

УНИВЕРЗИТЕТ У БЕОГРАДУ  
МАТЕМАТИЧКИ ФАКУЛТЕТ

Марија Р. Ивановић

**Нови приступи у решавању  
оптимизационог проблема  
римске доминације на  
графовима**

докторска дисертација

Београд, 2022.

UNIVERSITY OF BELGRADE  
FACULTY OF MATHEMATICS

Marija R. Ivanović

**New approaches for solving the  
Roman domination problem on  
graphs**

Doctoral Dissertation

Belgrade, 2022.

## Подаци о ментору и члановима комисије

### Ментор:

**др Александар Савић,**  
ванредни професор,  
Универзитет у Београду, Математички факултет

### Чланови комисије:

**др Милан Дражић**  
ванредни професор,  
Универзитет у Београду, Математички факултет

**др Татјана Давидовић**  
научни саветник,  
Математички институт САНУ

**др Драган Урошевић**  
научни саветник,  
Математички Институт САНУ

*Датум одбране:*

## *Захвалност*

Захвалност пре свега дугујем свом ментору, др Александру Савићу, за труд, савете и несебично пренесено знање како на послу, тако и током израде ове дисертације. Веома сам захвална на указаном поверењу. Такође, хтела бих да се захвалим члановима комисије: професору др Милану Дражићу, који ме је увео у свет математичког моделирања и пружио ми, дељењем свог знања и сјајних идеја, мотивацију да се бавим истраживањима у овој области, професору др Драгану Урошевићу на прилици за заједнички рад, на подршци и помоћи на путу мог научног усавршавања и професорки др Татјани Давидовић на, пре свега, дивној сарадњи, саветима и мотивацији за даљи напредак и усавршавање. Искористила бих прилику да се захвалим и колегиницама из Рачунарске лабораторије на техничкој подршци приликом тестирања инстанци и својим драгим колегиницама и колегама са Катедре за нумеричку математику и оптимизацију на изузетно лепој радној и пријатељској атмосфери. Задовољство је и част бити део таквог тима.

Најзад, желим да се захвалим свима који су били уз мене и који су ми својим саветима, идејама и сугестијама помогли при изради дисертације. Хвала свима од срца!

у Београду, 2022.  
Марија Ивановић

## Подаци о докторској дисертацији

**Назив докторске дисертације:** Нови приступи у решавању оптимизационог проблема римске доминације на графовима

**Резиме:** Фокус дисертације представљају оптимизациони проблем римске доминације и његове две модификације. Формулације целобројног линеарног програмирања проблема римске доминације, познате из литературе, су побољшане, доказано је да су побољшања еквивалентна оригиналним формулацијама упркос мањем броју ограничења и релаксацијама променљивих и извршена су поређења оригиналних и побољшаних формулација коришћењем стандардних оптимизационих решавача, CPLEX и Gurobi. Без обзира на податак да се побољшане формулације могу једнако користити као и оригиналне формулације, из нумеричких резултата се може приметити да су побољшане формулације корисније за одређени број инстанци. Будући да се оригиналне и побољшане формулације не могу решавати стандардним оптимизационим решавачима на неким класама графова великих димензија и да алгоритми који постоје у литератури најчешће решавају проблем римске доминације за конкретну класу графова, конструисан је нови алгоритам који проблем римске доминације решава за све класе и димензије графа. Иако је проблем римске доминације из класе NP проблема, представљеним алгоритмом су се, за већи број инстанци, пронашла решења која су једнака оптималним за чак мање од једне секунде. За прву модификацију проблема римске доминације, која се назива проблемом ограничене римске доминације, представљена је прва формулација мешовитог целобројног линеарног програмирања и, до момента писања ове тезе, то је једина математичка формулација овог проблема позната у литератури. За другу модификацију проблема римске доминације, проблем слабе римске доминације, представљена је побољшана целобројна формулација, показано је да је представљена формулација коректна и еквивалентна формулацији из литературе, а затим су обе формулације тестиране коришћењем стандардних оптимизационих решавача, CPLEX и Gurobi. На основу нумеричких резултата јасно се може видети предност примене побољшане формулације. Поред нових математичких формулација, представљен је и побољшан алгоритам линеарне комплексности за решавање

проблема слабе римске доминације на блок графовима, а будући да је проблем слабе римске доминације из класе NP проблема, конструисан је и нови алгоритам заснован на методи променљивих околина за решавање проблема слабе римске доминације. Представљеним алгоритмом достиже се решење једнако оптималном за скоро све тестиране инстанце за које је познато оптимално решење, док су за неке инстанце за које није доказана оптималност пронађеног решења описаним алгоритмом добијена боља решења. Поред бољих решења, одређени број инстанци је први пут решен с обзиром да оптимизациони решавачи нису успели да их реше пре прекидања програма.

**Кључне речи:** целобројно линеарно програмирање, мешовито целобројно линеарно програмирање, комбинаторна оптимизација, метахеуристика, метода променљивих околина

**Научна област:** Оптимизација

**Ужа научна област:** Дискретна оптимизација

**УДК број:** [519.852 + 519.854] : 519.863(043.3)

## Dissertation Data

**Doctorial dissertation title:** New approaches in solving the Roman domination problem on graphs

**Abstract:** This dissertation focuses on the Roman domination problem and its two modifications. Improvements and relaxations of two integer linear programming formulations for the Roman domination problem from the literature are introduced, proved to be equivalent to the existing ones despite of the variables relaxation and usage of fewer number of constraints and compared by standard optimization solvers, CPLEX and Gurobi. Improved formulations can be equally used as original ones, but, as it can be seen from numerical results, for some instances, they can be more useful. Given the fact that old and new formulations can not be used for some large size instances, and that algorithms for solving Roman domination problem are mostly defined for some particular graph classes, the aim of this research was to find a new algorithm that can be used for solving Roman domination problem on all graph classes and all graph sizes. Although the Roman domination problem belongs to the class NP, presented algorithm is able to find solution value equal to optimal solution value on very large number of instances in less then a second. For the first modification of the Roman domination problem, named Restrained Roman domination problem, a new mixed integer linear programming formulation is introduced and, to the best of the author's knowledge, this formulation is the first in the literature. For the second modification of the Roman domination problem, the Weak Roman domination problem, an improved integer linear programming formulation is presented. Improved formulation is also proved to be correct, equivalent to the existing formulation from the literature and compared using standard optimization solvers, CPLEX and Gurobi. Numerical results showed the advantage of the improved formulation on almost all tested instances. Additionally, an improved linear-time algorithm for solving the Weak Roman domination problem on block graphs is introduced and, similarly to the Roman domination problem, a new algorithm, based on the variable neighborhood search method is presented. With the new variable neighborhood search based algorithm we aimed to find solution of the Weak Roman domination problem equal to the optimal on very large number of tested instances. For instances for which some solution value is found but not proved to be an optimal, presented algorithm provided the new lower-bounds. Even more, for some instances, where optimization solvers were not able to prove optimality or

to find any solution, new solutions are found.

