: $sol \leftarrow sol'$
 - 10: **until** $obj(sol') \leq obj(sol)$
 - 11: **return** sol
-

2.4 Експериментални резултати

Сви тестови су извршени на рачунару са Intel Xeon E5-2640@2.5 GHz и 64 GB RAM меморије. У циљу добијања оптималних решења, математичка формулација проблема је коришћена у оквиру егзактног решавача Gurobi верзије 7. Предложена VNS имплементација, као и адаптивни VNS и VNS са стратешком осцилацијом, имплементирани су у програмском језику C++. Извршени су и прелиминарни експерименти како би се пронашао адекватан редослед истраживања околина и адекватна величина сваке околине коју треба истражити.

Адаптивни VNS и VNS са стратешком осцилацијом користе исти редослед и величине околина као и предложени VNS. Вредности параметра α у SVNS-у и параметар повећања буџета β VNS-а са стратешком осцилацијом су такође изабрани наведеним експериментима. Да би се обезбедило коректно поређење перформанси, свака од три имплементације VNS методе је покренута 5 пута на свакој разматраној инстанци.

Инстанце проблема

За потребе експеримената, генерисане су три групе инстанци проблема. Прва група инстанци је добијена модификацијом инстанци из [52], које су генерисане за MCLP са преференцијама корисника. Скуп података из [52] садржи 6 група по 10 инстанци, где свака група садржи инстанце са истом величином проблема и истим полупречником покривања. Величина проблема је дефинисана бројем корисника $n = |J|$ и бројем потенцијалних локација објеката $m = |I_1|$. Величине проблема $n \times m$ за разматране групе инстанци су следеће: 225×25 , 450×50 , 675×75 , 900×100 , 1350×150 и 1800×200 . Вредност полупречника покривања је постављена на 0.80, 0.70, 0.50, 0.30, 0.25 и 0.20, док се преференције корисника према објектима генеришу процедуром заснованом на троугаоној расподели вероватноћа, предложеном у раду [26].

У случају MCLP-а ограниченог буџета који је разматран у овој дисертацији, у инстанце је неопходно укључити трошкове за успостављање објеката и расположиви буџет у свакој инстанци из рада [52]. За сваку локацију $i \in I_1$, трошкови c_i су изабрани на случајан начин из интервала [1000, 1500]. Вредност B се израчунава као $1000p$, где је p број објеката које треба поставити, преузет из одговарајућих оригиналних MCLP инстанци из [52]. Вредност p је иста за инстанце који припадају истој групи: 3, 6, 10, 13, 20 и 27 за инстанце величине 225×25 , 450×50 , 675×75 , 900×100 , 1350×150 и 1800×200 .

У циљу испитивања перформанси VNS метахеуристике и поређења са егзактним методама када се димензија проблема повећава, додатно је генерисана група инстанци великих димензија пратећи процедуру из рада [52]. Величине проблема $n \times m$ за инстанце у овој другој групи су: 4500×500 , 7200×800 и 9000×1000 . Ове инстанце великих димензија се генеришу на следећи начин. Прво, $n + m$ тачака су на случајан начин постављене у јединични квадрат. Затим се 10% тачака бира на случајан начин као потенцијалне локације објеката, док преостале тачке представљају локације корисника. Полупречник покри-

2.4. ЕКСПЕРИМЕНТАЛНИ РЕЗУЛТАТИ

вања је постављен на 0.5 за све инстанце у овој групи. Потражња за сваког корисника се генерише равномерно из интервала $[1, m \cdot 10]$, док је буџет постављен на $100 \cdot m$. Трошкови локација објеката и преференције корисника према објектима добијају се на исти начин као и за прву групу инстанци.

Трећа група инстанци је генерисана на основу инстанци великих димензија које су добијене на основу реалних, демографских података, а представљене у раду [123] који се бави MCLP-ом. Овај скуп података састоји се од 24 инстанце које су засноване на подацима о демографској густини америчког становништва према попису из 2010. године, а генерисани су коришћењем исте методологије као у [97]. У овим инстанцама, свака локација може представљати локацију корисника и истовремено локацију потенцијалног или успостављеног објекта [123]. Свака локација је дата својим координатама (географска ширина и дужина) и познат је број становника који живе у придруженом региону. Региони су дефинисани блоком улица једног од следећих региона: Њујорк, Бронкс, Сан Франциско и Кингс. Сваки регион дефинише групу инстанци, са истим скупом чворова корисника и полупречником покривања, док број објеката који треба поставити варира. Оригиначне групе инстанци укључују 2713, 3839, 5137 и 7730 чворова корисника. Детаљне информације о оригиналним инстанцама могу се наћи у раду [123].

MCLP инстанце из [123] су измењене на следећи начин. Из скупа чворова потражње у оригиналној групи инстанци (Њујорк, Бронкс, Сан Франциско и Кингс), на случајан начин је изабрано 10% чворова као потенцијалне локације објеката. Из овог новодобијеног скупа изабрано је 10% чворова који представљају локације са већ успостављеним објектима. Величине проблема $n \times m$ за групе измењених инстанци Њујорк, Бронкс, Сан Франциско и Кингс су следеће: 2442×271 , 3456×383 , 4624×513 и 6957×773 . Удаљеност између две локације израчуната је коришћењем географских координата, као у раду [123]. Полупречник покривања је исти за све инстанце које припадају истој групи ($R = 4000, 5000, 6000, 7000$), док се буџет добија као број чворова потражње $n \times 10$. Потражња сваког чвора је једнака величини популације у овом чвору (број корисника). Троугаона расподела се користи за израчунавање преференција чворова потражње према објектима, као у радовима [26] и [52]. За сваку потенцијалну локацију објекта, трошкови успостављања објекта се генеришу нормалном расподелом из интервала $[1000, 2000]$. Средња вредност је пропорционална укупној потражњи која би се могла покрити ако се објекат налази на

овој локацији, док је стандардна девијација постављена на 0.1.

Прелиминарни експерименти

За потребе прелиминарних експеримената, коришћен је подскуп од 18 инстанци проблема из првог скупа података (три насумично одабране инстанце проблема из сваке од шест група). Инстанце из друге групе нису разматране у овим експериментима, због њихове сложености и дужег времена извршавања VNS-а на њима.

Прво су извршени прелиминарни експерименти у циљу одређивања редоследа истраживања околина у фази размрдавања. Величина сваке околине N^1 , N^2 , N^3 и N^4 је постављена на 1 у овим експериментима и све могуће комбинације редоследа околина су испитане. VNS је покренут 5 пута на свакој инстанци из изабраног подскупа, а време извршавања VNS-а је било ограничено на $m/10$ секунди. Резултати су представљени у Табели 2.1 на следећи начин. Прва колона са леве стране садржи редослед околина, док друга колона представља просечне вредности најбољег VNS решења - *avg best*. Просечно процентуално одступање VNS решења добијеног у сваком покретању, израчунато у односу на најбоље решење дато је у трећој колони - *avg gap(%)*. Просечно време покретања у секундама у којем је VNS постигао најбоље решење у сваком покретању приказано је у четвртој колони - *t(s)*. Треба приметити да су у Табели 2.1 просечне вредности у колонама 2-4 израчунате за све инстанце из подскупа података који се користи у прелиминарним експериментима. Последње четири колоне Табеле 2.1 садрже просечан број побољшања пронађених у околинама N^1 , N^2 , N^3 и N^4 .

Према подацима приказаним у Табели 2.1, претраживање околина по редоследу N^1 , N^4 , N^3 , N^2 довело је до највеће вредности функције циља (у просеку) и овај редослед је коришћен у преосталим експериментима. ОкоLINE N^1 и N^4 су се показале најбољим за истраживање, са просечним бројем побољшања од 1.63 и 2.24. Истраживање околина N^2 и N^3 било је мање успешно, јер су у просеку обезбедиле 1.06 и 1.29 побољшања. Затим је величина свих околина повећана на два, док је редослед био фиксиран на N^1 , N^4 , N^3 , N^2 . Добијена просечна вредност функције циља од 405312.61 је нешто мања у поређењу са истраживањем околина величине један истим редоследом. Могуће је приметити да повећање величине околина N^2 и N^3 са један на два није довело до побољшања. На основу овога величине околина N^2 и N^3 су постављене на 1.

2.4. ЕКСПЕРИМЕНТАЛНИ РЕЗУЛТАТИ

Табела 2.1: Резултати прелиминарних експеримената са редоследом околина

Редослед	$avg\ best$	$avg\ gap(\%)$	$avg\ t(s)$	просечан број побољшања			
				N^1	N^2	N^3	N^4
$N^1\ N^2\ N^3\ N^4$	405190.89	0.044	1.19	3.29	0.63	0.87	1.03
$N^1\ N^2\ N^4\ N^3$	405257.50	0.045	1.07	3.13	0.47	0.58	1.55
$N^1\ N^3\ N^2\ N^4$	405164.89	0.037	1.30	3.32	0.66	0.82	1.13
$N^1\ N^3\ N^4\ N^2$	405292.11	0.042	1.03	3.47	0.58	0.82	1.29
$N^1\ N^4\ N^2\ N^3$	405347.39	0.071	1.25	3.18	0.58	0.74	1.26
$N^1\ N^4\ N^3\ N^2$	405174.00	0.053	1.40	3.24	0.50	0.71	1.63
$N^2\ N^1\ N^3\ N^4$	405123.06	0.034	1.19	1.95	1.40	1.53	1.08
$N^2\ N^1\ N^4\ N^3$	405153.89	0.037	1.72	2.16	1.40	1.55	1.63
$N^2\ N^3\ N^1\ N^4$	405180.00	0.061	1.63	1.21	1.90	2.05	1.37
$N^2\ N^3\ N^4\ N^1$	405164.89	0.061	1.42	0.74	1.87	2.05	2.26
$N^2\ N^4\ N^1\ N^3$	405009.94	0.039	1.44	1.16	1.37	1.55	2.95
$N^2\ N^4\ N^3\ N^1$	405021.78	0.038	1.33	0.76	1.42	1.58	2.79
$N^3\ N^1\ N^2\ N^4$	405147.17	0.033	0.99	1.95	1.58	1.84	1.00
$N^3\ N^1\ N^4\ N^2$	405168.56	0.025	0.99	1.71	1.42	1.90	1.32
$N^3\ N^2\ N^1\ N^4$	405085.56	0.021	0.95	0.97	1.76	1.92	1.13
$N^3\ N^2\ N^4\ N^1$	405162.94	0.038	1.37	0.47	1.87	2.00	1.92
$N^3\ N^4\ N^1\ N^2$	405189.00	0.041	1.23	0.71	1.42	1.87	2.45
$N^3\ N^4\ N^2\ N^1$	405198.61	0.029	1.02	0.42	1.61	1.97	2.26
$N^4\ N^1\ N^2\ N^3$	405197.50	0.052	1.25	0.74	0.58	0.76	3.66
$N^4\ N^1\ N^3\ N^2$	405170.94	0.053	1.20	0.87	0.50	0.76	4.18
$N^4\ N^2\ N^1\ N^3$	405291.83	0.053	1.37	0.97	0.42	0.68	3.82
$N^4\ N^2\ N^3\ N^1$	405285.83	0.051	1.34	0.92	0.58	0.82	3.90
$N^4\ N^3\ N^1\ N^2$	405288.28	0.053	1.32	0.90	0.55	0.74	4.32
$N^4\ N^3\ N^2\ N^1$	405193.39	0.032	1.42	0.92	0.45	0.74	3.87
Просек:				1.63	1.06	1.29	2.24

Табела 2.2 приказује резултате експеримената са различитим величинама околина. У овим експериментима, коришћен је редослед околина N^1, N^4, N^3, N^2 , величина околина N^2 и N^3 је постављена на један, док је величина околина N^1 и N^4 повећана до 6 са кораком један. Резултати за сваку разматрану комбинацију су израчунати и представљени на исти начин као у Табели 2.1, с тим што је у табели 2.2 коришћена ознака N_k^i за околину N^i величине k . Према резултатима приказаним у табели 2.2, може се закључити да величину околина N^1 и N^4 треба фиксирати на 5, јер је комбинација $N_5^1, N_5^4, N_1^2, N_1^3$ дала највећу вредност функције циља (у просеку), са малим просечним процентуалним одступањем и кратким просечним временом извршавања. Као што се може видети из Табеле 2.2, даље повећање величине околина N^1 и N^4 довело је

2.4. ЕКСПЕРИМЕНТАЛНИ РЕЗУЛТАТИ

до деградације вредности функције циља (у просеку). На основу спроведених прелиминарних експеримената, вредност параметра k_{max} је постављена на 12, где је $k_1 = 5$, $k_2 = 1$, $k_3 = 1$ и $k_4 = 5$.

Табела 2.2: Резултати прелиминарних експеримената са величином околина

Околина	<i>avg best</i>	<i>avg gap(%)</i>	<i>avg t(s)</i>	Околина	<i>avg best</i>	<i>avg gap(%)</i>	<i>avg t(s)</i>
$N_1^1 N_1^4 N_1^2 N_1^3$	405347.39	0.071	1.25	$N_4^1 N_1^4 N_1^2 N_1^3$	405441.06	0.027	2.14
$N_1^1 N_2^4 N_1^2 N_1^3$	405262.56	0.023	2.16	$N_4^1 N_2^4 N_1^2 N_1^3$	405464.33	0.032	1.84
$N_1^1 N_3^4 N_1^2 N_1^3$	405488.17	0.034	2.65	$N_4^1 N_3^4 N_1^2 N_1^3$	405519.72	0.029	2.64
$N_1^1 N_4^4 N_1^2 N_1^3$	405369.28	0.041	2.33	$N_4^1 N_4^4 N_1^2 N_1^3$	405461.44	0.028	2.37
$N_1^1 N_5^4 N_1^2 N_1^3$	405527.94	0.043	2.92	$N_4^1 N_5^4 N_1^2 N_1^3$	405489.67	0.031	2.75
$N_1^1 N_6^4 N_1^2 N_1^3$	405488.17	0.042	2.32	$N_4^1 N_6^4 N_1^2 N_1^3$	405540.11	0.026	2.81
$N_2^1 N_1^4 N_1^2 N_1^3$	405453.33	0.044	1.94	$N_5^1 N_1^4 N_1^2 N_1^3$	405555.67	0.028	2.77
$N_2^1 N_2^4 N_1^2 N_1^3$	405291.00	0.036	2.02	$N_5^1 N_2^4 N_1^2 N_1^3$	405435.33	0.029	2.73
$N_2^1 N_3^4 N_1^2 N_1^3$	405519.17	0.036	2.93	$N_5^1 N_3^4 N_1^2 N_1^3$	405540.11	0.015	3.48
$N_2^1 N_4^4 N_1^2 N_1^3$	405499.33	0.030	3.15	$N_5^1 N_4^4 N_1^2 N_1^3$	405509.11	0.019	2.69
$N_2^1 N_5^4 N_1^2 N_1^3$	405505.50	0.035	2.44	$N_5^1 N_5^4 N_1^2 N_1^3$	405576.06	0.026	2.75
$N_2^1 N_6^4 N_1^2 N_1^3$	405499.44	0.040	2.76	$N_5^1 N_6^4 N_1^2 N_1^3$	405519.72	0.033	3.12
$N_3^1 N_1^4 N_1^2 N_1^3$	405441.06	0.032	2.41	$N_6^1 N_1^4 N_1^2 N_1^3$	405540.11	0.020	3.11
$N_3^1 N_2^4 N_1^2 N_1^3$	405464.33	0.032	2.48	$N_6^1 N_2^4 N_1^2 N_1^3$	405540.11	0.023	2.90
$N_3^1 N_3^4 N_1^2 N_1^3$	405529.50	0.038	2.51	$N_6^1 N_3^4 N_1^2 N_1^3$	405540.11	0.027	2.84
$N_3^1 N_4^4 N_1^2 N_1^3$	405453.56	0.032	2.58	$N_6^1 N_4^4 N_1^2 N_1^3$	405462.06	0.026	2.88
$N_3^1 N_5^4 N_1^2 N_1^3$	405519.72	0.032	2.91	$N_6^1 N_5^4 N_1^2 N_1^3$	405540.11	0.038	2.73
$N_3^1 N_6^4 N_1^2 N_1^3$	405497.22	0.038	2.34	$N_6^1 N_6^4 N_1^2 N_1^3$	405540.11	0.018	3.08

На крају, извршен је и скуп прелиминарних експеримената да би се одредиле вредности параметра α који се користи у адаптивној VNS методи и параметра процентуалног повећања буџета β из VNS-а са стратешком осцилацијом. Резултати су представљени у Табелама 2.3 и 2.4, где прва колона означава вредност разматраног параметра, а подаци у преостале три колоне су представљени на исти начин као подаци одговарајућих колона у Табели 2.1. На основу добијених резултата, вредност параметра α је постављена на 0.7, док је повећање буџета β фиксирано на 35%.

2.4. ЕКСПЕРИМЕНТАЛНИ РЕЗУЛТАТИ

Табела 2.3: Резултати прелиминарних експеримената са параметром α алгорита SVNS

α	<i>avg best</i>	<i>avg gap</i> (%)	<i>avg t</i> (s)
0.1	405540.111	0.010	2.724
0.2	405540.111	0.004	2.178
0.3	405576.056	0.015	2.199
0.4	405576.056	0.012	2.845
0.5	405576.056	0.008	2.757
0.6	405576.056	0.013	2.638
0.7	405576.056	0.005	2.884
0.8	405518.167	0.009	2.598
0.9	405574.500	0.011	1.968

Табела 2.4: Резултати прелиминарних експеримената са параметром β алгорита VNS-SO

β %	<i>avg best</i>	<i>avg gap</i> (%)	<i>avg t</i> (s)
10	405540.111	0.003	2.709
15	405540.111	0.006	2.633
20	405576.056	0.008	2.656
25	405576.056	0.009	2.358
30	405576.056	0.009	2.591
35	405576.056	0.006	2.740
40	405540.111	0.003	3.174
45	405540.111	0.004	2.972
50	405540.111	0.005	2.576

Резултати на инстанцама са ограниченим буџетом

Табела 2.5 приказује детаљне резултате Gurobi решавача, предложених VNS, SVNS и VNS-SO метода на првом скупу инстанци. За ове инстанце, Gurobi решавач је ограничен на 1 сат извршавања. Параметар критеријума заустављања I_{max} за све VNS методе има вредност $m/10$, где је m број потенцијалних локација објеката. Прва колона Табеле 2.5 садржи ознаку разматране инстанце у формату $n \times m - k$, где $n \times m$ означава величину проблема, а $k = 1, \dots, 10$ је број додељен посматраној инстанци у одговарајућој групи. Следеће две колоне показују вредност функције циља оптималног решења Opt или вредност

2.4. ЕКСПЕРИМЕНТАЛНИ РЕЗУЛТАТИ

доње границе LB , добијене Gurobi решавачем, као и време извршавања које је решавачу било потребно $t(s)$ за рад (у секундама). У следеће четири колоне представљени су подаци који се односе на VNS резултате: најбоља вредност функције циља $best$ добијена у 5 покретања, просечно укупно време извршавања (у секундама) које је VNS методи било потребно да заврши свако покретање - $tot.t(s)$, просечно време извршавања (у секундама) у којем је VNS пронашао своје најбоље решење у сваком покретању - $t(s)$, процентуално одступање средње и најлошије, односно најмање вредности функције циља добијене у 5 VNS покретања, рачунато у односу на вредност функције циља оптималног или најбољег решења ($mean\ gap(\%)$ and $worst\ gap(\%)$). У преосталим колонама Табеле 2.5, резултати VNS-SO и SVNS метода су представљени на исти начин као и за VNS. Оптимална и најбоља позната решења су подебљана у Табели 2.5.

Резултати приказани у Табели 2.5 показују да сва три VNS приступа достижу сва оптимална решења која даје Gurobi решавач. У случају две инстанце из групе 1350×150 и 8 инстанци из групе 1800×200 , све три VNS методе су побољшале доње границе које је дао Gurobi. За две инстанце величине 1800×200 , најбоља VNS, VNS-SO и SVNS решења се поклапају са допустивим које враћа Gurobi решавач у року од једног сата. Ниске вредности средњег и најгорег одступања од оптималних решења указују на високу стабилност све три варијанте VNS. За 46 од 60 тестираних инстанци, средње и највеће одступање VNS решења су 0%, што значи да је VNS достигао оптимално решење у свих 5 покретања. Слично, VNS-SO је вратио решење са средњим и најгорим одступањем од 0% за 49 од 60 инстанци. Предложени SVNS је показао нешто боље перформансе од VNS-а и VNS-SO-а у погледу стабилности решења, пошто су средње и највеће одступање били 0% за 55 од 60 инстанци. Средње вредности одступања најбољих VNS, VNS-SO и SVNS решења нису прелазила вредности редом 0.112%, 0.220% и 0.026%, а одступања најлошијих решења нису прелазиле вредности редом 0.366%, 0.366% и 0.068%.

Што се тиче времена извршавања, предложене VNS методе су се показале супериорним у односу на Gurobi за све разматране инстанце проблема. На свих 10 инстанци у групи највећих димензија 1800×200 , Gurobi решавач ради максимално 3600 секунди, док су VNS, VNS-SO и SVNS пронашли најбоље решење за мање од 7, 13 и 15 секунди. За другу највећу групу инстанци 1350×150 , Gurobi решавач је достигао временско ограничење у 5 од 10 инстанци. Са друге стране, VNS, VNS-SO и SVNS су дали најбоље решење за мање од 3, 5 и

2.4. ЕКСПЕРИМЕНТАЛНИ РЕЗУЛТАТИ

6 секунди. Укупно време извршавања сва три VNS приступа остало је кратко, чак и у случају инстанци великих димензија из првог скупа података. Међутим, може се приметити да су VNS-SO и SVNS били спорији (у просеку) у односу на VNS.

Табела 2.6 даје сажети приказ резултата добијених помоћу Gurobi решавача и предложених VNS метода на инстанцама из првог скупа података груписаних по величини проблема. Прва колона Табеле 2.6 садржи димензију проблема $n \times m$ разматране групе инстанци. У трећој колони приказане су просечне вредности функције циља (*avg obj*) решења добијених Gurobi решавачем, док је у четвртој колони приказано просечно време извршавања *avg t(s)*. За сваку од предложених VNS метода, дати су следећи подаци: просечне вредности функције циља најбољих решења, просечно укупно време извршавања *avg tot.t(s)*, просечно време потребно VNS методи да нађе најбоље решење у сваком покретању *avg. t(s)*, просечно процентуално одступање средње вредности решења у односу на оптимално или најбоље решење (*mean gap(%)*), просечно процентуално одступање најлошијег решења добијено такође у односу на оптимално или најбоље решење (*worst gap(%)*). Просечне вредности представљене у Табели 2.6 су израчунате за све инстанце које припадају истој групи (тј. које имају исту величину $n \times m$). За сваку групу инстанци у Табели 2.6, просечне вредности оптималних решења или најбољих решења су подебљане.

Из сажетог приказа резултата представљених у Табели 2.6, може се уочити да се решења која је дао Gurobi и предложени VNS поклапају за све инстанце у прве четири групе: 225×25 , 450×50 , 675×75 и 900×100 . У случају две групе са инстанцама највећих димензија (1350×150 и 1800×200), просечна вредност функције циља најбољих решења добијених сваком од предложених VNS варијанти била је боља од просечне вредности најбољег решења које је дао Gurobi. На ове две групе инстанци, просечно побољшање најбољег решења добијеног предложеним VNS методама у односу на најбоље решење које је дао Gurobi износи 0.07% и 0.16%, респективно.

Подаци представљени у колонама *avg t(s)* и *avg tot.t(s)* Табеле 2.6 указују на ефикасност све три VNS методе на првом скупу података. За инстанце највећих димензија 1800×200 , предложени VNS, VNS-SO и SVNS су постигли најбоље решење за 3.69, 5.93 и 5.94 секунде (у просеку), док је просечно укупно време извршавања било 16.33, 29.91 и 29.22 секунди. Са друге стране, Gurobi је потрошио максимално време извршавања за сваку инстанцу из ове групе, а про-

2.4. ЕКСПЕРИМЕНТАЛНИ РЕЗУЛТАТИ

сечни квалитет решења био је нижи у поређењу са квалитетом најбољих VNS решења. Колоне које се односе на просечне вредности одступања показују стабилност све три VNS имплементације. Просечне *mean gap*(%) и *worst gap*(%) вредности биле су значајно испод 0.1% за сва три VNS приступа. Из сажетог приказа резултата на првом скупу података уочава се да су разматране VNS варијанте показале веома сличне перформансе у погледу квалитета решења. Треба приметити да је, у просеку, SVNS добијао оптимално или најбоље решење са бољом стабилношћу у поређењу са VNS и VNS-SO. У погледу времена извршавања, VNS био бржи у поређењу са друга два VNS приступа, посебно на већим инстанцама проблема.

Детаљни резултати експеримената на новим инстанцама великих димензија (други скуп података) су представљени у Табели 2.7. Инстанце из другог скупа података захтевају више рачунарског времена, стога је ограничење за Gurobi решавач постављено на 3 сата. Табела 2.7 је организована на исти начин као Табела 2.5. Као што се може видети из података представљених у Табели 2.7, Gurobi решавач је дао само допустива решења (доње границе) за инстанце димензија 4500×500 и 7200×800 , док су најбоља решења добијена извршавањем све три VNS метахеуристике бољег квалитета. У просеку, побољшања најбољих VNS решења у односу на доње границе које је дао Gurobi на инстанцама 4500×500 и 7200×800 су редом 4.38% и 4.73%. За највећу групу инстанци 9000×1000 у другој групи, Gurobi решавач је остао без меморије (64 GB RAM меморије је искоришћено) и није добио чак ни допустиво решење. Са друге стране, све три VNS методе су дале своја најбоља решења за релативно кратко време, имајући у виду димензију проблема.

Сажети приказ експерименталних резултата на инстанцама другог скупа података дат у Табели 2.8 јасно оправдава употребу метахеуристичког приступа заснованог на VNS методи при решавању инстанци проблема великих димензија. Како се димензија проблема повећава, предност коришћења VNS метода у односу на Gurobi решавач постаје све очигледнија, у погледу квалитета добијених решења и потребних времена извршавања. У просеку, за инстанце 4500×500 и 7200×800 , SVNS је дао боља решења и показао се стабилнијим у односу на VNS и VNS-SO методе, док је VNS-SO био најбољи у погледу квалитета решења и стабилности на инстанцама 9000×1000 . Што се тиче времена извршавања, VNS је био најбржи, али су све три VNS методе дале решења у релативно кратком времену. На пример, приликом решавања највеће групе ин-

2.4. ЕКСПЕРИМЕНТАЛНИ РЕЗУЛТАТИ

станци 9000×1000 , просечне $t(s)$ вредности биле су око 5 минута за VNS, 41 минут за VNS-SO и 18 минута за SVNS.

Табела 2.5: Детаљни резултати добијени помоћу Gurobi решавача и метода заснованих на VNS методи на првом скупу података

Инстанца	Gurobi		VNS				VNS-SO				SVNS						
	Opt/LB	t (s)	best	avg tot.t (s)	avg gap (%)	mean worst gap (%)	best	avg tot.t (s)	avg gap (%)	mean worst gap (%)	best	avg tot.t (s)	avg gap (%)	mean worst gap (%)			
225x25-01	17658	5.71	17658	0.53	0.02	0.00	0.00	17658	0.42	0.01	0.00	0.00	17658	0.39	0.02	0.00	0.00
225x25-02	15823	3.07	15823	0.30	0.01	0.00	0.00	15823	0.44	0.01	0.00	0.00	15823	0.40	0.01	0.00	0.00
225x25-03	16751	4.52	16751	0.19	0.01	0.00	0.00	16751	0.29	0.00	0.00	0.00	16751	0.49	0.01	0.00	0.00
225x25-04	18321	3.81	18321	0.34	0.01	0.00	0.00	18321	0.49	0.02	0.00	0.00	18321	0.39	0.01	0.00	0.00
225x25-05	17601	1.69	17601	0.21	0.01	0.00	0.00	17601	0.29	0.00	0.00	0.00	17601	0.27	0.01	0.00	0.00
225x25-06	17009	5.18	17009	0.32	0.01	0.00	0.00	17009	0.40	0.01	0.00	0.00	17009	0.39	0.01	0.00	0.00
225x25-07	16880	4.51	16880	0.20	0.01	0.00	0.00	16880	0.28	0.01	0.00	0.00	16880	0.63	0.01	0.00	0.00
225x25-08	17436	4.76	17436	0.24	0.01	0.00	0.00	17436	0.29	0.01	0.00	0.00	17436	0.40	0.01	0.00	0.00
225x25-09	16668	5.65	16668	0.28	0.01	0.00	0.00	16668	0.44	0.01	0.00	0.00	16668	0.39	0.01	0.00	0.00
225x25-10	17753	6.32	17753	0.43	0.01	0.00	0.00	17753	0.42	0.01	0.00	0.00	17753	0.40	0.02	0.00	0.00
450x50-01	66527	15.99	66527	1.63	0.13	0.00	0.00	66527	1.53	0.08	0.00	0.00	66527	2.22	0.06	0.00	0.00
450x50-02	74217	17.99	74217	1.30	0.09	0.00	0.00	74217	1.37	0.11	0.00	0.00	74217	4.56	0.02	0.00	0.00
450x50-03	70565	17.68	70565	1.76	0.36	0.00	0.00	70565	1.77	0.30	0.037	0.183	70565	2.33	0.29	0.00	0.00
450x50-04	63103	9.55	63103	1.12	0.02	0.00	0.00	63103	1.27	0.03	0.00	0.00	63103	1.71	0.02	0.00	0.00
450x50-05	75519	15.30	68745	1.21	0.02	0.00	0.00	68745	1.32	0.02	0.00	0.00	68745	1.29	0.02	0.00	0.00
450x50-06	68745	15.66	67171	1.16	0.02	0.00	0.00	67171	1.39	0.02	0.00	0.00	67171	3.03	0.02	0.00	0.00
450x50-07	67171	13.57	73460	1.26	0.02	0.00	0.00	73460	1.44	0.03	0.00	0.00	73460	2.67	0.04	0.00	0.00
450x50-08	73460	11.93	69668	1.29	0.06	0.00	0.00	69668	1.59	0.02	0.00	0.00	69668	2.66	0.02	0.00	0.00
450x50-09	69668	16.86	69180	1.22	0.02	0.00	0.00	69180	1.33	0.02	0.00	0.00	69180	2.10	0.03	0.00	0.00
450x50-10	69180	12.50	75519	1.32	0.02	0.00	0.00	75519	1.32	0.02	0.00	0.00	75519	1.29	0.02	0.00	0.00

Табела 2.5: (Наставак) Детаљни резултати добијени помоћу Gurobi решавача и метода заснованих на VNS-у на првом скупу података

Инстанца	Gurobi			VNS			VNS-SO			SVNS							
	Opt/LB	t (s)	best	avg tot.t (s)	avg mean gap (%)	worst gap (%)	best	avg tot.t (s)	avg mean gap (%)	worst gap (%)	best	avg tot.t (s)	avg mean gap (%)	worst gap (%)			
675x75-01	174947	35.86	174947	2.46	0.05	0.000	0.000	174947	3.99	0.12	0.000	0.000	174947	4.52	0.08	0.000	0.000
675x75-02	162609	377.83	162609	2.35	0.16	0.000	0.000	162609	2.60	0.14	0.000	0.000	162609	5.03	0.07	0.000	0.000
675x75-03	170073	80.93	170073	2.84	0.83	0.096	0.366	170073	4.82	0.48	0.220	0.366	170073	5.32	1.37	0.000	0.000
675x75-04	163495	62.25	163495	2.48	0.35	0.000	0.000	163495	2.59	0.36	0.000	0.000	163495	4.82	0.82	0.000	0.000
675x75-05	171776	89.65	171776	2.85	0.41	0.000	0.000	171776	3.34	0.66	0.000	0.000	171776	5.70	0.83	0.000	0.000
675x75-06	161829	62.29	161829	2.81	0.31	0.000	0.000	161829	2.34	0.47	0.000	0.000	161829	5.31	0.60	0.000	0.000
675x75-07	187997	82.95	187997	2.53	0.39	0.000	0.000	187997	2.76	0.50	0.000	0.000	187997	4.53	0.66	0.000	0.000
675x75-08	172497	62.45	172497	2.36	0.29	0.000	0.000	172497	2.86	0.53	0.000	0.000	172497	5.70	0.65	0.000	0.000
675x75-09	188528	87.89	188528	2.54	0.47	0.000	0.000	188528	2.76	0.67	0.000	0.000	188528	4.46	0.79	0.000	0.000
675x75-10	173424	78.42	173424	2.14	0.05	0.000	0.000	173424	2.32	0.05	0.000	0.000	173424	4.74	0.10	0.000	0.000
900x100-01	278660	224.22	278660	5.06	0.53	0.000	0.000	278660	5.19	0.15	0.000	0.000	278660	6.05	0.18	0.000	0.000
900x100-02	304929	177.55	304929	5.91	1.60	0.025	0.127	304929	7.86	1.11	0.051	0.127	304929	6.39	0.51	0.000	0.000
900x100-03	305420	129.00	305420	3.71	0.14	0.000	0.000	305420	4.70	0.06	0.000	0.000	305420	6.95	0.13	0.000	0.000
900x100-04	285587	153.54	285587	3.59	0.29	0.000	0.000	285587	5.63	0.04	0.000	0.000	285587	6.52	0.22	0.000	0.000
900x100-05	304280	94.76	304280	3.56	0.34	0.000	0.000	304280	3.81	0.06	0.000	0.000	304280	7.26	0.27	0.000	0.000
900x100-06	300282	105.21	300282	3.66	0.30	0.000	0.000	300282	5.98	0.76	0.000	0.000	300282	6.14	0.09	0.000	0.000
900x100-07	292831	456.24	292831	3.81	0.50	0.000	0.000	292831	5.01	0.21	0.000	0.000	292831	6.17	0.16	0.000	0.000
900x100-08	307865	93.96	307865	3.65	0.39	0.000	0.000	307865	6.09	0.75	0.000	0.000	307865	6.11	0.33	0.000	0.000
900x100-09	294441	318.79	294441	3.78	0.63	0.005	0.012	294441	6.05	0.07	0.000	0.000	294441	8.01	1.59	0.000	0.000
900x100-10	281827	159.65	281827	3.39	0.43	0.107	0.134	281827	7.83	0.87	0.107	0.134	281827	8.34	1.15	0.000	0.000
1350x150-01	655207	3172.28	655207	7.22	0.21	0.000	0.000	655207	13.09	0.61	0.000	0.000	655207	13.68	1.09	0.000	0.000
1350x150-02	683018	3600.00	685479	7.68	0.64	0.000	0.000	685479	15.25	0.77	0.055	0.275	685479	12.71	0.72	0.000	0.000
1350x150-03	704045	3600.00	704045	8.90	1.49	0.003	0.015	704045	22.63	4.49	0.012	0.015	704045	15.20	2.04	0.000	0.000
1350x150-04	646917	3600.00	646917	9.14	2.17	0.112	0.200	646917	16.24	1.99	0.000	0.000	646917	17.65	5.93	0.000	0.000
1350x150-05	639754	1812.33	639754	8.31	1.75	0.012	0.030	639754	13.47	3.09	0.000	0.000	639754	17.56	4.89	0.000	0.000
1350x150-06	664771	85.99	664771	7.13	0.31	0.000	0.000	664771	10.26	0.22	0.000	0.000	664771	18.09	0.12	0.000	0.000
1350x150-07	628974	3600.00	631164	10.03	2.29	0.009	0.026	631164	23.85	3.27	0.004	0.019	631164	18.03	5.28	0.009	0.019
1350x150-08	649436	3600.00	649436	8.45	0.92	0.000	0.000	649436	15.86	2.52	0.000	0.000	649436	15.28	2.16	0.000	0.000
1350x150-09	690608	275.08	690608	7.43	0.43	0.000	0.000	690608	10.02	0.38	0.000	0.000	690608	15.03	0.28	0.000	0.000
1350x150-10	685665	546.49	685665	7.17	0.19	0.000	0.000	685665	7.76	0.13	0.000	0.000	685665	19.58	0.54	0.000	0.000

Табела 2.5: (Наставак) Детаљни резултати добијени помоћу Gurobi решавача и метода заснованих на VNS-у на првом скупу података

Инстанца	Gurobi			VNS			VNS-SO			SVNS							
	Opt/LB	t (s)	best	avg	avg	mean worst	best	avg	avg	mean worst	best	avg	avg	mean worst			
			(s)	tot.t	t(s)	gap	(s)	tot.t	t(s)	gap	(s)	tot.t	t(s)	gap			
			(%)	(s)	(%)	(%)	(s)	(s)	(%)	(%)	(s)	(s)	(%)	(%)			
1800x200-01	1191252	3600.00	1192836	15.28	2.02	0.105	0.253	1192836	38.22	12.19	0.009	0.045	1192836	26.04	2.30	0.000	0.000
1800x200-02	1283792	3600.00	1286009	14.80	3.84	0.094	0.135	1286009	24.00	8.78	0.000	0.000	1286009	36.80	10.84	0.014	0.068
1800x200-03	1178417	3600.00	1179235	16.55	4.98	0.000	0.000	1179235	24.99	5.02	0.042	0.069	1179235	23.18	0.51	0.000	0.000
1800x200-04	1246845	3600.00	1246845	15.87	4.15	0.007	0.036	1246845	25.07	0.44	0.000	0.000	1246845	24.69	4.06	0.000	0.000
1800x200-05	1235246	3600.00	1235246	15.96	4.37	0.000	0.000	1235246	29.88	1.96	0.000	0.000	1235246	23.71	2.03	0.000	0.000
1800x200-06	1179960	3600.00	1180532	15.52	1.07	0.000	0.000	1180532	19.63	1.02	0.000	0.000	1180532	23.77	1.27	0.000	0.000
1800x200-07	1192326	3600.00	1195530	17.78	4.69	0.073	0.158	1195530	38.39	5.96	0.000	0.000	1195530	39.89	14.23	0.026	0.066
1800x200-08	1192330	3600.00	1198190	23.71	6.90	0.014	0.017	1198190	36.67	12.70	0.000	0.000	1198190	33.48	9.91	0.003	0.015
1800x200-09	1215459	3600.00	1219321	13.77	2.53	0.068	0.252	1219321	29.30	2.59	0.024	0.030	1219321	29.51	5.69	0.006	0.030
1800x200-10	1249664	3600.00	1251010	14.01	2.35	0.000	0.000	1251010	32.95	8.67	0.012	0.060	1251010	31.09	8.57	0.000	0.000