**Keywords:** integer linear programming, mixed integer linear programming, combinatorial optimization, metaheuristics, variable neighborhood search

**Scientific field:** Optimization

**Scientific discipline:** Discrete optimization

**UDC number:** [519.852 + 519.854] : 519.863(043.3)



# Садржај

<b>1</b>	<b>Уводна разматрања и основни појмови</b>	<b>1</b>
1.1	Увод . . . . .	1
1.2	Основни појмови и дефиниције . . . . .	6
1.3	Ознаке коришћене у раду . . . . .	14
1.4	Проблеми доминације и римске доминације на графу . . .	18
<b>2</b>	<b>Преглед релевантне литературе о проблемима римске доминације</b>	<b>22</b>
2.1	Проблем римске доминације . . . . .	22
2.2	Проблем ограничене римске доминације . . . . .	32
2.3	Проблем слабе римске доминације . . . . .	37
2.4	Преглед алгоритама за решавање проблема римске доминације и неких његових модификација . . . . .	40
<b>3</b>	<b>Математичке формулације проблема римске доминације и неких његових модификација</b>	<b>43</b>
3.1	Проблем римске доминације на графовима . . . . .	43
3.1.1	Математичке формулације проблема римске доминације из релевантне литературе . . . . .	43
3.1.2	Побољшање математичких формулација проблема римске доминације из литературе . . . . .	45
3.2	Проблем ограничене римске доминације . . . . .	50
3.2.1	Нова математичка формулација проблема ограничене римске доминације . . . . .	50

3.3	Проблем слабе римске доминације . . . . .	54
3.3.1	Математичка формулација проблема слабе римске доминације из релевантне литературе . . . . .	54
3.3.2	Побољшана математичка формулација <i>WRD</i> проблема . . . . .	60
3.4	Поређења математичких формулација . . . . .	66
<b>4</b>	<b>Алгоритми за решавање проблема римске и слабе римске доминације на графу</b>	<b>87</b>
4.1	Егзактна метода за решавање проблема римске доминације . . . . .	88
4.1.1	Алгоритам линеарне комплексности за решавање проблема слабе римске доминације на блок-графовима	89
4.2	Недетерминистичка метода за решавање проблема римске доминације . . . . .	100
4.2.1	Метода променљивих околина . . . . .	101
4.2.2	Нова метода за решавање проблема римске доминације заснована на методи променљивих околина .	106
4.2.3	Нова метода за решавање проблема слабе римске доминације заснована на методи променљивих околина	154
<b>5</b>	<b>Закључак</b>	<b>170</b>
	<b>Литература</b>	<b>172</b>

*Људи који не верују да је математика једноставна  
заправо нису свесни колико је живот компликован.*

*John von Neuman (1903-1957)*

# 1 Уводна разматрања и основни појмови

## 1.1 Увод

Као једно од највећих царстава у историји, Римско царство је у I веку нове ере заузимало велике делове Европе, Азије и северне Африке. Реформе које је Октавијан Август (лат. Gaius Julius Caesar Octavianus Augustus) спровео после успешно завршених грађанских ратова, крајем I века старе ере, подразумевале су и реорганизацију царства и његове војске. Број легија, који је 30. године пре нове ере износио око 100, сведен је на 28. После катастрофалног пораза у Теутобуршкој шуми (енг. Teutoburg Forest) 9. године нове ере, када су германска племена уништила 3 од постојећих 28 легија, стратегија царства је прешла са изразито офанзивне на консолидацију и дефанзиву. Током I и II века нове ере наметнула су се два стратегијска приступа одбране царства. Током владавине Јулио-Клаудијеваца (енг. Julio Claudian dynasty), до 68. године нове ере, одбрана Римског царства била је заснована како на активном учешћу легија, тако и на значајном ослањању на трупе клијентских краљевстава распоређеним у широком луку, од пустиње Арабије до малих племенских држава Германа смештеним на крајњем северозападу европског континента. Стратегија се заснивала на перцепцији војне снаге Римског царства које је, у случају непослушности, могло да доведе до уништења било ког од потчињених клијентских краљевстава. Сазнање о моћи Рима спречавало је нападе на његову територију чак и када римска војска није могла адекватно да одговори на напад. Покошавање већег броја клијентских краљевстава, посебно на Блиском истоку и Малој Азији, као и освајање Дакије, довели су до преласка

на изузетно софистицирану стратегију одбране чији је темељ поставио цар Хадријан (117-138) (лат. Publius Aelius Traianus Hadrianus). Како би се предупредили напади на територију самог царства, стратегија цара Хадријана се заснивала на нападима на било коју већу војну формацију са друге стране границе и, као таква, успешно је функционисала до почетка кризе, односно до 235. године.

Велика криза III века, која се завршила владавинама цара Диоклецијана (лат. Gaius Aurelius Valerius Diocletianus Augustus) и цара Константина (лат. Flavius Valerius Aurelius Constantinus Augustus), довела је до драстичне промене у стратегији одбране царства. Почетком IV века царство није више било у стању да предупреди нападе на своју територију, тј. да ефикасно извршава ударе по ојачаним и знатно боље организованим племенским конфедерацијама Германа дуж европских граница, на Рајни и Дунаву, или знатно ојачалог персијског царства, под Сасанидима, на Блиском истоку. Због тога је цар Константин извео реформу одбране и свео је на одбрану по дубини територије. Одбраном по дубини територије би се, у најбољем случају, нападач, који је упао на територију царства, опколио тако што би пограничне трупе блокирале границу иза њега, а трупе распоређене по дубини територије блокирале пут ка суштински важним и популационо и економски најзначајнијим територијама царства. У циљу реализације оваквог вида одбране, било је потребно осмислити распоред што је могуће мањег броја јединица по целом царству тако да свака провинција буде заштићена, или, у случају успешног напада, упад буде локализован на што је могуће мању територију. За детаљнији опис стратегије одбране Римског царства погледати књигу Е. N. Luttwak-а, [1].

Идеја за математичким приказом проблема распоређивања легија Римског царства први пут се спомиње у раду John Arquilla и Hal Fredricksen-а "Graphing' an Optimal Grand Strategy" објављеном у часопису "Military Operations Research" 1995. године. Убрзо потом, С.S. ReVelle априла 1997. године објављује свој рад под именом "Да ли можеш да заштитиш Римско царство? (енг. Can you protect the Roman Empire?). Као одговор на питање које је ReVelle поставио, Ian Stewart објављује текст под насловом "Одбранити Римско царство!" (енг. "Defend the Roman Empire!") децембра 1999. године и тиме инспирише велики број научника да овај проблем реше. Већ у издању август-септембар 2000. године С.S. ReVelle и К.Е.

Rossing објављују свој рад ”Одбрана Римског царства: Класичан проблем у војној стратегији” (енг. ”Defendens Imperium Romanum: A Classical Problem in Military Strategy”) који се сматра једним од првих радова на ту тему. Том приликом ReVelle и Rossing поставили су два питања:

- Колико је трупа потребно да се заштити царство и како да се трупе распореде тако да свака регија царства буде или сигурна или заштићена?
- У случају да постоји ограничен број трупа, који је оптимални распоред трупа који ће омогућити да највећи број регија буде сигуран или заштићен?

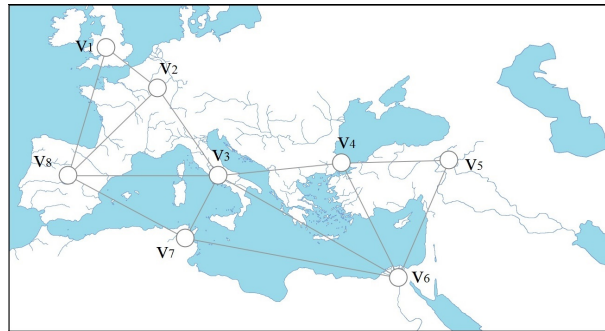
Као одговор на постављена питања у истом раду дати су математички модели за решавање описаних проблема. Са обзиром да се за сваку регију може поставити питање да ли је или није испуњен неки услов, проблем организовања трупа припада класи проблема одлучивања. Стога су ReVelle и Rossing, ако је сваки регион или сигуран (има стационирану трупу) или сусед са регијом која има две трупе, проблем одређивања најмањег броја трупа потребних да свака регија царства буде или сигурна или заштићена назвали *проблемом покривања скупа* (енг. the set covering deployment problem). Слично, одговор на друго питање представља *проблем максималног покривања скупа* (енг. the maximal covering deployment problem). Проблеми покривања скупа и максималног покривања скупа, као класични проблеми комбинаторне оптимизације, припадају класи NP комплетних проблема.

Будући да се свака регија Римског царства може представити чвором графа и да се за сваке две регије које имају заједничку границу (или се морским путем може директно стићи од једне регије до друге), одговарајући чворови могу повезати граном, Римско царство се може представити графовски. При том би број трупа стационираних у регији коју дати чвор симболизује заправо представљао тежину тог чвора. Нека је, зато, свака регија Римског царства представљена чвором графа и нека су два чвора повезана уколико између одговарајућих регија постоји заједничка граница или директна водена веза. Дакле, нека се, на пример, граф састоји из 8 чворова:  $v_1 \equiv$  Британија,  $v_2 \equiv$  Галија,  $v_3 \equiv$  Рим,  $v_4 \equiv$  Тракија,  $v_5 \equiv$  Азија,  $v_6 \equiv$  Египат,  $v_7 \equiv$  Африка и  $v_8 \equiv$  Шпанија

и нека је матрица суседства датих чворова дефинисана са

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}.$$

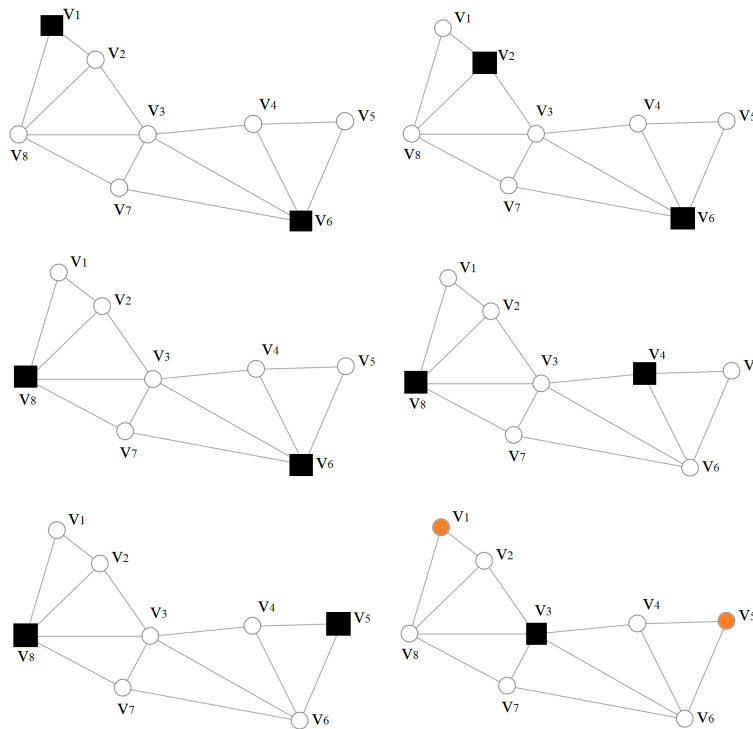
Обележимо овако дефинисан граф са  $G_{\mathcal{R}}$ , Слика 1. Проблем одређи-



Слика 1: Приказ графа  $G_{\mathcal{R}}$  дефинисаног за Римско царство

вања броја трупа и њиховог распореда којим би се заштитило 8 регија може се оптимално решити на шест начина. Узимајући да тежина чвора заправо представља број стационарираних трупа у одговарајућој регији, обележимо чвор тежине 2 црним квадратом, чвор тежине један црвеним кругом и чвор тежине нула белим кругом. Тада се свих шест оптималних решења проблема распоређивања трупа на графу  $G_{\mathcal{R}}$  могу приказати као на Слици 2.