Табела 2.6: Сажети резултати добијени помоћу Gurobi решавача и метода заснованих на VNS-у на првом скупу података

Група	Gurobi			VNS			VNS-SO			SVNS							
	avg obj	avg t(s)	best	avg	avg	mean worst	best	avg	avg	mean worst	best	avg	avg	mean worst			
			(s)	tot.t	t(s)	gap	(s)	tot.t	t(s)	gap	(s)	tot.t	t(s)	gap			
			(%)	(s)	(%)	(%)	(s)	(s)	(%)	(%)	(s)	(s)	(%)	(%)			
225x25	17190.0	4.52	17190.00	0.30	0.01	0.000	0.000	17190.0	0.38	0.01	0.000	0.000	17190.0	0.42	0.01	0.000	0.000
450x50	69815.5	14.7	69815.50	1.33	0.08	0.000	0.000	69815.5	1.43	0.07	0.004	0.018	69815.5	2.39	0.05	0.000	0.000
675x75	172717.5	102.05	172717.50	2.54	0.33	0.010	0.037	172717.5	3.04	0.40	0.022	0.037	172717.5	5.01	0.60	0.000	0.000
900x100	295612.2	191.29	295612.20	4.01	0.52	0.014	0.027	295612.2	5.82	0.41	0.016	0.026	295612.2	6.79	0.46	0.000	0.000
1350x150	664839.5	2389.22	665304.60	8.15	1.04	0.014	0.027	665304.6	14.84	1.75	0.007	0.031	665304.6	16.28	2.31	0.001	0.002
1800x200	1216529.1	3600.00	1218475.40	16.33	3.69	0.036	0.085	1218475.4	29.91	5.93	0.009	0.020	1218475.4	29.22	5.94	0.005	0.018

Табела 2.7: Детаљни резултати добијени помоћу Gurobi решавача и метода заснованих на VNS-у на другом скупу података

Инстанца	Gurobi			VNS			VNS-SO			SVNS								
	Opt/LB	t (s)		best	avg tot.t (s)	mean gap (%)	worst gap (%)	best	avg tot.t (s)	mean gap (%)	worst gap (%)	best	avg tot.t (s)	mean gap (%)	worst gap (%)			
4500x500	5738930	10800		6129520	163.18	40.55	0.035	0.173	6129520	369.64	86.92	0.035	0.173	6129520	245.56	53.32	0.000	0.000
4500x500	6072596	10800		6372212	194.90	79.09	0.065	0.071	6375800	389.89	68.42	0.051	0.071	6375800	240.17	68.47	0.000	0.000
4500x500	5911177	10800		6131535	150.79	33.93	0.022	0.067	6131535	282.75	27.66	0.000	0.000	6131535	191.46	11.30	0.000	0.000
4500x500	5842170	10800		6118252	123.22	15.32	0.079	0.395	6118252	342.79	24.85	0.018	0.089	6118252	227.11	44.57	0.018	0.089
4500x500	5950470	10800		6247778	147.95	32.82	0.159	0.294	6247778	403.02	146.19	0.100	0.268	6247778	238.36	51.31	0.069	0.344
4500x500	5959484	10800		6199175	132.98	24.81	0.092	0.350	6199175	401.45	77.77	0.022	0.112	6199175	212.27	37.06	0.000	0.000
4500x500	5888663	10800		6118776	151.12	40.94	0.077	0.325	6118776	353.75	90.68	0.024	0.032	6118776	236.94	61.36	0.011	0.028
4500x500	5886271	10800		6076589	156.83	44.56	0.061	0.083	6077432	458.60	121.48	0.046	0.058	6079401	265.63	94.56	0.016	0.046
4500x500	5872482	10800		6113945	153.87	45.18	0.037	0.149	6113945	490.37	194.53	0.000	0.000	6113945	267.36	97.25	0.000	0.000
4500x500	6201406	10800		6398528	134.96	25.25	0.037	0.184	6398528	323.70	27.27	0.000	0.000	6398528	193.69	16.87	0.000	0.000
7200x800	14982896	10800		15834433	690.56	174.87	0.085	0.238	15834433	2597.52	171.27	0.000	0.000	15834433	896.64	107.55	0.031	0.154
7200x800	15331141	10800		15922329	723.04	213.31	0.063	0.099	15923753	2649.16	677.33	0.055	0.125	15925772	1218.99	490.68	0.024	0.038
7200x800	14904938	10800		15636762	638.04	145.93	0.156	0.257	15650269	2704.49	747.96	0.086	0.224	15650269	881.20	133.70	0.084	0.119
7200x800	15597689	10800		16206741	838.41	357.55	0.027	0.051	16206741	2943.30	885.20	0.006	0.011	16206741	1024.67	283.02	0.006	0.011
7200x800	14892523	10800		15767790	697.80	200.26	0.067	0.110	15776189	2359.08	239.51	0.044	0.054	15776366	1185.96	419.43	0.001	0.001
7200x800	14649811	10800		15535813	699.01	221.40	0.140	0.370	15535813	1797.10	181.78	0.000	0.000	15535813	884.84	103.98	0.000	0.000
7200x800	14873705	10800		15604646	652.06	179.90	0.147	0.239	15604646	2880.54	362.73	0.006	0.024	15604646	1408.97	688.42	0.003	0.013
7200x800	15022405	10800		15684700	752.30	271.58	0.341	0.442	15728760	2547.50	690.61	0.040	0.099	15728760	1245.93	454.57	0.076	0.281
7200x800	15044417	10800		15576476	568.53	104.66	0.102	0.348	15574396	3512.52	975.95	0.016	0.029	15574396	1157.05	421.07	0.013	0.013
7200x800	15106019	10800		15670322	783.65	282.15	0.022	0.039	15672403	2492.44	400.78	0.027	0.044	15672403	1092.60	339.18	0.012	0.032
9000x1000	Out of memory			25131112	1214.33	277.81	0.091	0.127	25131112	8360.50	2992.390	0.077	0.108	25133424	2285.50	837.13	0.054	0.098
9000x1000	Out of memory			24253848	1180.90	266.33	0.247	0.343	24301323	8095.52	2703.500	0.124	0.207	24261996	2690.87	1118.49	0.181	0.193
9000x1000	Out of memory			24666191	1074.54	176.94	0.169	0.223	24678149	7623.44	2224.600	0.108	0.215	24678149	2306.45	689.89	0.000	0.000
9000x1000	Out of memory			24278504	1331.80	461.12	0.142	0.361	24278504	7577.87	2332.630	0.034	0.124	24278504	3145.20	1536.34	0.109	0.152
9000x1000	Out of memory			24756205	1311.11	419.91	0.044	0.069	24757098	10239.443	080.240	0.035	0.049	24763227	2924.80	1297.08	0.025	0.035
9000x1000	Out of memory			24776614	1553.58	646.07	0.147	0.205	24805066	7809.07	2506.020	0.012	0.060	24805066	2402.01	769.87	0.049	0.125
9000x1000	Out of memory			24575588	1167.54	310.63	0.093	0.116	24590130	8026.78	2335.630	0.045	0.069	245883397	2715.18	1037.45	0.035	0.049
9000x1000	Out of memory			24073537	978.72	144.19	0.211	0.326	24071536	8002.23	1968.850	0.028	0.067	24075325	3118.68	1478.45	0.020	0.083
9000x1000	Out of memory			24873079	1032.59	209.20	0.224	0.315	24895201	7756.79	2470.810	0.073	0.120	24908127	2916.43	1256.27	0.056	0.084
9000x1000	Out of memory			24642581	1011.08	180.34	0.060	0.172	24642715	6817.01	1856.660	0.025	0.074	24642581	2376.17	788.58	0.016	0.074

Табела 2.9: (Наставак) Детаљни резултати добијени помоћу Gurobi решавача и VNS метода на трећем скупу података

Инстанца	Дим.	Gurobi			VNS			VNS-SO			SVNS		
		<i>Opt/LB(s)</i>	<i>best</i>	<i>avg</i> <i>tot.t</i> (s)	<i>avg</i> <i>tot.t</i> (s)	<i>mean</i> <i>gap</i> (%)	<i>worst</i> <i>gap</i> (%)	<i>best</i>	<i>avg</i> <i>tot.t</i> (s)	<i>avg</i> <i>tot.t</i> (s)	<i>mean</i> <i>gap</i> (%)	<i>worst</i> <i>gap</i> (%)	
San Fr-01	4624x513	35486910800	358459	107.80 5.32	0.000 0.000	0.000 0.000	70.91	421.38	358459	230.59	49.92	0.000 0.000	
San Fr-02	4624x513	32458310800	332922	94.33 8.63	0.115 0.452	0.007 0.034	143.24	466.38	332922	192.81	14.76	0.014 0.034	
San Fr-03	4624x513	35299710800	366041	160.14 28.41	0.177 0.239	0.094 0.094	68.23	540.76	365922	307.26	129.64	0.081 0.134	
San Fr-04	4624x513	35187510800	360217	147.62 41.45	0.000 0.000	0.000 0.000	80.25	530.40	360217	212.21	37.30	0.000 0.000	
San Fr-05	4624x513	34646310800	356207	196.97 40.13	0.321 0.856	0.020 0.098	85.93	466.43	356227	245.94	72.46	0.000 0.000	
San Fr-06	4624x513	34406710800	348580	208.29 71.04	0.176 0.435	0.110 0.435	218.01	596.34	348580	265.57	81.49	0.000 0.000	
San Fr-07	4624x513	33656710800	348239	121.49 23.96	0.000 0.000	0.000 0.000	52.11	413.57	348239	240.08	51.89	0.000 0.000	
San Fr-08	4624x513	34640910800	353592	123.20 24.08	0.156 0.319	0.000 0.000	70.50	484.16	353592	202.67	24.55	0.000 0.000	
San Fr-09	4624x513	34046210800	348081	149.01 37.37	0.065 0.247	0.016 0.079	144.66	539.85	348081	260.27	70.16	0.000 0.000	
San Fr-10	4624x513	34283710800	353046	187.64 26.67	0.101 0.160	0.008 0.013	200.03	680.85	353167	302.39	119.76	0.000 0.000	
Kings-01	6957x773	Out of memory	1091380	608.96 210.93	0.407 0.468	0.239 0.419	533.75	2128.62	1094406	962.83	428.55	0.025 0.094	
Kings-02	6957x773	Out of memory	1090874	519.49 166.21	0.086 0.203	0.064 0.103	363.20	2072.88	1090874	961.88	433.55	0.027 0.044	
Kings-03	6957x773	Out of memory	1105930	486.10 147.95	0.055 0.140	0.029 0.100	343.50	1936.48	1106057	1303.21	784.80	0.007 0.011	
Kings-04	6957x773	Out of memory	1109133	568.51 193.37	0.248 0.405	0.062 0.171	380.36	2011.95	1110046	900.98	380.08	0.037 0.090	
Kings-05	6957x773	Out of memory	1092066	544.01 99.31	0.162 0.261	0.059 0.164	1049.51	2755.55	1092066	684.39	178.22	0.000 0.000	
Kings-06	6957x773	Out of memory	1108099	529.19 126.35	0.354 0.542	0.000 0.002	731.25	2497.30	1110709	1047.60	502.53	0.000 0.000	
Kings-07	6957x773	Out of memory	1083568	827.72 304.46	0.179 0.263	0.216	1398.68	3311.17	1084903	968.17	454.00	0.060 0.082	
Kings-08	6957x773	Out of memory	1119334	587.84 176.49	0.132 0.234	0.070 0.119	305.07	2155.96	1118731	926.88	333.38	0.057 0.064	
Kings-09	6957x773	Out of memory	1102587	573.08 300.75	0.059 0.065	0.077	1290.82	3014.10	1103092	1142.64	595.16	0.034 0.056	
Kings-10	6957x773	Out of memory	1090944	416.87 175.54	0.023 0.024	0.014 0.069	902.39	2718.88	1091168	766.14	259.59	0.004 0.021	

Табела 2.10: Сажети резултати добијени помоћу Gurobi решавача и метода заснованих на VNS методи на трећем скупу података

Група	Дим.	Gurobi			VNS			VNS-SO			SVNS		
		<i>obj</i>	<i>t(s)</i>	<i>avg</i> <i>best</i> <i>tot.t</i> (s)	<i>avg</i> <i>t(s)</i> (s)	<i>avg</i> <i>mean</i> <i>gap</i> (%)	<i>avg</i> <i>mean</i> <i>worst</i> <i>gap</i> (%)	<i>avg</i> <i>t(s)</i> (s)	<i>avg</i> <i>t(s)</i> (s)	<i>avg</i> <i>mean</i> <i>worst</i> <i>gap</i> (%)	<i>avg</i> <i>t(s)</i> (s)	<i>avg</i> <i>mean</i> <i>worst</i> <i>gap</i> (%)	
New York	2442x271	662540.4	10800	668126.8	30.97	5.10	0.008 0.010	6.00	668126.8	49.61	6.00	0.000 0.000	
Bronx	3456x383	605135.2	10800	614418.0	72.09	14.94	0.053 0.116	156.88	614441.1	156.88	33.76	0.029 0.057	
San Fr.	4624x513	344112.9	10800	352538.4	149.65	30.71	0.111 0.271	514.01	352540.6	514.01	113.39	0.026 0.075	
Kings	6957x773	Out of memory	1099391.5	566.18	190.13	0.170 0.260	2460.29	1100106.2	1100180.89666.47	2460.29	729.85	0.065 0.144	

2.4. ЕКСПЕРИМЕНТАЛНИ РЕЗУЛТАТИ

Детаљни резултати Gurobi решавача и предложеног VNS-а на инстанцама проблема изведених на основу демографских података (трећи скуп података) приказани су у Табели 2.9. За ове инстанце такође је постављено временско ограничење од 3 сата за Gurobi решавач. Табела 2.9 је организована на следећи начин. Прва колона означава групу и број разматране инстанце (на пример, NewYork-01), док друга колона приказује њену величину у формату $n \times m$. Преостале колоне које се односе на Gurobi и VNS резултате имају исту структуру и значење као и одговарајуће колоне у Табели 2.5.

Резултати представљени у Табели 2.9 указују на предности предложених VNS метода у односу на Gurobi решавач када се ради о захтевним инстанцама проблема великих димензија. За инстанце у прве три групе (Њујорк, Бронкс и Сан Франциско), побољшане су све доње границе које је Gurobi дао у датом временском ограничењу. Само у два случаја (NewYork-03 и NewYork-06), најбоља VNS решења се поклапају са LB добијеним Gurobi решавачем. Побољшања најбољих VNS решења у поређењу са доњим границама које је дао Gurobi за прве три групе инстанци износе 0.85%, 1.54% и 2.62%, респективно. За инстанце у четвртој групи (Кингс), Gurobi решавачу је понестало меморије, иако је 64 GB RAM меморије било на располагању. Са друге стране, предложене VNS методе су дале најбоље решење за релативно кратко време извршавања, док максимална меморија коју користи VNS никада није прелазила 200 MB RAM меморије.

Сажети приказ VNS резултата на трећем скупу података дат је у Табели 2.10, која представља просечне податке израчунате за све инстанце које припадају истој групи. Као што се може приметити, SVNS је био у просеку супериорнији у односу на VNS и VNS-SO и по квалитету решења и по стабилности, док је VNS имао најкраће просечно време извршавања. Вредности средњег и најлошијег одступања од најбољег решења су у просеку веома мале ($avg\ mean\ gap \leq 0.170\%$ и $avg\ worst\ gap \leq 0.271\%$) што указује на стабилност све три предложене VNS методе при решавању реалних инстанци. Чак и за највеће (Кингс) инстанце, просечно време извршавања у којем је VNS дао своја најбоља решења било је мање од 730 секунди. Просечно укупно време извршавања VNS-а је такође било релативно кратко (до 2461 секунде) имајући у виду сложеност и димензију разматраних инстанци из трећег скупа тест података.

Резултати на инстанцама са фиксним бројем објеката

У оквиру овог одељка испитане су перформансе предложених VNS метода при решавању варијанте MCLP која је најсличнија проблему разматраном у овој дисертацији. Прецизније, у питању је варијанта MCLP-а са преференцијама корисника и фиксним бројем p објеката за постављање, која је уведена у раду [52] и решавана хибридном GRASP-TS метахеуристиком. Треба приметити да је ова MCLP варијанта специјалан случај проблема MCLP ограниченог буџета са преференцијама корисника који се разматра у овој дисертацији, јер се постављањем трошкова отварања објеката на 1 и ограничења буџета на p (број објеката који ће бити постављени) проблем своди на MCLP варијанту из рада [52]. Иста MCLP варијанта је разматрана и у раду [133], где је предложен хибридни RVNS-TS-VNS алгоритам као приступ решавању проблема.

Прилагођавање VNS, VNS-SO и SVNS имплементација предложених у овој дисертацији за решавање MCLP варијанте из [52] и [133] је једноставно. Довољно је користити само околине добијене на основу операције *замене*. У овом случају, фаза размрдавања не производи недопустива решења, стога процедура похлепног поправљања решења након размрдавања није потребна. Приликом прилагођавања VNS-SO методе, процедура похлепне конструкције решења се мора позвати након повећања буџета како би се обезбедило да простор решења са повећаним бројем лоцираних објеката буде претражен. Остали елементи прилагођених VNS имплементација и вредности параметара остају исти као у оригиналним.

Прилагођене VNS методе су тестиране на MCLP скупу података генерисаних у раду [52] на истом рачунару као оригиналне VNS имплементације предложене за решавање варијанте MCLP са ограниченим буџетом и преференцијама корисника. Табели 2.11 представља сажети приказ резултата добијених помоћу две најбоље познате методе из литературе, GRASP-TS [52] и RVNS-TS-VNS [133] и модификације два предложена алгоритма са најбољим перформансама, VNS и SVNS. Прилагођена VNS-SO метода за MCLP варијанту из [52] показала се мање стабилном у односу на прилагођене VNS и SNVS методе и захтевала је значајно дуже време извршавања. Из тог разлога, резултати прилагођеног VNS-SO-а нису укључени у Табелу 2.11.

Све метахеуристике чији су резултати приказани у Табели 2.11, су имплементирани у C++ програмском језику, али тестиране на различитим рачунарима. Да би се обезбедило коректно поређење, укупна времена извршавања

GRASP-TS-a и RVNS-TS-VNS-a су скалирана и приказана са звездом, тј. $tot.t^*$. Фактор скалирања је израчунат коришћењем једнонитног *PassMark* резултата из базе *www.cpubenchmark.com*. GRASP-TS је тестиран на Intel Xeon E3-1220 @ 3.10 GHz процесору, RVNS-TS-VNS је покренут на AMD A6-3670 APU @ 2.7 GHz процесору, док су модификоване VNS методе извршене на Intel Xeon E5-2640 @ 2.5 GHz процесору. Према подацима из *www.cpubenchmark.com*, једнонитне вредности *PassMark* резултата за ова три рачунара су 1526, 1123 и 1348, респективно. Стога се скалирано време за GRASP-TS израчунава као $tot.t^* = tot.t \times 1526/1348$, док се скалирано време за RVNS-TS-VNS добија као $tot.t^* = tot.t \times 1123/1348$.

Резултати представљени у Табели 2.11 показују да све четири методе дају исте најбоље познате вредности функције циља на свим инстанцама. RVNS-TS-VNS и модификовани SVNS су показали бољу стабилност у поређењу са друге две методе. За све инстанце, RVNS-TS-VNS и модификовани SVNS су дали најбоље познате вредности функције циља у сваком од 5 покретања. Такође се може приметити да су модификације VNS и SVNS захтевале значајно краће време извршавања у поређењу са GRASP-TS и RVNS-TS-VNS методама из литературе. Стога се може закључити да се методе VNS и SVNS предложене у овој дисертацији могу успешно применити (уз мање неопходне измене) и на MCLP варијанту из [52], [133].

Табела 2.11: Сажети резултати добијени GRASP-TS [52], RVNS-TS-VNS [133] и прилагођеним VNS и SVNS методама на MCLP инстанцама са фиксним бројем објеката

Inst. size	GRASP-TS					RVNS-TS-VNS					VNS					SVNS				
	avg best	avg (s)	avg tot.t (s)	avg tot.t.* (s)	avg mean worst gap (%)	avg best	avg (s)	avg tot.t (s)	avg tot.t.* (s)	avg mean worst gap (%)	avg best	avg (s)	avg tot.t (s)	avg tot.t.* (s)	avg mean worst gap (%)	avg best	avg (s)	avg tot.t (s)	avg tot.t.* (s)	avg mean worst gap (%)
225x25	20061.1	0.00	0.00	0.00	0.000	20061.1	1.70	1.42	0.000	0.000	20061.1	0.32	0.03	0.000	0.000	20061.1	0.30	0.02	0.000	0.000
450x50	75463.0	1.40	1.58	0.000	0.000	75463.0	4.40	3.67	0.000	0.000	75463.0	0.69	0.05	0.000	0.000	75463.0	1.07	0.06	0.000	0.000
675x75	185876.0	7.60	8.60	0.006	0.031	185876.0	12.80	10.66	0.000	0.000	185876.0	1.36	0.14	0.006	0.031	185876.0	2.39	0.16	0.000	0.000
900x100	315939.9	13.90	15.74	0.000	0.000	315939.9	25.80	21.49	0.000	0.000	315939.9	2.43	0.28	0.000	0.000	315939.9	5.03	0.26	0.000	0.000
1350x150	706653.0	58.80	66.56	0.000	0.000	706653.0	72.80	60.65	0.000	0.000	706653.0	5.40	0.72	0.001	0.002	706653.0	11.54	1.06	0.000	0.000
1800x200	1295728.3	172.80	195.620	0.001	0.001	1295728.3	168.30	140.210	0.000	0.000	1295728.3	9.59	1.31	0.001	0.003	1295728.3	21.24	2.23	0.000	0.000

Глава 3

Двокритеријумски проблем максималног покривања локација са преференцијама корисника

3.1 Опис проблема и преглед релевантне литературе

У овом поглављу разматрана је варијанта дво критеријумског MCLP-а са преференцијама корисника, која је први пут представљена у раду [135]. Улазни подаци за ову варијанту MCLP-а су следећи: скуп потенцијалних локација објеката, скуп корисника са додељеним потражњама и преференције корисника према локацијама објеката. У питању је хомогена варијанта MCLP-а неограничених капацитета, што значи да сви објекти које треба отворити пружају исту врсту услуге и да објекти немају ограничења у погледу капацитета. Претпоставља се да треба успоставити тачно p објеката и да ће сваки корисник бити додељен успостављеном објекту који преферира. Како би се постигла равнотежа између ефикасности успостављене мреже и правичности у пружању услуге корисницима, истовремено се посматрају два циља. Први циљ је максимизација тежинске суме покривене потражње, при чему су тежине једнаке или директно пропорционалне вредностима преференција корисника. Други циљ је минимизација броја непокривених корисника. Описана варијанта дво критеријумског MCLP решавана је у раду [136] у којем су предложене три ефикасне методе вишекритеријумске оптимизације засноване на методи променљивих околина.

Описана MCLP варијанта пре публикавања рада [135] није истраживана у литератури, иако постоје разне области потенцијалне примене. Једна од њих је оптимизација сервисних мрежа различитог типа, у ситуацијама где је неопходно да се максимизује количина покривене потражње и да минимизује број корисника који остају неуслужени, узимајући у обзир преференције корисника. На пример, у системима испоруке неопходно је максимизовати укупну покривену потражњу уз обраћање пажње на то да корисници са већом потражњом буду покривени објектима које они преферирају. У исто време, неопходно је смањити број корисника који нису покривени објектима, што је више могуће.

Још једна примена посматраног проблема јесте у дизајну телекомуникационих мрежа, на пример, при планирању постављања антена за бежичну комуникацију. Потражња се представља количином протока података додељених корисницима, док су преференције пропорционалне квалитету сигнала које корисник добија ако је додељен лоцираној антени (објекту). У реалним ситуацијама, квалитет сигнала не зависи само од растојања између корисника и антене, већ се може мењати уколико постоје препреке између два чвора. Како постоји ограничени број антена које треба поставити, ако се лоцирају на местима погодним по кориснике са великим потражњама, велики број корисника са мањим протоком података може остати непокривен. Са друге стране, ако се посматра искључиво минимизација броја непокривених корисника, важни корисници са великим протоцима података могу остати непокривени или да имају лош квалитет сигнала.

У литератури постоји више радова који се баве проширивањем проблема MCLP у погледу увођења више функција циља. Најсличнија варијанта разматраној варијанти у овој дисертацији предложена је у раду [170] и има две функције циља: тежинска сума покривене потражње и укупна цена отварања објеката. Двокритеријумски MCLP, који максимизује покривену потражњу и минимизује тежинску суму растојања између сваког непокривеног корисника и њему најближег услужног центра, проучаван је у раду [36]. Badri и сар. [12] су формулисали проблем лоцирања ватрогасних станица као вишекритеријумски модел покривања локација. Примарни циљ је максимизовати услугу најзахтевнијим подручјима, али су у модел укључени и други циљеви који се односе на време транспорта, удаљеност и различите техничке захтеве.

Варијанта вишекритеријумског MCLP-а која је предложена у раду [8], има за циљ да исторемено максимизује примарну и секундарну (резервну) покри-

веност корисника, као и да минимизује суму растојања између непокривених корисника и њихових преферираних објеката. Karasakal и Silva [98] уводе дво-критеријумски MCLP који подразумева парцијално покривање непокривених корисника. Ако за корисника постоји бар један отворени центар на растојању које није веће од неке задате константе (полупречника покривања), тај корисник се сматра у потпуности покривеним. Са даљим порастом удаљености до најближег отвореног центра, степен покривености опада према некој неоппадајућој функцији (најчешће линеарној) и корисник се сматра парцијално (делимично покривеним). Уколико је растојање корисника до њему најближег центра веће од неке задате константе, корисник је непокривен. Модел укључује два циља: максимизација потпуног и парцијалног покривања и минимизација максималне дистанце између непокривених корисника и њихових најближих објеката. Дво-критеријумски MCLP модел из [98] може да се користи за дизајнирање мрежа брзе испоруке где нису сви чворови потпуно покривени објектима. У раду [139] је изучавана двокритеријумска варијанта динамичког MCLP.

Chanta и сар. [31] су проучавали проблем локација станица хитне медицинске помоћи (енгл. emergency medical services, EMS) са циљем да нађу баланс између ефикасности услуге и правичности за кориснике. Као резултат формулисана су три модела двокритеријумског покривања локација [31]. У сва три модела примарни циљ је максимизација покривене потражње, док је секундарни циљ остваривање правичности у руралним и урбаним подручјима. Секундарни циљеви предложени у моделу су: минимизација максималне дистанце између непокривене области потражње и њене најближе станице, минимизација броја непокривених руралних области потражње и минимизација броја непокривених области потражње. Проблем постављања станица за пуњење електричних возила је формулисан као вишефазни вишекритеријумски MCLP у раду [167]. Из економских разлога, аутори моделирају постављање инфраструктуре за пуњење у више фаза: прве фазе подстичу транзицију путника, док економски фактори доминирају у каснијим фазама. Варијанта двокритеријумског MCLP која такође укључује преференције корисника, а најсличнија је варијанти представљеној у овој дисертацији, разматрана је у раду [11] објављеном годину дана после рада [135]. Први циљ варијанте из рада [11] је максимизација укупне покривене потражње корисника, а други циљ максимизација укупног збира преференција покривених корисника. Међутим, модел из рада [11] не укључује правичност као циљ, тј. не разматра се број непокривених

корисника. Поред тога, методологија истраживања из рада је другачија, уместо поређења скупова решења, пореде се само изабрана појединачна решења, те резултати из рада [11] и [135] нису директно упоредиви.

МОЕА метахеуристике су успешно примењене на различитим дискретним вишекритеријумским локацијским проблемима, укључујући неке варијанте вишекритеријумског MCLP-а. На пример, Karasakal и Silva [98] су дизајнирали вишекритеријумски еволутивни алгоритам, означен као модификовани SPEA-II (mSPEA-II), за решавање двокритеријумског MCLP-а предложеног у истом истраживању. У раду [167] аутори истражују перформансе генетских алгоритама са две различите конструктивне стратегије (инкременталне и декременталне), при решавању вишефазних вишекритеријумских MCLP-а за постављање станица за пуњење електричних возила.

Пратећи идеју примена МОЕА приступа за друге вишекритеријумске локацијске проблеме, у раду [135], прилагођена су три МОЕА алгоритама из литературе за двокритеријумски MCLP са преференцијама корисника. Прецизније, прилагођени су: NSGA-II [51], SPEA-II [187] и ϵ -МОЕА [50]. Експерименти су извршени на четири групе MCLP инстанци из праксе и резултати су поређени на основу три стандардна индикатора перформанси. Сва три МОЕА приступа показала добре перформансе, међу њима није постојала значајна разлика.

МОЕА алгоритми коришћени у [135] су популацијске метахеуристике, прилагођене разматраном двокритеријумском MCLP-у са преференцијама корисника. У оквиру ове дисертације, испитује се другачији приступ, где се користи метахеуристика заснована на побољшању једног решења и примени локалне претраге. Полазећи од општег принципа примене VNS методе за вишекритеријумску оптимизацију из рада [59], дизајниран је вишекритеријумски VNS за решавање двокритеријумске варијанте MCLP-а из [135]. Пратећи концепт основног, редукованог и општег VNS-а, имплементирани су три варијанте вишекритеријумског VNS-а: вишекритеријумски основни VNS (енгл. multi-objective basic VNS, MO-BVNS), вишекритеријумски редуковани VNS (енгл. multi-objective reduced VNS, MO-RVNS) и вишекритеријумски уопштени VNS (енгл. multi-objective generalized VNS, MO-GVNS).

3.2 Математичка формулација проблема

Нека је J скуп корисника, док је I скуп потенцијалних локација за отварање нових услужних центара. Корисник може бити покривен отвореним објектом, ако њихова удаљеност није већа од унапред одређене вредности, која је означена као полупречник покривања. За сваког корисника $j \in J$, скуп локација центра које покривају овог корисника означен је са $I(j)$. Са друге стране, $J(i)$ означава скуп корисника које је могуће покрити успостављањем центра на локацији $i \in I$. Сваки корисник $j \in J$ има одређену потражњу $D_j > 0$, док вредности $g_{ij} \in (0, 1]$ означавају преференцију корисника $j \in J$ да га опслужује центар $i \in I$. Веће вредности преференције означавају већи афинитет корисника према објекту. Прецизније, $g_{i_1j} > g_{i_2j}$ значи да би корисник j радије био опслужен од стране услужног центра i_1 него од центра i_2 . Број нових услужних центара које треба отворити је фиксиран на p .

Математички модел разматраног двокритеријумског MCLP-а са преференцијама корисника користи два скупа бинарних променљивих одлучивања. Променљиве y_i узима вредност 1 ако је центар успостављен на локацији $i \in I$, а 0 у супротном. Променљива x_{ij} је једнака 1 ако је корисник $j \in J$ покривен од стране центра $i \in I$, а 0 у супротном.

Користећи наведену нотацију, двокритеријумски MCLP са преференцијама корисника [135] се може бити записан као следећи целобројни линеарни програм:

$$\max \quad \sum_{i \in I} \sum_{j \in J(i)} g_{ij} D_j x_{ij} \quad (3.1)$$

$$\min \quad |J| - \sum_{j \in J} \sum_{i \in I(j)} x_{ij} \quad (3.2)$$

при условима

$$\sum_{i \in I} y_i = p, \quad (3.3)$$

$$\sum_{i \in I(j)} x_{ij} \leq 1 \quad \forall j \in J, \quad (3.4)$$

$$x_{ij} \leq y_i \quad \forall i \in I, j \in J(i), \quad (3.5)$$

$$\sum_{s=k+1}^{|I(j)|} x_{i_s, j} + y_{i_k} \leq 1 \quad \forall j \in J, k \in 1, \dots, |I(j)| - 1, \quad (3.6)$$

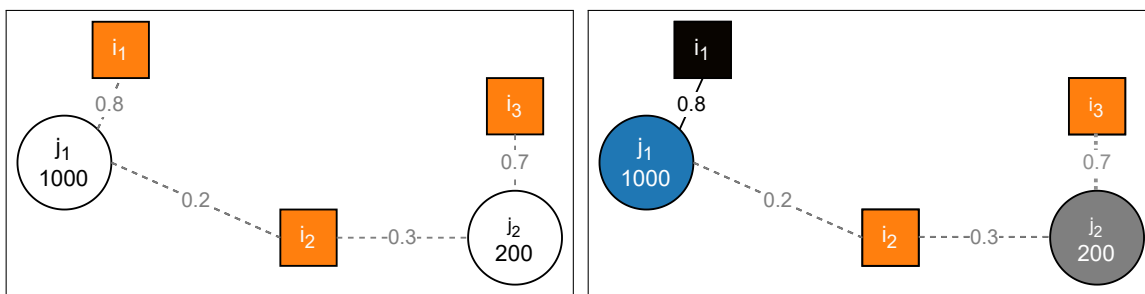
$$y_i \in \{0, 1\} \quad \forall i \in I, \quad (3.7)$$

$$x_{ij} \in \{0, 1\} \quad \forall i \in I, j \in J. \quad (3.8)$$

Прва функција циља (3.1) је укупна тежинска суму потражње покривених корисника, односно оних који су додељени отвореном центру и циљ је њена максимизација. Друга функција циља (3.2) представља број непокривених корисника, а циљ је њена минимизација. Тежине укључене у функцију циља (3.1) су једнаке преференцијама корисника ка њиховом додељеном центру. На овај начин постиже се да већа вредност преференције корисника $j \in J$ према објекту $i \in I$ имплицира да ће утицај покривене потражње D_j у укупној суми бити већи. Број услужних центара које треба отворити је постављен на p , што је обезбеђено ограничењем (3.3). Ограничење (3.4) означава да је сваки корисник покривен са највише једним центром, док ограничење (3.5) обезбеђује да сваки корисник може бити опслужен само од једног отвореног центра. Ограничење (3.6) обезбеђује да је сваки корисник додељен свом најпожељнијем отвореном центру. Променљиве y_i и x_{ij} су дефинисане као бинарне ограничењима (3.7) и (3.8). Разматрани двокритеријумски MCLP је НП-тежак, као генерализација класичног једнокритеријумског MCLP за који је доказано да је НП-тежак у раду [125].

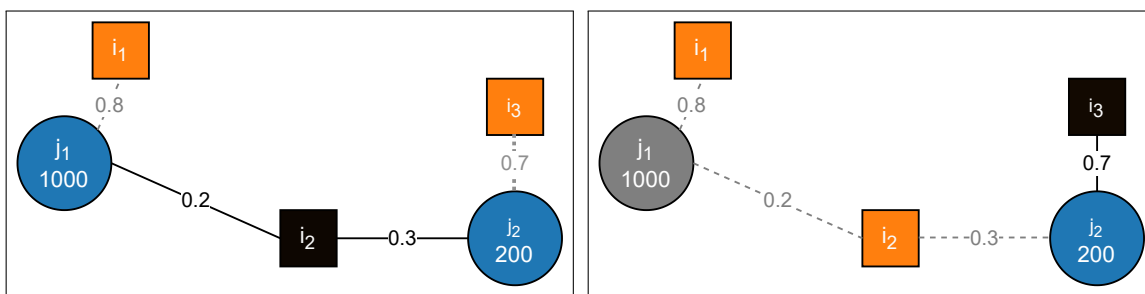
У циљу илустрације ефекта разматраних функција циља и преференција корисника, на слици 3.1 приказана су различита решења добијена на једној инстанци малих димензија. На слици 3.1(a) представљена је разматрана инстанца са три потенцијалне локације (означене као квадрати) и два корисника (означени круговима у којима уписане вредности означавају потражњу корисника). Вредности испрекиданих линија означавају преференције корисника према објектима. У Парето оптималном решењу приказаном на слици 3.1(b), центар i_1 је отворен да би се покрила велика потражња корисника j_1 који пре-

3.2. МАТЕМАТИЧКА ФОРМУЛАЦИЈА ПРОБЛЕМА



(a) Инстанца малих димензија са $|I| = 3$ и $|J| = 2$.

(b) Парето оптимално решење. Центар је постављен на i_1 . Вредности функција циља су: $F_1 = 800, F_2 = 1$.



(c) Парето оптимално решење. Центар је постављен на i_2 . Вредности функција циља су: $F_1 = 260, F_2 = 0$.

(d) Решење није Парето оптимално. Центар је постављен на i_3 . Вредности функција циља су: $F_1 = 140, F_2 = 1$.

Слика 3.1: Инстанца малих димензија (a). Решења (b), (c), (d) приказују утицај избора локације објекта и преференција корисника на вредност функција циља F_1 и F_2 .