Имајући у виду да је дати пример мањих размера, лако се може видети да је најмањи број трупа потребних да се заштити царство 4, односно да са мањим бројем трупа то није могуће. Међутим, да ли се увођењем нових или изменом постојећих услова Римско царство може заштити са мањим бројем трупа? Односно, да ли наведени услови заиста гарантују безбедност и како спречити побуне у регијама које немају стационариране трупе? Као одговор на ова питања настао је велики број сличних проблема. Под претпоставком да је регија заштићена уколико има стационарирану трупу или је суседна са регијом у којој је бар једна



Слика 2: Илустрација оптималног распореда трупа по регијама Римског царства на графу  $G_{\mathcal{R}}$

трупа стационирана, може се дозволити померање трупе из провинције у којој је стационирана ка суседној, нападнутој провинцији. Проблем који дозвољава описано померање тако да је царство безбедно пре и након померања трупе, назива се *проблемом слабе римске доминације*. Увођењем услова којим свака регија која нема стационирану трупу, поред суседа који има две стациониране трупе, мора имати најмање једног суседа који такође нема стационирану трупу, настаје *проблем ограничене римске доминације*. Слично, ако се трупе, уместо по чворовима, распоређују по гранама графа, добија се *проблем римске доминације на гранама*, односно, ако је дозвољено распоређивање трупа и по чворовима и гранама графа, добија се *проблем мешовите римске доминације*.

Иако је основа овог проблема историјског карактера, прави разлог проучавања проблема римске доминације се заправо огледа у њиховој примени данас. Наиме, проблем римске доминације се еквивалентно може пресликати на велики број проблема који се тичу распоређивања различитих врста ресурса.

## 1.2 Основни појмови и дефиниције

Нека су функције  $f, g_i : S \rightarrow \mathbb{R}$ ,  $i = 1, \dots, m$ , дефинисане на неком скупу  $S$ . Проблем одређивања решења  $x$  из скупа допустивих решења  $X \subseteq S$ ,  $X = \{x \in S \mid g_i(x) \leq 0, \quad i = 1, 2, \dots, m\}$ , за које функција  $f$  достиже минималну или максималну вредност, назива се *проблемом оптимизације*. Постоји велики број проблема који се математички могу моделовати као проблеми оптимизације: минимизација трошкова у производњи, минимизација трошкова при распоређивању запослених, при транспорту робе, максимизација профита, као и многи други.

Проблем оптимизације чији се *простор решења* може представити коначним или пребројиво бесконачним, дискретним скупом  $S$ , назива се *проблемом дискретне оптимизације* и у његовом најапстрактнијем математичком облику, може представити следећом формом:

$$\begin{array}{l} \min \text{ (или } \max \text{ )} \quad f(x) \\ \text{при ограничењима} \quad x \in X. \end{array} \quad (*)$$

Функција  $f$  проблема (\*) назива се *функцијом циља*, а решење  $x^* \in X$  *оптималним решењем*, уколико има најмању (највећу) вредност функције циља у односу на сва решења из допустивог скупа  $X$ ,  $X \subseteq S$ .

Са обзиром да проучавање простора решења представља основу поступка одређивања оптималног решења, дискретна оптимизација се сматра граном *комбинаторике*. Овај став произилази из податка да у појму дискретне оптимизације апсолутно централну улогу има неизбежан избор између узајамно искључивих случајева.

Проблеми дискретне оптимизације код којих су функција циља  $f$  и функције ограничења  $g_i$ ,  $i = 1, \dots, m$ , дефинисане као линеарне функције, могу се класификовати у проблеме *линеарног програмирања* (енг. Linear Programming, LP). Иначе, уколико макар једна од поменутих функција није линеарна, тада се такви проблеми класификују у проблеме *нелинеарног програмирања* (енг. Non Linear Programming, NLP). Проблеми линеарног програмирања као и проблеми који ће бити разматрани у тези



могу се представити на следећи начин:

$$\min \text{ или } \max \sum_{i=1}^n c_i x_i$$

при ограничењима:

$$\begin{aligned} \sum_{i=1}^n a_{ij} x_i &\geq b_j, & j = 1, \dots, m, \\ x_i &\geq 0, & i = 1, \dots, n, \\ x_i &\in Z, & i \in I. \end{aligned} \quad (**)$$

Променљиве одлучивања  $x_i$  условљене су ненегативношћу, са  $m$  линеарних неједнакости одређених коефицијентима  $a_{ij}$  и  $b_j$  и захтевом да свако  $x_i$ , где је  $i \in I$  ( $I \subseteq \{1, 2, \dots, n\}$ ), буде цео број. Циљ овако дефинисаног проблема је да се одреде вредности променљивих  $x_i$  такве да сума  $\sum_{i=1}^n c_i x_i$  буде минимална (или максимална). Проблем (\*\*), назива се *проблемом целобројног линеарног програмирања* (енг. Integer Linear Programming, ILP) уколико је  $I = \{1, 2, \dots, n\}$ , у супротном се, ако  $I$  није празан скуп, назива *проблемом мешовитог целобројног линеарног програмирања* (енг. Mixed Integer Linear Programming, MILP).

Интуитивно је јасно да су проблеми дискретне оптимизације најчешће тешки за решавање. Наиме, кардиналност скупа допустивих решења који треба испитати, а који је великих димензија, расте експоненцијално са порастом броја променљивих. На пример, ако узмемо да неки проблем има 200 бинарних променљивих, тада број могућих решења тог проблема износи  $2^{200}$ , односно око  $1.6 \cdot 10^{60}$ . Ако сада узмемо да исти проблем има 201 променљиву, кардиналност његовог скупа решења је два пута већа. Дакле, ако се овај проблем решава испитивањем вредности свих решења, тада је за 201 променљиву потребно проверити два пута више могућих решења. Ова огромна и експоненцијално растућа величина скупа допустивих решења заправо онемогућава комплетну проверу свих решења чак и за проблеме малих димензија, а чест је случај да је провера и најмањег дела скупа решења рачунски неизводљива.

**Фундаментални концепти теорије комплексности.** Као што је раније поменуто, у савременом животу постоји велики број проблема који се могу формулисати као проблеми оптимизације, а са тим у вези, и велики број различитих класа проблема оптимизације. Примера ради, проблем

рутирања возила представља једну класу проблема, док проблем распо-  
ређивања возача сасвим другу. Уколико проблем рутирања решавамо  
за неки конкретно задат возни парк, тада се може рећи да решавамо  
проблем за конкретно задату *инстанцу*. Инстанце које припадају ис-  
тој класи проблема могу се међусобно разликовати на основу више ра-  
зличитих параметара. Најчешће их карактерише њихова величина, тј.  
број улазних података који називамо *димензијом*. Имплицитно је јасно  
да проблем може да обухвата инстанце изузетно различитих величина.  
Остављајући по страни питање о томе како се за једну инстанцу де-  
финише величина, а како се заиста дефинише обим рачунања те ин-  
станце, сигурно се може очекивати да ће се обим рачунања повећати са  
повећањем величине инстанце. Са тим у вези, фундаменталне разлике  
у решивости између различитих класа проблема представљају предмет  
проучавања *теорије комплексности израчунавања* (енг. complexity theory).  
Теоријом комплексности се обим рачунања одговарајуће математичке  
форме изражава у функцији од димензије саме инстанце и на тај начин  
врши категоризација проблема. Време решавања  $t(n)$  проблема димен-  
зије  $n$  зависи од брзине рачунара на коме се тај проблем решава и  
може се изразити у секундама, минутима или некој другој сличној вели-  
чини. Иако  $t(n)$  представља процесорско време потребно да алгоритам  
реша неки проблем димензије  $n$ , често се уместо тачног записа функције  
 $t(n)$  за процену ефикасности алгоритама и асимптотску анализу користи  
Bachmann-Landau-ова нотација *велико*  $O$  и каже да се алгоритам извршава  
у времену  $O(g(n))$ <sup>1</sup>. На овај начин, процена времена израчунавања “у  
најгорем случају”, одражава да се баш свака инстанца може ефикасно  
решити у том времену. На пример, ако се број рачунских радњи неког  
алгоритма, при решавању инстанце димензије  $n$ , може изразити поли-  
номом петог степена, тада се може рећи да је тај алгоритам сложености  
 $n^5$ , односно  $O(n^5)$ . Слично, сложеност алгоритама може се оценити са  
 $O(1)$ ,  $O(n)$ ,  $O(n \log(n))$ ,  $O(2^n)$ , итд.

Алгоритам се сматра ефикасним уколико се његова сложеност може  
оценити са  $O(n^k)$  ( $k$  је константа која зависи од проблема који се решава,  
док  $n$  представља димензију инстанце):

---

<sup>1</sup>За две позитивне функције  $t, g : \mathbb{N} \rightarrow \mathbb{R}$  кажемо да је алгоритам брзине  $t(n)$ , односно  
асимптотске сложености  $O(g(n))$  ако постоје позитивне константе  $n_0$  и  $c$  такве да за  
свако  $n \geq n_0$  важи  $t(n) \leq c \cdot g(n)$ .

- Претпоставимо да за  $n = 10$  постоји алгоритам логаритамске сложености коме је потребно сат времена рада, алгоритам сложености  $O(n^5)$  коме је потребан само један минут и шема сложености  $O(2^n)$  којој је потребна само једна секунда за налажење решења ове инстанце. Тада ће за  $n = 100$  алгоритам логаритамске сложености захтевати два сата, алгоритам полиномијалне сложености потрошити 69.4 дана а експоненцијалној шеми ће бити потребно више од  $10^{17}$  векова за решавање истог проблема<sup>2</sup>.

Дакле, са повећањем димензије проблема могу се видети драстичне разлике у времену извршавања алгоритама, који алгоритам логаритамске сложености чине много пожељнијим него алгоритам полиномијалне а овај много пожељнији од алгоритма експоненцијалне сложености.

Поред физичког рачунара, сложеност алгоритма може се оценити коришћењем апстрактних рачунарских система (на пример, Тјурингове машине). Заправо, имајући у виду модел Тјурингове машине, може се јасно дефинисати појам *величина* инстанце. Тјурингова машина не решава проблем директно. Уместо тога, Тјурингова машина, по ћелијама бесконачне траке, кодира проблем симболима машинског алфабета и под величином инстанце подразумева дужину записа те инстанце, односно број заузетих ћелија на траци Тјурингове машине. Како је величина инстанце дефинисана дужином кода проблема, ова чињеница сугерише да, потенцијално, исти проблем, у зависности од избора кодирања, може да има веома различите величине. Међутим, у већини случајева на анализу комплексности не утичу разлике у кодирањима јер су сва разумна кодирања еквивалентна у смислу да једно кодирање може бити изражено као полиномијална функција неког другог. Ипак, постоје разлике у кодирањима које се не могу игнорисати. На пример, позната бинарна, октална, децимална и слична кодирања, која за кодирања константи проблема користе фиксирани број цифара, и, рецимо, унарни запис бројева (број се записује одговарајућим бројем цртица), дају различите сложености алгоритама за исти проблем. Како у унарном запису дужина записа проблема полиномијално расте са повећањем величине константи а код записа са фиксним бројем цифара дужина записа проблема расте логаритамски, потпуно је замисливо да алгоритам решава проблем у унарном запису у полиномијалном времену а да проблем у запису са фиксним бро-

---

<sup>2</sup>Подаци су преузети из [2].

јем цифара решава у експоненцијалном времену. Због тога је уобичајено да се проблем класификује као решив у полиномијалном времену само уколико је полиномијалан када су константе проблема задате у запису са фиксним бројем цифара. Није немогуће да неки проблеми, који се данас сматрају полиномијално решивим, престану то да буду уколико се дође до открића новог супер густог начина кодирања који ће логаритамски скратити дужину улазних константи проблема.

**Проблеми одлучивања.** Међу проблемима дискретне оптимизације велику пажњу привлаче проблеми који, уместо потпуног вектора решења или оптималне вредности, као решење проблема добијају одговор у облику "ДА" или "НЕ". Овакви проблеми називају се *проблемима одлучивања*. Сваки проблем оптимизације може се свести на проблем одлучивања, а да би се видела директна веза између проблема оптимизације и проблема исказаног у облику проблема одлучивања, посматрајмо познати проблем трговачког путника:

*Нека је дат граф  $G = (V, E)$  чије су гране  $(i, j)$  тежине  $w_{ij}$ . Да ли за дати граф постоји затворени пут дужине не веће од  $w$  којим се обилазе сви чворови тог графа?*

*Класа P проблема.* Проблеми одлучивања могу бити различите сложености и обима, од лако решивих па све до проблема за које је доказано да се не могу решити ни једним алгоритмом. Проблеми за које је показано да су полиномијалне сложености припадају класи најбоље решених проблема, класи P проблема. Према томе, класи P проблема припадају проблеми за које постоји Тјурингова машина која даје тачан "ДА" или "НЕ" одговор у ограниченом броју корака рада машине, а чија је граница изражена полиномом од дужине записа проблема.

*Класа CoP проблема.* Уколико се у претходном примеру постави обрнуто питање, односно уколико се пита да ли за дати граф не постоји затворени пут који обилази сваки чвор графа дужине не веће од  $w$ , добија се комплемент проблема P. Дакле, комплемент проблема одлучивања представља проблем у коме су одговори "ДА" и "НЕ" заменили места. Овакав прелаз са питања у коме се проверава да ли за неки проблем постоји решење на питање у којима се проверава да ли тај проблем нема решење заправо је значајнији него што се чини. Међутим, за проблеме из класе P овакав прелаз није од великог значаја јер њихови

комплементи такође припадају класи  $P$  проблема.