ферира центар i_1 (преференција j_1 ка i_1 износи 0.8). Приметимо да корисник j_2 са пет пута мањом потражњом у односу на j_1 остаје непокривен. У другом Парето оптималном решењу приказаном на слици 3.1(c), отворени центар на i_2 покрива оба корисника j_1 и j_2 . На крају, слика 3.1(d) приказује решење које није Парето оптимално, при чему је центар успостављен на чвору i_3 . Треба приметити да центар i_3 покрива само корисника j_2 , који има високу преференцију према отвореном центру, али и ниску потражњу у исто време. Са друге стране, корисник j_1 са много већом потражњом остаје непокривен.

У сврху илустрације ефекта коришћења две посматране функције циља, на Слици 3.2 су приказана три решења добијена на једној реалној инстанци са великим бројем чворова. На слици 3.2(a) приказана је инстанца добијена на

основу демографских података из градске области Бронкс у Њу Јорку, преузета из рада [123]. Инстанца садржи скуп од 3839 корисничких чворова (означени као плави кругови) од којих је 10% изабрано за потенцијалне локације објеката (означене као наранџасти кругови). Транспарентност боје која репрезентује кориснички чвор је пропорционална његовој потражњи. Слика 3.2(c) приказује екстремно решење у којој је доносилац одлуке само заинтересован да максимизује тежинску суму потражње (функција циља F_1), те последично значајан број корисника остаје непокривен. Успостављени објекти су означени са црним квадратима, покривени корисници са плавим круговима, а непокривени са црвеним крстовима. Ако је само циљ минимизација броја непокривених корисника (функција циља F_2), добија се још једно екстремно решење приказано на Слици 3.2(d) са објектима који су сада распоређени равномерније преко целе мапе и релативно малим бројем непокривених корисника. Ако доносилац одлука тежи да пронађе баланс између тежинске покривене потражње и броја непокривених корисника, добија се Парето оптимално решење приказано на Слици 3.2(b).

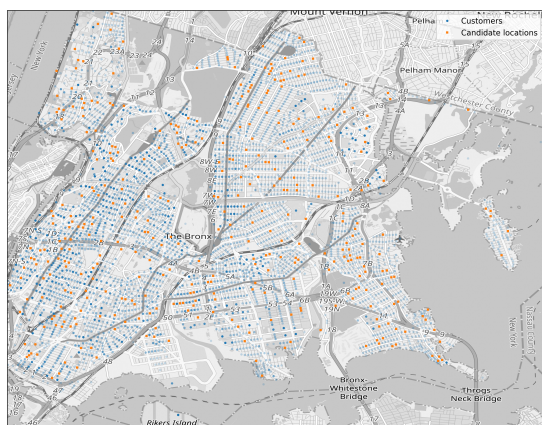
3.3 Вишекритеријумска метода променљивих околина

Представљање решења

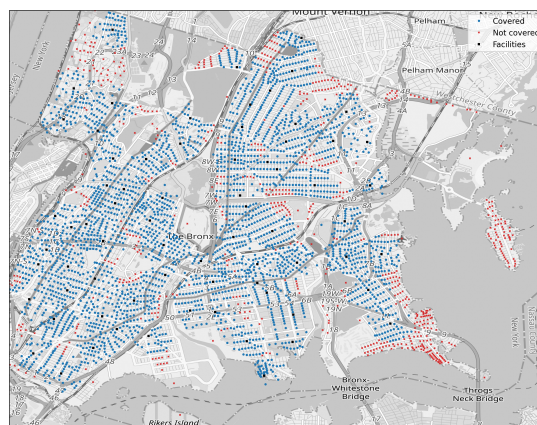
У оквиру овог одељка описане су три варијанте MO-VNS методе (означене са MO-RVNS, MO-BVNS и MO-GVNS) за разматрани двокритеријумски MCLP са преференцијама корисника. Све три варијанте MO-VNS методе за овај проблем су изложене у раду [136], заједно са експерименталним резултатима. У све три предложене варијанте MO-VNS методе, решење проблема је представљено бинарним низом дужине $|I|$ где сваки бит одговара потенцијалној локацији објекта. Ако је бит на i -тој позицији у низу једнак 1, то значи да је објекат $i \in I$ отворен и вредност променљиве одлучивања y_i постављена на 1. Иначе, објекат на локацији i је затворен и $y_i = 0$. Како тачно p објеката треба отворити, решење се сматра допустивим ако тачно p битова у низу има вредност 1. На пример, ако је $p = 3$, а $|I| = 10$, једно од допустивих решења може бити $s = [0100110000]$

Околина се заснивају на операцији замене (енгл. swap), односно затварању

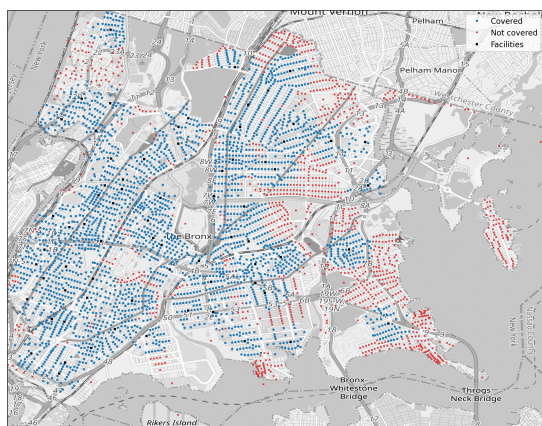
3.3. ВИШЕКРИТЕРИЈУМСКА МЕТОДА ПРОМЕНЉИВИХ ОКОЛИНА



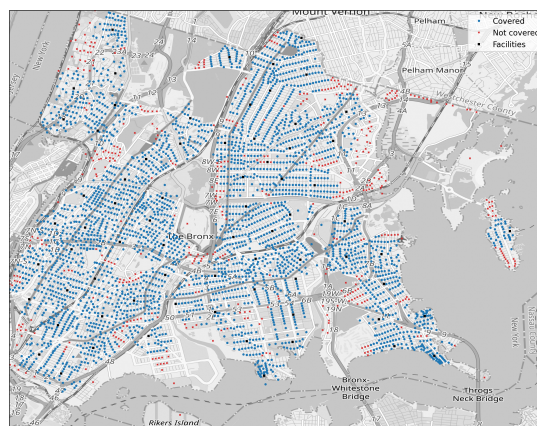
(a) Графички приказ Бронкс инстанце.



(b) Пример Парето оптималног решења.



(c) Екстремно решење где је прва функција циља приоритет.



(d) Екстремно решење где је друга функција циља приоритет.

Слика 3.2: Бронкс инстанца и три различита решења. Мапа Бронкса је преузета са www.openstreetmap.org.

једног објекта на једној локацији и отварању на другој. Околина реда 1 се добија применом једне операције замене, док се применом 2 операције замене добија околина реда 2, итд. На пример, ако на решење s применимо операцију замене елемената на позицијама 1 и 9 добијамо ново допустиво решење $s' = [0000110010]$

Вишекритеријумски основни VNS

Алгоритам 18 Вишекритеријумски основни VNS (МО-BVNS)

```
1: Input:  $k_{max}, t_{max}$ 
2: Output: set of non-dominated solutions  $S$ 
3:  $S \leftarrow$  generate random set of non-dominated solutions
4: while  $t_{max}$  is not reached do
5:    $k \leftarrow 1$ 
6:   while  $k \leq k_{max}$  do
7:      $N \leftarrow S$ 
8:     for each solution  $x \in S$  do
9:        $x' \leftarrow \text{SHAKE}(x, k, N)$ 
10:       $\text{LOCALSEARCH}_1(x', N)$ 
11:       $\text{LOCALSEARCH}_2(x', N)$ 
12:      if  $N \setminus S \neq \emptyset$  then
13:         $S \leftarrow \text{CLEAN}(N)$ 
14:         $k \leftarrow 1$ 
15:      else
16:         $k \leftarrow k + 1$ 
```

Структура МО-BVNS заснована је на корацима BVNS-а за једнокритеријумску оптимизацију и приказана је Алгоритмом 18. Представљени МО-BVNS почиње са насумично генерисаним скупом решења S , у којем су доминирана решења уклоњена, те алгоритам ради само са скупом недоминираних решења. Овај скуп се назива апроксимативним скупом, који током извршавања алгоритма треба да се приближи што више правом Парето оптималном скупу.

Током МО-BVNS итерација извршавају се две фазе, размрдавање и локална претрага, заједно са кораком промене околине. На крају сваке итерације, алгоритам проверава да ли је дошло до побољшања, тј. да ли је додат нови недоминирани елемент у тренутни скуп N . Ако је решење побољшано, алгоритам започиње претрагу од прве околине, у супротном наставља претрагу од следеће околине из датог скупа. Ако је побољшање постигнуто, алгоритам

ажурира тренутни апроксимативни скуп S , тј. уклања решења која су сада доминирана извршавањем CLEAN процедуре. MO-BVNS итерације се извршавају све док се не испуни критеријум заустављања, у овом случају док се не достигне максимално време извршавања. У сваком кораку, над сваким решењем у апроксимативном скупу се изводи размрдавање (имплементирано као SNAKE процедура). Решење измењено размрдавањем се прослеђује у две процедуре локалне претраге, по једна за сваку функцију циља (LOCALSEARCH₁ и LOCALSEARCH₂).

Фаза размрдавања за свако решење из апроксимативног скупа N је иста као у случају једнокритеријумског BVNS-а. За свако решење апроксимативног скупа x , мења се k насумичних локација (употребом операције swap) и на тај начин се генерише ново решење из k -те околине тренутног решења x' . Фаза размрдавања је имплементирана као SNAKE процедура и приказана је Алгоритмом 19. За разлику од једнокритеријумске варијанте, за свако решење x' добијено овом процедуром проверава се да ли је недоминирано, тј. да ли га треба укључити у тренутни апроксимативни скуп N (CHECKNONDOMINANCE процедура).

Алгоритам 19 Процедура размрдавања (SNAKE)

- 1: **Input:** solution x , neighborhood size k , set of solutions N
 - 2: **Output:** perturbed solution x' , updated set of solutions N
 - 3: $x' \leftarrow x$
 - 4: **for** $i \leftarrow 1$ to k **do**
 - 5: $x' \leftarrow \text{RANDOMSWAP}(x')$
 - 6: **if** CHECKNONDOMINANCE(N, x') **then**
 - 7: $N \leftarrow N \cup \{x'\}$
-

Фаза локалне претраге покушава да побољша апроксимативни скуп N односу на једну од функција циља. Док истражује околину тренутног решења x , процедура локалне претраге изводи проверу недоминатности решења (CHECKNONDOMINANCE процедура) после сваке операције замене. Решење из околине x' које највише побољшава разматрану функцију циља се бира као ново тренутно решење. Процедура се завршава уколико се не побољшава вредност разматране функције циља (CHECKOBJECTIVE процедуре). Структура локалне претраге је описана Алгоритмом 20.

Алгоритам 20 Локална претрага

```

1: Input: solution  $x$ , set of solutions  $N$ 
2: Output: updated set of solutions  $N$ , set of new non-dominated solutions  $N'$ 
3:  $improvement \leftarrow true$ 
4:  $best \leftarrow x$ 
5:  $N' \leftarrow \emptyset$ 
6: while  $improvement$  do
7:    $improvement \leftarrow false$ 
8:   while there is new solution  $x'$  in neighborhood of  $x$  do
9:      $x' \leftarrow \text{SWAP}(x)$ 
10:    if  $\text{CHECKNONDOMINANCE}(N, x')$  then
11:       $N \leftarrow N \cup x'$ 
12:       $N' \leftarrow N' \cup x'$ 
13:    if  $\text{CHECKOBJECTIVE}(best, x')$  then
14:       $best \leftarrow x'$ 
15:       $improvement \leftarrow true$ 
16:    $x \leftarrow best$ 

```

Вишекритеријумски редуковани VNS

У циљу испитивања утицаја фазе размдравања на квалитет решења, имплементирана је и редукована варијанта алгоритма означена са MO-RVNS. Циљ је појачати експлорацију простора претраге што је више могуће, повећавајући број итерација и позива SHAKE процедуре. Структура MO-RVNS варијанте је приказана Алгоритмом 21. Може се приметити да се MO-RVNS добија од MO-BVNS методе уклањањем процедура LOCALSEARCH₁ и LOCALSEARCH₂ (линије 8 и 9 Алгоритма 18).

Вишекритеријумски уопштени VNS

На основу идеје општег VNS-а за једнокритеријумску оптимизацију, имплементирана је још једна вишекритеријумска VNS варијанта за разматрани проблем, вишекритеријумска општа метода променљивих околина (MO-GVNS), која је приказана Алгоритмом 22. Разлика између MO-BVNS и MO-GVNS метода је у фази локалне претраге, прецизније, у начину на који се LOCALSEARCH₁ и LOCALSEARCH₂ процедуре извршавају. Треба напоменути да у оригиналном раду [59], где је представљен концепт за вишекритеријумски VNS, са MO-GVNS је означена варијанта која користи процедуре променљивог спуста (VND) по појединачним функцијама циља уместо локалних претрага. Додатно, предста-

Алгоритам 21 Вишекритеријумски редуковани VNS (МО-RVNS)

```

1: Input:  $k_{max}, t_{max}$ 
2: Output: set of non-dominated solutions  $S$ 
3:  $S \leftarrow$  generate random set of non-dominated solutions
4: while  $t_{max}$  is not reached do
5:    $k \leftarrow 1$ 
6:   while  $k \leq k_{max}$  do
7:      $N \leftarrow S$ 
8:     for each solution  $x \in S$  do
9:       SHAKE( $x, k, N$ )
10:    if  $N \setminus S \neq \emptyset$  then
11:       $S \leftarrow$  CLEAN( $N$ )
12:       $k \leftarrow 1$ 
13:    else
14:       $k \leftarrow k + 1$ 

```

вљена варијанта МО-BVNS није разматрана у оригиналном раду [59].

МО-GVNS користи два додатна скупа N_1 и N_2 за чување недоминираних решења добијених позивањем LOCALSEARCH₁ и LOCALSEARCH₂. Иницијално, N_1 и N_2 садрже само решења из фазе размрадавања. Две процедуре локалне претраге се наизменично извршавају све док скупови N_1 и N_2 не постану празни, што значи да ни једно ново недоминирано решење није пронађено. Следеће недоминирано решење које ће бити прослеђено процедурама LOCALSEARCH₁ и LOCALSEARCH₂ се бира процедуром SELECT. Ова процедура насумично бира једно решење из скупа недоминираних решења, шаље га процедурама локалне претраге и након тога уклања решење из скупа.

Алгоритам 22 Вишекритеријумски општи VNS (MO-GVNS)

```
1: Input:  $k_{max}, t_{max}$ 
2: Output: set of non-dominated solutions  $S$ 
3:  $S \leftarrow$  generate random set of non-dominated solutions
4: while  $t_{max}$  is not reached do
5:    $k \leftarrow 1$ 
6:   while  $k \leq k_{max}$  do
7:      $N \leftarrow S$ 
8:     for each solution  $x \in S$  do
9:        $x' \leftarrow \text{SHAKE}(x, k, N)$ 
10:       $N_1 \leftarrow \{x'\}$ 
11:       $N_2 \leftarrow \{x'\}$ 
12:      while  $N_1 \cup N_2 \neq \emptyset$  do
13:         $N_1 \leftarrow N_1 \cup \text{LOCALSEARCH}_1(\text{SELECT}(N_2), N)$ 
14:         $N_2 \leftarrow N_2 \cup \text{LOCALSEARCH}_2(\text{SELECT}(N_1), N)$ 
15:      if  $N \setminus S \neq \emptyset$  then
16:         $S \leftarrow \text{CLEAN}(N)$ 
17:         $k \leftarrow 1$ 
18:      else
19:         $k \leftarrow k + 1$ 
```

3.4 Експериментални резултати

Три предложена MO-VNS алгоритма имплементирана су у програмском језику Java. У експериментима се такође користе вишекритеријумски еволутивни приступи (NSGA-II, SPEA-II и ε -MOEA) приказани у раду [135] који су коришћени у сврху поређења са предложеним MO-VNS варијантама. Ове три MOEA имплементације су преузете из *Java* оквира за вишекритеријумску оптимизацију [80] и прилагођене су за посматрани проблем. У постојећој литератури, NSGA-II и SPEA-II су постали стандардни приступи у решавању вишекритеријумских оптимизационих проблема, стога је постало уобичајено да се перформансе неког новог вишекритеријумског приступа упоређују са ове две MOEA методе.

У наставку је дат сажети опис елемената MOEA метода из [135] коришћених за решавање двокритеријумског MCLP са преференцијама корисника:

- Иницијална популација се у сваком од три MOEA алгоритма генерише насумично.

- Након генерисања иницијалне популације јединки, сваки од три МОЕА алгоритма понављају следеће кораке: селекција, укрштање, мутација, евалуација потомства и унапређење популације. Ови кораци се понављају док се не испуни услов заустављања након чега се недоминирана популација враћа као коначни резултат.
- У фази селекције користи се бинарни турнир са правилима за имплементацију ограничења.
- За креирање нове генерације користи се једнопозиционо укрштање, док се за мутацију користи инвертовање битова.
- Политика замене генерација зависи од датог МОЕА алгоритма.

Све разматране методе (три предложене MO-VNS методе и три МОЕА методе коришћене у [135]) су извршене на истим тест инстанцама и на истом рачунару са *Intel i3-4170 3.70GHz CPU* и *8GB RAM*-а.

Инстанце

За потребе експеримената, коришћене су тест инстанце из [135], које су добијене модификацијом реалних инстанци великих димензија приказаних у раду [123] за једнокритеријумски MCLP. Инстанце из [123] су такође модификоване и за тестирање метода за једнокритеријумски MCLP са преференцијама корисника и ограниченим буџетом који је представљен у претходном поглављу.

Рад [135] се бави вишекритеријумским MCLP-ом и разматра реалније претпоставке у поређењу са [123]. У реалним ситуацијама, број потенцијалних локација објеката је много мањи у поређењу са бројем локација корисника [45]. У свакој групи инстанци, 10% корисничких чворова се насумично бира за потенцијалне локације објеката, што даје $|I| = 271, 383, 513$ и 773 за Њујорк, Бронкс, Сан Франциско и Кингс, респективно. Полупречник покривања је исти за инстанце које припадају истој групи $R = 400, 600, 600, 800$, док је број објеката које је потребно успоставити $p = 70$ за све групе. Пратећи идеје из радова [26] и [52], у раду [135] се користи троугаона расподела за израчунавање преференција корисника према потенцијалним објектима. Овако генерисане преференције се на крају скалирају у интервал $(0, 1]$. Када се генеришу преференције за сваког од корисника j према свакој локацији i , следеће вредности се користе у троугаоној расподели:

3.4. ЕКСПЕРИМЕНТАЛНИ РЕЗУЛТАТИ

- доња граница - минимално растојање од локације i до било ког корисника у $J(i)$,
- горња граница - максимално растојање од локације i до било ког корисника у $J(i)$,
- модус - растојање од локације i до корисника j .

За сваку од четири групе инстанци (Њујорк, Бронкс, Сан Франциско и Кингс), генерисано је по 5 тест инстанци са различитим скуповима потенцијалних локација објеката и различитим вредностима преференција корисника.

Резултати

Да би се обезбедило коректно поређење посматраних шест алгоритама (МО-BVNS, МО-RVNS, МО-GVNS, NSGA-II, SPEA-II и ε -МОЕА), максимално процесорско време је ограничено на t_{max} секунди за сваки алгоритам. Максимално време извршавања зависи од величине инстанце, то значи да t_{max} расте са порастом $|I|$ и $|J|$. У извршеним експериментима, вредност t_{max} је постављена на 200, 300, 500 и 700 секунди за Њујорк, Бронкс, Сан Франциско и Кингс групе инстанци, респективно. Параметар свих МО-VNS алгоритама k_{max} представља максималну величину околине и постављен је на 3. Код МОЕА алгоритама, величина популације је постављена на 100, вероватноћа укрштања на 1, док је вероватноћа мутације 0.01, 0.008, 0.006 и 0.004 за инстанце из Њујорк, Бронкс, Сан Франциско и Кингс групе, респективно. Вредност параметра k који користи SPEA-II алгоритам је фиксирана на 1, што значи да се густина рачуна у односу на растојање од најближег суседа. Број јединки нове генерације је постављен на 100 у оквиру SPEA-II алгоритма. Вредност параметра ε коришћеног за критеријум доминантности у оквиру ε -МОЕА алгоритма узима вредност 0.01.

Како је прави Парето фронт непознат, комбиновани су Парето фронтови из свих покретања свих шест разматраних алгоритама и коришћени као референтни скуп. Решење алгоритма добијено из једног покретања означено је као апроксимативни скуп. Да би се упоредиле перформансе имплементираних алгоритама, коришћено је пет индикатора перформанси [109]:

- **Хиперзапремина** (енгл. Hypervolume - HV) се израчунава као величина (запремина, површина) простора решења над којим доминирају решења из апроксимативног скупа.

- **Инвертовано генеративно растојање** (енгл. Inverted generational distance - IGD) се добија као просечно растојање између решења из референтног скупа и најближег решења из апроксимативног скупа.
- **Адитивни епсилон индикатор** (енгл. Additive epsilon indicator - EPS+) је најмања вредност за коју треба транслирати апроксимативни скуп како би он доминирао над референтним.
- **Допринос** (енгл. Contribution - C) је однос величине пресека референтног и апроксимативног скупа и величине референтног скупа. Ако је апроксимативни скуп једнак референтном, онда C износи 1.
- **Величина** (енгл. Size) је број решења у апроксимативном скупу.

Избор индикатора HV, IGD и EPS+ се заснива на њиховој широкој употреби у вишекритеријумској оптимизацији. На пример, извршена анализа у раду [154] указује да је HV највише коришћена метрика, а затим следе EPS и IGD. Додатно, укључена су два интуитивна и једноставна индикатора: допринос и величина апроксимативног скупа. Циљ је да се максимизују HV, C и Size, а да се минимизују IGD и EPS+.

Сваки од шест алгоритама је покренут 10 пута за сваку инстанцу из сваке од четири групе инстанци. Добијени резултати су приказани у Табели 3.1 на следећи начин. За сваки индикатор перформанси (HV, IGD, EPS+, C и Size), чувају се средња вредност (mean), минимум (min), максимум (max) и медијана (med.) вредности добијених током 10 извршавања посматраног алгорита на свакој тест инстанци. Стандардна девијација (std) добијених вредности је такође израчуната да би се оценила стабилност алгоритама. Просек вредности mean, std, med., min и max статистика за сваки индикатор перформанси је израчунат по групама инстанци. Ове просечне вредности су приказане у Табели 3.1, а најбоље mean вредности су подебљане.

3.4. ЕКСПЕРИМЕНТАЛНИ РЕЗУЛТАТИ

Табела 3.1: Поређење предложених MO-VNS варијанти и MOEA алгоритама за двокритеријумски MCLP на групама инстанци - Њујорк (NY), Бронкс (Bronx), Сан Франциско (SF) и Кингс (Kings)

Група		NSGA-II				SPEA-II					
		HV	IGD	EPS+ C	Size	HV	IGD	EPS+ C	Size		
NY	mean	0.721	0.064	0.130	0.002	87.5	0.704	0.080	0.152	0.000	73.8
	std	0.028	0.018	0.026	0.004	16.2	0.027	0.018	0.029	0.000	14.9
	min	0.676	0.039	0.091	0.000	62.4	0.658	0.051	0.105	0.000	48.4
	med.	0.727	0.061	0.127	0.000	88.9	0.706	0.081	0.155	0.000	74.6
	max	0.756	0.096	0.171	0.014	110.8	0.739	0.104	0.197	0.001	93.2
Bronx	mean	0.650	0.115	0.177	0.000	103.5	0.633	0.131	0.196	0.000	86.7
	std	0.025	0.018	0.028	0.000	22.4	0.028	0.020	0.023	0.000	18.7
	min	0.607	0.093	0.140	0.000	74.4	0.593	0.100	0.162	0.000	59.6
	med.	0.655	0.112	0.176	0.000	100.3	0.631	0.132	0.197	0.000	89.7
	max	0.680	0.147	0.227	0.000	141.6	0.681	0.161	0.231	0.000	116.2
SF	mean	0.460	0.237	0.311	0.000	63.8	0.465	0.237	0.306	0.000	59.8
	std	0.053	0.040	0.044	0.000	19.4	0.040	0.030	0.035	0.000	16.1
	min	0.381	0.176	0.250	0.000	34.2	0.409	0.189	0.255	0.000	34.4
	med.	0.464	0.231	0.312	0.000	63.9	0.468	0.234	0.302	0.000	57.1
	max	0.543	0.300	0.385	0.000	94.6	0.530	0.280	0.369	0.000	90.0
Kings	mean	0.229	0.444	0.536	0.000	40.6	0.235	0.440	0.526	0.000	39.3
	std	0.052	0.062	0.069	0.000	13.4	0.053	0.065	0.059	0.000	14.4
	min	0.130	0.369	0.447	0.000	23.2	0.147	0.361	0.455	0.000	21.0
	med.	0.236	0.434	0.527	0.000	39.3	0.242	0.433	0.518	0.000	37.2
	max	0.297	0.568	0.674	0.000	66.0	0.304	0.557	0.632	0.000	64.8

Табела 3.1: Поређење предложених MO-VNS варијанти и MOEA алгоритама за двокритеријумски MCLP (наставак)

Група		c-MOEA				MO-RVNS					
		HV	IGD	EPS+ C	Size	HV	IGD	EPS+ C	Size		
NY	mean	0.684	0.101	0.173	0.000	61.8	0.802	0.010	0.035	0.041	137.4
	std	0.030	0.022	0.028	0.000	10.0	0.009	0.004	0.019	0.060	7.6
	min	0.629	0.070	0.133	0.000	45.6	0.784	0.006	0.014	0.000	122.4
	med.	0.690	0.099	0.170	0.000	62.1	0.805	0.009	0.031	0.016	138.3
	max	0.723	0.138	0.217	0.000	78.8	0.811	0.018	0.071	0.176	148.2
Bronx	mean	0.606	0.153	0.220	0.000	71.5	0.802	0.014	0.040	0.078	215.2
	std	0.032	0.021	0.033	0.000	15.1	0.006	0.004	0.011	0.088	16.4
	min	0.552	0.122	0.175	0.000	49.2	0.792	0.009	0.024	0.001	186.0
	med.	0.611	0.150	0.217	0.000	69.8	0.803	0.014	0.040	0.059	217.5
	max	0.649	0.186	0.277	0.000	96.4	0.811	0.021	0.058	0.294	240.4
SF	mean	0.448	0.253	0.332	0.000	51.3	0.720	0.047	0.104	0.037	102.4
	std	0.048	0.035	0.043	0.000	14.0	0.023	0.015	0.026	0.059	12.8
	min	0.371	0.199	0.270	0.000	31.6	0.680	0.029	0.059	0.000	82.0
	med.	0.448	0.252	0.336	0.000	50.4	0.723	0.044	0.104	0.009	103.1
	max	0.527	0.312	0.394	0.000	76.4	0.751	0.072	0.145	0.177	120.6
Kings	mean	0.219	0.457	0.554	0.000	35.8	0.685	0.059	0.128	0.048	118.1
	std	0.058	0.071	0.080	0.000	12.0	0.029	0.018	0.024	0.110	15.7
	min	0.125	0.364	0.453	0.000	22.4	0.637	0.035	0.089	0.000	94.0
	med.	0.229	0.443	0.539	0.000	34.2	0.684	0.058	0.129	0.002	117.9
	max	0.302	0.581	0.698	0.000	61.6	0.725	0.092	0.161	0.348	144.4

3.4. ЕКСПЕРИМЕНТАЛНИ РЕЗУЛТАТИ

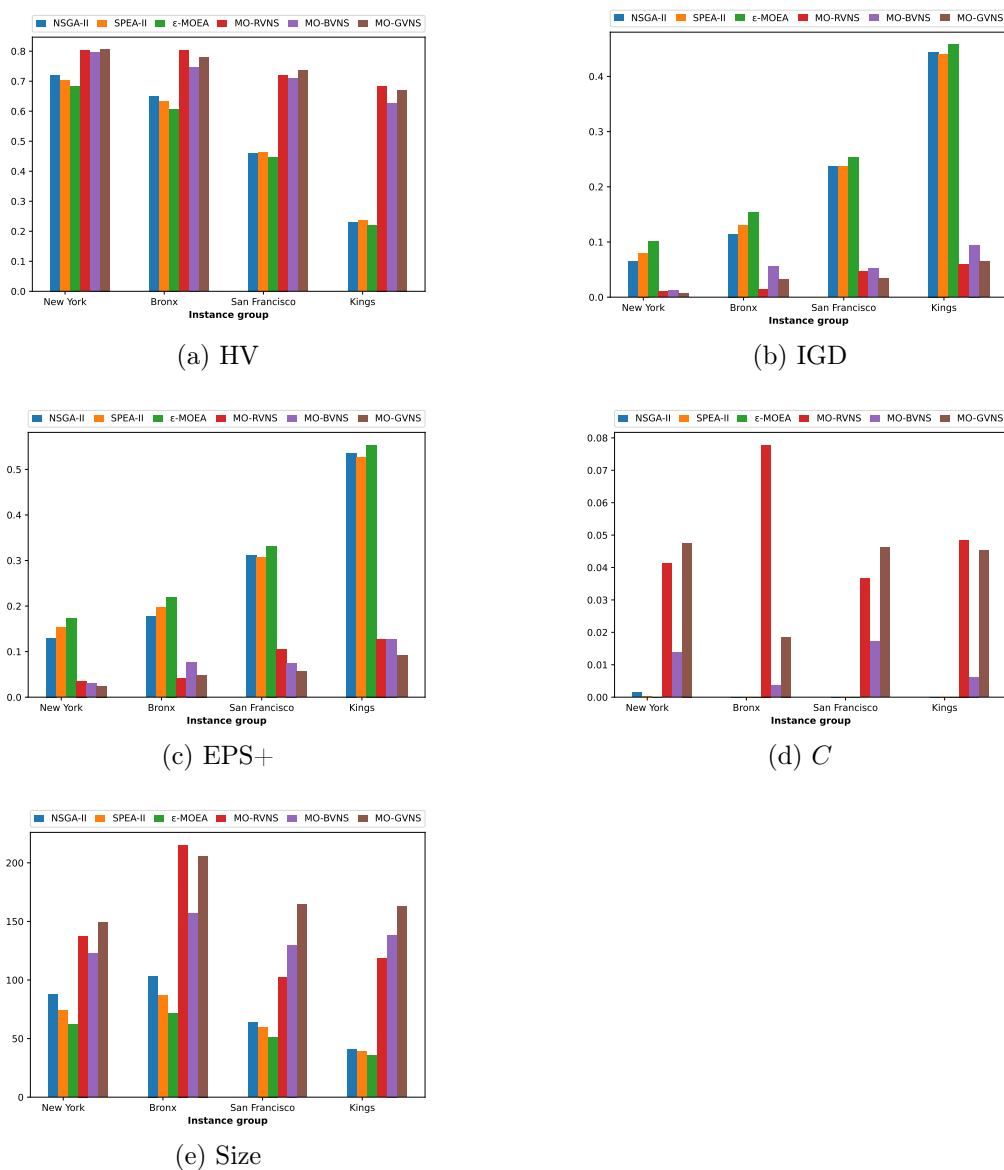
Табела 3.1: Поређење предложених MO-VNS варијанти и MOEA алгоритама за двокритеријумски MCLP (наставак)

Група		MO-BVNS				MO-GVNS					
		HV	IGD	EPS+ C	Size	HV	IGD	EPS+ C	Size		
NY	mean	0.798	0.012	0.030	0.014	122.6	0.806	0.007	0.025	0.048	148.9
	std	0.005	0.003	0.007	0.024	8.3	0.005	0.002	0.009	0.099	5.6
	min	0.790	0.008	0.020	0.000	107.0	0.798	0.004	0.012	0.000	139.6
	med.	0.798	0.012	0.028	0.001	123.4	0.806	0.007	0.024	0.002	149.1
	max	0.804	0.017	0.043	0.067	135.0	0.813	0.010	0.038	0.310	157.6
Bronx	mean	0.748	0.055	0.077	0.004	156.6	0.779	0.032	0.049	0.019	206.0
	std	0.021	0.016	0.018	0.007	16.6	0.010	0.007	0.009	0.033	18.2
	min	0.706	0.036	0.054	0.000	128.4	0.763	0.022	0.035	0.000	181.0
	med.	0.750	0.052	0.076	0.000	156.4	0.779	0.032	0.048	0.006	203.2
	max	0.774	0.089	0.114	0.023	182.0	0.792	0.044	0.061	0.103	240.2
SF	mean	0.710	0.052	0.075	0.017	129.9	0.736	0.034	0.057	0.046	164.3
	std	0.024	0.016	0.017	0.035	14.2	0.022	0.015	0.016	0.068	15.5
	min	0.671	0.027	0.048	0.000	106.0	0.698	0.015	0.030	0.000	141.2
	med.	0.708	0.053	0.076	0.000	130.5	0.734	0.035	0.059	0.014	163.0
	max	0.748	0.077	0.102	0.107	153.0	0.767	0.058	0.081	0.205	191.2
Kings	mean	0.626	0.094	0.128	0.006	138.4	0.671	0.065	0.092	0.045	162.6
	std	0.038	0.026	0.031	0.014	19.0	0.041	0.027	0.027	0.108	17.0
	min	0.554	0.060	0.086	0.000	109.6	0.611	0.025	0.048	0.000	136.8
	med.	0.633	0.089	0.124	0.000	136.3	0.675	0.064	0.093	0.002	164.0
	max	0.681	0.144	0.185	0.044	173.4	0.735	0.107	0.135	0.330	192.0

Прво су анализиране и поређене (просечне) средње вредности (mean) индикатора перформанси за сваки алгоритам из посматране групе инстанци. На основу вредности свих пет индикатора перформанси презентованих у Табели 3.1, закључује се да сва три MOEA алгоритма имају сличне перформансе. Са друге стране, три предложена MO-VNS алгоритма су показали боље резултате у односу на MOEA алгоритме на све 4 групе инстанци. Разлика у перформансама између MO-VNS имплементација и MOEA приступа се посебно уочава на две највеће групе инстанци - Сан Франциско и Кингс. Један од разлога за супериорност MO-VNS приступа у односу на MOEA алгоритме је чињеница да MO-VNS изводи ажурирања вредности две функције циља након сваке операције замене (отварање или затварање објекта), док MOEA алгоритми не користе ове операције и морају да рачунају функције циља за сва решења у новој генерацији.

Међусобним поређењем перформанси три предложене MO-VNS имплементације, може се закључити да су MO-RVNS и MO-GVNS показале боље перформансе у односу на MO-BVNS. На основу просечних средњих вредности свих пет индикатора, MO-GVNS је супериорнији на NY и SF групама инстанци у односу на MO-RVNS и MO-BVNS, док је MO-RVNS бољи на Бронкс групи инстанци у поређењу са MO-GVNS и MO-BVNS. На највећој групи инстанци (Кингс), MO-

3.4. ЕКСПЕРИМЕНТАЛНИ РЕЗУЛТАТИ



Слика 3.3: Поређење MO-VNS и MOEA метода коришћењем средњих вредности пет индикатора перформанси.

GVNS и MO-RVNS су показали сличне перформансе. Прецизније, MO-RVNS је дао боље средње вредности за три, а MO-GVNS за две од пет индикатора перформанси на Кингс групи инстанци.

На Слици 3.3 приказани су графици који дају бољи увид у перформансе предложених MO-VNS метода и три MOEA приступа из [135]. Са Сlike 3.3

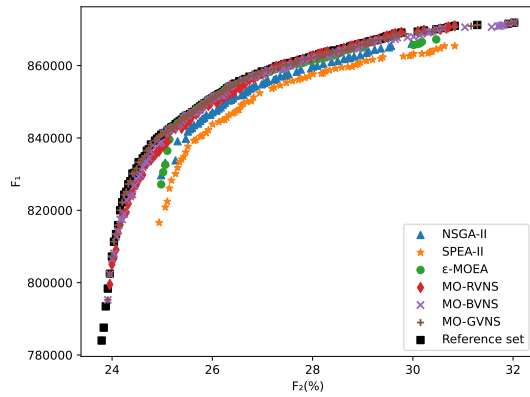
3.4. ЕКСПЕРИМЕНТАЛНИ РЕЗУЛТАТИ

јасно се може видети да MO-VNS приступ даје боље резултате у односу на MOEA приступе у погледу свих 5 индикатора перформанси. Када се међусобно пореде MO-VNS варијанте и анализирају вредности индикатора HV , IGD и C , приказани на графиконима Сликe 3.3(a), (b) и (d), респективно, закључује се да је MO-RVNS супериорнији у односу на друге две MO-VNS методе на Бронкс и Кингс групама инстанци, док MO-GVNS има боље перформансе на Њујорк и Сан Франциско групама инстанци. Према графикону на Слици 3.3(c), где је приказан $EPS+$ индикатор, рангирање је следеће: MO-GVNS, затим MO-BVNS и на крају MO-RVNS. Треба напоменути да мала одступања од овог рангирања постоје код Бронкс групе инстанци на којој MO-RVNS има најбољу вредност индикатора, затим у случају Кингс групе, где MO-BVNS и MO-RVNS имају исте вредности индикатора $EPS+$. На основу резултата на Слици 3.3(e), где су дате вредности $Size$, MO-GVNS и MO-RVNS су супериорнији за групе инстанци Њујорк и Бронкс, респективно. За две највеће групе инстанци Сан Франциско и Кингс, рангирање по индикатору $Size$ је следеће: MO-GVNS, MO-BVNS, MO-RVNS.