*Класа  $NP$  проблема.* Да би се проблеми класификовали као проблеми који припадају класи  $P$  проблема, проблем одлучивања мора бити решив у полиномијалном времену. Међутим, шта се дешава ако је познато само неко решење проблема? Често је провера којом би проблем одлучивања за конкретно задато решење дао одговор "ДА" релативно лака. Узимајући проблем трговачког путника за пример, релативно брзо може да се провери да ли предложени пут задовољава све услове допустивости. Ако је предложени пут допустив, питање његовог постојања је решено. Уколико предложени пут не задовољава неко ограничење, све што може да се уради је да се тражи неко друго решење.

Основна карактеристика  $NP$  проблема је да се не могу лако решити али зато провера њихових потенцијалних решења може бити релативно лака. Конкретно, класа  $NP$  проблема укључује све оне проблеме одлучивања на које се може одговорити у полиномијалном времену (у односу на проблем и предложено решење) будући да се одговарајућом Тјуринговом машином кодирају и проблем и његово потенцијано решење. Скраћеница  $NP$  потиче од *недетерминистички у полиномијалном времену*.

*Класа  $CoNP$  проблема.* Комплементи проблема класе  $NP$  проблема, тј. проблеми одлучивања из класе  $NP$  код којих одговору "ДА" одговара одговор "НЕ" припадају класи  $CoNP$ . Раније је већ напоменуто да се алгоритми за решавање проблема класе  $P$  могу прилагодити тако да решавају своје комплементарне "НЕ" парове, тј. да је  $P = CoP$ . За сваки проблем класе  $NP$  постоји Тјурингова машина таква да се њено извршавање прекида у полиномијалном времену за сваки улаз, а за разлику од класе  $P$  проблема, потенцијално решење представља део тог улаза. Тако, ако се машина заустави са одговором "ДА", знамо да је одговор на основни проблем "ДА". Међутим, ако се машина заустави а да није доказала позитиван одговор, јако мало се може закључити о том проблему. Можда је тачан одговор "ДА" али је на улазу погрешно потенцијално решење а можда је и одговор "НЕ". Једини закључак који може да се изведе је да кодирани проблем и укључено потенцијално решење не доводе до одговора "ДА". Како се за одређени проблем може са сигурношћу тврдити да има одговор "НЕ", тј. како се решавају проблеми из класе  $CoNP$ ? У потпуности је могуће да је једини ваљан

приступ испитивање свих могућих потенцијалних решења, закључујући "НЕ" уколико ни једно од потенцијалних решења не води до доказивог "ДА". Будући да је број потенцијалних решења експоненцијалне природе, можда није могуће потврдити "НЕ" одговоре са потенцијалним решењима полиномијалне дужине. Стога, постоје многи чланови из класе CoNP који не припадају класи NP проблема.

Иако је потпуно јасно да  $CoP = P$ , однос између класа P и NP проблема и даље представља нерешен проблем. Заправо, одговор на питање "да ли се проблеми, код којих се за неко решење може добити "ДА" или "НЕ" одговор у полиномијалном времену, такође могу решити у полиномијалном времену?" је и даље непознат.

Са тим у вези, идеја да се један проблем решава применом ефикасног потпрограма коришћеног у решавању неког другог проблема има кључну улогу код анализе комплексности проблема. На пример, инверз матрице димензије  $n \times n$  може се решити поступком који се користи за решавање система од  $n$  једначина са  $n$  непознатих.

**Дефиниција 1.1.** *Кажемо да се проблем (A) може свести на проблем (B) уколико се било који алгоритам за решавање проблема (B) може превести на алгоритам за решавање проблема (A). Проблем (A) се полиномијално своди на проблем (B) ако алгоритам полиномијалне сложености за решавање проблема (B) повлачи постојање алгоритма полиномијалне сложености за решавање проблема (A). Полиномијално свођење проблема (A) на проблем (B) обележава се са  $(A) \propto (B)$ .*

**Теорема 1.1.** *(Лема о сводљивости, [3]) Ако проблем (A) може да се сведе на проблем (B) у полиномијалном времену, тада важи:*

- $(B) \in P \Rightarrow (A) \in P;$
- $(B) \in NP \Rightarrow (A) \in NP.$

**NP-еквивалентни проблеми.** Проблеми на које се полиномијално сведе сви проблеми из класе NP називају се *NP-тешким*. Иако се ни један од тих проблема не може решити полиномијално, испоставило се да је фамилија NP-тешких проблема изузетно велика. Она, практично, укључује све проблеме који већ дуго заокупљају пажњу људи који се

баве дискретном оптимизацијом. Међу њима су проблем трговачког путника, бојење графова, покривање скупова, проблем ранца и разни други проблеми.

Изузетан допринос за откривање великог броја NP-тешких проблема и развој теорије комплексности имају Stephen Cook [4] и Richard Karp [5]. Cook је дефинисао проблем задовољивости (енг. satisfiability problem, (SAT) проблем) и показао да је тако дефинисан проблем NP-тежак, док је Karp [5] демонстрирао да се проблем задовољивости може свести на велики број проблема дискретне оптимизације из литературе.

Уведимо скуп бинарних променљивих  $\{x_1, \dots, x_n\}$  и њихове комплементе обележимо са  $\bar{x}_i$ ,  $i = 1, \dots, n$ . Свакој променљивој додељује се вредност "тачно" или "нетачно" у смислу да је  $x_i$  тачно ако и само ако је  $\bar{x}_i$  нетачно. Проблемом задовољивости поставља се следеће питање:

Ако је над датим скупом променљивих дата конјункција низа дисјункција, да ли се датим променљивим могу доделити вредности тако да задата конјункција буде таутологија?

Уколико такво додељивање постоји, за тај израз кажемо да је *задовољив*. Сада се Cook-ова теорема о припадности (SAT) проблема класи NP тешких проблема може изразити на следећи начин.

**Теорема 1.2.** *Сваки проблем  $Q \in NP$  може се полиномијално свести на (SAT).*

На основу Cook-ове теореме може се видети да је (SAT) једнако тежак за полиномијално решавање као и било који други проблем из класе NP проблема.

*NP комплетни проблеми.* Проблем који припадају и класи NP-тешких и класи NP проблема називају се NP комплетним проблемима. Свака два NP комплетна проблема ( $Q$ ) и ( $R$ ) имају својства  $(Q) \propto (R)$  и  $(R) \propto (Q)$ . Са тим у вези, како је релација  $\propto$  већ рефлексивна и транзитивна, следи да NP-комплетни проблеми формирају класу еквиваленције, тј. ако се за један проблем покаже да постоји алгоритам који га решава у полиномијалном времену, тада такав алгоритам постоји за све проблеме из ове класе.

Појмови и значења описани у овом одељку настали су под утицајем и делимично су преузети из књиге [6]. Будући да прича о P и NP проблемима, као и о релацијама  $P=NP$  и  $P \neq NP$  превазилази оквире ове тезе, дате су само основне дефиниције и теорема без доказа. Много детаљније се о овим проблемима може наћи у [2] и [6].

Имајући у виду да време налажења оптималних решења одређених проблема комбинаторне оптимизације може бити недопустиво велико, јавља се потреба за алгоритмима који ће одредити приближно решење тих проблема у допустивом временском интервалу. Такви алгоритми називају се хеуристикама и они не гарантују да је пронађено решење оптимално. Заправо, хеуристика се сматра добром уколико је добијено субоптимално решење одговарајуће по неком критеријуму (време извршавања, процени приближности оптималном решењу итд.).

### 1.3 Ознаке коришћене у раду

Нека је дат граф  $G = (V, E)$ . Користимо следеће ознаке:

$V = \{v_1, v_2, \dots, v_n\}$  - скуп чворова графа  $G$ ;

$E = \{e_1, e_2, \dots, e_m\}$  - скуп грана графа  $G$ ;

$|V|$  - кардиналност скупа  $V$ , најчешће у ознаци  $n$ , за граф са  $n$  чворова каже се да је *реда*  $n$ ;

$|E|$  - кардиналност скупа  $E$ , најчешће у ознаци  $m$ ;

$A = [a_{ij}]_{n \times n}$  - матрица повезаности чворова графа  $G$ ,

$$a_{ij} = \begin{cases} 1, & \text{ако су чворови } v_i \text{ и } v_j \text{ повезани граном} \\ 0, & \text{иначе;} \end{cases}$$

$e = (u, v)$  - оријентисана грана,  $e = \{u, v\}$  - неоријентисана грана,  $e = (u, u)$  - петља;

$deg_G(u)$  - степен чвора  $u \in V$  (број грана инцидентних са чвором  $u$ );

$\delta(G) = \min \{deg_G(u) \mid u \in V\}$  - минимални степен чворова у графу ( $\delta$  ако је граф познат);



$\Delta(G) = \max \{deg_G(u) \mid u \in V\}$  - максимални степен чворова у графу ( $\Delta$  ако је граф познат);

$N_G(v) = \{u \in V \mid \{u, v\} \in E\}$  - отворена околина чвора  $v \in V$  ( $N_v$  ако је граф познат);

$N_{G[v]} = N_G(v) \cup \{v\}$  - затворена околина чвора  $v \in V$  ( $N_{[v]}$  ако је граф познат);

$N(S) = (\cup_{v \in S} N_v) \setminus S$  - отворена околина скупа  $S$ ,  $S \subseteq V$ ;

$N[S] = N(S) \cup S$  - затворена околина скупа;

$\overline{G}$  - комплемент графа  $G$  (граф са истим бројем чворова као и  $G$  и особином да су два чвора суседна у  $\overline{G}$  ако нису суседна у  $G$ ).

$d(u, v)$  - дужина најкраћег пута између чворова  $u$  и  $v$ ;

$ecc(v) = \max \{d(v, w) : w \in V\}$  - ексцентрицитет чвора (енг. eccentricity);

$diam(G) = \max \{ecc(v) : v \in V\}$  - дијаметар графа (енг. diameter);

$rad(G) = \min \{ecc(v) : v \in V\}$  - радијус графа (енг. radius);

$S - pn$  приватно суседство чвора (чвор  $u$  се назива *приватним суседом* (енг. private neighbor) чвора  $v$  ( $v \in S \subseteq V$ ), у односу на  $S$  (у ознаци:  $u$  је  $S - pn$  од  $v$ ), уколико је  $u \in N_{[v]} \setminus N[S \setminus \{v\}]$ );

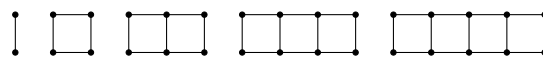
$pn(v, S) = N_{[v]} \setminus N[S \setminus \{v\}]$  приватна околина чвора  $v \in S$  у односу на  $S$ ;

$epn(v, S) = pn(v, S) \setminus \{v\}$  спољна приватна околина (енг. external private neighbor set of  $v$ ) чвора  $v$ .

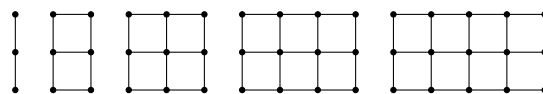
Појмови коршћени у релевантним радовима из литературе:

- скуп  $S$  назива се *нередундантним* ако је  $pn(v, S) \neq \emptyset$  за свако  $v \in S$ ;
- скуп  $S \subseteq V$  назива се *независним* или *стабилним* ако ни која два чвора у скупу  $S$  нису суседна;
- број елемената, у ознаци  $i(G)$ , представља кардиналност максималног независног скупа  $S$  у  $G$ ;