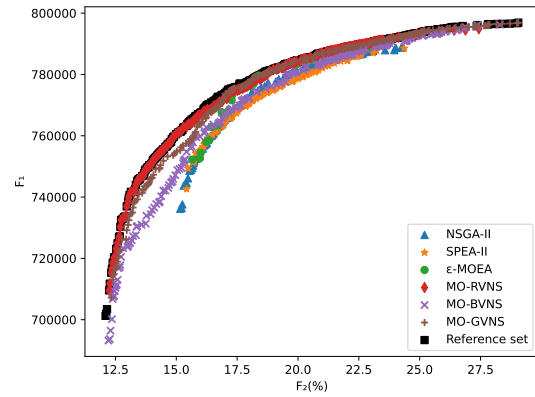
Имајући у виду да доносиоци одлука бирају компромисно решење на основу конкретне реалне ситуације, улога алгоритама је да пруже што више различитих алтернатива. Графичка репрезентација апроксимативних скупова решења се користи како би се дао бољи опис добијених решења на једној инстанци из сваке групе. Слика 3.4 приказује апроксимативне скупове (фронтове) са највећом вредношћу индикатора HV добијене у 10 извршавања на једној инстанци из сваке групе (Њујорк, Бронкс, Сан Франциско и Кингс). Решења из референтног скупа су означена црним квадратима на све четири слике. Вредности F_2 на хоризонталној оси су приказане као проценат непокривених корисника из скупа J .

На основу графичке репрезентације може се приметити да су MO-VNS решења боље распрострањена по простору функција циља у поређењу са онима које дају MOEA методе. На Њујорк и Бронкс инстанцама (видети Сliku 3.4a и Сliku 3.4b) постоји одређено преклапање MO-VNS и MOEA решења на средини простора функција циља, док на инстанцама већих димензија Сан Франциско и Кингс (Слика 3.4c и Слика 3.4d), решења три MO-VNS варијанте очигледно доминирају над решењима MOEA метода. Може се такође приметити да MO-RVNS често даје боља решења на средини простора функција циља, док су MO-BVNS и MO-GVNS решења равномерније распрострањена по про-

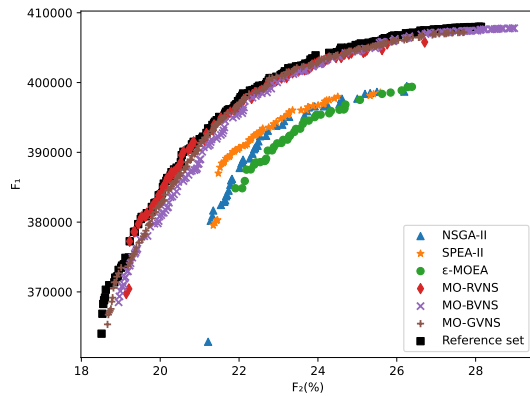
3.4. ЕКСПЕРИМЕНТАЛНИ РЕЗУЛТАТИ



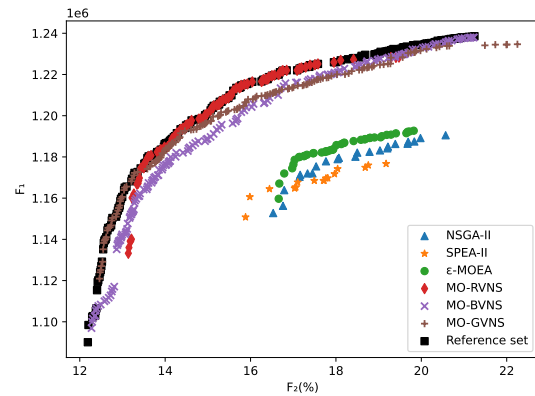
(a) Њујорк ($|I| = 271$, $|J| = 2442$)



(b) Бронкс ($|I| = 383$, $|J| = 3456$)



(c) Сан Франциско ($|I| = 513$, $|J| = 4624$)



(d) Кингс ($|I| = 773$, $|J| = 6957$)

Слика 3.4: Референтни скуп и апроксимативни скуп са највећом хиперзапремином добијени за прву инстанцу из сваке групе

стору. За боље разумевање добијених решења, може се анализирати избор доносиоца одлука када се користи референтни скуп за Бронкс инстанцу (видети Сliku 3.4b). Ако доносилац одлука преферира мање непокривених корисника, он ће одабрати решење са $\approx 12\%$ непокривених корисника и $\approx 700\,000$ покривене тежинске потражње. У супротном, он може да повећа покривену тежинску потражњу до $\approx 800\,000$, али $\approx 30\%$ корисника ће остати непокривено. Између ова два екстремна решења, доносилац одлука може изабрати између многих решења са компромисом између покривене тежинске потражње и броја непокривених корисника.

Глава 4

Уопштени проблем постављања регенератора у оптичким мрежама

4.1 Опис проблема и преглед релевантне литературе

Оптичке мреже су тип телекомуникационих мрежа које користе оптичке каблове за пренос података у виду светлосних импулса (оптичких сигнала) између два чвора. Употреба светлости као медија за пренос чини оптичку мрежу једном од најбржих телекомуникационих мрежа. Са друге стране, квалитет преноса сигнала опада при повећању удаљености од изворног чвора. Стога, специјални и скупи уређаји, који се називају *регенератори*, морају бити уграђени у мрежу да би се повратио квалитет оптичког сигнала. Имајући у виду високе трошкове инсталације и одржавања регенератора, важно је минимизовати број уграђених регенератора, а истовремено одржати поуздану комуникацију између свих чворова у мрежи. Овај проблем су представили Chen и сар. [35] и означен је као проблем постављања регенератора (енгл. Regenerator Location Problem - RLP). У RLP-у, претпоставља се да су трошкови постављања исти за све регенераторе и да сви чворови у мрежи могу да комуницирају. Касније, Chen и сар. [34] су разматрали случај када само подскуп чворова у мрежи комуницира (крајњи чворови) и добијени проблем је означен као уопштени проблем постављања регенератора (енгл. Generalized Regenerator Location Problem - GRLP). Мотивисани чињеницом да трошкови инсталације регенератора на различитим локацијама могу варирати због трошкова некретнина или из других

практичких разлога, Chen и сар. [34] су представили варијанту GRLP-а, под називом тежински уопштени проблем постављања регенератора (енгл. *Weighted Generalized Regenerator Location Problem - WGRLP*). WGRLP боље одражава реалну ситуацију када регенератори морају бити уграђени на нехомогеном простору (на пример, трошкови постављања регенератора у центру града могу бити знатно већи у односу на приградско насеље или рурално подручје). Све претпоставке и захтеви у WGRLP-у су исти као и у GRLP-у, са изузетком трошкова инсталације који могу варирати у зависности од потенцијалне локације за постављање регенератора. Циљ WGRLP-а је минимизовати укупан збир трошкова инсталације регенератора. У случајевима када су трошкови уградње регенератора једнаки за сваку потенцијалну локацију (тј. регенератори су распоређени на хомогеном простору), коришћење GRLP-а је прикладније. И GRLP и WGRLP су НП-тешки проблеми оптимизације, имајући у виду да је RLP доказано НП-тежак [35, 71].

Због сложености разматраних проблема постављања регенератора, као и чињенице да оптичке мреже имају велики број чворова, егзактне методе су у могућности да обезбеде решења само за инстанце малих и средњих димензија. Стога, хеуристичке методе представљају природан избор за проблеме који произилазе из дизајна оптичких мрежа великих димензија [96]. Chen и сар. [35] су предложили три хеуристичке методе и методу гранања и одсецања (BnC) за решавање RLP-а. На основу уочене везе RLP-а и проблема разапињућег стабла са максималним бројем листова (енгл. *Maximum Leaf Spanning Tree Problem - MLSTP*), закључено је да се процедура стратешке осцилације предложена за MLSTP [158] може применити и на RLP, што је и урађено у раду [35]. Касније, Duarte и сар. [58] су предложили GRASP и пристрасни генетски алгоритам са насумичним кључевима (енгл. *Biased Random Key Genetic Algorithm*) за решавање RLP-а. Методу гранања и одсецања, две конструктивне хеуристике (GH1 и GH2), и процедуру локалне претраге су развили Chen и сар. [34] за решавање GRLP, док су Quintana и сар. [147] предложили GRASP за исти проблем.

У циљу решавања проблема WGRLP, Chen и сар [34] су модификовали BnC и две хеуристике GH1 и GH2 које су претходно биле развијене за GRLP. У раду [134] предложене су SVNS и BVNS методе за решавање WGRLP. Обе методе предложене у раду [134] могу бити примењене и за решавање проблема GRLP, тако што се при примени предложених метода за решавање WGRLP цена сваке локације постави на 1, чиме се WGRLP своди на GRLP.

У овој дисертацији, предложена је основна варијанта VNS метахеуристике, која се може успешно применити на оба проблема, WGRLP и GRLP. Предложена метода представља побољшање у односу на постојеће варијанте VNS метода које су претходно имплементирани у [134], те надаље означен као побољшани BVNS (iBVNS). Предложени iBVNS је тестиран на GRLP и WGRLP инстанцама из литературе [34, 134]. Добијени iBVNS резултати су упоређени са резултатима егзактне VnC методе, похлепних хеуристика GH1 и GH2 [34], постојећих SVNS и BVNS метода [134], као и најбоље познате метахеуристике GRASP [147] за GRLP. Предложена iBVNS имплементација, као и њени резултати на GRLP и WGRLP инстанцама изложени су у раду [132].

4.2 Формулација проблема на графу

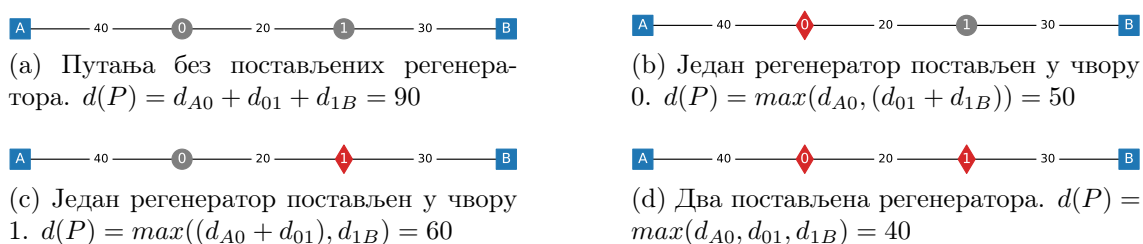
Оба разматрана проблема, GRLP и WGRLP, полазе од оптичке мреже представљене графом $G = (V, E)$ са скупом чворова V и скупом грана E . Скуп чворова V је подељен на два дисјунктна скупа: скуп крајњих (терминалних) чворова T који комуницирају међусобно и скуп S потенцијалних локација за постављање регенератора. Постоји ограничење $d_{max} > 0$ за удаљеност коју оптички сигнал може да пређе пре почетка слабљења квалитета сигнала. Путања P која повезује два крајња чвора је представљена низом i, v_1, \dots, v_m, j , где је $i, j \in T, v_k \in V, k = 1, \dots, m$. Претпоставља се да постоји грана која повезује сваки пар суседних чворова у овом низу. Дужина путање P , означена као $l(P)$, је добијена сабирањем дужина грана између парова суседних чворова у низу који представља P . Ако важи $l(P) \leq d_{max}$, сигнал се може пренети од i до j без регенерисања. У супротном, регенератори морају бити постављени у једном или више унутрашњих чворова путање P .

Нека је $L \subseteq S$ скуп постављених регенератора и претпоставимо да путања P садржи регенераторе $\{r_1, \dots, r_k\} \subseteq L$. Удаљеност између два крајња чвора i и j дуж P се рачуна као $d(P) = \max\{l(P_{i,r_1}), l(P_{r_1,r_2}), \dots, l(P_{r_k,j})\}$. Овде је $P_{k,l}$ подпутања P која повезује k и l , где је $k, l \in \{i, r_1, \dots, r_k, j\}$. Рачунање удаљености између два крајња чвора је илустровано на Слици 4.1. Растојања су израчуната за путању без регенератора (Слика 4.1(a)), један регенератор (Слика 4.1(b) и 4.1(c)) и два регенератора (Слика 4.1(d)).

Код оба проблема, GRLP и WGRLP, скуп регенератора $L \subseteq S$ мора бити изабран тако да сваки крајњи чвор i може да пошаље сигнал другом крајњем

4.2. ФОРМУЛАЦИЈА ПРОБЛЕМА НА ГРАФУ

чвору j дуж неке путање P тако да је $d(P) \leq d_{max}$. Једина разлика између GRLP и WGRLP је претпоставка која се односи на трошкове постављања регенератора. Код проблема WGRLP, сваки чвор $r \in S$ има додељену тежину $w_r > 0$ која одражава цену инсталације регенератора на r . У случају GRLP-а, трошкови инсталације су једнаки за све чворове из S , што значи $w_r = 1$ за сваки $r \in S$. Циљ WGRLP-а је минимизација суме тежина постављених регенератора $\sum_{r \in L} w_r$, док је циљ GRLP-а минимизација броја постављених регенератора $\sum_{r \in L} 1$.



Слика 4.1: Израчунавање дужине путање између два крајња чвора. Дужина путање зависи од броја и позиција регенератора.

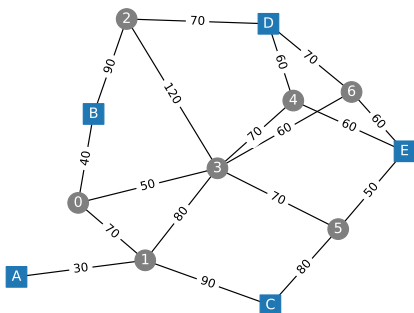
Почетни граф $G = (V, E)$ се трансформише у једноставнији граф $B = (V, E')$, означен као *комуникациони граф*. Овај концепт су увели Chen и сар. [34], а коришћен је у радовима [34, 134, 147] за решавање проблема GRLP и WGRLP. Кораци конструисања комуникационог графа су следећи:

- 1) Уклонити све гране дужине веће од d_{max} из скупа E ;
- 2) За сваки несуседни пар чворова $i, j \in V$, додати нову грану e која их повезује, и израчунати дужину e као дужину одговарајуће најкраће путање између i и j ;
- 3) Уклонити све нове гране дужине веће од d_{max} и означити преостали скуп грана са E' .

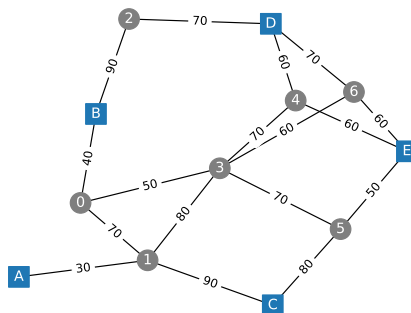
Пар крајњих чворова $i, j \in T$ се сматра *повезаним* ако постоји грана $e \in E'$ која их повезује. Ово значи да је могућа директна комуникација између i и j . У супротном, овај пар се означава као пар који *није директно повезан* (енгл. not directly connected - NDC). Решење за GRLP (WGRLP) је допустиво ако су сви парови крајњих чворова повезани. Кораци за изградњу комуникационог графа су приказани на Слици 4.2.

За боље разумевање GRLP и WGRLP проблема, једна инстанца малих димензија из литературе [34] и њена два оптимална решења су приказани на

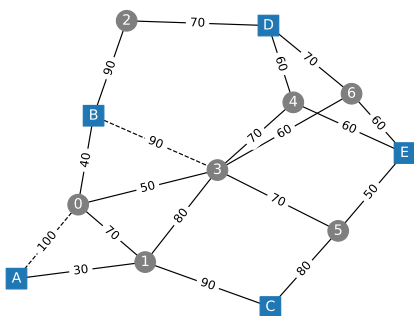
4.3. УНАПРЕЂЕНА МЕТОДА ПРОМЕНЉИВИХ ОКОЛИНА ЗА WGRLP



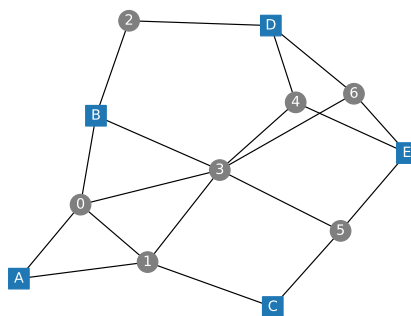
(a) Почетни граф са тежинама грана једнаким дужини између два чвора и $d_{max} = 100$.



(b) Гране са дужином већом од d_{max} су уклоњене. У овом случају, грана (2,3) са дужином 120 је уклоњена из графа.



(c) Најкраће путање између чворова дужине не веће од d_{max} су додате као нове гране. Гране (B,3) и (A,0) се додају у граф.



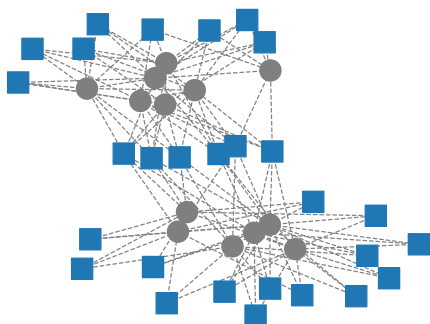
(d) Добијени комуникациони граф.

Слика 4.2: Конструкција комуникационог графа

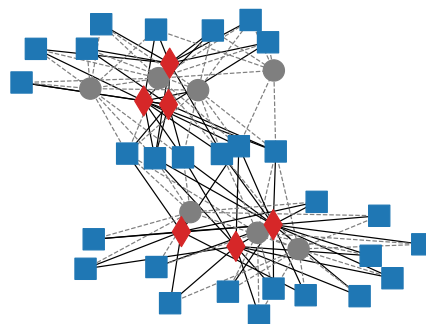
Слици 4.3. Разлика у трошковима постављања регенератора код GRLP и WGRLP може резултирати различитим оптималним решењима, што се може видети на сликама 4.3b и 4.3c.

4.3 Унапређена метода променљивих околина за WGRLP

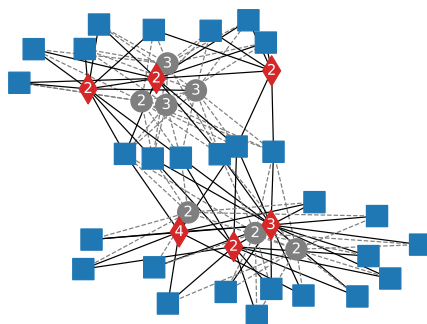
Метода променљивих околина је већ успешно примењивана у постојећој литератури за решавање проблема оптимизације телекомуникационих мрежа [91,112,114,120,126,127]. У овој дисертацији, предложена је побољшана основна



(а) Инстанца са 50 чворова од којих 13 (25%) чине потенцијалне локације за регенераторе. Број NDC парова је 266.



(б) Оптимално решење за основну варијанту проблема (GRLP). Вредност функције циља је 6.



(с) Оптимално решење за тежинску варијанту проблема (WGRLP). За сваку локацију регенератора назначена је цена постављања. Вредност функције циља је 15.

Слика 4.3: Инстанца малих димензија из литературе и оптимална решења за GRLP и WGRLP проблеме. Успостављени регенератори означени су црвеном бојом, док је свака потенцијална локација обележена ценом постављања регенератора.

VNS метахеуристике у односу на VNS варијанту из рада [134], означена са iBVNS, која може да се користи за решавање оба проблема, WGRLP и GRLP. Предложени iBVNS даје решења бољег квалитета за краће време извршавања у поређењу са постојећим хеуристичким приступима за решавање GRLP и WGRLP. У наставку овог одељка детаљно су објашњени елементи iBVNS методе.

Репрезентација решења

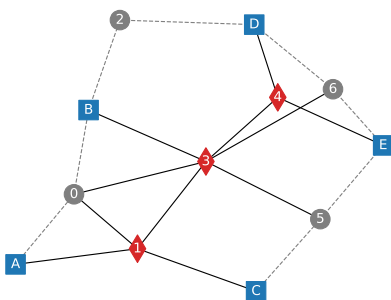
Предложени iBVNS користи низ целих бројева за кодирање решења за WGRLP. Сваки елемент низа представља индекс локације са постављеним регенератором. Када су локације регенератора познате, вредност функције циља WGRLP-а се добија сабирањем тежина ових локација, док је вредност функције циља GRLP-а једнака броју постављених регенератора.

У предложеној iBVNS методи, комуникациони граф је представљен листом чворова. За сваки чвор из листе, хеш скуп чува информације о његовим суседним чворовима. На овај начин, алгоритам може ефикасно проверити да ли разматрана грана постоји у графу и истражити суседне чворове датог чвора.

Решење проблема за GRLP (WGRLP) се сматра допустивим ако су сви *NDC* парови у одговарајућем комуникационом графу повезани. Због тога је за сваки пар чворова потребно на ефикасан начин испитати да ли је он повезан или *NDC* пар. Потребна ефикасност се постиже чувањем свих *NDC* парова у хеш мапи. Тачније, хеш мапа садржи сваки *NDC* пар (индексе чворова који чине пар) са додељеним логичким вредностима (*true* за повезане и *false* за неповезане *NDC* парове). Ако су све вредности у хеш мапи једнаке *true*, одговарајуће решење је допустиво. Графички прикази једног допустивог и једног недопустивог решења проблема WGRLP дати су на Слици 4.4, као и одговарајуће табеле повезаности *NDC* парова за свако од приказаних решења.

ОкоLINE решења

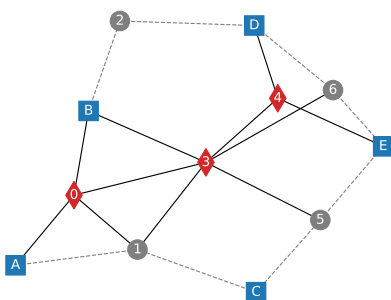
Предложена iBVNS имплементација користи две окоLINE решења, означене као N_1 и N_2 . ОкоLINE N_1 решења x састоји се од свих решења добијених од x уклањањем једног постављеног регенератора. Ова окоLINE се истражује у фази размрдавања предложеног iBVNS-а, где се регенератор за уклањање бира насумично. Понављањем описаног уклањања регенератора k пута добија се окоLINE величине k . Требало би напоменути да се уклањањем регенератора



(a) Графички приказ допустивог решења. Локације регенератора су приказане црвеном бојом, а индекси (1, 3, 4) се чувају у низу.

A - B	True	B - D	True
A - C	True	B - E	True
A - D	True	C - D	True
A - E	True	C - E	True
B - C	True	D - E	True

(b) Табела NDC парова који одговарају решењу приказаном под (a). Сви NDC парови су повезани (ознака true).



(c) Графички приказ недопустивог решења. Локације регенератора су приказане црвеном бојом а индекси (0, 3, 4) се чувају у низу. Крајњи чвор C је одвојен од других крајњих чворова.

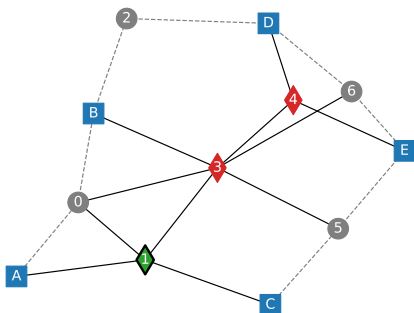
A - B	True	B - D	True
A - C	False	B - E	True
A - D	True	C - D	False
A - E	True	C - E	False
B - C	False	D - E	True

(d) Табела NDC парова који одговарају решењу приказаном под (c). NDC парови који укључују чвор C означено је да нису повезани (ознака false).

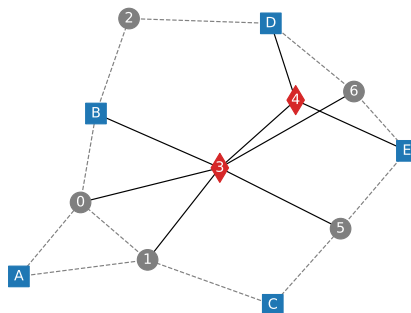
Слика 4.4: Репрезентација допустивог (a, b) и недопустивог (c, d) решења. Решење је приказано графички са означеним локацијама регенератора (црвена боја и облик ромба). Активне гране (повезане на регенератор) су приказане са црним пуним линијама, док су неактивне приказане сивим испрекиданим линијама. Сви крајњи чворови морају бити повезани да би решење било допустиво (видети табеле NDC парова).

4.3. УНАПРЕЂЕНА МЕТОДА ПРОМЕНЉИВИХ ОКОЛИНА ЗА WGRLP

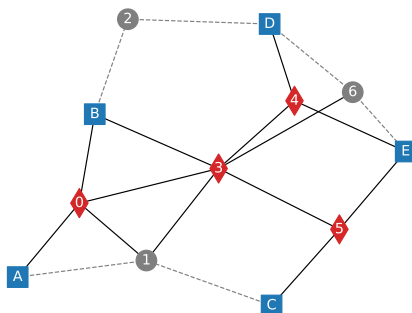
може добити недопустиво решење, које се у том случају поправља процедуром описаном у Одељку 4.3. На Слици 4.5. дат је пример једног допустивог решења - Слика 4.5(a), његовог недопустивог суседа из околине N_1 - Слика 4.5(b) и поправљеног суседног решења до допустивог - Слика 4.5(c)



(a) Почетно решење са изабраним регенератором (на чвору 1) за уклањање.



(b) Решење добијено уклањањем регенератора на локацији 1. Добијено решење је недопустиво. Крајњи чворови А и С су одвојени од других чворова.

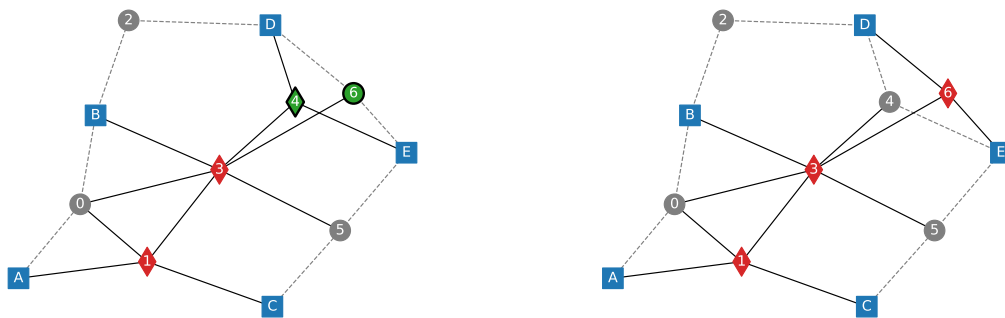


(c) Решење добијено поправљањем недопустивог решења (b) тако што су додата два регенератора (0 и 5). Добијено решење је допустиво, али користи више регенератора у поређењу са почетним решењем (a).

Слика 4.5: Илустрација околине N_1 .

Околина N_2 решења x садржи сва решења добијена од x уклањањем једног постављеног регенератора и постављањем регенератора на другу потенцијалну локацију са мањим трошковима постављања у поређењу са почетном локацијом (операција замене). Околина N_2 се користи у фази локалне претраге предло-

женог iBVNS-a. Описана операција замене се извршава само ако решење остаје допустиво. Све операције замене које могу да доведу до побољшања функције циља се испитују процедуром провере допустивости описаном у Одељку 4.3. За илустрацију, једно решење добијено операцијом замене приказано је на Слици 4.6.



(a) Почетно допустиво решење са два чвора (4 и 6) која су изабрана за операцију замене.

(b) Решење након операције замене чворова 4 и 6. Ново решење је остало допустиво.

Слика 4.6: Илустрација околине N_2 .

Ефикасно ажурирање функције циља

Две главне операције које се изводе током извршавања iBVNS-a су: додавање регенератора у решење и уклањање регенератора из решења. Када додајемо регенератор у решење, претрага графа по ширини (енгл. Breadth-first search - BFS) се примењује почевши од локације на коју се регенератор $r \in S$ поставља. Током BFS претраге, низ логичких вредности *visited* се креира и ажурира да би се сачувала информација о посећеним чворовима. Ако је чвор посећен, његов одговарајући елемент у овом низу је *true*, а у супротном *false*.

У предложеној iBVNS имплементацији, BFS претрага је модификована на следећи начин. Када се разматра чвор v који је суседан локацији регенератора u , његови суседи ће бити испитани само ако је регенератор инсталиран на u . У супротном, v се означава као посећен, и његови суседи се даље не разматрају. На овај начин, није потребно вршити промене у комуникационом графу (брисање и уметање нових вештачких грана), као што је то случај у GRASP имплементацији из рада [147] и VNS имплементацијама из рада [134]. Додавање

нових грана у комуникациони граф негативно утиче на ефикасност алгоритама што је у предложеној iBVNS имплементацији избегнуто модификацијом BFS претраге. Алгоритам 23 показује псеудокод имплементиране модификоване BFS процедуре почевши од локације r .

Алгоритам 23 Модификована BFS претрага за комуникациони граф

```

1: Input: starting location  $r$ 
2: Output: boolean array  $visited$ 
3:  $queue \leftarrow$  empty queue;
4:  $visited \leftarrow$  false;
5:  $visited(r) \leftarrow$  true;
6:  $queue.add(r)$ ;
7: while  $queue$  is not empty do
8:    $u \leftarrow queue.poll()$ ;
9:   for  $v$  adjacent to  $u$  do
10:    if not  $visited(v)$  then
11:       $visited(v) \leftarrow$  true;
12:      if location  $v$  has a regenerator then
13:         $queue.add(v)$ ;
14: return  $visited$ 

```

Приликом уклањања регенератора из решења или извршавања операције замене, може се десити да добијено решење није допустиво. Део који одузима највише времена је испитивање допустивости новодобијеног решења након сваке од ове две операције. Стога, ефикасна процедура провере допустивости може значајно смањити укупно време извршавања. У предложеној iBVNS методи, модификовани BFS се користи да се ефикасно провери допустивост решења након уклањања регенератора или извршавања операције замене. Процедура за проверу допустивости на основу BFS-а је описана Алгоритмом 24. Процедура полази од новодобијеног решења x и његовог скупа постављених регенератора, означеног као $x.regenerators$. Локације регенератора које су посећене се чувају у скупу $visitedRegenerators$. На почетку извршавања BFS процедуре, скуп $visitedRegenerators$ је празан и бројач повезаних парова $pairs$ из $NDCpairs$ је постављен на 0. Процедура провере допустивости разматра један по један непосећени регенератор r из скупа $x.regenerators$ и примењује BFS почевши од локације r . BFS претрага враћа скуп посећених чворова $visited$ као излаз. У случају да је регенератор r већ био посећен током BFS претраге неког од претходно разматраних регенератора, он се не разматра поново и алгоритам једноставно додаје r у скуп $visitedRegenerators$. Пар $pair$ из $NDCpairs$

је означен као повезан ако су оба крајња чвора пара $pair$ посећена, а бројач повезаних NDC парова се повећава за један. Описаним поступком, сви сада повезани парови из $NDCpairs$ скупа су идентификовани на ефикасан начин. Решење је означено као допустиво ако је сваки пар $pair$ из скупа NDC парова $NDCpairs$ повезан. Процедура допустивости је илустрована једноставним примером приказаним на Слици 4.7.

Алгоритам 24 Процедура провере допустивости решења

```

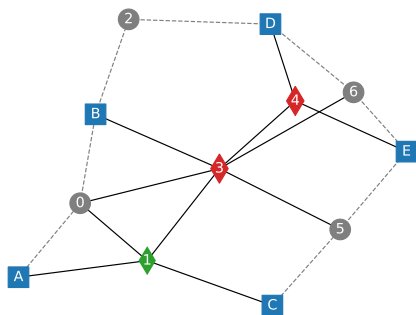
1: Input: solution  $x$ 
2: Output: boolean feasibility
3:  $visitedRegenerators \leftarrow \emptyset$ ;
4:  $connected \leftarrow 0$ ;
5: for  $r$  in  $x.regenerators$  do
6:   if  $r \notin visitedRegenerators$  then
7:      $visited \leftarrow \text{BFS}(r)$ ;
8:     for  $pair$  in  $NDCpairs$  do
9:       if  $pair$  is not connected and  $visited(pair.start)$  and  $visited(pair.end)$  then
10:        mark  $pair$  as connected;
11:         $connected \leftarrow connected + 1$ ;
12:   for  $p$  in  $x.regenerators$  do
13:     if  $visited(p)$  then
14:       add  $p$  to  $visitedRegenerators$ ;
15: return  $connected == \text{size of } NDCpairs$ ;

```

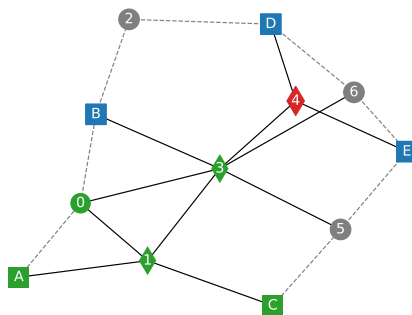
Примењена стратегија за проверу допустивости заснована на BFS процедури не зависи од редоследа додавања регенератора у решење, као што је то био случај са провером допустивости имплементираном у VNS методама из рада [134] и GRASP методи из рада [147]. Поред тога, може се десити да процедуре провере допустивости примењене у постојећим метахеуристичким приступима [134, 147] означе допустиво решење као недопустиво. Са друге стране, нови поступак провере допустивости решења имплементиран у iBVNS методи не допушта могућност да се допустиво решење означи као недопустиво, у чему се огледа још једна предност предложене iBVNS методе.

Похлепна конструкција и поправка решења

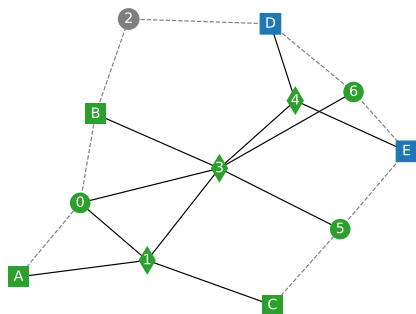
Предложени iBVNS користи похлепну процедуру, која се примењује за конструисање почетног решења и за исправљање недопустивих решења која се могу појавити приликом брисања регенератора. У првом случају, похлепна процедура почиње од празног решења, док је у другом случају улаз решење које је



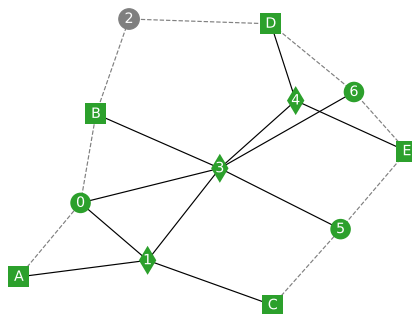
(a) Процедура провере допустивости почиње од првог чвора регенератора у низу решења (изабран је регенератор у чвору 1).



(b) Суседни чворови регенератора 1 су означени као посећени (обојени зеленом бојом). Регенератор на чвору 3 је сада посећен и процедура се наставља од њега.



(c) Суседни чворови регенератора 3 су означени као посећени. Регенератор на чвору 4 је сада посећен и процедура се наставља од њега.



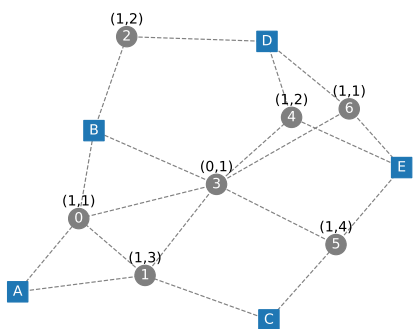
(d) Суседни чворови регенератора 4 су означени као посећени. Сви крајњи чворови су сада посећени, што значи да су сви NDC парови сада повезани и решење је означено као допустиво.

Слика 4.7: Илустрација поступка провере допустивости.

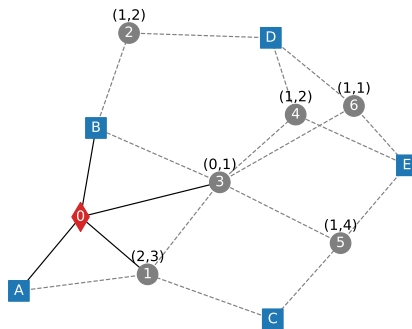
провером допустивости означено као недопустиво. Предложена похлепна процедура, описана Алгоритмом 25, итеративно додаје један по један регенератор почевши од улазног решења s . За сваку потенцијалну локацију i без постављеног регенератора, процедура израчунава вредност $gain$ (добит), тј. укупан број NDC парова који би били повезани ако се регенератор постави на i . Потенцијалне локације су сортиране у листи $list$ у опадајућем редоследу по односу добити и трошкова (у случају GRLP-а, трошак сваке потенцијалне локације регенератора је 1). Нова локација регенератора се насумично бира из скупа који чине првих $count$ елемената из листе $list$. Вредност $count$ је једнака броју потенцијалних локација са односом добити и трошкова који није мањи од одређене вредности. У предложеној похлепној процедури та вредност се рачуна као однос добити и трошкова прве локације max из листе $list$ помноженом параметром $\gamma \in (0, 1)$. Мотивација за примену ове стратегије за избор нове локације регенератора је обезбеђивање боље диверсификације претраге. Локација са новим постављеним регенератором се брише из листе потенцијалних локација. Након додавања регенератора, није потребно из почетка покретати проверу допустивости (тј. довољно је испитати да ли су сви NDC парови означени као повезани), што обезбеђује додатну уштеду времена. Описани кораци се понављају све док се не конструише допустиво решење. Описана похлепна процедура је илустрована примером на Слици 4.8.

Основна структура унапређене методе променљивих околина

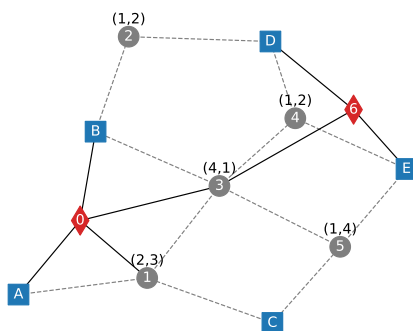
Основна структура предложене iBVNS методе је приказана Алгоритмом 26. Параметар критеријума заустављања I_{max} означава максималан број итерација без побољшања решења, док параметар k_{max} означава максималну величину околина. Параметар $weighted$ узима вредност $true$ ако се iBVNS примењује на WGRLP, а $false$ ако се решава GRLP. Почетно решење се конструише похлепном процедуром (Алгоритам 25), која полази од празног решења. Број итерација без побољшања I_{count} је у почетку постављен на 0. Главна VNS петља почиње постављањем бројача величине околина k на 1. Затим се итеративно примењују три основне фазе алгоритма, размрдавање, локална претрага и промена околина (Move or not) док је $k \leq k_{max}$. Главна VNS петља се понавља све док I_{count} не достигне I_{max} .



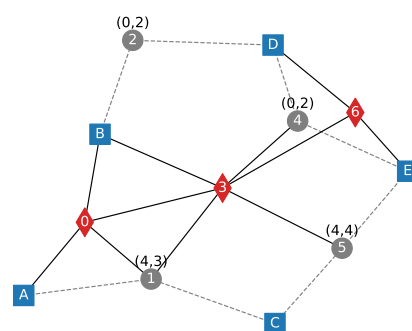
(a) За сваку потенцијалну локацију, тј. чвор је дата вредност $cost$. Поред тога, $gain$ чвора се израчунава као број нових NDC парова који би били повезани ако се регенератор постави на ту локацију. Ова два броја су приказана изнад чвора као $(gain, cost)$.



(b) Први регенератор је постављен на чвору 0, са највећим односом $gain$ према $cost$ (исто као чвор 6). У овом случају, насумично се одређује који ће чвор бити изабран. Вредности $gain$ за сваки чвор се ажурирају. Један нови NDC пар је сада повезан (A-B). Вредност функције циља решења је 1.

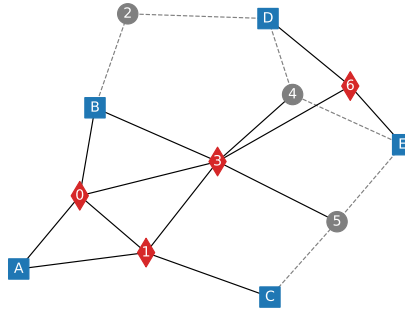


(c) Нови регенератор је постављен на чвор 6. Један нови NDC пар је сада повезан (D-E). Вредност функције циља решења је 2.



(d) Нови регенератор је постављен на чвор 3. Четири нова NDC пара су сада повезана (A-D, A-E, B-D и B-E). Вредност функције циља решења је 3.

Слика 4.8: Илустрација поступка конструкције решења. Ради једноставности, параметар γ је постављен на 1.



(e) Нови регенератор је постављен на чвор 1. Четири нова NDC пара су сада повезана (A-C, B-C, C-D и C-E). Решење је сада допуштиво и вредност функције циља је 6. Може се видети да је оптимално решење другачије (1, 3 и 6), где је вредност функције циља једнака 5.

Слика 4.8: Илустрација поступка конструкције решења (наставак)

Алгоритам 25 Процедура похлепне конструкције и поправке решења

```

1: Input: solution  $s$ 
2: Output: feasible solution  $s$ 
3: while solution  $s$  is infeasible do
4:   for  $i$  in candidate locations do
5:      $visited \leftarrow \text{BFS}(i)$ ;
6:      $gain_i \leftarrow 0$ 
7:      $pairs_i = \text{empty list}$ ;
8:     for  $pair$  in  $NDCpairs$  do
9:       if  $pair$  is not connected and  $visited(pair.start)$  and  $visited(pair.end)$  then
10:         $gain_i \leftarrow gain_i + 1$ ;
11:        add  $pair$  to  $pairs_i$ ;
12:    $list \leftarrow \text{sort candidate locations in decreasing order by gain to cost ratio}$ ;
13:    $max \leftarrow \text{first element of } list$ ;
14:    $count \leftarrow 0$ ;
15:   for  $e$  in  $list$  do
16:     if gain to cost ratio of  $e \geq \gamma \cdot \text{gain to cost ratio of } max$  then
17:        $count = count + 1$ ;
18:    $selected \leftarrow \text{select random location from the first } count \text{ elements of } list$ ;
19:   add regenerator at location  $selected$ ;
20:   mark all pairs from  $pairs_{selected}$  as connected;
21:   remove  $selected$  from candidate locations;

```

Фаза размрдавања се реализује у околини N_1 величине k . Прецизније, k насумично изабраних регенератора се уклања из тренутног решења x , што даје

ново решење x' , које представља улаз за процедуру провере допустивости описану Алгоритмом 24. Ако је решење x' недопустиво, похлепна процедура се примењује на x' и поправља га до допустивог. Алгоритам наставља са фазом локалне претраге, чији је улаз допустиво решење x' . Фаза локалне претраге се састоји од процедура CLEAN и LOCALSEARCH у случају WGRLP-а. Прво се примењује процедура CLEAN чија је улога уклањање непотребних регенератора из x' , ако их има. Регенератор се сматра *непотребним* ако његово уклањање не утиче на допустивост решења. Затим се примењује процедура LOCALSEARCH да би се пронашло потенцијално боље решење у околини N_2 величине $k = 1$ решења x' . То значи да се у сваком кораку изводи једна операција замене (уклањање једног регенератора из x' и постављање јефтинијег), али само ако се том заменом чува допустивост решења. Код имплементације процедуре LOCALSEARCH користи се стратегија првог побољшања. У случају GRLP-а, описана операција замене не може да побољша вредност функције циља, док операција затварања два и постављања једног регенератора има кубну сложеност, што би значајно утицало на време извршавања алгоритма. Стога, при решавању GRLP, користи се само CLEAN процедура за уклањање непотребних регенератора, како би се очувала ефикасност iBVNS-а.

При извршавању корака промене околине (Move or Not), алгоритам ће прећи у решење x' добијено у фази локалне претраге само ако је оно боље од тренутно најбољег решења x . У овом случају, бројач итерација без побољшања I_{count} се ресетује на 0, а k се поставља на 1. У супротном, k се повећава за 1, а претрага се наставља са размрдавањем у околини N_1 величине $k + 1$. Описани кораци се понављају док се не достигне максимални број итерација без побољшања решења I_{max} .

Алгоритам 26 Предложени iBVNS за GRLP и WGRLP

```
1: Input:  $I_{max}$ ,  $k_{max}$ , weighted
2: Output: solution  $x$ 
3:  $x \leftarrow$  empty solution;
4: GreedyProcedure( $x$ ) ;
5:  $I_{count} \leftarrow 0$ ;
6: while  $I_{count} \leq I_{max}$  do
7:    $k \leftarrow 1$ ;
8:   while  $k \leq k_{max}$  do
9:      $x' \leftarrow$  Shake in  $N_1(x, k)$ ;           // Shaking phase
10:     $x' \leftarrow$  Clean ( $x'$ );                 // Local search phase
11:    if weighted then
12:       $x' \leftarrow$  LocalSearch in  $N_2(x', 1)$ ;
13:    if  $f(x') < f(x)$  then                   // Move or not step
14:       $x \leftarrow x'$ ;
15:       $k \leftarrow 1$ ;
16:       $I_{count} \leftarrow 0$ ;
17:    else
18:       $k \leftarrow k + 1$ ;
19:     $I_{count} \leftarrow I_{count} + 1$ ;
return  $x$ 
```

4.4 Експериментални резултати

За тестирање предложеног iBVNS-а, коришћено је пет скупова инстанци из литературе: скупови 1, 2 и 3 које су увели Chen и сар. [34] и скупови 4 и 5 који су генерисани у раду [134]. Сваки скуп података је дефинисан бројем чворова $n = |V|$ у мрежи и процентом $p(\%)$ крајњих чворова у V . Скуп 1 садржи GRLP инстанце малих и средњих димензија са $n \in \{50, 75, 100, 125, 150\}$ и $p \in \{25, 50, 75\}$. Скуп 2 се добија из скупа 1 додавањем насумично одабраних тежина из скупа $\{2, 3, 4\}$ потенцијалним локацијама регенератора. Скуп 3 се састоји од GRLP инстанци великих димензија са $n \in \{175, 200, 300, 400, 500\}$ и $p \in \{25, 50, 75\}$ пратећи иста правила као за скуп 1. На крају, скупови 4 и 5 садрже WGRLP инстанце великих димензија са $n \in \{175, 200, 300, 400, 500\}$ и $p \in \{25, 50, 75\}$, респективно. Скупови 4 и 5 из рада [134] се добијају на основу скупа 3 укључивањем насумично одабраних тежина из $\{2, 3, 4\}$ у случају скупа 4 и $\{1, 2, 3, 4, 5, 6, 7, 8\}$ за скуп 5. Сваки од разматраних скупова садржи по 10 инстанци за сваку комбинацију n и p . Детаљно објашњење о томе како су инстанце генерисане се може наћи у радовима [34, 134].

Предложена iBVNS метода је имплементирана у програмском језику Java.

4.4. ЕКСПЕРИМЕНТАЛНИ РЕЗУЛТАТИ

Сви експерименти су извршени на рачунару Intel Core i3-4170 CPU 3.7GHz са 8GB RAM-а. На свакој разматраној инстанци из скупова 1-5, предложени iBVNS је покренут пет пута. На основу резултата прелиминарних експеримента, вредности iBVNS параметара се постављају на следећи начин: $k_{max} = 2, \gamma = 0.95, I_{max} = 50 + n/5$ за GRLP, и $k_{max} = 4, \gamma = 0.95, I_{max} = 100 + n/10$ за WGRLP.

Табела 4.1: Резултати и поређења на GRLP инстанцама из скупа 1

Група	B&C [34]			GH1 [34]		GH2 [34]		BVNS [134]			SVNS [134]			iBVNS			
	<i>UB</i>	<i>LB</i>	<i>t(s)*</i>	<i>best</i>	<i>t(s)*</i>	<i>best</i>	<i>t(s)*</i>	<i>best</i>	<i>gap(%)</i>	<i>t(s)</i>	<i>best</i>	<i>gap(%)</i>	<i>t(s)</i>	<i>best</i>	<i>gap(%)</i>	<i>t(s)</i>	<i>t.tot(s)</i>
25	7.2	7.2	0.20	7.2	0.00	7.2	0.00	7.3	1.11	1.23	7.3	1.11	0.88	7.2	0.05	0.00	0.00
50	7.5	7.5	0.20	7.7	0.00	7.7	0.00	7.5	0.00	2.74	7.5	0.00	2.67	7.5	0.07	0.00	0.00
75	4.3	4.3	0.10	4.3	0.00	4.3	0.00	4.3	0.00	1.06	4.3	0.00	1.14	4.3	0.04	0.00	0.00
25	9.7	9.7	1.40	9.8	0.00	9.8	0.00	9.7	0.00	2.96	9.7	0.00	1.93	9.7	0.10	0.00	0.00
50	9.2	9.2	2.20	9.6	0.00	9.5	0.00	9.2	0.00	3.74	9.2	0.00	6.03	9.2	0.12	0.01	0.20
75	5.5	5.5	0.60	5.5	0.00	5.6	0.00	5.5	0.00	2.96	5.5	0.00	3.09	5.5	0.06	0.00	0.00
25	11.7	11.7	6.70	11.8	0.00	11.8	0.00	11.8	0.67	5.92	12.2	2.33	7.73	11.7	0.44	0.04	0.00
100	10.8	10.8	21.50	10.9	0.00	11.0	0.00	10.8	0.55	11.96	10.8	1.15	9.76	10.8	0.51	0.11	0.18
75	7.0	7.0	3.20	7.2	0.00	7.3	0.00	7.0	0.47	14.82	7.0	1.12	12.76	7.0	0.21	0.02	0.00
25	14.1	14.1	31.70	14.1	0.00	14.1	0.00	14.2	0.53	12.05	14.1	0.90	10.31	14.1	0.76	0.08	0.00
125	11.9	11.9	237.10	12.1	0.20	12.0	0.10	11.9	0.38	12.91	11.9	1.38	8.84	11.9	0.89	0.15	0.30
75	8.1	8.1	14.20	8.4	0.00	8.5	0.10	8.1	1.28	13.35	8.1	1.87	13.27	8.1	0.36	0.07	1.82
25	14.5	14.5	259.80	14.5	0.10	14.5	0.10	14.5	0.15	14.99	14.5	0.89	12.31	14.5	1.13	0.16	0.15
150	12.2	12.2	838.60	12.3	0.40	12.3	0.50	12.2	0.76	18.88	12.2	2.27	17.98	12.2	0.95	0.21	0.21
75	9.1	9.1	403.40	9.5	0.10	9.5	0.20	9.1	1.89	17.75	9.2	2.17	9.54	9.1	0.46	0.07	1.95
Просек	9.52	9.52	121.39	9.66	0.05	9.67	0.07	9.54	0.52	9.16	9.57	1.01	7.88	9.52	0.32	0.06	0.41

*на Intel Core 2 Duo са 3 GHz и 3.25 GB RAM-а

У Табелама 4.1-4.2, представљени су резултати на GRLP инстанцама из скупова 1 и 3, респективно. Приказани су резултати предложеног iBVNS-а, егзактне VnC методе и две хеуристике (GH1 и GH2) из рада [34], као и резултати VNS метода (BVNS и SVNS) предложених у раду [134]. Прве две колоне Табела 4.1-4.2 садрже параметре n и $p(\%)$ за сваку групу инстанци. Преостале колоне садрже просечне резултате добијене одговарајућом методом на инстанцама из исте групе (тј. инстанцама са истим n и p). Следеће три колоне садрже горње (UB) и доње границе (LB) оптималног решења добијеног егзактном VnC методом и време извршавања у секундама - $t(s)$ које захтева VnC на рачунару са Intel Core 2 Duo 3 GHz и 3.25 GB RAM-а. Следеће четири колоне представљају најбоље резултате добијене хеуристикама GH1 и GH2 ($best$) и одговарајућа времена извршавања - $t(s)$ на истом рачунару као и VnC. Резултати који се односе на перформансе предложеног iBVNS-а и постојећих BVNS и SVNS метода су приказани у последњих десет колона: најбоље решење добијено у пет циклуса - $best$, просечно процентуално одступање од оптималног (или најбољег познатог

4.4. ЕКСПЕРИМЕНТАЛНИ РЕЗУЛТАТИ

решења) - $gap(\%)$, просечно време извршавања у секундама које је потребно за постизање најбољег решења - $t(s)$, и просечно укупно време извршавања - $t.tot(s)$. Просечне вредности оптималних (или најбољих познатих) решења за сваку групу примера су подебљане у Табелама 4.1-4.2.

Табела 4.2: Резултати и поређења на GRLP инстанцама из скупа 3

Група	В&С [34]			GH1 [34]		GH2 [34]		BVNS [134]			SVNS [134]			iBVNS			
	<i>UB</i>	<i>LB</i>	<i>t(s)*</i>	<i>best</i>	<i>t(s)*</i>	<i>best</i>	<i>t(s)*</i>	<i>best</i>	<i>gap(%)</i>	<i>t(s)</i>	<i>best</i>	<i>gap(%)</i>	<i>t(s)</i>	<i>best</i>	<i>gap(%)</i>	<i>t(s)</i>	<i>t.tot(s)</i>
25	16.3	16.2	1625.50	16.4	0.10	16.5	0.30	16.3	0.00	15.43	16.3	0.59	12.45	16.3	0.39	0.20	1.73
175	13.1	12.3	2667.00	13.2	1.00	13.4	1.00	13.1	3.08	21.70	13.1	4.97	11.16	13.1	1.11	0.39	1.77
75	9.1	9.1	245.30	9.7	0.40	9.6	0.40	9.1	2.93	18.02	9.2	2.92	12.38	9.1	1.46	0.18	0.96
25	17.4	16.2	2869.00	17.4	0.30	17.5	0.40	17.4	0.17	21.54	17.4	1.11	13.34	17.4	0.29	0.48	2.40
200	12.8	11.3	3142.00	13.0	2.60	13.0	2.80	12.8	2.04	28.48	12.9	2.31	20.47	12.8	0.62	0.68	2.74
75	10.0	10.0	1116.00	10.6	1.70	10.5	1.70	10.1	1.96	14.55	10.1	2.32	15.02	10.0	0.52	0.35	1.37
25	16.8	15.3	3600.00	16.8	1.00	17.4	1.30	19.3	3.85	79.11	19.7	2.65	54.63	19.2	1.14	2.21	8.52
300	15.3	12.6	3600.00	15.4	1.20	16.1	2.40	16.5	2.87	86.08	16.9	4.69	27.27	16.3	2.48	2.52	7.95
75	12.4	9.9	3600.00	12.5	1.50	12.9	2.10	12.0	3.46	49.03	12.2	4.94	12.07	11.8	2.21	0.79	2.99
25	21.9	13.3	3600.00	22.4	7.40	22.4	8.90	19.9	2.72	285.61	20.5	3.10	57.06	19.9	1.14	6.67	22.23
400	19.6	11.1	3600.00	19.8	20.30	20.0	18.80	17.2	2.17	248.30	18.0	4.24	54.28	17.1	1.27	5.62	19.37
75	14.4	9.5	3600.00	14.7	4.70	14.4	6.00	13.1	3.94	84.87	13.5	3.66	18.09	13.0	2.01	1.96	7.38
25	24.8	12.1	3600.00	25.0	21.60	24.8	28.40	21.9	3.91	543.42	22.8	4.84	185.79	21.9	1.68	21.02	60.82
500	21.4	8.8	3600.00	21.4	20.30	21.4	33.20	18.8	3.04	532.96	19.7	3.32	81.34	18.6	2.69	13.84	44.05
75	12.7	7.6	3600.00	12.7	32.40	15.0	28.20	11.9	6.03	129.08	12.6	4.06	39.27	11.7	3.63	4.20	15.30
Просек	15.87	11.69	2937.65	16.07	7.77	16.33	9.06	15.29	2.81	143.88	15.66	3.32	40.97	15.21	1.51	4.07	13.31

*на Intel Core 2 Duo са 3 GHz ианд 3.25 GB RAM-а

Из резултата приказаних у Табели 4.1, може се видети да iBVNS скоро тренутно достиже оптимална решења за свих 15 група инстанци из скупа 1 (0.06s у просеку) и да је његово укупно време рада такође веома кратко (0.41s у просеку). Хеуристике GH1 и GH2 су постигле оптимална решења само за 4 и 5 (од 15) група GRLP инстанци из скупа 1, респективно. Методе BVNS и SVNS су постигле оптимална решења за 12 (од 15) група, а обе методе су потрошиле више времена у односу на iBVNS да дођу до најбољих решења (9.16s BVNS за и 7.88s за SVNS у просеку).

У случају GRLP инстанци из скупа 3 које нису решене до оптималности, предложени iBVNS достиже горње границе које је дао егзактни VnC за 6 група инстанци, док су горње границе добијене VnC-ом побољшане за 7 од 15 група инстанци (видети Табелу 4.2). Обе хеуристике, GH1 и GH2 из рада [34] су показале лоше перформансе на инстанцама из скупа 3. Просечно време извршавања у којем iBVNS први пут даје своје најбоље решење и просечно укупно време извршавања су веома кратки (4.07s и 13.31s, респективно), имајући у виду димензију инстанци из скупа 3. Претходно предложене методе, BVNS и SVNS из рада [134] су показале лошије перформансе у погледу и квалитета решења и

4.4. ЕКСПЕРИМЕНТАЛНИ РЕЗУЛТАТИ

времена извршавања у поређењу са iBVNS-ом.

У Табели 4.3, приказано је поређење iBVNS резултата на највећим GRLP инстанцама са резултатима добијеним GRASP методом коју су предложили Quintana и сар. [147]. Аутори рада [147] су објавили само резултате GRASP методе на највећим GRLP инстанцама. Да би се обезбедило коректно поређење iBVNS-а са GRASP-ом, тј. како би просечно време извршавања било мање за iBVNS, задато је ограничење од 20 секунди на укупно iBVNS време извршавања и постављено је $I_{max} = n/5$. Као што се може видети из Табеле 4.3, предложени iBVNS је побољшао најбоља GRASP решења на свих 6 група GRLP инстанци са $n = 400, 500$ чворова. У просеку, време које је iBVNS методи требало да достигне своје најбоље решење било је 4.87, а укупно време рада iBVNS било је 12.78. Са друге стране, просечно време извршавања GRASP-а [147] је било 16.38 секунди на Intel Core i5 2410M CPU са 2.30 GHz и 8GB RAM-а.

Табела 4.3: Резултати и поређења GRASP-а и iBVNS-а на највећим GRLP инстанцама из скупа 3

Група	GRASP [147]		iBVNS				
	<i>best</i>	<i>t(s)*</i>	<i>best</i>	<i>gap(%)</i>	<i>t(s)</i>	<i>t.tot(s)</i>	
400	25	20.9	19.9	2.68	3.66	13.06	
	50	18.3	17.2	2.29	4.09	11.96	
	75	13.9	4.62	12.9	2.93	1.56	4.88
500	25	23.4	22.2	3.16	8.98	19.04	
	50	19.6	19.19	18.7	5.17	9.13	18.69
	75	12.4	29.82	11.8	4.96	1.83	9.06
Просечно	18.08	16.38	17.12	3.53	4.87	12.78	

*на Intel Core i5 2410M CPU са 2.30 GHz и 8GB RAM-а

Резултати iBVNS на инстанцама из скупа 2 упоређени су са резултатима егзактне VnC методе и две похлепне хеуристике (GH1 и GH2) [34], као и са резултатима SVNS и BVNS метахеуристика [134]. Предложени iBVNS и постојећи SVNS и BVNS [134] су тестирани на истом рачунару. Резултати експеримената на скупу 2 су представљени у Табели 4.4 на исти начин као у Табелама 4.1-4.2. Резултати приказани у Табели 4.4 показују да само iBVNS метода и SVNS из [134] достижу сва оптимална решења која су претходно добијена егзактном VnC методом на свакој групи инстанци у скупу 2, док GH1, GH2, и BVNS дају решења лошијег квалитета. SVNS је био нешто бољи од iBVNS-а у погледу про-

4.4. ЕКСПЕРИМЕНТАЛНИ РЕЗУЛТАТИ

Табела 4.4: Резултати и поређења на WGRLP инстанцама из скупа 2

Група	B&C [34]			GH1 [34]		GH2 [34]		BVNS [134]			SVNS [134]			iBVNS			
	<i>UB</i>	<i>LB</i>	<i>t(s)*</i>	<i>best</i>	<i>t(s)*</i>	<i>best</i>	<i>t(s)*</i>	<i>best</i>	<i>gap(%)</i>	<i>t(s)</i>	<i>best</i>	<i>gap(%)</i>	<i>t(s)</i>	<i>best</i>	<i>gap(%)</i>	<i>t(s)</i>	<i>t.tot(s)</i>
25	21.3	21.3	0.20	21.4	0.00	21.4	0.00	21.3	0.00	2.52	21.3	0.00	2.29	21.3	0.00	0.00	0.36
50	21.3	21.3	0.20	21.4	0.00	21.3	0.00	21.3	0.00	2.93	21.3	0.00	3.81	21.3	0.00	0.01	0.54
75	11.3	11.3	0.10	11.5	0.00	11.5	0.00	11.3	0.00	1.61	11.3	0.00	1.97	11.3	0.00	0.00	0.43
25	28.2	28.2	1.20	28.4	0.00	28.4	0.00	28.2	0.00	3.99	28.2	0.00	2.57	28.2	0.00	0.01	0.76
75	25.7	25.7	2.20	26.1	0.00	25.8	0.00	25.7	0.00	9.92	25.7	0.07	8.82	25.7	0.00	0.04	1.43
75	13.9	13.9	0.40	14.5	0.00	14.2	0.00	13.9	0.00	5.73	13.9	0.00	4.71	13.9	0.33	0.04	0.55
25	35.0	35.0	5.20	35.1	0.00	35.0	0.00	35.0	0.28	15.04	35.0	0.04	15.42	35.0	0.00	0.06	2.28
100	28.6	28.6	15.90	28.8	0.00	28.9	0.20	28.6	0.69	19.06	28.6	0.05	29.03	28.6	0.00	0.14	3.11
75	18.7	18.7	2.30	19.1	0.00	18.9	0.10	18.7	0.11	12.52	18.7	0.19	16.27	18.7	1.02	0.10	1.39
25	42.6	42.6	35.10	42.6	0.10	42.6	0.10	42.7	0.12	19.91	42.6	0.00	27.03	42.6	0.24	0.28	4.10
125	31.6	31.6	124.70	32.5	0.20	32.3	0.50	31.6	0.52	61.33	31.6	0.44	80.92	31.6	0.31	0.67	5.45
75	20.1	20.1	9.60	20.8	0.20	20.9	0.00	20.1	0.75	33.86	20.1	0.47	42.64	20.1	0.87	0.38	3.46
25	41.9	41.9	146.10	42.0	0.10	42.1	0.30	42.2	0.04	15.88	41.9	0.00	16.73	41.9	0.00	0.36	7.58
150	32.3	32.3	562.30	33.3	1.20	32.8	0.60	32.3	0.75	23.28	32.3	0.59	21.25	32.3	0.10	1.10	8.36
75	21.4	21.4	31.20	22.1	0.20	21.9	0.50	21.4	0.87	50.1	21.4	0.64	37.67	21.4	0.27	0.48	3.21
Просек	26.26	26.26	62.45	26.64	0.13	26.53	0.15	26.29	0.28	18.51	26.26	0.17	20.74	26.26	0.21	0.25	2.87

*на Intel Core 2 Duo са 3 GHz и 3.25 GB RAM-а

сечних вредности одступања на свим инстанцама из скупа 2 (0.17% за SVNS и 0.21% за iBVNS). Међутим, у погледу времена извршавања, iBVNS се показао супериорнијим у односу на SVNS и BVNS. Подаци представљени у колони $t(s)$ показују да је просечно време потребно iBVNS методи да постигне најбоље решење било 0.25s, док је SVNS методи било потребно 20.74s, а BVNS методи 18.51s. Чак је и просечно укупно време рада iBVNS-а (2.87s) било неколико пута краће у поређењу са просечним временом SVNS-а и BVNS-а за постизање својих најбољих решења. Хеуристике GH1 и GH2 су биле брзе у проналажењу решења, али је њихов квалитет решења био значајно лошији у поређењу са iBVNS-ом и другим метахеуристикама за WGRLP.

Експериментални резултати за скупове података 4 и 5 представљени су у Табелама 4.5 и 4.6, респективно. За ове WGRLP инстанце великих димензија, до сада нису позната оптимална решења. Стога, Табеле 4.5-4.6 садрже само резултате SVNS-а и BVNS-а [134] и резултате предложеног iBVNS-а. На основу приказаних података, може се закључити да је iBVNS надмашио и SVNS и BVNS у погледу квалитета решења, стабилности и времена рада. Предложени iBVNS је дао најбоља позната решења за свих 15 група инстанци из скупа 4, док су BVNS и SVNS достигли најбоља позната решења само на осмој и трећој групи инстанци, респективно.

Сличан закључак се може извести и за инстанце из скупа 5, где је iBVNS постигао најбоља позната решења на свих 15, а BVNS и SVNS на 9 група инстанци. Мале вредности просечног одступања (0.62% за скуп 4 и 0.06% за скуп 5) ука-

4.4. ЕКСПЕРИМЕНТАЛНИ РЕЗУЛТАТИ

зују на стабилност предложеног iBVNS приступа. Просечно време извршавања за које је iBVNS постигао своје најбоље решење по први пут (18.87s за скуп 4 и 11.61s за скуп 5) је знатно ниже у поређењу са одговарајућим временима извршавања BVNS-а (372.18s за скуп 4 и 217.95s за скуп 5) и SVNS-а (300.64s за скуп 4 и 226.62s за скуп 5). У просеку, укупно iBVNS време извршавања је кратко (67.78s за скуп 4 и 75.90s за скуп 5), што указује на ефикасност предложене метахеуристике при решавању инстанци WGRLP великих димензија.

Табела 4.5: Резултати и поређења на WGRLP инстанцама из скупа 4

Група	SVNS [134]			BVNS [134]			iBVNS			
	<i>best</i>	<i>gap</i> (%)	<i>t</i> (s)	<i>best</i>	<i>gap</i> (%)	<i>t</i> (s)	<i>best</i>	<i>gap</i> (%)	<i>t</i> (s)	<i>t.tot</i> (s)
25	43.6	0.55	28.44	43.6	0.37	33.55	43.6	0.00	2.10	15.85
175 50	34.4	0.92	67.02	34.3	0.64	77.74	34.3	0.22	2.90	12.26
75	21.6	0.68	25.50	21.4	1.13	22.65	21.4	0.28	0.41	3.14
25	48.0	0.49	58.62	48.0	0.40	64.49	48.0	0.05	2.22	20.89
200 50	32.8	0.88	52.09	32.7	1.06	58.89	32.7	0.17	3.19	16.34
75	25.0	1.81	53.50	25.0	0.96	54.07	25.0	0.37	1.85	8.52
25	50.9	0.61	223.31	50.8	0.78	346.04	50.8	0.62	13.98	83.63
300 50	39.9	1.70	314.77	39.9	1.58	448.48	39.7	0.78	10.39	51.78
75	28.9	1.85	202.92	28.9	1.95	282.75	28.7	1.74	5.65	20.43
25	50.9	1.82	614.73	50.6	1.26	674.45	50.1	0.39	40.57	152.13
400 50	40.6	1.98	606.37	40.3	2.65	626.66	40.3	0.85	36.99	121.98
75	29.5	4.20	390.84	29.6	3.26	479.98	29.3	1.12	8.59	34.88
25	54.3	1.57	757.32	54.5	2.20	786.44	53.0	1.05	79.33	219.57
500 50	44.8	2.89	690.11	44.7	4.01	927.09	43.1	1.22	64.93	194.22
75	27.3	3.83	424.06	26.8	4.95	699.36	26.5	0.48	9.93	56.59
Просек	38.17	1.72	300.64	38.07	1.81	372.18	37.77	0.62	18.87	67.48

4.4. ЕКСПЕРИМЕНТАЛНИ РЕЗУЛТАТИ

Табела 4.6: Резултати и поређења на WGRLP инстанцама из скупа 5

Група	SVNS [134]			BVNS [134]			iBVNS			
	<i>best</i>	<i>gap</i> (%)	<i>t</i> (s)	<i>best</i>	<i>gap</i> (%)	<i>t</i> (s)	<i>best</i>	<i>gap</i> (%)	<i>t</i> (s)	<i>t.tot</i> (s)
25	56.1	0.00	17.06	56.1	0.00	14.30	56.1	0.00	1.63	20.37
175 50	37.3	0.44	40.62	37.3	0.34	43.34	37.3	0.03	2.79	17.67
75	19.7	0.07	17.19	19.7	0.07	27.31	19.7	0.00	0.32	4.18
25	64.1	0.00	42.48	64.1	0.10	27.35	64.1	0.00	1.01	29.64
200 50	31.3	0.19	40.17	31.3	0.00	35.42	31.3	0.00	1.51	19.11
75	22.9	0.05	29.99	22.9	0.05	20.83	22.9	0.05	0.46	6.34
25	55.1	0.33	240.05	55.1	0.05	231.45	55.1	0.00	11.06	112.09
300 50	37.8	0.50	170.46	37.8	0.29	244.32	37.8	0.07	7.75	80.29
75	22.8	1.15	173.30	22.9	1.01	148.42	22.8	0.07	4.08	24.54
25	52.6	0.59	384.67	52.6	0.26	290.12	52.3	0.19	32.05	171.07
400 50	34.9	0.78	433.95	34.7	0.25	458.64	34.7	0.04	11.17	136.31
75	21.8	0.62	227.74	21.8	0.47	263.24	21.7	0.00	3.09	36.82
25	58.5	0.92	630.76	58.7	0.66	555.41	58.1	0.16	45.36	224.76
500 50	34.1	1.15	542.47	34.3	0.91	563.93	33.6	0.27	47.05	197.16
75	18.4	0.34	408.37	18.4	0.26	345.23	18.3	0.00	4.83	58.17
Просек	37.83	0.48	226.62	37.85	0.31	217.95	37.72	0.06	11.61	75.90

Глава 5

Двокритеријумски уопштени проблем постављања регенератора у оптичким мрежама

5.1 Опис проблема и преглед релевантне литературе

Код претходно описаних GRLP и WGRLP, у случају ограниченог буџета, није увек могуће остварити комуникацију између свих крајњих корисника путем оптичке мреже. То је управо мотивација за увођење двокритеријумског GRLP модела који узима у обзир и трошкове инсталације (тј. трошкове постављања регенератора) и комуникацију парова крајњих корисника мреже. Како у пракси, сви парови не морају имати исти приоритет, за другу функцију циља узима се збир тежинских коефицијената повезаних парова крајњих чворова. Описану варијанту вишекритеријумског GRLP представили смо у раду [140] и до тада у литератури ова варијанта GRLP није била разматрана.

Нека је S скуп локација за постављање регенератора и нека је $L \subseteq S$ скуп чворова на којима су постављени регенератори. Са T је означен скуп парова крајњих чворова. Прва функција циља f_1 је сума трошкова c_l постављених регенератора из скупа L . Друга функција циља f_2 се добија сабирањем тежина w_{ij} по свим паровима повезаних крајњих чворова $i, j \in T$. За рачунање друге функције циља треба пронаћи који су парови крајњих чворова повезани након постављања регенератора и сабрати њихове тежине. Посматрамо истовремено

два циља: минимизацију функције f_1 и максимизацију функције f_2 .

У литератури постоји неколико радова који се односе на проблем постављања регенератора у оптичким мрежама, који укључују две или више функција циља. Међутим, постојећи радови користе другачије моделе у односу на GRLP и уско су везани за специфичне технологије оптичких мрежа. Поред тога, разматрају само број регенератора који треба минимизовати, али не узимају у обзир разлике у цени постављања регенератора на различитим локацијама. На пример, у раду [29], аутори применом варијанте генетског алгоритма за вишекритеријумску оптимизацију NSGA-II оптимизују две функције циља: вероватноћу блокирања и укупан број регенератора. У раду [32] се оптимизују исте функције циља под претпоставком да је могуће инсталирати више регенератора на једној локацији. Трећи проблем, разматран у [156], укључује трошкове рутирања, број операција регенерације сигнала и број чворова који врше регенерацију. Резултујући математички модел је у [156] решаван програмирањем ограничења (енгл. Constraint Programming).

За решавање разматраног проблема у раду [140] примењена су три позната уопштена еволутивна алгоритма који се користе у вишекритеријумској оптимизацији: NSGA-II [51], SPEA-II [188] и SMS-EMOA [17]. Имплементације алгоритама у раду [140] су прилагођене разматраном проблему.

Модификовани алгоритми NSGA-II, SPEA-II и SMS-EMOA за решавање двокритеријумског GRLP користе бинарни низ за репрезентацију решења, где је број елемената низа једнак броју потенцијалних локација за регенераторе: $r = n(1 - p/100)$, где је n број чворова, а p проценат крајњих чворова (корисника) у конкретном тест примеру. Решења почетне популације се генеришу на случајан начин. За свако решење се на случајан начин бира број k ($0 \leq k \leq r$), а затим се k случајно изабраних битова у бинарној репрезентацији решења поставља на 1. Као оператор селекције се користи бинарна турнирска селекција. За оператор укрштања се користи једнопозиционо укрштање, док се мутација реализује једноставном променом вредности изабраних битова са одређеном вероватноћом.

Основни корак у имплементацији наведених алгоритама је обезбедити ефикасну евалуацију решења. Прва функција циља се једноставно рачуна на основу цена постављених регенератора. За рачунање друге функције циља треба пронаћи који су парови повезани након постављања регенератора и сабрати њихове тежине. За ефикасну евалуацију решења користи се процедура заснована на

претрази графа у ширину, која се покреће из сваког постављеног регенератора као што је описано у претходном Поглављу 4, Алгоритмом 23. Описани кораци евалуације решења приказани су Алгоритмом 27. Осим у модификованим еволутивним алгоритмима NSGA-II, SPEA-II и SMS-EMOA, описана процедура се користи и за ефикасно рачунање функција циља у предложеним алгоритмима заснованим на методи променљивих околина за двокритеријумски GRLP, који су описани у насатавку Поглавља 5.

Алгоритам 27 Евалуација функција циља двокритеријумског GRLP

```

1: Input: solution  $x$ , costs, weights
2: Output:  $f_1, f_2$ 
3:  $visitedRegenerators \leftarrow \emptyset$ 
4:  $f_1 \leftarrow 0$ 
5:  $f_2 \leftarrow 0$ 
6: for  $r$  in  $x.regenerators$  do
7:   if  $r \notin visitedRegenerators$  then
8:      $visited \leftarrow BFS(r)$ ;
9:     for  $pair$  in  $NDCpairs$  do
10:      if  $pair$  is not connected and  $visited(pair.start)$  and  $visited(pair.end)$  then
11:        mark  $pair$  as connected;
12:         $f_2 \leftarrow f_2 + weights[pair]$ 
13:      for  $p$  in  $x.regenerators$  do
14:        if  $visited(p)$  then
15:          add  $p$  to  $visitedRegenerators$ ;
16:       $f_1 \leftarrow f_1 + costs[r]$ 

```

5.2 Вишекритеријумска метода променљивих околина

У овој дисертацији предложено је неколико алгоритама заснованих на методи променљивих околина за решавање двокритеријумског GRLP, MO-BVNS, MO-RVNS, MO-GVNS чија је основна структура иста као и структура наведених алгоритама за решавање двокритеријумског проблема максималног покривања локација, описаних у Поглављу 3. Код вишекритеријумског GRLP нема недопустивих решења, стога број постављених регенератора може бити од 0 до n . Код фазе размрдавања SNAKE инвертује се k битова, где се са једнаком вероватноћом од 0.5 уклања регенератор или поставља нови. Код локалне претраге LOCALSEARCH₁ за побољшање f_1 , решење из околине се добија укла-

њањем једног регенератора. У случају локалне претраге LOCALSEARCH₂ за побољшање функције f_2 , решење из околине се добија постављањем једног регенератора. Обе локалне претраге се покрећу само ако већ нису покретане на датом решењу из фазе размрдавања. Као што је већ поменуто, имплементација варијанти вишекритеријумске VNS методе такође користи Алгоритам 27 за евалуацију решења.

5.3 Експериментални резултати

За експерименте су модификоване инстанце из скупова 2 и 4 из радова [34] и [134]. Димензије инстанце једне групе су одређене бројем чворова n , као и процентом чворова који представљају крајње чворове p . Инстанце су подељене на групе са ознаком $n(r)$, где је $r = n(1-p/100)$, при чему свака група садржи по 10 инстанци. Инстанце су проширене тежинама за сваки пар крајњих чворова, при чему свака тежина узима случајну вредност из скупа 1, 2, 3, 4, 5.

За све алгоритме критеријум заустављања је ограничено време извршавања на $n/10$ секунди. За еволутивне алгоритме параметри су следећи: величина популације је 100, вероватноћа мутације $2/r$, док је вероватноћа укрштања 0.9. За све VNS методе параметар k_{max} је постављен на 3. Имплементирани алгоритми су упоређени на исти начин као у Поглављу 3, помоћу метрика HV , IGD , $EPS+$, C , $Size$. Такође, како прави Парето фронт није познат, и овде су комбиновани Парето фронтови из свих покретања свих шест разматраних алгоритама и коришћени као референтни скуп.

Сваки имплементирани алгоритам је покренут 10 пута за сваку инстанцу из сваке групе. Експерименти су покретани на рачунару са процесором Intel i7-8650U@1.90GHz и 16 GB RAM меморије. Добијени резултати су приказани у Табелама 5.1- 5.12 на следећи начин. За сваки индикатор перформанси (HV , IGD , $EPS+$, C и $Size$), чувају се средња вредност (mean), минимум (min), максимум (max) и медијана (med.) вредности добијених током 10 извршавања посматраног алгоритма на свакој тест инстанци. Стандардна девијација (std) добијених вредности је такође приказана. Просек вредности mean, std, med., min и max статистика за сваки индикатор перформанси је израчунат по групама инстанци. Ове просечне вредности су приказане у Табелама 5.1- 5.12, а најбоље mean вредности су подебљане.

Табела 5.1: Резултати МОЕА алгоритма на МО-GRLP инстанцама скупа 2

	NSGA2				SPEA2				SMS-EMOA						
	HV	IGD	EPS	C	Size	HV	IGD	EPS	C	Size	HV	IGD	EPS	C	Size
mean	0.70993	0.00000	0.00000	1.00000	16.60000	0.70993	0.00004	0.00002	0.99900	16.58000	0.70941	0.00345	0.00803	0.89271	16.46000
std	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00009	0.00004	0.00200	0.04000	0.00033	0.00223	0.00723	0.05319	0.79675
50x13 min	0.70993	0.00000	0.00000	1.00000	16.60000	0.70992	0.00000	0.00000	0.99500	16.50000	0.70871	0.00049	0.00176	0.80073	15.30000
med.	0.70993	0.00000	0.00000	1.00000	16.60000	0.70993	0.00000	0.00000	1.00000	16.60000	0.70954	0.00335	0.00422	0.90108	16.30000
max	0.70993	0.00000	0.00000	1.00000	16.60000	0.70993	0.00022	0.00010	1.00000	16.60000	0.70976	0.00702	0.02018	0.96314	17.60000
mean	0.71347	0.00014	0.00046	0.99643	15.29000	0.71341	0.00042	0.00125	0.98332	15.30000	0.71170	0.00673	0.01176	0.75389	15.32000
std	0.00001	0.00027	0.00078	0.00679	0.03000	0.00011	0.00060	0.00161	0.02595	0.18482	0.00130	0.00341	0.00666	0.09007	0.92072
50x25 min	0.71345	0.00000	0.00000	0.98269	15.20000	0.71320	0.00000	0.00000	0.92980	15.10000	0.70851	0.00256	0.00481	0.58950	13.90000
med.	0.71347	0.00000	0.00000	1.00000	15.30000	0.71347	0.00026	0.00130	0.99412	15.30000	0.71196	0.00617	0.01055	0.76046	15.35000
max	0.71347	0.00067	0.00188	1.00000	15.30000	0.71347	0.00164	0.00426	1.00000	15.60000	0.71303	0.01358	0.02756	0.88489	16.80000
mean	0.58663	0.00000	0.00000	1.00000	9.00000	0.58663	0.00000	0.00000	1.00000	9.00000	0.58567	0.00509	0.01425	0.91048	9.09000
std	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00139	0.00507	0.01201	0.07714	0.56325
50x38 min	0.58663	0.00000	0.00000	1.00000	9.00000	0.58663	0.00000	0.00000	1.00000	9.00000	0.58205	0.00014	0.00074	0.76582	8.40000
med.	0.58663	0.00000	0.00000	1.00000	9.00000	0.58663	0.00000	0.00000	1.00000	9.00000	0.58622	0.00312	0.01102	0.93203	9.15000
max	0.58663	0.00000	0.00000	1.00000	9.00000	0.58663	0.00000	0.00000	1.00000	9.00000	0.58663	0.01472	0.03585	0.99091	9.90000
mean	0.72250	0.00017	0.00024	0.99266	20.56000	0.72251	0.00010	0.00008	0.99546	20.59000	0.72131	0.00489	0.01157	0.75001	20.13000
std	0.00002	0.00023	0.00029	0.01069	0.12000	0.00001	0.00016	0.00015	0.00838	0.07000	0.00109	0.00229	0.00932	0.07932	1.21854
75x19 min	0.72245	0.00000	0.00000	0.97079	20.40000	0.72248	0.00000	0.00000	0.97514	20.50000	0.71885	0.00227	0.00340	0.60884	18.40000
med.	0.72251	0.00008	0.00004	0.99783	20.60000	0.72251	0.00000	0.00000	1.00000	20.60000	0.72157	0.00412	0.00841	0.75478	20.10000
max	0.72251	0.00070	0.00069	1.00000	20.80000	0.72251	0.00041	0.00043	1.00000	20.70000	0.72229	0.00904	0.03075	0.85720	22.30000
mean	0.72608	0.00083	0.00146	0.93061	22.18000	0.72603	0.00095	0.00221	0.92150	22.07000	0.72186	0.00970	0.02150	0.54122	21.03000
std	0.00007	0.00034	0.00075	0.03654	0.30697	0.00011	0.00030	0.00152	0.03423	0.26393	0.00230	0.00378	0.00733	0.09143	1.54512
75x38 min	0.72595	0.00043	0.00053	0.86564	21.80000	0.72583	0.00054	0.00101	0.86936	21.80000	0.71693	0.00439	0.01120	0.39778	18.60000
med.	0.72609	0.00078	0.00131	0.93529	22.20000	0.72606	0.00090	0.00163	0.92904	22.00000	0.72224	0.00902	0.02007	0.54133	21.15000
max	0.72617	0.00147	0.00257	0.97059	22.50000	0.72614	0.00137	0.00488	0.96366	22.50000	0.72457	0.01651	0.03391	0.68374	23.60000
mean	0.63665	0.00037	0.00140	0.98326	12.81000	0.63648	0.00066	0.00351	0.97047	12.80000	0.63204	0.01333	0.03619	0.72295	11.85000
std	0.00006	0.00052	0.00136	0.01739	0.09000	0.00046	0.00105	0.00415	0.03533	0.15583	0.00370	0.00898	0.02917	0.11079	1.01034
75x57 min	0.63649	0.00000	0.00000	0.94672	12.70000	0.63526	0.00003	0.00047	0.88497	12.60000	0.62482	0.00388	0.00760	0.51229	10.00000
med.	0.63668	0.00008	0.00107	0.98819	12.80000	0.63666	0.00036	0.00225	0.98462	12.80000	0.63329	0.01025	0.02409	0.73856	12.00000
max	0.63670	0.00138	0.00400	1.00000	13.00000	0.63668	0.00320	0.01269	0.99375	13.00000	0.63577	0.03159	0.08944	0.86050	13.30000

Табела 5.2: Резултати МОЕА алгоритма на MO-GRLP инстанцама скупа 2 (наставка)

	NSGA2				SPEA2				SMS-EMOA						
	HV	IGD	EPS	C	Size	HV	IGD	EPS	C	Size	HV	IGD	EPS	C	Size
mean	0.73417	0.00066	0.00137	0.94921	28.77000	0.73417	0.00059	0.00139	0.95385	28.89000	0.73209	0.00653	0.01556	0.62875	26.58000
std	0.00002	0.00038	0.00085	0.01643	0.37633	0.00002	0.00038	0.00093	0.01866	0.37269	0.00090	0.00221	0.00690	0.07050	1.52830
100x25 min	0.73413	0.00025	0.00032	0.92589	28.20000	0.73414	0.00011	0.00027	0.92010	28.30000	0.73040	0.00325	0.00712	0.51167	23.90000
med.	0.73417	0.00072	0.00184	0.94941	28.75000	0.73417	0.00065	0.00190	0.95283	28.95000	0.73213	0.00642	0.01499	0.63281	26.70000
max	0.73420	0.00131	0.00238	0.97225	29.40000	0.73421	0.00117	0.00265	0.98042	29.40000	0.73340	0.01068	0.02813	0.73996	28.90000
mean	0.72818	0.00108	0.00396	0.81581	26.45000	0.72822	0.00095	0.00387	0.82689	26.39000	0.72273	0.00990	0.02371	0.43426	24.90000
std	0.00021	0.00058	0.00119	0.05630	0.76870	0.00030	0.00065	0.00204	0.06419	0.72978	0.00266	0.00336	0.00879	0.07954	1.88712
100x50 min	0.72773	0.00047	0.00295	0.71580	25.00000	0.72755	0.00022	0.00191	0.69565	25.30000	0.71757	0.00521	0.01341	0.30175	22.10000
med.	0.72822	0.00092	0.00359	0.82284	26.45000	0.72835	0.00077	0.00313	0.84572	26.50000	0.72354	0.00923	0.02222	0.45035	24.90000
max	0.72839	0.00214	0.00638	0.88162	27.50000	0.72845	0.00214	0.00768	0.90769	27.40000	0.72591	0.01678	0.04413	0.56589	28.30000
mean	0.68593	0.00174	0.00589	0.88974	17.61000	0.68534	0.00244	0.00962	0.83699	17.74000	0.67869	0.01486	0.03686	0.56235	16.63000
std	0.00021	0.00060	0.00260	0.03011	0.42769	0.00065	0.00103	0.00422	0.06823	0.49827	0.00464	0.00682	0.01712	0.12084	1.56305
100x75 min	0.68548	0.00096	0.00330	0.84127	17.00000	0.68448	0.00111	0.00421	0.73842	17.00000	0.66897	0.00603	0.01752	0.34377	13.80000
med.	0.68598	0.00171	0.00541	0.89224	17.55000	0.68543	0.00238	0.01053	0.83165	17.85000	0.67956	0.01413	0.03476	0.56999	16.50000
max	0.68609	0.00293	0.01009	0.92828	18.10000	0.68599	0.00420	0.01455	0.92575	18.30000	0.68384	0.02820	0.07280	0.74349	19.20000
mean	0.75901	0.00141	0.00296	0.81388	37.46000	0.75901	0.00127	0.00336	0.84958	37.86000	0.75721	0.00535	0.01445	0.51710	34.83000
std	0.00013	0.00072	0.00211	0.05425	1.02590	0.00016	0.00076	0.00299	0.05753	1.16202	0.00097	0.00166	0.00511	0.07086	2.00943
125x32 min	0.75875	0.00065	0.00080	0.71677	35.90000	0.75870	0.00022	0.00047	0.74410	36.40000	0.75513	0.00248	0.00686	0.40006	31.20000
med.	0.75904	0.00123	0.00235	0.82255	37.40000	0.75904	0.00114	0.00252	0.85059	37.75000	0.75743	0.00539	0.01415	0.51613	34.85000
max	0.75916	0.00284	0.00714	0.87899	39.10000	0.75920	0.00261	0.00961	0.92813	40.10000	0.75841	0.00835	0.02508	0.64313	38.00000
mean	0.73323	0.00278	0.00633	0.70522	27.92000	0.73333	0.00222	0.00618	0.71906	28.18000	0.72585	0.01200	0.02633	0.38517	26.93000
std	0.00057	0.00099	0.00289	0.06157	0.84926	0.00043	0.00082	0.00242	0.05728	0.86973	0.00316	0.00342	0.00701	0.08608	1.89201
125x63 min	0.73208	0.00142	0.00352	0.59626	26.60000	0.73255	0.00100	0.00284	0.63389	27.00000	0.72025	0.00725	0.01628	0.25142	23.70000
med.	0.73335	0.00269	0.00586	0.71166	27.65000	0.73333	0.00213	0.00595	0.71839	28.15000	0.72635	0.01195	0.02597	0.37574	26.85000
max	0.73382	0.00470	0.01201	0.79498	29.50000	0.73388	0.00351	0.01027	0.80347	29.60000	0.72992	0.01744	0.03691	0.52208	30.20000
mean	0.69547	0.00222	0.00517	0.83170	18.36000	0.69508	0.00273	0.00742	0.80827	18.38000	0.68722	0.01945	0.05240	0.45125	17.00000
std	0.00065	0.00106	0.00331	0.08334	0.51380	0.00106	0.00178	0.00521	0.09702	0.47404	0.00648	0.00922	0.02740	0.10560	1.90486
125x94 min	0.69391	0.00118	0.00143	0.67693	17.60000	0.69314	0.00094	0.00163	0.64564	17.60000	0.67275	0.00707	0.01319	0.26182	13.60000
med.	0.69557	0.00184	0.00482	0.84606	18.35000	0.69544	0.00234	0.00634	0.81897	18.30000	0.68948	0.01708	0.04967	0.46696	17.05000
max	0.69603	0.00453	0.01238	0.93624	19.20000	0.69601	0.00608	0.01562	0.93926	19.00000	0.69395	0.03742	0.09432	0.61693	20.00000

Табела 5.3: Резултати МОЕА алгоритма на MO-GRLP инстанцама скупа 2 (наставак)

	NSGA2				SPEA2				SMS-EMOA						
	HV	IGD	EPS	C	Size	HV	IGD	EPS	C	Size	HV	IGD	EPS	C	Size
mean	0.76919	0.00131	0.00195	0.86057	36.46000	0.76923	0.00102	0.00176	0.88863	37.08000	0.76706	0.00545	0.01302	0.53313	34.91000
std	0.00008	0.00037	0.00055	0.04513	0.85744	0.00005	0.00031	0.00030	0.03290	0.56472	0.00133	0.00215	0.00504	0.09239	1.95818
150x38 min	0.76903	0.00078	0.00154	0.77889	34.90000	0.76911	0.00054	0.00152	0.83002	36.10000	0.76398	0.00244	0.00602	0.38971	31.70000
med.	0.76924	0.00120	0.00188	0.87174	36.60000	0.76925	0.00101	0.00164	0.90189	37.10000	0.76736	0.00545	0.01278	0.51905	35.05000
max	0.76926	0.00188	0.00325	0.92179	37.60000	0.76926	0.00146	0.00238	0.92156	38.00000	0.76851	0.00954	0.02281	0.69256	38.00000
mean	0.74665	0.00203	0.00740	0.68507	30.85000	0.74652	0.00193	0.00746	0.69627	30.97000	0.74173	0.01085	0.03125	0.39963	29.25000
std	0.00089	0.00126	0.00502	0.11627	1.20908	0.00110	0.00137	0.00543	0.12808	1.06334	0.00350	0.00494	0.01746	0.10406	2.35535
150x75 min	0.74503	0.00049	0.00165	0.53176	28.70000	0.74463	0.00038	0.00147	0.51108	29.50000	0.73414	0.00438	0.01200	0.22231	24.70000
med.	0.74690	0.00174	0.00616	0.67232	30.95000	0.74708	0.00147	0.00536	0.71927	31.00000	0.74283	0.01046	0.02956	0.40924	29.65000
max	0.74759	0.00448	0.01512	0.86782	32.60000	0.74759	0.00432	0.01681	0.87957	32.70000	0.74554	0.02092	0.06742	0.56097	32.50000
mean	0.69678	0.00245	0.00698	0.77372	20.03000	0.69646	0.00282	0.00737	0.75480	19.80000	0.68965	0.01288	0.02913	0.43105	20.18000
std	0.00086	0.00154	0.00446	0.08956	0.80432	0.00119	0.00184	0.00540	0.09352	0.89239	0.00359	0.00436	0.00908	0.09621	1.82976
150x113 min	0.69495	0.00038	0.00218	0.62546	18.90000	0.69425	0.00075	0.00251	0.59523	18.50000	0.68291	0.00674	0.01520	0.27086	17.60000
med.	0.69708	0.00219	0.00610	0.79003	19.85000	0.69676	0.00239	0.00584	0.76688	19.90000	0.69013	0.01247	0.02834	0.43086	20.20000
max	0.69753	0.00529	0.01457	0.89995	21.50000	0.69751	0.00605	0.01751	0.88292	21.10000	0.69432	0.02085	0.04589	0.57814	23.70000

Табела 5.4: Резултати MO-VNS алгоритма на MO-GRLP инстанцама скупа 2

MO-RVNS										MO-BVNS					MO-GVNS				
	HV	IGD	EPS	C	Size	HV	IGD	EPS	C	Size	HV	IGD	EPS	C	Size				
mean	0.70993	0.00011	0.00005	0.99750	16.55000	0.70992	0.00003	0.00023	0.99910	16.59000	0.70991	0.00003	0.00029	0.99894	16.59000				
std	0.00000	0.00011	0.00005	0.00250	0.05000	0.00002	0.00009	0.00068	0.00270	0.03000	0.00004	0.00009	0.00088	0.00319	0.03000				
50x13 min	0.70992	0.00000	0.00000	0.99500	16.50000	0.70986	0.00000	0.00000	0.99100	16.50000	0.70980	0.00000	0.00000	0.98936	16.50000				
med.	0.70993	0.00011	0.00005	0.99750	16.55000	0.70993	0.00000	0.00000	1.00000	16.60000	0.70993	0.00000	0.00000	1.00000	16.60000				
max	0.70993	0.00022	0.00010	1.00000	16.60000	0.70993	0.00030	0.00225	1.00000	16.60000	0.70993	0.00031	0.00294	1.00000	16.60000				
mean	0.71339	0.00037	0.00085	0.98404	15.35000	0.71346	0.00000	0.00008	0.99841	15.30000	0.71347	0.00000	0.00005	0.99765	15.30000				
std	0.00007	0.00038	0.00073	0.01624	0.05000	0.00001	0.00001	0.00022	0.00360	0.00000	0.00000	0.00000	0.00006	0.00288	0.00000				
50x25 min	0.71332	0.00000	0.00000	0.96392	15.30000	0.71343	0.00000	0.00000	0.98995	15.30000	0.71346	0.00000	0.00000	0.99412	15.30000				
med.	0.71339	0.00045	0.00136	0.98235	15.35000	0.71347	0.00000	0.00000	1.00000	15.30000	0.71347	0.00000	0.00000	1.00000	15.30000				
max	0.71347	0.00090	0.00153	1.00000	15.40000	0.71347	0.00003	0.00070	1.00000	15.30000	0.71347	0.00001	0.00011	1.00000	15.30000				
mean	0.58663	0.00000	0.00000	1.00000	9.00000	0.58663	0.00000	0.00000	1.00000	9.00000	0.58663	0.00000	0.00000	1.00000	9.00000				
std	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000				
50x38 min	0.58663	0.00000	0.00000	1.00000	9.00000	0.58663	0.00000	0.00000	1.00000	9.00000	0.58663	0.00000	0.00000	1.00000	9.00000				
med.	0.58663	0.00000	0.00000	1.00000	9.00000	0.58663	0.00000	0.00000	1.00000	9.00000	0.58663	0.00000	0.00000	1.00000	9.00000				
max	0.58663	0.00000	0.00000	1.00000	9.00000	0.58663	0.00000	0.00000	1.00000	9.00000	0.58663	0.00000	0.00000	1.00000	9.00000				
mean	0.72250	0.00006	0.00015	0.99696	20.59000	0.72251	0.00002	0.00006	0.99925	20.58000	0.72250	0.00001	0.00038	0.99783	20.64000				
std	0.00001	0.00011	0.00023	0.00646	0.05385	0.00001	0.00007	0.00018	0.00224	0.06000	0.00003	0.00004	0.00112	0.00590	0.12000				
75x19 min	0.72248	0.00000	0.00000	0.97826	20.50000	0.72249	0.00000	0.00000	0.99253	20.40000	0.72240	0.00000	0.00000	0.98139	20.60000				
med.	0.72251	0.00000	0.00000	1.00000	20.60000	0.72251	0.00000	0.00000	1.00000	20.60000	0.72251	0.00000	0.00000	1.00000	20.60000				
max	0.72251	0.00033	0.00055	1.00000	20.70000	0.72251	0.00023	0.00060	1.00000	20.60000	0.72251	0.00013	0.00373	1.00000	21.00000				
mean	0.72608	0.00073	0.00137	0.93347	22.27000	0.72612	0.00056	0.00109	0.95314	22.30000	0.72612	0.00058	0.00134	0.95330	22.31000				
std	0.00005	0.00018	0.00067	0.02709	0.16506	0.00005	0.00015	0.00066	0.02397	0.12482	0.00007	0.00018	0.00139	0.02090	0.14284				
75x38 min	0.72603	0.00043	0.00044	0.90603	22.10000	0.72604	0.00041	0.00027	0.91463	22.10000	0.72599	0.00027	0.00022	0.92796	22.10000				
med.	0.72605	0.00080	0.00150	0.92170	22.30000	0.72612	0.00058	0.00120	0.96082	22.30000	0.72612	0.00058	0.00120	0.95882	22.30000				
max	0.72617	0.00092	0.00230	0.97052	22.50000	0.72618	0.00081	0.00192	0.98059	22.40000	0.72618	0.00086	0.00442	0.98089	22.50000				
mean	0.63669	0.00020	0.00103	0.99500	12.80000	0.63669	0.00017	0.00090	0.99445	12.81000	0.63669	0.00014	0.00074	0.99611	12.80000				
std	0.00005	0.00020	0.00120	0.00772	0.00000	0.00003	0.00019	0.00113	0.00923	0.03000	0.00005	0.00020	0.00120	0.00772	0.00000				
75x57 min	0.63653	0.00000	0.00000	0.97778	12.80000	0.63663	0.00000	0.00000	0.97391	12.80000	0.63653	0.00000	0.00000	0.97778	12.80000				
med.	0.63670	0.00031	0.00143	0.99444	12.80000	0.63670	0.00015	0.00071	0.99722	12.80000	0.63670	0.00000	0.00000	1.00000	12.80000				
max	0.63670	0.00048	0.00311	1.00000	12.80000	0.63670	0.00042	0.00259	1.00000	12.90000	0.63670	0.00048	0.00311	1.00000	12.80000				

Табела 5.5: Резултати MO-VNS алгоритма на MO-GRLP инстанцама скупа 2 (наставак)

	MO-RVNS					MO-BVNS					MO-GVNS				
	HV	IGD	EPS	C	Size	HV	IGD	EPS	C	Size	HV	IGD	EPS	C	Size
mean	0.73417	0.00031	0.00060	0.96919	29.00000	0.73418	0.00024	0.00050	0.97003	29.00000	0.73417	0.00022	0.00066	0.97408	29.04000
std	0.00001	0.00034	0.00062	0.01547	0.31596	0.00002	0.00018	0.00035	0.01491	0.30353	0.00008	0.00022	0.00123	0.01896	0.34830
100x25 min	0.73416	0.00004	0.00019	0.94140	28.50000	0.73415	0.00004	0.00011	0.94679	28.40000	0.73398	0.00000	0.00001	0.94441	28.50000
med.	0.73417	0.00024	0.00037	0.97181	29.00000	0.73417	0.00021	0.00045	0.96878	29.05000	0.73419	0.00018	0.00031	0.97160	29.00000
max	0.73420	0.00096	0.00179	0.98571	29.30000	0.73421	0.00059	0.00123	0.99643	29.30000	0.73421	0.00068	0.00399	0.99778	29.60000
mean	0.72839	0.00046	0.00258	0.88915	26.68000	0.72840	0.00052	0.00283	0.87422	26.72000	0.72835	0.00050	0.00322	0.87019	26.74000
std	0.00021	0.00039	0.00155	0.05298	0.73390	0.00013	0.00038	0.00148	0.05167	0.73802	0.00019	0.00044	0.00210	0.05944	0.80143
100x50 min	0.72791	0.00010	0.00080	0.78735	25.80000	0.72814	0.00006	0.00123	0.77916	25.70000	0.72798	0.00007	0.00101	0.75741	25.80000
med.	0.72846	0.00043	0.00210	0.89865	26.30000	0.72842	0.00047	0.00261	0.87737	26.65000	0.72841	0.00034	0.00261	0.87449	26.60000
max	0.72857	0.00137	0.00578	0.94782	27.80000	0.72855	0.00138	0.00574	0.94366	27.80000	0.72854	0.00142	0.00723	0.94380	28.10000
mean	0.68606	0.00079	0.00380	0.93555	17.56000	0.68604	0.00102	0.00413	0.94079	17.35000	0.68613	0.00085	0.00287	0.94574	17.48000
std	0.00034	0.00073	0.00365	0.04645	0.32963	0.00014	0.00027	0.00228	0.02513	0.17216	0.00013	0.00054	0.00171	0.03074	0.26748
100x75 min	0.68522	0.00029	0.00151	0.82741	17.30000	0.68579	0.00064	0.00167	0.89429	17.10000	0.68594	0.00002	0.00043	0.90230	17.30000
med.	0.68621	0.00037	0.00224	0.94702	17.55000	0.68607	0.00100	0.00398	0.94049	17.35000	0.68611	0.00097	0.00330	0.94192	17.35000
max	0.68626	0.00235	0.01209	0.97121	18.10000	0.68618	0.00145	0.00739	0.97222	17.60000	0.68630	0.00150	0.00531	0.99444	17.90000
mean	0.75910	0.00057	0.00223	0.91768	38.76000	0.75892	0.00106	0.00375	0.81574	37.92000	0.75900	0.00091	0.00294	0.83848	37.99000
std	0.00008	0.00051	0.00165	0.03713	0.80192	0.00023	0.00053	0.00306	0.05281	0.95483	0.00016	0.00050	0.00198	0.05298	1.18912
125x32 min	0.75901	0.00004	0.00035	0.86313	37.50000	0.75844	0.00037	0.00095	0.70831	36.60000	0.75869	0.00026	0.00065	0.73765	36.20000
med.	0.75908	0.00040	0.00166	0.92393	38.85000	0.75899	0.00099	0.00304	0.82388	38.00000	0.75907	0.00075	0.00232	0.85769	38.05000
max	0.75921	0.00144	0.00447	0.96783	40.10000	0.75913	0.00197	0.00986	0.88431	39.60000	0.75916	0.00168	0.00662	0.90272	39.60000
mean	0.73343	0.00174	0.00569	0.76280	28.40000	0.73315	0.00224	0.00681	0.69497	28.48000	0.73318	0.00194	0.00619	0.70331	28.80000
std	0.00056	0.00089	0.00272	0.06758	0.81241	0.00065	0.00100	0.00309	0.07772	1.16581	0.00053	0.00084	0.00271	0.07782	0.98821
125x63 min	0.73232	0.00058	0.00272	0.64970	27.20000	0.73202	0.00087	0.00350	0.57451	26.70000	0.73218	0.00085	0.00304	0.56822	27.20000
med.	0.73342	0.00166	0.00577	0.75164	28.40000	0.73322	0.00218	0.00607	0.69339	28.30000	0.73327	0.00190	0.00556	0.70817	28.70000
max	0.73401	0.00335	0.01092	0.85572	29.90000	0.73386	0.00392	0.01236	0.81967	30.60000	0.73377	0.00359	0.01165	0.81617	30.40000
mean	0.69574	0.00158	0.00351	0.89906	18.20000	0.69572	0.00192	0.00477	0.89132	18.33000	0.69565	0.00184	0.00424	0.88959	18.25000
std	0.00042	0.00092	0.00203	0.05929	0.34865	0.00047	0.00101	0.00321	0.06573	0.32607	0.00049	0.00114	0.00305	0.07522	0.39325
125x94 min	0.69491	0.00014	0.00015	0.79301	17.70000	0.69471	0.00060	0.00095	0.77096	17.90000	0.69478	0.00042	0.00052	0.76039	17.70000
med.	0.69592	0.00163	0.00364	0.90808	18.25000	0.69590	0.00179	0.00366	0.90944	18.25000	0.69587	0.00156	0.00399	0.90400	18.20000
max	0.69608	0.00297	0.00618	0.97786	18.70000	0.69608	0.00344	0.00951	0.97540	18.80000	0.69607	0.00372	0.00965	0.97810	18.90000

Табела 5.6: Резултати MO-VNS алгоритма на MO-GRLP инстанцама скупа 2 (наставак)

	MO-RVNS				MO-BVNS				MO-GVNS						
	HV	IGD	EPS	C	Size	HV	IGD	EPS	C	Size	HV	IGD	EPS	C	Size
mean	0.76915	0.00074	0.00174	0.91267	37.03000	0.76921	0.00080	0.00177	0.87291	36.63000	0.76920	0.00079	0.00158	0.88082	36.88000
std	0.00009	0.00043	0.00067	0.03916	0.40596	0.00005	0.00046	0.00120	0.04464	0.93397	0.00008	0.00047	0.00132	0.04527	0.55252
150x38 min	0.76897	0.00017	0.00100	0.84404	36.20000	0.76912	0.00019	0.00065	0.79607	35.20000	0.76901	0.00023	0.00065	0.78869	35.90000
med.	0.76916	0.00065	0.00158	0.91452	37.10000	0.76922	0.00079	0.00174	0.87057	36.60000	0.76924	0.00064	0.00102	0.89072	36.95000
max	0.76929	0.00143	0.00318	0.97050	37.50000	0.76926	0.00171	0.00407	0.93874	38.20000	0.76927	0.00163	0.00438	0.94166	37.80000
mean	0.74702	0.00133	0.00594	0.76939	30.80000	0.74714	0.00132	0.00503	0.75916	30.59000	0.74684	0.00163	0.00641	0.72250	30.52000
std	0.00067	0.00086	0.00475	0.10440	0.85167	0.00058	0.00093	0.00409	0.09016	0.97383	0.00056	0.00087	0.00378	0.07815	0.94352
150x75 min	0.74561	0.00025	0.00118	0.57455	29.40000	0.74582	0.00033	0.00140	0.58788	29.30000	0.74593	0.00038	0.00212	0.61169	28.80000
med.	0.74717	0.00121	0.00511	0.77364	30.85000	0.74737	0.00116	0.00358	0.77618	30.60000	0.74699	0.00154	0.00537	0.72568	30.45000
max	0.74763	0.00302	0.01378	0.89773	32.40000	0.74758	0.00318	0.01382	0.87079	32.60000	0.74754	0.00290	0.01229	0.84558	32.00000
mean	0.69721	0.00126	0.00459	0.84796	19.90000	0.69736	0.00136	0.00428	0.84819	19.78000	0.69740	0.00109	0.00344	0.87765	19.77000
std	0.00056	0.00148	0.00420	0.07951	0.71792	0.00023	0.00122	0.00247	0.06667	0.54220	0.00048	0.00137	0.00331	0.08083	0.62249
150x113 min	0.69614	0.00021	0.00106	0.70400	18.50000	0.69703	0.00009	0.00114	0.74628	19.10000	0.69622	0.00003	0.00038	0.71659	18.70000
med.	0.69742	0.00072	0.00351	0.86360	19.95000	0.69738	0.00095	0.00378	0.85430	19.65000	0.69749	0.00079	0.00316	0.87526	19.80000
max	0.69767	0.00452	0.01316	0.93721	21.00000	0.69769	0.00348	0.00817	0.93761	20.70000	0.69774	0.00402	0.01059	0.97464	20.80000

Табела 5.7: Резултати МОЕА алгоритма на МО-GRLP инстанцама скупа 4

	NSGA2					SPEA2					SMS-EMOA				
	HV	IGD	EPS	C	Size	HV	IGD	EPS	C	Size	HV	IGD	EPS	C	Size
mean	0.78314	0.00285	0.00963	0.59697	38.64000	0.78323	0.00239	0.00916	0.63562	38.90000	0.77893	0.00878	0.02176	0.31548	36.64000
std	0.00048	0.00088	0.00283	0.08241	1.63329	0.00045	0.00096	0.00320	0.08256	1.58377	0.00198	0.00249	0.00435	0.07643	2.48451
175x44 min	0.78210	0.00143	0.00440	0.44616	36.00000	0.78238	0.00097	0.00429	0.49925	36.20000	0.77486	0.00569	0.01419	0.20283	32.30000
med.	0.78327	0.00289	0.00948	0.60548	38.50000	0.78334	0.00225	0.00893	0.64806	39.10000	0.77928	0.00841	0.02229	0.29918	36.75000
max	0.78369	0.00423	0.01396	0.71241	41.60000	0.78373	0.00419	0.01388	0.75583	41.80000	0.78156	0.01349	0.02839	0.45276	40.40000
mean	0.74442	0.00315	0.00932	0.55370	33.60000	0.74429	0.00281	0.00938	0.54865	33.53000	0.73354	0.01674	0.03229	0.24999	31.19000
std	0.00078	0.00175	0.00309	0.07317	1.42594	0.00079	0.00118	0.00276	0.07921	1.27590	0.00422	0.00423	0.00739	0.05930	2.58668
175x88 min	0.74263	0.00127	0.00549	0.42599	31.20000	0.74264	0.00120	0.00578	0.40541	31.50000	0.72592	0.00934	0.02025	0.17755	26.90000
med.	0.74462	0.00258	0.00903	0.54790	33.75000	0.74444	0.00258	0.00900	0.55035	33.50000	0.73354	0.01708	0.03248	0.23764	31.30000
max	0.74519	0.00627	0.01546	0.66446	35.80000	0.74529	0.00516	0.01426	0.66039	35.50000	0.73998	0.02379	0.04434	0.36352	35.50000
mean	0.72075	0.00341	0.00600	0.74389	18.95000	0.72002	0.00364	0.00786	0.71876	19.29000	0.70353	0.03064	0.08246	0.30263	17.38000
std	0.00061	0.00150	0.00302	0.08268	0.76617	0.00148	0.00229	0.00556	0.12764	0.99447	0.01369	0.01787	0.05060	0.14336	2.31685
175x132 min	0.71935	0.00127	0.00206	0.58189	17.90000	0.71718	0.00105	0.00167	0.49240	18.00000	0.67120	0.01036	0.02315	0.06048	13.10000
med.	0.72096	0.00322	0.00613	0.76930	18.85000	0.72033	0.00295	0.00824	0.72993	19.40000	0.70783	0.02605	0.06760	0.31500	17.60000
max	0.72130	0.00581	0.01147	0.84236	20.10000	0.72145	0.00780	0.01631	0.88305	21.00000	0.71638	0.06748	0.18173	0.50985	21.10000
mean	0.78659	0.00288	0.00867	0.49443	42.42000	0.78670	0.00254	0.01016	0.53652	43.32000	0.78291	0.00863	0.02044	0.28734	40.11000
std	0.00048	0.00088	0.00301	0.06597	1.86389	0.00057	0.00104	0.00440	0.07809	1.66900	0.00150	0.00201	0.00447	0.05285	2.68254
200x50 min	0.78575	0.00162	0.00512	0.38012	39.20000	0.78546	0.00141	0.00450	0.41206	40.50000	0.77993	0.00605	0.01394	0.20936	35.80000
med.	0.78666	0.00281	0.00807	0.49627	42.55000	0.78684	0.00225	0.00959	0.53776	43.25000	0.78319	0.00825	0.01991	0.28376	39.75000
max	0.78718	0.00427	0.01380	0.59517	45.20000	0.78731	0.00486	0.01761	0.65700	46.00000	0.78484	0.01269	0.02789	0.37395	44.20000
mean	0.73782	0.00488	0.01321	0.50502	30.93000	0.73784	0.00435	0.01261	0.49753	31.49000	0.72695	0.01763	0.03375	0.25310	30.08000
std	0.00087	0.00152	0.00397	0.05962	1.79609	0.00094	0.00139	0.00359	0.07638	1.34972	0.00443	0.00403	0.00757	0.05945	2.41740
200x100 min	0.73636	0.00234	0.00713	0.41265	27.90000	0.73646	0.00197	0.00681	0.38608	29.40000	0.71864	0.01114	0.02185	0.15898	26.20000
med.	0.73794	0.00477	0.01270	0.50747	30.70000	0.73785	0.00432	0.01226	0.48878	31.50000	0.72736	0.01716	0.03348	0.25308	30.10000
max	0.73908	0.00781	0.01994	0.60658	34.10000	0.73941	0.00621	0.01840	0.63150	33.50000	0.73371	0.02554	0.04580	0.36280	34.20000
mean	0.72669	0.00414	0.01273	0.58160	23.70000	0.72665	0.00386	0.01291	0.56318	23.92000	0.70550	0.03099	0.08200	0.21216	21.27000
std	0.00202	0.00287	0.00873	0.10478	1.12043	0.00185	0.00221	0.00664	0.11223	1.00841	0.01663	0.01490	0.04094	0.12015	2.53418
200x150 min	0.72356	0.00102	0.00338	0.43941	22.20000	0.72356	0.00132	0.00453	0.41573	22.10000	0.66993	0.01203	0.03288	0.03609	17.10000
med.	0.72772	0.00277	0.00923	0.59114	23.70000	0.72678	0.00335	0.01320	0.54650	24.00000	0.71108	0.02776	0.06929	0.20688	20.90000
max	0.72862	0.00857	0.02444	0.73635	25.60000	0.72868	0.00754	0.02341	0.73777	25.20000	0.72368	0.06021	0.16376	0.40619	25.80000

Табела 5.8: Резултати МОЕА алгоритма на MO-GRLP инстанцама скупа 4 (наставка)

	NSGA2					SPEA2					SMS-EMOA				
	HV	IGD	EPS	C	Size	HV	IGD	EPS	C	Size	HV	IGD	EPS	C	Size
mean	0.80699	0.00653	0.01371	0.28041	44.21000	0.80724	0.00654	0.01659	0.28047	45.53000	0.80263	0.01206	0.02786	0.15475	42.85000
std	0.00108	0.00155	0.00385	0.05103	2.72952	0.00143	0.00208	0.00743	0.07370	2.60770	0.00234	0.00257	0.00776	0.05623	3.13839
300x75 min	0.80530	0.00414	0.00749	0.20308	40.20000	0.80477	0.00349	0.00724	0.15866	41.20000	0.79793	0.00783	0.01709	0.07701	38.10000
med.	0.80701	0.00642	0.01364	0.27978	44.05000	0.80719	0.00636	0.01465	0.27492	45.80000	0.80288	0.01224	0.02757	0.15663	42.60000
max	0.80868	0.00914	0.02096	0.36682	49.30000	0.80924	0.01001	0.02863	0.39531	49.30000	0.80588	0.01656	0.04045	0.24516	48.40000
mean	0.77870	0.00794	0.02077	0.25294	38.28000	0.77850	0.00789	0.02280	0.23414	39.41000	0.76276	0.02575	0.07266	0.07718	34.38000
std	0.00220	0.00256	0.00789	0.06279	2.40827	0.00296	0.00370	0.01133	0.08094	2.45225	0.01204	0.01065	0.04052	0.06342	3.89254
300x150 min	0.77364	0.00519	0.01259	0.15870	34.80000	0.77181	0.00351	0.01000	0.10582	35.60000	0.73765	0.01181	0.02684	0.00000	26.90000
med.	0.77925	0.00727	0.01875	0.25097	38.15000	0.77903	0.00683	0.02103	0.23752	39.40000	0.76514	0.02452	0.06184	0.06530	34.85000
max	0.78110	0.01388	0.03810	0.36191	42.50000	0.78181	0.01565	0.04756	0.36777	43.60000	0.77674	0.04641	0.15269	0.19513	39.80000
mean	0.75544	0.00384	0.01122	0.46280	28.79000	0.75418	0.00521	0.01516	0.41405	29.07000	0.74003	0.02453	0.05179	0.16368	27.17000
std	0.00147	0.00188	0.00473	0.08405	1.66976	0.00283	0.00306	0.00790	0.12744	1.64007	0.00923	0.01070	0.02407	0.09481	3.13784
300x225 min	0.75207	0.00151	0.00487	0.33291	26.10000	0.74875	0.00156	0.00578	0.25684	26.30000	0.72112	0.00884	0.02397	0.01107	21.50000
med.	0.75573	0.00359	0.01138	0.45034	28.80000	0.75535	0.00447	0.01449	0.42224	29.15000	0.74156	0.02229	0.04660	0.15235	27.55000
max	0.75698	0.00773	0.01900	0.61252	31.90000	0.75696	0.01115	0.02729	0.60275	31.50000	0.75243	0.04521	0.09402	0.32664	32.10000
mean	0.78619	0.00816	0.01650	0.24240	41.38000	0.78681	0.00699	0.01600	0.24573	43.68000	0.78175	0.01250	0.02621	0.18257	42.15000
std	0.00138	0.00189	0.00421	0.05375	2.88669	0.00147	0.00184	0.00352	0.04746	2.57126	0.00259	0.00231	0.00534	0.03977	2.84084
400x100 min	0.78369	0.00512	0.00990	0.17405	36.50000	0.78402	0.00417	0.01080	0.17467	39.10000	0.77739	0.00882	0.01696	0.11883	37.60000
med.	0.78643	0.00810	0.01717	0.23551	41.45000	0.78699	0.00679	0.01587	0.24430	44.00000	0.78194	0.01252	0.02619	0.18048	41.95000
max	0.78817	0.01161	0.02324	0.34104	46.20000	0.78867	0.01015	0.02234	0.32486	47.50000	0.78608	0.01588	0.03367	0.24286	47.20000
mean	0.77380	0.00961	0.01919	0.23266	36.98000	0.77410	0.00839	0.01836	0.24840	39.07000	0.76506	0.02001	0.03807	0.10613	34.95000
std	0.00229	0.00306	0.00529	0.06560	2.69182	0.00228	0.00257	0.00477	0.06974	2.54139	0.00481	0.00450	0.01269	0.04736	3.86753
400x200 min	0.76914	0.00551	0.01246	0.12998	32.30000	0.76994	0.00510	0.01251	0.14169	34.90000	0.75596	0.01274	0.02320	0.03052	28.70000
med.	0.77419	0.00924	0.01832	0.22746	37.05000	0.77477	0.00777	0.01772	0.25024	39.05000	0.76510	0.01973	0.03571	0.10992	35.05000
max	0.77685	0.01587	0.02903	0.34325	41.00000	0.77669	0.01308	0.02696	0.34476	43.30000	0.77257	0.02831	0.06337	0.18114	41.60000
mean	0.73524	0.00895	0.02162	0.30665	28.92000	0.73452	0.00992	0.02517	0.27679	28.50000	0.69014	0.05122	0.14214	0.06911	23.42000
std	0.00242	0.00314	0.00717	0.08792	2.36470	0.00309	0.00404	0.00999	0.10867	2.39587	0.03072	0.02481	0.07997	0.06241	4.33237
400x300 min	0.73110	0.00430	0.01132	0.18421	25.10000	0.72869	0.00468	0.01481	0.13908	24.90000	0.63021	0.01806	0.03991	0.00000	15.60000
med.	0.73584	0.00880	0.02140	0.31153	28.80000	0.73499	0.00930	0.02216	0.26179	28.70000	0.69425	0.04710	0.13265	0.05372	23.70000
max	0.73856	0.01458	0.03543	0.45551	32.50000	0.73837	0.01683	0.04565	0.45937	32.20000	0.73028	0.09762	0.27797	0.19964	30.00000

Табела 5.9: Резултати МОЕА алгоритма на MO-GRLP инстанцама скупа 4 (наставак)

	NSGA2				SPEA2				SMS-EMOA						
	HV	IGD	EPS	C	Size	HV	IGD	EPS	C	Size	HV	IGD	EPS	C	Size
mean	0.80099	0.00723	0.01385	0.18805	46.33000	0.80156	0.00602	0.01463	0.20913	47.89000	0.79856	0.01013	0.02004	0.13723	45.41000
std	0.00132	0.00154	0.00341	0.06609	3.03250	0.00152	0.00172	0.00406	0.07257	2.91556	0.00230	0.00208	0.00551	0.04849	3.99569
500x125 min	0.79862	0.00474	0.00906	0.10399	41.00000	0.79848	0.00367	0.00903	0.10301	43.00000	0.79426	0.00696	0.01233	0.07636	39.00000
med.	0.80100	0.00722	0.01352	0.17663	46.55000	0.80176	0.00593	0.01482	0.19189	48.10000	0.79847	0.01005	0.01987	0.13211	45.20000
max	0.80312	0.00996	0.02050	0.32948	51.00000	0.80359	0.00923	0.02244	0.31832	52.30000	0.80214	0.01345	0.02981	0.24142	52.70000
mean	0.77629	0.00842	0.01560	0.22886	40.78000	0.77687	0.00701	0.01381	0.22864	41.40000	0.77058	0.01418	0.02571	0.15648	40.44000
std	0.00161	0.00246	0.00394	0.05707	2.91319	0.00143	0.00247	0.00327	0.07196	2.89539	0.00327	0.00297	0.00599	0.04156	3.63273
500x250 min	0.77363	0.00399	0.00906	0.16473	35.70000	0.77474	0.00308	0.00850	0.15622	36.70000	0.76487	0.00978	0.01727	0.09027	34.30000
med.	0.77626	0.00903	0.01540	0.21226	40.90000	0.77674	0.00755	0.01384	0.20542	41.50000	0.77084	0.01430	0.02573	0.15747	40.50000
max	0.77894	0.01185	0.02175	0.34011	45.70000	0.77899	0.01048	0.01837	0.37287	46.00000	0.77561	0.01897	0.03467	0.22322	46.70000
mean	0.69953	0.00799	0.01940	0.35833	26.21000	0.69933	0.00830	0.02212	0.33624	26.50000	0.66889	0.03908	0.10789	0.11280	22.51000
std	0.00171	0.00320	0.00718	0.07155	1.80698	0.00235	0.00322	0.00975	0.08352	1.62999	0.01954	0.01669	0.05436	0.08104	3.57591
500x375 min	0.69658	0.00360	0.00929	0.26083	22.50000	0.69549	0.00334	0.00886	0.22028	24.00000	0.63122	0.01536	0.03165	0.00000	16.40000
med.	0.69935	0.00762	0.01915	0.34365	26.45000	0.69916	0.00867	0.02251	0.32255	26.50000	0.67354	0.03575	0.09355	0.10962	22.55000
max	0.70252	0.01455	0.03357	0.48614	28.70000	0.70268	0.01314	0.03604	0.48544	29.40000	0.69535	0.06931	0.19912	0.25616	28.40000

Табела 5.10: Резултати MO-VNS алгоритма на MO-GRLP инстанцама скупа 4

	MO-RVNS					MO-BVNS					MO-GVNS				
	HV	IGD	EPS	C	Size	HV	IGD	EPS	C	Size	HV	IGD	EPS	C	Size
mean	0.78348	0.00148	0.00686	0.72059	39.66000	0.78183	0.00434	0.01088	0.47889	38.63000	0.78260	0.00312	0.01004	0.55330	39.64000
std	0.00046	0.00081	0.00371	0.10723	1.16523	0.00127	0.00165	0.00439	0.08692	1.99079	0.00097	0.00135	0.00437	0.11521	1.71842
175x44 min	0.78255	0.00045	0.00259	0.52315	38.00000	0.77916	0.00223	0.00514	0.33758	35.20000	0.78061	0.00143	0.00374	0.37328	37.30000
med.	0.78362	0.00133	0.00620	0.73134	39.75000	0.78213	0.00399	0.01028	0.47354	38.60000	0.78288	0.00286	0.00938	0.56161	39.60000
max	0.78391	0.00290	0.01249	0.85279	41.40000	0.78319	0.00732	0.01915	0.62334	42.10000	0.78362	0.00599	0.01712	0.73185	42.50000
mean	0.74494	0.00182	0.00740	0.62741	33.66000	0.74320	0.00417	0.01173	0.50683	33.94000	0.74350	0.00379	0.01128	0.51997	33.69000
std	0.00073	0.00107	0.00332	0.09649	0.99457	0.00114	0.00147	0.00294	0.06054	1.47199	0.00121	0.00166	0.00339	0.08237	1.38811
175x88 min	0.74323	0.00052	0.00318	0.45083	32.30000	0.74106	0.00212	0.00724	0.41707	31.50000	0.74121	0.00161	0.00654	0.38909	31.70000
med.	0.74507	0.00161	0.00696	0.63479	33.60000	0.74334	0.00417	0.01198	0.50603	33.90000	0.74374	0.00356	0.01117	0.52673	33.50000
max	0.74570	0.00409	0.01498	0.77170	35.50000	0.74482	0.00651	0.01583	0.61264	36.40000	0.74513	0.00679	0.01618	0.63824	36.20000
mean	0.72093	0.00258	0.00452	0.80906	18.94000	0.72067	0.00234	0.00598	0.80252	19.24000	0.72076	0.00268	0.00503	0.79984	19.04000
std	0.00061	0.00142	0.00373	0.08905	0.69405	0.00107	0.00193	0.00501	0.13188	0.75748	0.00090	0.00184	0.00411	0.12479	0.83415
175x132 min	0.71978	0.00082	0.00147	0.68684	17.90000	0.71857	0.00028	0.00135	0.56249	18.20000	0.71920	0.00033	0.00157	0.60575	17.90000
med.	0.72120	0.00221	0.00324	0.82503	18.95000	0.72116	0.00186	0.00440	0.83992	19.10000	0.72127	0.00259	0.00273	0.83921	18.95000
max	0.72148	0.00478	0.01216	0.93437	20.00000	0.72153	0.00554	0.01461	0.95489	20.70000	0.72149	0.00540	0.01170	0.95670	20.30000
mean	0.78714	0.00136	0.00638	0.65016	44.79000	0.78349	0.00641	0.01328	0.37167	43.50000	0.78484	0.00525	0.01167	0.40711	44.24000
std	0.00037	0.00057	0.00292	0.09463	1.40286	0.00161	0.00172	0.00352	0.06142	2.60958	0.00169	0.00179	0.00426	0.09137	2.44820
200x50 min	0.78641	0.00051	0.00252	0.48356	42.70000	0.78041	0.00403	0.00886	0.26678	39.30000	0.78148	0.00256	0.00523	0.28467	40.40000
med.	0.78723	0.00130	0.00633	0.65602	44.65000	0.78373	0.00637	0.01281	0.36833	43.65000	0.78506	0.00513	0.01179	0.39243	44.05000
max	0.78760	0.00240	0.01100	0.79520	47.50000	0.78554	0.00965	0.02044	0.46310	47.30000	0.78692	0.00884	0.01930	0.58662	48.40000
mean	0.73867	0.00288	0.01061	0.56939	31.84000	0.73526	0.00727	0.01759	0.42909	32.23000	0.73571	0.00628	0.01708	0.43449	31.83000
std	0.00079	0.00139	0.00419	0.08479	1.35761	0.00204	0.00259	0.00413	0.06529	2.23773	0.00233	0.00272	0.00534	0.09278	2.15778
200x100 min	0.73728	0.00115	0.00523	0.44255	29.60000	0.73146	0.00335	0.00941	0.33940	28.40000	0.73082	0.00278	0.01006	0.26939	28.10000
med.	0.73880	0.00272	0.01074	0.56837	31.75000	0.73532	0.00737	0.01790	0.41573	32.15000	0.73601	0.00603	0.01703	0.43567	32.00000
max	0.73970	0.00558	0.01663	0.70966	34.00000	0.73827	0.01167	0.02328	0.54308	36.00000	0.73866	0.01147	0.02702	0.58164	34.90000
mean	0.72827	0.00183	0.00694	0.70846	23.38000	0.72767	0.00282	0.00883	0.60954	23.78000	0.72746	0.00291	0.00975	0.60755	23.83000
std	0.00070	0.00123	0.00423	0.10080	0.75075	0.00096	0.00146	0.00528	0.09666	1.03040	0.00167	0.00226	0.00860	0.11782	0.97013
200x150 min	0.72691	0.00048	0.00258	0.54021	22.50000	0.72599	0.00094	0.00300	0.46538	22.20000	0.72372	0.00064	0.00276	0.42394	22.60000
med.	0.72844	0.00136	0.00616	0.72052	23.25000	0.72795	0.00244	0.00697	0.60856	23.80000	0.72796	0.00232	0.00742	0.61982	23.80000
max	0.72895	0.00382	0.01393	0.84746	24.70000	0.72872	0.00517	0.01780	0.75945	25.70000	0.72879	0.00763	0.02860	0.76616	25.40000

Табела 5.11: Резултати MO-VNS алгоритма на MO-GRLP инстанцама скупа 4 (наставка)

	MO-RVNS					MO-BVNS					MO-GVNS				
	HV	IGD	EPS	C	Size	HV	IGD	EPS	C	Size	HV	IGD	EPS	C	Size
mean	0.80859	0.00322	0.00917	0.39990	49.06000	0.79384	0.01810	0.03033	0.15869	52.13000	0.79825	0.01437	0.02456	0.18711	54.31000
std	0.00087	0.00110	0.00289	0.09722	1.89162	0.00387	0.00357	0.00626	0.03885	3.67446	0.00517	0.00461	0.00795	0.05795	3.82206
300x75 min	0.80683	0.00171	0.00514	0.23937	45.80000	0.78760	0.01172	0.02094	0.11235	45.80000	0.78742	0.00742	0.01288	0.11250	48.00000
med.	0.80879	0.00302	0.00847	0.39733	49.10000	0.79402	0.01820	0.02965	0.14969	51.70000	0.79922	0.01352	0.02364	0.17898	55.05000
max	0.80981	0.00532	0.01470	0.56758	52.10000	0.80059	0.02373	0.04289	0.23959	58.20000	0.80529	0.02336	0.04117	0.31544	60.00000
mean	0.78113	0.00400	0.01403	0.35801	39.85000	0.76567	0.01911	0.03332	0.15727	45.49000	0.76674	0.01782	0.03323	0.16063	46.17000
std	0.00127	0.00155	0.00552	0.09975	2.27230	0.00425	0.00374	0.00568	0.04746	3.88279	0.00788	0.00646	0.00933	0.04704	4.08907
300x150 min	0.77885	0.00162	0.00602	0.20758	35.60000	0.75882	0.01258	0.02439	0.09178	39.50000	0.75024	0.00914	0.01955	0.09747	39.70000
med.	0.78129	0.00389	0.01373	0.35982	40.15000	0.76554	0.01929	0.03308	0.15367	45.35000	0.76776	0.01695	0.03365	0.15185	46.05000
max	0.78291	0.00685	0.02226	0.54273	43.30000	0.77296	0.02512	0.04336	0.24193	52.60000	0.77691	0.03119	0.04960	0.24632	53.60000
mean	0.75650	0.00242	0.00774	0.56762	28.41000	0.75264	0.00689	0.01583	0.39777	29.12000	0.75279	0.00678	0.01726	0.38488	28.61000
std	0.00106	0.00143	0.00436	0.11777	1.30483	0.00342	0.00373	0.00655	0.10811	1.84888	0.00298	0.00316	0.00662	0.10608	1.76184
300x225 min	0.75408	0.00083	0.00290	0.35734	26.40000	0.74542	0.00255	0.00819	0.22011	26.00000	0.74697	0.00271	0.00837	0.20806	26.20000
med.	0.75676	0.00232	0.00742	0.55726	28.30000	0.75376	0.00572	0.01454	0.41585	29.05000	0.75381	0.00606	0.01576	0.40312	28.40000
max	0.75747	0.00540	0.01668	0.76258	30.50000	0.75629	0.01407	0.02727	0.56192	32.40000	0.75582	0.01249	0.02721	0.53078	31.80000
mean	0.78872	0.00346	0.01148	0.35759	49.28000	0.76912	0.02209	0.03639	0.20115	55.92000	0.76861	0.02247	0.03621	0.20622	61.91000
std	0.00114	0.00136	0.00340	0.09353	2.26080	0.00472	0.00407	0.00762	0.04799	4.38870	0.00643	0.00512	0.00890	0.04846	6.46238
400x100 min	0.78675	0.00186	0.00693	0.22187	45.70000	0.76167	0.01463	0.02416	0.11933	48.80000	0.75693	0.01444	0.02192	0.14542	52.00000
med.	0.78903	0.00301	0.01092	0.35911	49.00000	0.76886	0.02212	0.03701	0.19960	56.10000	0.76986	0.02183	0.03517	0.19469	61.75000
max	0.79003	0.00565	0.01662	0.53090	53.30000	0.77735	0.02829	0.05007	0.29023	63.10000	0.77784	0.03129	0.05058	0.30164	74.20000
mean	0.77634	0.00493	0.01380	0.33335	40.46000	0.75636	0.02411	0.03649	0.19471	47.19000	0.75274	0.02651	0.03988	0.18266	50.03000
std	0.00134	0.00179	0.00372	0.10131	2.31236	0.00468	0.00474	0.00788	0.05049	4.68071	0.00862	0.00709	0.01024	0.04554	4.70277
400x200 min	0.77380	0.00235	0.00829	0.20528	36.30000	0.74826	0.01730	0.02588	0.11547	38.80000	0.73790	0.01644	0.02507	0.11482	41.70000
med.	0.77678	0.00465	0.01361	0.31612	40.45000	0.75724	0.02341	0.03496	0.20276	47.35000	0.75429	0.02530	0.03733	0.17916	50.35000
max	0.77794	0.00830	0.01964	0.53926	44.60000	0.76294	0.03255	0.05057	0.26544	54.20000	0.76523	0.03930	0.05886	0.25625	58.00000
mean	0.78662	0.00711	0.01843	0.37936	27.98000	0.72885	0.01504	0.02891	0.21717	30.36000	0.73048	0.01331	0.02807	0.25331	29.52000
std	0.00208	0.00291	0.00541	0.11886	1.43865	0.00408	0.00434	0.00692	0.08126	2.93156	0.00480	0.00489	0.00884	0.10100	2.71845
400x300 min	0.73304	0.00256	0.00958	0.21984	25.50000	0.72236	0.00830	0.01920	0.13884	25.90000	0.72294	0.00599	0.01543	0.13384	25.10000
med.	0.73670	0.00715	0.01879	0.37768	27.95000	0.72832	0.01560	0.03014	0.17849	30.60000	0.73139	0.01320	0.02581	0.24655	30.00000
max	0.73961	0.01168	0.02673	0.58504	30.20000	0.73492	0.02128	0.04044	0.35820	35.00000	0.73711	0.02067	0.04273	0.41431	33.60000

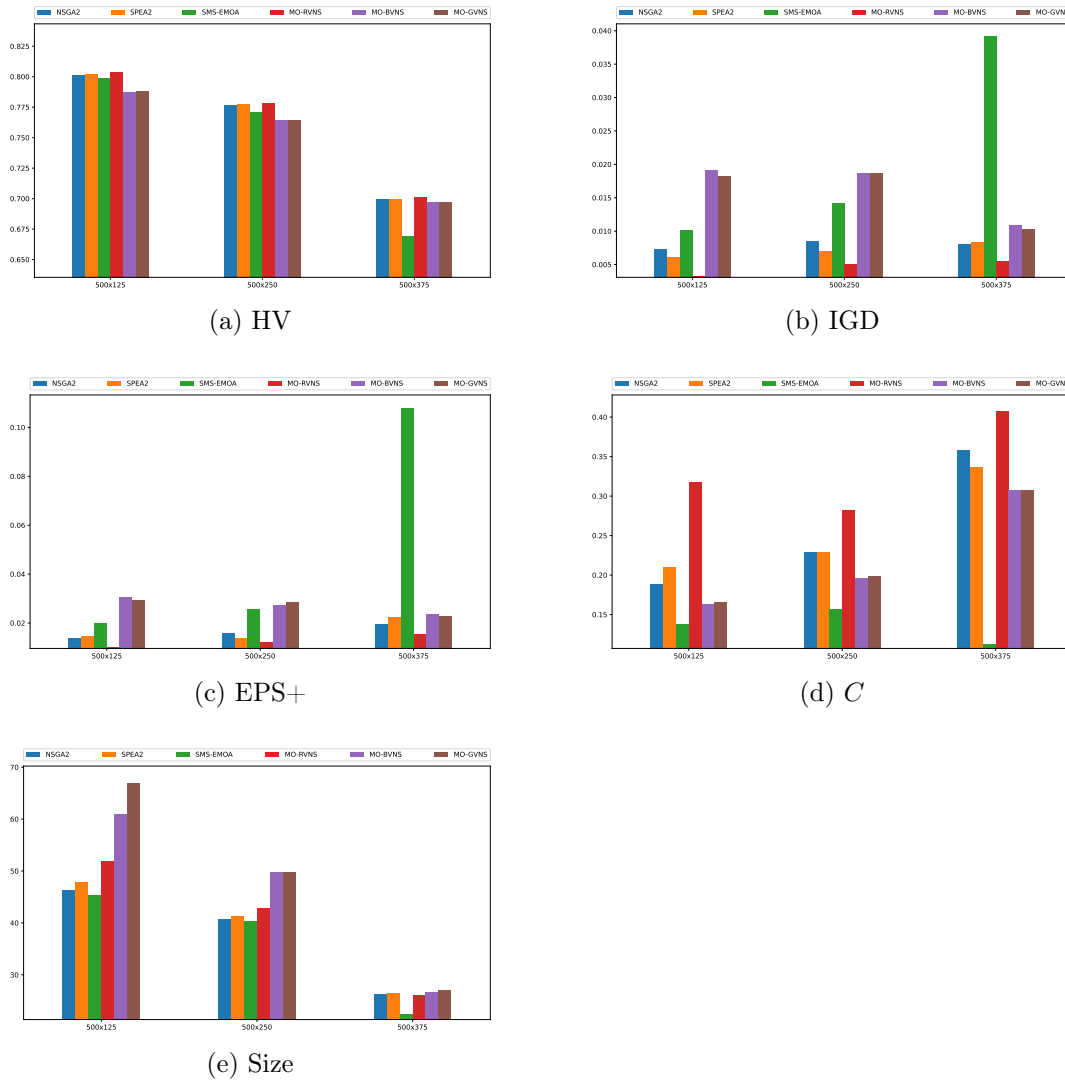
Табела 5.12: Резултати MO-VNS алгоритма на MO-GRLP инстанцама скупа 4 (наставкак)

	MO-RVNS				MO-BVNS				MO-GVNS						
	HV	IGD	EPS	C	Size	HV	IGD	EPS	C	Size	HV	IGD	EPS	C	Size
mean	0.80320	0.00325	0.01011	0.31698	51.87000	0.78689	0.01914	0.03035	0.16249	61.03000	0.78780	0.01819	0.02927	0.16520	66.88000
std	0.00096	0.00123	0.00286	0.10965	2.88659	0.00312	0.00298	0.00496	0.04851	5.38297	0.00534	0.00460	0.00799	0.05624	6.08095
500x125	0.80158	0.00144	0.00550	0.18191	46.40000	0.78094	0.01408	0.02251	0.08889	52.70000	0.77710	0.01180	0.01838	0.08478	57.40000
med.	0.80325	0.00341	0.01055	0.29890	51.85000	0.78668	0.01947	0.03072	0.16141	60.95000	0.78895	0.01736	0.02854	0.15636	66.20000
max	0.80449	0.00512	0.01451	0.51971	56.70000	0.79246	0.02440	0.03836	0.25100	70.50000	0.79495	0.02722	0.04392	0.25598	76.90000
mean	0.77780	0.00498	0.01186	0.28238	42.89000	0.76396	0.01871	0.02713	0.19547	49.78000	0.76386	0.01864	0.02852	0.19771	49.86000
std	0.00118	0.00169	0.00288	0.08684	1.88465	0.00356	0.00358	0.00560	0.04395	3.83368	0.00520	0.00477	0.00728	0.05357	4.92078
500x250	0.77593	0.00250	0.00726	0.18063	39.60000	0.75753	0.01360	0.01971	0.12497	43.70000	0.75550	0.01210	0.01983	0.11273	42.30000
med.	0.77766	0.00493	0.01223	0.26287	42.60000	0.76436	0.01817	0.02672	0.20286	49.95000	0.76456	0.01816	0.02701	0.19708	49.40000
max	0.77950	0.00759	0.01637	0.46279	45.90000	0.76928	0.02450	0.03789	0.25304	57.20000	0.77097	0.02620	0.04156	0.28444	58.60000
mean	0.70096	0.00552	0.01549	0.40767	26.17000	0.69654	0.01090	0.02358	0.30705	26.59000	0.69699	0.01019	0.02255	0.30711	27.05000
std	0.00158	0.00201	0.00525	0.08525	1.34669	0.00271	0.00299	0.00635	0.06421	2.30952	0.00283	0.00321	0.00654	0.06719	2.00182
500x375	0.69802	0.00301	0.00786	0.30723	23.80000	0.69146	0.00615	0.01399	0.21377	23.30000	0.69204	0.00600	0.01318	0.21884	24.20000
med.	0.70102	0.00523	0.01554	0.40810	26.20000	0.69674	0.01066	0.02381	0.30118	26.30000	0.69715	0.01001	0.02200	0.30284	26.85000
max	0.70328	0.00960	0.02359	0.56937	28.00000	0.70076	0.01605	0.03442	0.41665	30.60000	0.70089	0.01594	0.03381	0.41550	30.40000

На основу приказаних резултата може се закључити да варијанта MO-RVNS даје најбоље резултате на инстанцама великих димензија. На инстанцама мањих димензија алгоритми су упоредиви, где се посебно не истиче неки од тестираних.

На слици 5.1 су приказане средње вредности за сваку метрику и сваки алгоритам на инстанцама димензије $n = 500$. Анализом средњих вредности разматраних метрика на овим инстанцама које су највећи изазов за решавање, може се закључити да је MO-RVNS показао најбоље перформансе у односу на остале алгоритме по свим метрикама. Изузетак је величина апроксимативног скупа, где су варијанте MO-BVNS и MO-GVNS биле боље.

5.3. ЕКСПЕРИМЕНТАЛНИ РЕЗУЛТАТИ



Слика 5.1: Средње вредности метрика алгоритама на инстанцама највећих димензија $n = 500$

Закључак

У овој дисертацији су разматрана два локацијска проблема, дефинисане њихове двокритеријумске варијанте и предложене ефикасне метахеуристичке методе за решавање једнокритеријумских и двокритеријумских варијанти ових проблема.

Уведена је нова варијанта локацијског проблема максималног покривања (енгл. Maximal Covering Location Problem, MCLP) са преференцијама корисника и ограниченим буџетом, која до сада није била разматрана у литератури. Дата је математичка формулација ове нове варијанте MCLP проблема и предложене су варијанте методе променљивих околина за њено решавање. Такође, предложена је и двокритеријумска варијанта MO-MCLP, која тражи баланс између тежинске суме потражње покривених корисника (где тежине одговарају преференцијама) и броја непокривених корисника. Развијене су вишекритеријумске варијанте методе променљивих околина за решавање MO-MCLP које су упоређене са уопштеним еволутивним алгоритмима.

Предложена је ефикасна метода променљивих околина за решавање уопште-ног проблема постављања регенератора у оптичким мрежама (енгл. Generalized Regenerator Location Problem, GRLP), као и његове тежинске варијанте WGRLP. Резултати имплементираних методе су упоређени са најбољим из литературе. Поред тога, дефинисана је и двокритеријумска варијанта MO-GRLP, где је цена постављања регенератора једна функција циља, док другу чини сума тежина повезаних парова крајњих чворова. Вишекритеријумска варијанта методе променљивих околина је прилагођена за решавање MO-GRLP и резултати су упоређени са уопштеним еволутивним алгоритмима.

Научни допринос ове дисертације огледа се у следећим резултатима:

- Предложен је нови, унапређени начин решавања MCLP са преференцијама корисника у односу на постојеће алгоритме из литературе.

- Дефинисана је нова варијанта MCLP проблема са преференцијама корисника и ограниченим буџетом.
- Предложене су варијанте методе променљивих околина за решавање нове MCLP варијанте чији су резултати упоређени са егзактним решавачем.
- Дефинисан је нови двокритеријумски MCLP са преференцијама корисника са циљем остваривања баланса између покривене потражње корисника и броја непокривених корисника.
- У циљу решавања предложеног MO-MCLP, развијене су различите варијанте вишекритеријумске методе променљивих околина, чији су резултати упоређени са резултатима постојећих уопштених еволутивних алгоритама за вишекритеријумску оптимизацију.
- Генерисане су нове инстанце великих димензија на основу реалних података за MCLP са преференцијама корисника. Ове тест инстанце су коришћене за евалуацију ефикасности метода за решавање једнокритеријумског и двокритеријумског MCLP.
- Унапређен је начин решавања WGRLP у односу на постојећа решења из литературе.
- Уведен је нови ефикасни начина за проверу допустивости решења и ажурирање вредности функције циља приликом локалне претраге код решавања WGRLP.
- Дефинисан је нови двокритеријумски GRLP, решаван уопштеним еволутивним алгоритмима, као и предложеном вишекритеријумском методом променљивих околина.

Део ових доприноса објављен је у радовима у часописима индексираним на SCI листи [132, 136, 137], као и радовима у зборницима конференција [134, 135, 138–140, 170].

Будући правци истраживања могу да укључе хибридизацију метода за решавање варијанти MCLP и WGRLP са хеуристикама или егзактним методама. Затим, проширивање двокритеријумских проблема новим функцијама циља које би још боље одсликавале ситуације у пракси. Поред тога, могуће је применити предложене вишекритеријумске методе променљивих околина на сличне вишекритеријумске проблеме из литературе.

Библиографија

- [1] Mohammad Amin Adibi, Mostafa Zandieh, and Maghsoud Amiri. Multi-objective scheduling of dynamic job shop using variable neighborhood search. *Expert Systems with Applications*, 37(1):282–287, 2010.
- [2] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network flows: theory, algorithms, and applications*. Prentice Hall, New Jersey, USA, 1993.
- [3] Deniz Aksen, Nuray Piyade, and Necati Aras. The budget constrained r-interdiction median problem with capacity expansion. *Central European Journal of Operations Research*, 18:269–291, 2010.
- [4] Ekatarina Alekseeva and Yury Kochetov. Genetic local search for the p-median problem with client’s preferences. *Дискретный анализ и исследование операций*, 14(1):3–31, 2007.
- [5] Khalil Amine. Multiobjective simulated annealing: Principles and algorithm variants. *Advances in Operations Research*, 2019:1–13, 2019.
- [6] Claus Aranha, Christian L Camacho Villalón, Felipe Campelo, Marco Dorigo, Rubén Ruiz, Marc Sevaux, Kenneth Sörensen, and Thomas Stützle. Metaphor-based metaheuristics, a call for action: the elephant in the room. *Swarm Intelligence*, 16(1):1–6, 2022.
- [7] Ceyhun Araz, Hasan Selim, and Irem Ozkarahan. A fuzzy multi-objective covering-based vehicle location model for emergency services. *Computers & Operations research*, 34(3):705–726, 2007.
- [8] Ceyhun Araz, Hasan Selim, and Irem Ozkarahan. A fuzzy multi-objective covering-based vehicle location model for emergency services. *Computers & Operations Research*, 34(3):705–726, 2007.

- [9] José Elias Claudio Arroyo, Rafael dos Santos Ottoni, and Alcione de Paiva Oliveira. Multi-objective variable neighborhood search algorithms for a single machine scheduling problem with distinct due windows. *Electronic Notes in Theoretical Computer Science*, 281:5–19, 2011.
- [10] Soumen Atta, Priya Ranjan Sinha Mahapatra, and Anirban Mukhopadhyay. Multi-objective uncapacitated facility location problem with customers' preferences: Pareto-based and weighted sum ga-based approaches. *Soft Computing*, 23(23):12347–12362, 2019.
- [11] Soumen Atta, Priya Ranjan Sinha Mahapatra, and Anirban Mukhopadhyay. A multi-objective formulation of maximal covering location problem with customers' preferences: Exploring pareto optimality-based solutions. *Expert Systems with Applications*, 186:115830, 2021.
- [12] Masood A Badri, Amr K Mortagy, and Colonel Ali Alsayed. A multi-objective model for locating fire stations. *European Journal of Operational Research*, 110(2):243–260, 1998.
- [13] Cynthia Barnhart, Ellis L Johnson, George L Nemhauser, Martin WP Savelsbergh, and Pamela H Vance. Branch-and-price: Column generation for solving huge integer programs. *Operations Research*, 46(3):316–329, 1998.
- [14] Sumanta Basu, Megha Sharma, and Partha Sarathi Ghosh. Metaheuristic applications on discrete facility location problems: a survey. *Opsearch*, 52:530–561, 2015.
- [15] Adil Baykasoglu, Stephen Owen, and Nabil Gindy. A taboo search based approach to find the pareto optimal set in multiple objective optimization. *Engineering Optimization*, 31(6):731–748, 1999.
- [16] Oded Berman and Dmitry Krass. The generalized maximal covering location problem. *Computers and Operations Research*, 29(6):563–581, 2002.
- [17] Nicola Beume, Boris Naujoks, and Michael Emmerich. SMS-EMOA: Multiobjective selection based on dominated hypervolume. *European Journal of Operational Research*, 181(3):1653–1669, 2007.
- [18] Hans-Georg Beyer and Hans-Paul Schwefel. Evolution strategies—a comprehensive introduction. *Natural Computing*, 1:3–52, 2002.

- [19] Ranjan Bhattacharya and Susmita Bandyopadhyay. Solving conflicting bi-objective facility location problem by nsga ii evolutionary algorithm. *The International Journal of Advanced Manufacturing Technology*, 51(1-4):397–414, 2010.
- [20] Julian Blank and Kalyanmoy Deb. Pymoo: Multi-objective optimization in python. *IEEE Access*, 8:89497–89509, 2020.
- [21] Aymeric Blot, Marie-Éléonore Kessaci, and Laetitia Jourdan. Survey and unification of local search techniques in metaheuristics for multi-objective combinatorial optimisation. *Journal of Heuristics*, 24(6):853–877, 2018.
- [22] Christian Blum and Andrea Roli. *Metaheuristics*. Springer, 2003.
- [23] Jürgen Branke. *Multiobjective optimization: Interactive and evolutionary approaches*, volume 5252. Springer Science & Business Media, 2008.
- [24] Jack Brimberg, Said Salhi, Raca Todosijević, and Dragan Urošević. Variable neighborhood search: The power of change and simplicity. *Computers & Operations Research*, 155:106221, 2023.
- [25] José-Fernando Camacho-Vallejo, Álvaro Eduardo Cordero-Franco, and Rosa G González-Ramírez. Solving the bilevel facility location problem under preferences by a stackelberg-evolutionary algorithm. *Mathematical Problems in Engineering*, 2014(1):430243, 2014.
- [26] Lázaro Cánovas, Sergio García, Martine Labbé, and Alfredo Marín. A strengthened formulation for the simple plant location problem with order. *Operational Research Letters*, 35(2):141–150, 2007.
- [27] Martha-Selene Casas-Ramírez and José-Fernando Camacho-Vallejo. Solving the p-median bilevel problem with order through a hybrid heuristic. *Applied Soft Computing*, 60:73–86, 2017.
- [28] Martha-Selene Casas-Ramírez, José-Fernando Camacho-Vallejo, Juan A Díaz, and Dolores E Luna. A bi-level maximal covering location problem. *Operational Research: An International Journal*, 20:827–855, 2017.
- [29] Matheus A Cavalcante, Helder A Pereira, Daniel AR Chaves, and Raul C Almeida. Evolutionary multiobjective strategy for regenerator placement in

- elastic optical networks. *IEEE Transactions on Communications*, 66(8):3583–3596, 2018.
- [30] Derya Çelik Türkoğlu and Müjde Erol Genevois. A comparative survey of service facility location problems. *Annals of Operations Research*, 292:399–468, 2020.
- [31] Sunarin Chanta, Maria E Mayorga, and Laura A McLay. Improving emergency service in rural areas: a bi-objective covering location model for EMS systems. *Annals of Operations Research*, 221(1):133–159, 2014.
- [32] Daniel AR Chaves, Caio FCLC Ayres, Renan VB Carvalho, Helder A Pereira, Carmelo JA Bastos-Filho, and Joaquim F Martins-Filho. Multiobjective sparse regeneration placement algorithm in optical networks considering network performance and capex. In *2010 12th International Conference on Transparent Optical Networks*, pages 1–4. IEEE, 2010.
- [33] Chandra Chekuri and Amit Kumar. Maximum coverage problem with group budget constraints and applications. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 72–83. Springer, 2004.
- [34] Si Chen, Ivana Ljubić, and S Raghavan. The generalized regenerator location problem. *INFORMS Journal on Computing*, 27(2):204–220, 2015.
- [35] Si Chen, Ivana Ljubić, and Srinivasacharya Raghavan. The regenerator location problem. *Networks*, 55(3):205–220, 2010.
- [36] Richard Church, John Current, and James Storbeck. A bicriterion maximal covering location formulation which considers the satisfaction of uncovered demand. *Decision Sciences*, 22(1):38–52, 1991.
- [37] Richard Church and Charles R Velle. The maximal covering location problem. *Papers in Regional Science*, 32(1):101–118, 1974.
- [38] Richard L Church. The planar maximal covering location problem. *Journal of Regional Science*, 24(2):185–201, 1984.
- [39] Carlos A Coello Coello, Gary B Lamont, and David A Van Veldhuizen. *Evolutionary algorithms for solving multi-objective problems*. Springer, 2007.

- [40] Reuven Cohen and Liran Katzir. The generalized maximum coverage problem. *Information Processing Letters*, 108(1):15–22, 2008.
- [41] Yann Collette and Patrick Siarry. *Multiobjective optimization: principles and case studies*. Springer Science & Business Media, 2013.
- [42] José Manuel Colmenar, Rafael Martí, and Abraham Duarte. Multi-objective memetic optimization for the bi-objective obnoxious p-median problem. *Knowledge-Based Systems*, 144:88–101, 2018.
- [43] Fabio Colombo, Roberto Cordone, and Guglielmo Lulli. A variable neighborhood search algorithm for the multimode set covering problem. *Journal of Global Optimization*, 63(3):461–480, 2015.
- [44] Fabio Colombo, Roberto Cordone, and Guglielmo Lulli. The multimode covering location problem. *Computers & Operations Research*, 67:25–33, 2016.
- [45] Jean-François Cordeau, Fabio Furini, and Ivana Ljubić. Benders decomposition for very large scale partial set covering and maximal covering location problems. *European Journal of Operational Research*, 275(3):882–896, 2019.
- [46] Kerstin Dächert, Kathrin Klamroth, Renaud Lacour, and Daniel Vanderpooten. Efficient computation of the search region in multi-objective optimization. *European Journal of Operational Research*, 260(3):841–855, 2017.
- [47] Soheil Davari, Mohammad Hossein Fazel Zarandi, and Burhan Turksen. A greedy variable neighborhood search heuristic for the maximal covering location problem with fuzzy coverage radii. *Knowledge-Based Systems*, 41:68–76, 2013.
- [48] Dušan Teodorović Šelmić Milica Davidović, Tatjana. Bee colony optimization part i: The algorithm overview. *Yugoslav Journal of Operations Research*, 25(1), 2015.
- [49] Mansoor Davoodi. k -balanced center location problem: A new multi-objective facility location problem. *Computers & Operations Research*, 105:68–84, 2019.

- [50] Kalyanmoy Deb, Manikanth Mohan, and Shikhar Mishra. Evaluating the ε -domination based multi-objective evolutionary algorithm for a quick computation of pareto-optimal solutions. *Evolutionary Computation*, 13(4):501–525, 2005.
- [51] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and Tamt Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE transactions on Evolutionary Computation*, 6(2):182–197, 2002.
- [52] Juan A Díaz, Dolores E Luna, José-Fernando Camacho-Vallejo, and Martha-Selene 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.
- [53] Aleksandar Djenić, Nina Radojičić, Miroslav Marić, and Marko Mladenović. Parallel vns for bus terminal location problem. *Applied Soft Computing*, 42:448–458, 2016.
- [54] Karl Doerner, Walter Gutjahr, and Pamela Nolz. Multi-criteria location planning for public facilities in tsunami-prone coastal areas. *OR Spectrum*, 31(3):651–678, 2009.
- [55] Marco Dorigo, Mauro Birattari, and Thomas Stutzle. Ant colony optimization. *IEEE Computational Intelligence Magazine*, 1(4):28–39, 2006.
- [56] Darko Drakulić, Aleksandar Takači, and Miroslav Marić. New model of maximal covering location problem with fuzzy conditions. *Computing and Informatics*, 35(3):635–652, 2016.
- [57] Zvi Drezner and Horst W Hamacher. *Facility location: applications and theory*. Springer Science & Business Media, 2004.
- [58] Abraham Duarte, Rafael Martí, Mauricio GC Resende, and Ricardo MA Silva. Improved heuristics for the regenerator location problem. *International Transactions in Operational Research*, 21(4):541–558, 2014.
- [59] Abraham Duarte, Juan J Pantrigo, Eduardo G Pardo, and Nenad Mladenović. Multi-objective variable neighborhood search: an application to combinatorial optimization problems. *Journal of Global Optimization*, 63:515–536, 2015.

- [60] Pablo Maya Duque, Sofie Coene, Peter Goos, Kenneth Sörensen, and Frits Spieksma. The accessibility arc upgrading problem. *European Journal of Operational Research*, 224(3):458–465, 2013.
- [61] Matthias Ehrgott. *Multicriteria Optimization*. Springer, 2005.
- [62] Philippe Engrand. A multi-objective approach based on simulated annealing and its application to nuclear fuel management. In *5th International Conference on Nuclear Engineering*, 1997.
- [63] Güneş Erdogan, Maria Battarra, Gilbert Laporte, and Daniele Vigo. Metaheuristics for the traveling salesman problem with pickups, deliveries and handling costs. *Computers & Operations Research*, 39(5):1074–1086, 2012.
- [64] Levent Eriskin, Mumtaz Karatas, and Yu-Jun Zheng. A robust multi-objective model for healthcare resource management and location planning during pandemics. *Annals of Operations Research*, pages 1–48, 2022.
- [65] Yongsheng Fang and Jun Li. A review of tournament selection in genetic programming. In *International Symposium on Intelligence Computation and Applications*, pages 181–192. Springer, 2010.
- [66] Reza Zanjirani Farahani, Nasrin Asgari, Nooshin Heidari, Mahtab Hosseini, and Mark Goh. Covering problems in facility location: A review. *Computers & Industrial Engineering*, 62(1):368–407, 2012.
- [67] Reza Zanjirani Farahani, Masoud Hekmatfar, Behnam Fahimnia, and Narges Kazemzadeh. A review of some computational approaches to the capacitated location allocation problems. *Computers & Industrial Engineering*, 62(2):670–686, 2012.
- [68] Reza Zanjirani Farahani, Maryam SteadieSeifi, and Nasrin Asgari. Multiple criteria facility location problems: A survey. *Applied mathematical modelling*, 34(7):1689–1709, 2010.
- [69] Reza Zanjirani Farahani, Maryam SteadieSeifi, and Nasrin Asgari. Multiple criteria facility location problems: A survey. *Applied Mathematical Modelling*, 34(7):1689–1709, 2010.
- [70] Thomas Feo and Mauricio Resende. Greedy randomized adaptive search procedures. *Journal of Global Optimization*, 6:109–133, 1995.

- [71] Michele Flammini, Alberto Marchetti-Spaccamela, Gianpiero Monaco, Luca Moscardelli, and Shmuel Zaks. On the complexity of the regenerator placement problem in optical networks. *IEEE/ACM Transactions on Networking*, 19(2):498–511, 2011.
- [72] Kostas Florios and George Mavrotas. Generation of the exact pareto set in multi-objective traveling salesman and set covering problems. *Applied Mathematics and Computation*, 237:1–19, 2014.
- [73] Xavier Gandibleux, Nazik Mezdaoui, and Arnaud Fréville. A tabu search procedure to solve multiobjective combinatorial optimization problems. In *Advances in Multiple Objective and Goal Programming: Proceedings of the Second International Conference on Multi-Objective Programming and Goal Programming, Torremolinos, Spain, May 16–18, 1996*, pages 291–300. Springer, 1997.
- [74] Samrat Ganguly, Nirod Chandra Sahoo, and Debabrata Das. Multi-objective planning of electrical distribution systems using dynamic programming. *International Journal of Electrical Power & Energy Systems*, 46:65–78, 2013.
- [75] Martin Josef Geiger. Randomised variable neighbourhood search for multi objective optimisation. *arXiv preprint arXiv:0809.0271*, 2008.
- [76] Fred Glover. Future paths for integer programming and links to artificial intelligence. *Computers & Operations Research*, 13(5):533–549, 1986.
- [77] Fred Glover and Jin-Kao Hao. The case for strategic oscillation. *Annals of Operation Research*, 183(1):163–173, 2011.
- [78] David Goldberg. *Genetic algorithms in search, optimization, and machine learning*. Addison-wesley Reading Menlo Park, 1989.
- [79] L.E. Gorbachevskaya. Polynomially solvable and np-hard bilevel standardization problems. *PhD Thesis (in Russian), Sobolev Institute of Mathematics, Novosibirsk*, 1998.
- [80] David Hadka. Moea framework: A free and open source java framework for multiobjective optimization, 2019.

- [81] Pierre Hanjoul and Dominique Peeters. A facility location problem with clients' preference orderings. *Regional Science and Urban Economics*, 17(3):451–473, 1987.
- [82] Michael Pilegaard Hansen. Tabu search for multiobjective optimization: Mots. In *Proceedings of the 13th international conference on multiple criteria decision making*, pages 574–586, 1997.
- [83] Pierre Hansen, Yuri Kochetov, and Nenad Mladenović. *Lower bounds for the uncapacitated facility location problem with user preferences*. HEC Montréal, 2004.
- [84] Pierre Hansen and Nenad Mladenović. Variable neighborhood search: Principles and applications. *European Journal of Operational Research*, 130(3):449–467, 2010.
- [85] Pierre Hansen and Nenad Mladenović. Variable neighborhood search. In *Search methodologies: Introductory Tutorials in Optimization and Decision Support Techniques*, pages 313–337. Springer-Verlag, New York, 2014.
- [86] Pierre Hansen, Nenad Mladenović, Jack Brimberg, and José A Moreno Pérez. Variable neighborhood search. In Michel Gendreau and Jean-Yves Potvin, editors, *Handbook of Metaheuristics*, pages 57–97. Springer, 2019.
- [87] Pierre Hansen, Nenad Mladenović, and José Moreno Pérez. Variable neighbourhood search: methods and applications. *Annals of Operations Research*, 175(1):367–407, 2010.
- [88] Irina Harris, Christine Mumford, and Mohamed Naim. The multi-objective uncapacitated facility location problem for green logistics. In *2009 IEEE Congress on Evolutionary Computation*, pages 2732–2739. IEEE, 2009.
- [89] John Holland. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, 1975.
- [90] John H Holland. *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT press, 1992.

- [91] Holger Höller, Belén Melián, and Stefan Voß. Applying the pilot method to improve VNS and GRASP metaheuristics for the design of SDH/WDM networks. *European Journal of Operational Research*, 191(3):691–704, 2008.
- [92] Vasilyev Igor, Klimentova Xenia, , and Boccia Maurizio. Polyhedral study of simple plant location problem with order. *Operations Research Letters*, 41(2):153–158, 2013.
- [93] Hiroaki Ishii, Yung Lung Lee, and Kuang Yih Yeh. Fuzzy facility location problem with preference of candidate sites. *Fuzzy Sets and Systems*, 158(17):1922–1930, 2007.
- [94] Olivera Janković, Stefan Mišković, Zorica Stanimirović, and Raca 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.
- [95] Andrzej Jaskiewicz. A comparative study of multiple-objective metaheuristics on the bi-objective set covering problem and the pareto memetic algorithm. *Annals of Operations Research*, 131:135–158, 2004.
- [96] Mata Javier, de Miguel Ignacio, J. Durán Ramón, Merayo Noemí, Kumar Singh Sandeep, Jukan Admela, and Chamania Mohit. Artificial intelligence (AI) methods in optical networks: A comprehensive survey. *Optical Switching and Networking*, 28:43–57, 2018.
- [97] Hongzhong Jia, Fernando Ordóñez, and Maged M Dessouky. Solution approaches for facility location of medical supplies for large-scale emergencies. *Computers & Industrial Engineering*, 52(2):257–276, 2007.
- [98] Esra Karasakal and Ahmet Silav. A multi-objective genetic algorithm for a bi-objective facility location problem with partial coverage. *TOP*, 24(1):206–232, 2016.
- [99] Mumtaz Karatas and Ertan Yakıcı. An iterative solution approach to a multi-objective facility location problem. *Applied Soft Computing*, 62:272–287, 2018.
- [100] André A Keller. *Multi-objective optimization in theory and practice i: classical methods*. Bentham Science Publishers, 2017.

- [101] James Kennedy and Russell Eberhart. Particle swarm optimization. In *Proceedings of IEEE International Conference on Neural Networks*, pages 1942–1948. IEEE, 1995.
- [102] Samir Khuller, Anna Moss, and Joseph Seffi Naor. The budgeted maximum coverage problem. *Information Processing Letters*, 70(1):39–45, 1999.
- [103] Scott Kirkpatrick, Daniel Jr Gelatt, and Mario Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.
- [104] Sadan Kulturel-Konak, Alice E Smith, and Bryan A Norman. Multi-objective tabu search using a multinomial probability mass function. *European Journal of Operational Research*, 169(3):918–931, 2006.
- [105] Marco Laumanns, Lothar Thiele, Kalyanmoy Deb, and Eckart Zitzler. Combining convergence and diversity in evolutionary multiobjective optimization. *Evolutionary Computation*, 10(3):263–282, 2002.
- [106] Jung Man Lee and Young Hoon Lee. Facility location and scale decision problem with customer preference. *Computers & Industrial Engineering*, 63(1):184–191, 2012.
- [107] Yun-Chia Liang and Chia-Yin Chuang. Variable neighborhood search for multi-objective resource allocation problems. *Robotics and Computer-Integrated Manufacturing*, 29(3):73–78, 2013.
- [108] Yun-Chia Liang and Min-Hua Lo. Multi-objective redundancy allocation optimization using a variable neighborhood search algorithm. *Journal of Heuristics*, 16:511–535, 2010.
- [109] Arnaud Liefooghe and Bilel Derbel. A correlation analysis of set quality indicator values in multiobjective optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference 2016*, pages 581–588, 2016.
- [110] Qi Liu, Xiaofeng Li, Haitao Liu, and Zhaoxia Guo. Multi-objective metaheuristics for discrete optimization problems: A review of the state-of-the-art. *Applied Soft Computing*, 93:106382, 2020.
- [111] Freerk A. Lootsma, editor. *Multi-Objective Linear Programming*, pages 229–257. Springer US, Boston, MA, 1999.

- [112] Samir Loudni, Patrice Boizumault, and Philippe David. On-line resources allocation for ATM networks with rerouting. *Computers & Operations Research*, 33(10):2891–2917, 2006.
- [113] Helena Lourenço, Olivier Martin, and Thomas Stützle. Iterated local search. *Handbook of Metaheuristics*, pages 320–353, 2003.
- [114] Isaac Lozano-Osorio, Jesus Sanchez-Oro, Miguel Ángel Rodríguez-García, and Abraham Duarte. Optimizing computer networks communication with the band collocation problem: A variable neighborhood search approach. *Electronics*, 9(11):1860, 2020.
- [115] Thibaut Lust and Daniel Tuytens. Variable and large neighborhood search to solve the multiobjective set covering problem. *Journal of Heuristics*, 20(2):165–188, 2014.
- [116] Rita Macedo, Cláudio Alves, Saïd Hanafi, Bassem Jarboui, Nenad Mladenović, Bruna Ramos, and JM Valério de Carvalho. Skewed general variable neighborhood search for the location routing scheduling problem. *Computers & Operations Research*, 61:143–152, 2015.
- [117] Miroslav Marić, Zorica Stanimirović, Nikola Milenković, and Aleksandar Denić. Metaheuristic approaches to solving large-scale bilevel uncapacitated facility location problem with clients’ preferences. *Yugoslav Journal of Operations Research*, 25(3):361–368, 2015.
- [118] R. Timothy Marler and Jasbir S. Arora. Survey of multi-objective optimization methods for engineering. *Structural and Multidisciplinary Optimization*, 26(6):369–395, 2004.
- [119] Rafael Martí, Vicente Campos, Mauricio GC Resende, and Abraham Duarte. Multiobjective grasp with path relinking. *European Journal of Operational Research*, 240(1):54–71, 2015.
- [120] Hela Masri, Saoussen Krichen, and Adel Guitouni. A multi-start variable neighborhood search for solving the single path multicommodity flow problem. *Applied Mathematics and Computation*, 251:132–142, 2015.

- [121] George Mavrotas. Effective implementation of the ε -constraint method in multi-objective mathematical programming problems. *Applied Mathematics and Computation*, 213(2):455–465, 2009.
- [122] George Mavrotas and Kostas Florios. An improved version of the augmented ε -constraint method (augmecon2) for finding the exact pareto set in multi-objective integer programming problems. *Applied Mathematics and Computation*, 219(18):9652–9669, 2013.
- [123] Vinícius R Máximo, Mariá CV Nascimento, and André CPLF Carvalho. Intelligent-guided adaptive search for the maximum covering location problem. *Computers & Operations Research*, 78:129–137, 2017.
- [124] Andrés L Medaglia, Juan G Villegas, and Diana M Rodríguez-Coca. Hybrid biobjective evolutionary algorithms for the design of a hospital waste management network. *Journal of Heuristics*, 15(2):153, 2009.
- [125] Nimrod Megiddo, Eitan Zemel, and S Louis Hakimi. The maximum coverage location problem. *SIAM Journal on Algebraic Discrete Methods*, 4(2):253–261, 1983.
- [126] Belén Melián. Using memory to improve the VNS metaheuristic for the design of SDH/WDM networks. In Francisco Almeida, María J. Blesa Aguilera, Christian Blum, José Marcos Moreno Vega, Melquíades Pérez Pérez, Andrea Roli, and Michael Sampels, editors, *Hybrid Metaheuristics*, pages 82–93, Heidelberg, 2006. Springer.
- [127] Belén Melián-Bautista, Holger Höller, and Stefan Voß. Designing WDM networks by a variable neighborhood search. *Journal of Telecommunications & Information Technology*, 4, 2006.
- [128] Achille Messac, Amir Ismail-Yahaya, and Christopher A Mattson. The normalized normal constraint method for generating the pareto frontier. *Structural and Multidisciplinary Optimization*, 25:86–98, 2003.
- [129] Stefan Mišković. A VNS-LP algorithm for the robust dynamic maximal covering location problem. *OR Spectrum*, 39(4):1011–1033, 2017.
- [130] Melanie Mitchell. *An introduction to genetic algorithms*. MIT press, 1998.

- [131] Nenad Mladenović and Pierre Hansen. Variable neighborhood search. *Computers & Operations Research*, 24(11):1097–1100, 1997.
- [132] Lazar Mrkela. An efficient variable neighborhood search for generalized regenerator location problems. *International Journal on Artificial Intelligence Tools*, 31(05):2250033, 2022.
- [133] Lazar Mrkela and Zorica Stanimirović. VNS-based solution approaches to the maximal covering location problem with customer preference ordering. In *Proceedings of the XIII Balkan Conference on Operational Research, BALCOR 2018*, pages 133–140. The Mathematical Institute of the Serbian Academy of Sciences and Arts (SANU), Belgrade, Serbia, 2018.
- [134] Lazar Mrkela and Zorica Stanimirović. Skewed variable neighborhood search method for the weighted generalized regenerator location problem. *Lecture Notes in Computer Science*, 11328:182–201, 2019.
- [135] Lazar Mrkela and Zorica Stanimirović. A bi-objective maximal covering location problem: a service network design application. In *2020 International Conference on Innovations in Intelligent Systems and Applications (INISTA)*, pages 1–7. IEEE, 2020.
- [136] Lazar Mrkela and Zorica Stanimirović. A multi-objective variable neighborhood search for the maximal covering location problem with customer preferences. *Cluster Computing*, 25(3):1677–1693, 2022.
- [137] Lazar Mrkela and Zorica Stanimirović. A variable neighborhood search for the budget-constrained maximal covering location problem with customer preference ordering. *Operational Research*, 22(5):5913–5951, 2022.
- [138] Lazar Mrkela and Zorica Stanimirović. A metaheuristic approach to solving the maximal covering location problem. In *Proceedings of the XLVI International Symposium on Operational Research (SYM-OP-IS)*, Kladovo, Serbia, 2019. Faculty of Organizational Sciences, University of Belgrade.
- [139] Lazar Mrkela and Zorica Stanimirović. Bi-objective dynamic maximal covering location problem. In *Proceedings of the XLVIII International Symposium on Operational Research (SYM-OP-IS)*, Banja Koviljača, Serbia, 2021. Faculty of Mathematics, University of Belgrade.

- [140] Lazar Mrkela, Filip Vidojević, and Zorica Stanimirović. Dvokriterijumski uopšteni problem postavljanja regeneratora u optičkim mrežama. In *50. Međunarodni simpozijum o operacionim istraživanjima*. Beograd: Medija centar,, Odbrana”, 2023.
- [141] Daniel Mueller-Gritschneider, Helmut Graeb, and Ulf Schlichtmann. A successive approach to compute the bounded pareto front of practical multiobjective optimization problems. *SIAM Journal on Optimization*, 20(2):915–934, 2009.
- [142] Dongkyung Nam and Cheol Hoon Park. Multiobjective simulated annealing: A comparative study to evolutionary algorithms. *International Journal of Fuzzy Systems*, 2(2):87–97, 2000.
- [143] Peter Nijkamp and Jaap Spronk. Interactive multidimensional programming models for locational decisions. *European Journal of Operational Research*, 6(2):220–223, 1981.
- [144] Diego Oliva, Essam H Houssein, and Salvador Hinojosa. *Metaheuristics in machine learning: theory and applications*. Springer, 2021.
- [145] Kivanç Onan, Füsün Ülengin, and Bahar Sennaroğlu. An evolutionary multi-objective optimization approach to disaster waste management: A case study of istanbul, turkey. *Expert Systems with Applications*, 42(22):8850–8857, 2015.
- [146] Anthony Przybylski and Xavier Gandibleux. Multi-objective branch and bound. *European Journal of Operational Research*, 260(3):856–872, 2017.
- [147] Juan David Quintana, Jesús Sánchez-Oro, and Abraham Duarte. Efficient greedy randomized adaptive search procedure for the generalized regenerator location problem. *International Journal of Computational Intelligence Systems*, 9(6):1016–1027, 2016.
- [148] Hari K Rajagopalan, Cem Saydam, and Jing Xiao. A multiperiod set covering location model for dynamic redeployment of ambulances. *Computers & Operations Research*, 35(3):814–826, 2008.
- [149] L.A. Rastrigin. The convergence of the random search method in the extremal control of a many parameter system. *Automaton & Remote Control*, 24:1337–1342, 1963.

- [150] Ingo Rechenberg. Cybernetic solution path of an experimental problem. *Royal Aircraft Establishment*, 1122, 1965.
- [151] Charles ReVelle and Kathleen Hogan. The maximum availability location problem. *Transportation Science*, 23(3):192–200, 1989.
- [152] Charles ReVelle, Joseph Schweitzer, and Stephanie Snyder. The maximal conditional covering problem. *INFOR*, 34(2):77–91, 1996.
- [153] Charles S ReVelle and Horst A Eiselt. Location analysis: A synthesis and survey. *European Journal of Operational Research*, 165(1):1–19, 2005.
- [154] Nery Riquelme, Christian Von Lüken, and Benjamin Baran. Performance metrics in multi-objective optimization. In *2015 Latin American Computing Conference (CLEI)*, pages 1–11. IEEE, 2015.
- [155] Herbert Robbins and Sutton Monro. A stochastic approximation method. *The Annals of Mathematical Statistics*, pages 400–407, 1951.
- [156] Sébastien Rumley, Christian Gaumier, and Radoslaw Szymanek. Multi-objective optimization of regenerator placement using constraint programming. In *15th International Conference on Optical Network Design and Modeling-ONDM 2011*, pages 1–6. IEEE, 2011.
- [157] Majid Salari. An iterated local search for the budget constrained generalized maximal covering location problem. *Journal of Mathematical Modelling and Algorithms in Operations Research*, 13(3):301–313, 2014.
- [158] Jesús Sánchez-Oro and Abraham Duarte. Beyond unfeasibility: Strategic oscillation for the maximum leaf spanning tree problem. In José M. Puerta, José A. Gámez, Bernabe Dorronsoro, Edurne Barrenechea, Alicia Troncoso, Bruno Baruque, and Mikel Galar, editors, *Advances in Artificial Intelligence*, pages 322–331, Cham, 2015. Springer International Publishing.
- [159] Jesús Sánchez-Oro, Ana D López-Sánchez, and J Manuel Colmenar. A multi-objective parallel variable neighborhood search for the bi-objective obnoxious p-median problem. *Optimization Letters*, pages 1–31, 2022.
- [160] James David Schaffer. *Some experiments in machine learning using vector evaluated genetic algorithms (artificial intelligence, optimization, adaptation, pattern recognition)*. PhD thesis, Vanderbilt University, 1984.

- [161] Verena Schmid and Karl F Doerner. Ambulance location and relocation problems with time-dependent travel times. *European Journal of Operational Research*, 207(3):1293–1303, 2010.
- [162] Paolo Serafini. Simulated annealing for multi objective optimization problems. In *Multiple Criteria Decision Making: Proceedings of the Tenth International Conference: Expand and Enrich the Domains of Thinking and Application*, pages 283–292. Springer, 1994.
- [163] Ron Shamir. The efficiency of the simplex method: a survey. *Management Science*, 33(3):301–334, 1987.
- [164] Pradyumn Kumar Shukla. On the normal boundary intersection method for generation of efficient front. In *Computational Science–ICCS 2007: 7th International Conference, Beijing, China, May 27–30, 2007, Proceedings, Part I 7*, pages 310–317. Springer, 2007.
- [165] Lawrence V Snyder. Covering problems. In H. A. Eiselt and Vladimir Marianov, editors, *Foundations of location analysis. International series in operations research & management science, vol 155*, pages 109–135. Springer, Boston, MA, 2011.
- [166] Jan A Snyman, Daniel N Wilke, et al. *Practical mathematical optimization*. Springer, 2005.
- [167] Helge Spieker, Alexander Hagg, Adam Gaier, Stefanie Meilinger, and Alexander Asteroth. Multi-stage evolution of single-and multi-objective mclp. *Soft Computing*, 21(17):4859–4872, 2017.
- [168] N. Srinivas and K. Deb. Multi-objective function optimization using non-dominated sorting genetic algorithms. *Evolutionary computation*, 2(3):221–248, 1995.
- [169] Zorica Stanimirović, Stefan Misković, Darko Trifunović, and Veselin Veljović. A two-phase optimization method for solving the multi-type maximal covering location problem in emergency service networks. *Information Technology and Control.*, 46(1):100–117, 2017.
- [170] Zorica Stanimirović and Lazar Mrkela. Evolutionary algorithm approach to bi-objective maximal covering location problem with customer preferences. In

Proceedings of the XLVII International Symposium on Operational Research (SYM-OP-IS), Belgrade, Serbia, 2020. Faculty of Transport and Traffic Engineering, University of Belgrade.

- [171] Rainer Storn and Kenneth Price. Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 11(4):341–359, 1995.
- [172] A Suppapitnarm, Keith A Seffen, Geoff T Parks, and PJ Clarkson. A simulated annealing algorithm for multiobjective optimization. *Engineering Optimization*, 33(1):59–85, 2000.
- [173] El-Ghazali Talbi. *Metaheuristics: from design to implementation*. John Wiley & Sons, 2009.
- [174] Tetsuzo Tanino, Tamaki Tanaka, and Masahiro Inuiguchi. *Multi-objective programming and goal programming: theory and applications*, volume 21. Springer Science & Business Media, 2003.
- [175] Ye Tian, Langchun Si, Xingyi Zhang, Ran Cheng, Cheng He, Kay Chen Tan, and Yaochu Jin. Evolutionary large-scale multi-objective optimization: A survey. *ACM Computing Surveys (CSUR)*, 54(8):1–34, 2021.
- [176] Igor Vasilyev, K.B. Klimentova, and Yury Kochetov. New lower bounds for the facility location problem with clients’ preferences. *Computational Mathematics and Mathematical Physics*, 49:1010–1020, 2009.
- [177] Dalessandro Soares Vianna and José Elias Claudio Arroyo. A grasp algorithm for the multi-objective knapsack problem. In *XXIV International Conference of the Chilean Computer Science Society*, pages 69–75. IEEE, 2004.
- [178] Juan G Villegas, Fernando Palacios, and Andrés L Medaglia. Solution methods for the bi-objective (cost-coverage) unconstrained facility location problem with an illustrative example. *Annals of Operations Research*, 147(1):109–141, 2006.
- [179] Christian Von Lüken, Benjamín Barán, and Carlos Brizuela. A survey on multi-objective evolutionary algorithms for many-objective problems. *Computational Optimization and Applications*, 58(3):707–756, 2014.

- [180] CDT Watson-Gandy. Heuristic procedures for the m-partial cover problem on a plane. *European Journal of Operational Research*, 11(2):149–157, 1982.
- [181] John A White and Kenneth E Case. On covering problems and the central facilities location problem. *Geographical Analysis*, 6(3):281–294, 1974.
- [182] Laurence A Wolsey. *Integer programming*. John Wiley & Sons, 2020.
- [183] Jian-Bo Yang and Pratyush Sen. Preference modelling by estimating local utility functions for multiobjective optimization. *European Journal of Operational Research*, 95(1):115–138, 1996.
- [184] Xin-She Yang. *Engineering optimization: an introduction with metaheuristic applications*. John Wiley & Sons, 2010.
- [185] Mohammad Hossein Fazel Zarandi, Soheil Davari, and Seyyed Ali Haddad Sisakht. The large-scale dynamic maximal covering location problem. *Mathematical and Computer Modelling*, 57(3-4):710–719, 2013.
- [186] Aimin Zhou, Bo-Yang Qu, Hui Li, Shi-Zheng Zhao, Ponnuthurai Nagaratnam Suganthan, and Qingfu Zhang. Multiobjective evolutionary algorithms: A survey of the state of the art. *Swarm and Evolutionary Computation*, 1(1):32–49, 2011.
- [187] Eckart Zitzler, Marco Laumanns, and Lothar Thiele. SPEA 2: improving the strength pareto evolutionary algorithm, TIK report 103. *Computer Engineering and Networks Laboratory (TIK), ETH Zurich, Zurich, Switzerland*, 2001.
- [188] Eckart Zitzler and Lothar Thiele. Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. *IEEE transactions on Evolutionary Computation*, 3(4):257–271, 1999.

Биографија аутора

Лазар Мркела рођен је 3. августа 1992. године у Книну, Хрватска. Основну школу и гимназију завршио је у Панчеву. Школске 2011/2012. године уписао је основне академске студије на Математичком факултету у Београду (студијски програм Информатика) и дипломирао школске 2013/2014. године са просечном оценом 8,93.

Школске 2014/2015. године уписао је мастер академске студије на Математичком факултету, студијски програм Информатика. Мастер академске студије завршио је 2016. године одбраном мастер рада под насловом „Решавање проблема оптималног планирања бежичних меш мрежа применом метакскурсних метода” под руководством проф. др Зорице Станимировић. Просечна оцена на мастер студијама је 9,62. Школске 2016/2017. године уписао је докторске академске студије на Математичком факултету, студијски програм Информатика. Положио је све испите предвиђене планом и програмом докторских студија са просечном оценом 10,00.

Од 2016. до 2017. године радио је као софтверски инжењер у Институту „Михајло Пупин” у Београду на развоју и одржавању софтвера за ЈП Електро мрежа Србије. Од 2017. године запослен је на Факултету за информационе технологије, Универзитета Метрополитан у Београду. Као асистент учествовао је у извођењу вежби из више предмета на основним студијама у области рачунарства. Поред наставе био је ангажован и у Лабораторији за софтверско инжењерство Универзитета Метрополитан, где је радио на развоју софтвера за креирање и анализу интерактивних наставних материјала.

Од 2022. до 2024. године био је запослен у компанији Hooiooovoo, где је радио као софтверски инжењер на пројектима за шведску компанију Kindred group. Као истраживач био је ангажован на пројекту „Математички модели и методе оптимизације великих система”, пројекат бр. 174010 Министарства просвете, науке и технолошког развоја Републике Србије, чији је координатор био Математички институт САНУ. Био је предавач у оквиру пројеката Преквалификације за ИТ 2018 (Пројекат Владе Републике Србије подржан од UNDP-a). Био је рецензент неколико радова за часописе IEEE Transactions on Medical Imaging, Soft Computing, Nature Scientific Reports и Journal of Ambient Intelligence and Humanized Computing. Био је члан Организационог одбора 48. Међународног симпозијума о операционим истраживањима (SYM-OP-IS 2021).

Прилог 2.

Изјава о истоветности штампане и електронске верзије докторског рада

Име и презиме аутора Лазар Мркела

Број индекса 2012/2016

Студијски програм Иинформатика

Наслов рада „Метахеуристичке методе вишекритеријумске оптимизације и примене на дискретне локацијске проблеме“

Ментор проф. др Зорица Станимировић

Потписани/а Лазар Мркела

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, 25.12.2024



Прилог 3.

Изјава о коришћењу

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

„Метахеуристичке методе вишекритеријумске оптимизације и примене на дискретне локацијске проблеме“

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).

У Београду, 25.12.2024

Потпис докторанда



1. Ауторство - Дозвољавање умножавање, дистрибуцију и јавно саопштавање дела, и прераде, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце, чак и у комерцијалне сврхе. Ово је најслободнија од свих лиценци.

2. Ауторство – некомерцијално. Дозвољавање умножавање, дистрибуцију и јавно саопштавање дела, и прераде, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце. Ова лиценца не дозвољава комерцијалну употребу дела.

3. Ауторство - некомерцијално – без прераде. Дозвољавање умножавање, дистрибуцију и јавно саопштавање дела, без промена, преобликовања или употребе дела у свом делу, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце. Ова лиценца не дозвољава комерцијалну употребу дела. У односу на све остале лиценце, овом лиценцом се ограничава највећи обим права коришћења дела.

4. Ауторство - некомерцијално – делити под истим условима. Дозвољавање умножавање, дистрибуцију и јавно саопштавање дела, и прераде, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце и ако се прерада дистрибуира под истом или сличном лиценцом. Ова лиценца не дозвољава комерцијалну употребу дела и прерада.

5. Ауторство – без прераде. Дозвољавање умножавање, дистрибуцију и јавно саопштавање дела, без промена, преобликовања или употребе дела у свом делу, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце. Ова лиценца дозвољава комерцијалну употребу дела.

6. Ауторство - делити под истим условима. Дозвољавање умножавање, дистрибуцију и јавно саопштавање дела, и прераде, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце и ако се прерада дистрибуира под истом или сличном лиценцом. Ова лиценца дозвољава комерцијалну употребу дела и прерада. Слична је софтверским лиценцама, односно лиценцама отвореног кода.