- граф  $G = (V, E)$  је *регуларан*, односно  $r$ -регуларан ако су сви чворови графа  $G$  степена  $r$  ( $\delta(G) = \Delta(G)$ , односно  $\delta(G) = \Delta(G) = r$ );
- граф  $G = (V, E)$  је *комплетан* ако је  $\delta(G) = \Delta(G) = n - 1$ ,  $|V| = n$ . Комплетни графови степена  $n$  означавају се са  $K_n$ ;
- граф се назива *простим* уколико нема петљи нити паралелних грана.
- граф  $G = (V, E)$  са особином  $V = V_1 \cup V_2$ ,  $V_1 \cap V_2 = \emptyset$ , при чему за сваку грану из  $E$  важи да јој је један крај из скупа  $V_1$  а други из скупа  $V_2$  назива се *бипартитним* или *биграфом*;
- бипартитни граф  $G = (V_1 \cup V_2, E)$  се назива *комплетно бипартитним* графом, ако су сви елементи скупа  $V_1$  повезани са свим елементима скупа  $V_2$ . За  $n_1 = |V_1|$  а  $n_2 = |V_2|$  комплетно бипартитни граф се обележава са  $K_{n_1 \times n_2}$ . Комплетно бипартитни граф  $K_{1,n}$  се назива *звездом*;
- нека је дат граф  $G = (V, E)$ . Граф облика  $\Gamma = (U, T)$  при чему је  $U \subseteq V$  и  $T \subseteq E$  такав да су крајеви грана из  $T$  чворови из  $U$  назива се *подграфом графа  $G$*  (енг. subgraph). Ако је  $U \neq V$ , граф  $\Gamma$  је прави подграф графа  $G$ ;
- За два произвољна графа  $G$  и  $H$  Декартов производ (енг. Cartesian product) се дефинише као граф  $G \square H$  са скупом чворова  $\{(u, v) : u \in G, v \in H\}$ . При том, два чвора  $(u_1, v_1)$  и  $(u_2, v_2)$  су суседна у  $G \square H$  ако и само ако је једно од следећих тврђења тачно: 1)  $u_1 = u_2$  и  $v_1$  је сусед са  $v_2$  у  $H$ ; 2)  $v_1 = v_2$  и  $u_1$  је сусед са  $u_2$  у  $G$ . Ако је  $G = P_m$  и ако је  $H = P_n$  тада се Декартов производ  $G \square H$  назива  $m \times n$  *грид графом* и обележава са  $G_{m,n}$ . На сликама 3 и 4 приказани су Декартови производи  $P_2 \square P_n$  и  $P_3 \square P_n$ , за  $1 \leq n \leq 5$ .

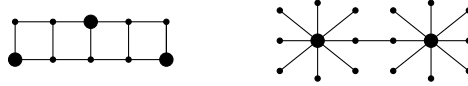


Слика 3: Грид графови  $G_{2,n}$ ,  $1 \leq n \leq 5$ .



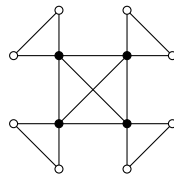
Слика 4: Грид графови  $G_{3,n}$ ,  $1 \leq n \leq 5$ .

- Скуп чворова  $S$ ,  $S \subset V$  графа  $G = (V, E)$  назива се *2-покривањем* (енг. 2-packing graph) ако за сваки пар чворова  $u, v \in S$  важи  $N[u] \cap N[v] = \emptyset$ . Максимална кардиналност скупа  $S$  који представља *2-покривање* графа  $G$  означава се са  $P_2(G)$  (илустрације ради, на слици 5 су за графове  $G_1$  и  $G_2$  чворови из скупова  $P_2(G_1)$  и  $P_2(G_2)$  увећани).



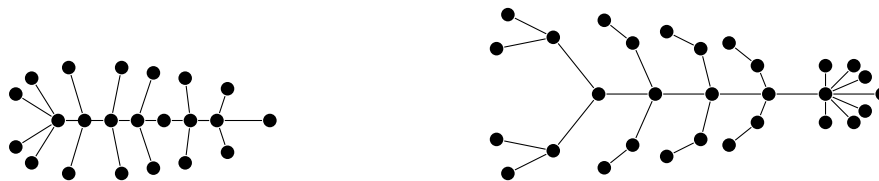
Слика 5: Графови  $G_1$  (лево) и  $G_2$  (десно)

- За два графа  $G$  и  $H$ , њихов кардинални производ  $G \times H$  дефинише се на следећи начин: за скуп чворова важи релација  $V(G \times H) = V(G) \times V(H)$ , док су два чвора  $(g, h), (g', h') \in V(G \times H)$  суседна ако и само ако је  $g$  суседно са  $g'$  у  $G$  и  $h$  сусед са  $h'$  у  $H$ .
- Унија два графа  $G$  и  $H$ , у ознаци  $G + H$ , дефинише се скупом чворова  $V(G) \cup V(H)$  и скупом грана  $E(G) \cup E(H) \cup \{vw : v \in V(G), w \in V(H)\}$ .
- Венац граф (енг. согона graph), у ознаци  $G \star H$ , се формира од једне копије графа  $G$  и  $|V(G)|$  копија графа  $H$ , тако да  $i$ -ти чвор од  $G$  буде суседан са свим чворовима  $i$ -те копије од  $H$ . Конкретно,  $G \star K_1$  представља граф сачињен од једне копије графа  $G$  и  $|V(G)|$  копија графа  $K_1$ , односно сачињен је од графа  $G$  и  $|V(G)|$  копија чвора  $v'$  повезаним на следећи начин: сваки чвор графа  $G$  је повезан са једном копијом графа  $K_1$ , тј. за сваки чвор  $v \in G$  и тачно једну копију  $v'$  графа  $K_1$  постоји грана  $vv'$  која их повезује. На слици 6 приказан је венац графова  $G = K_4$  (чији су чворови обојени црном бојом) и  $H = P_2$  (чији су чворови обојени белом бојом).



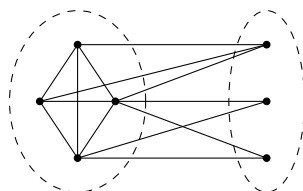
Слика 6: Венац граф  $K_4 \star P_2$ .

- *Гусенични граф* има структуру стабла које, када се уклоне листови и њима инцидентне гране, постаје пут (настали пут назива се *кичмом* гусенице), Слика 7 (лево).



Слика 7: Гусенични граф (лево) и Јастог граф (десно)

- *Јастог граф* има структуру стабла које, када се уклоне сви листови и њима инцидентне гране, постаје гусенични граф. Кичму јастога представља кичма њему одговарајуће гусенице, Слика 7 (десно).
- Граф  $G = (V, E)$ , чији се чворови могу партиционисати у два скупа,  $X$  и  $Y$ , тако да чворови скупа  $X$  чине независан скуп а да чворови скупа  $Y$  формирају комплетан граф, назива се *подељеним графом* (енг. split graph), Слика 8.



Слика 8: Подељени граф

## 1.4 Проблеми доминације и римске доминације на графу

Због своје примене, проблем доминације на графу представља један од најпроучаванијих проблема комбинаторне оптимизације.

**Дефиниција 1.2.** Скуп чворова  $S \subseteq V$  неког графа  $G = (V, E)$  назива се доминантним скупом уколико је сваки чвор  $u \in V$  тог графа или у скупу  $S$  или има бар једног суседа који је у скупу  $S$ . Најмања кардиналност међу свим доминантним скуповима дефинисаним за граф  $G$  обележава се са  $\gamma(G)$ . Број  $\gamma(G)$  назива се доминантним бројем за граф  $G$ .

**Дефиниција 1.3.** Нека је дат граф  $G = (V, E)$ . Функција  $f : V \rightarrow \{0, 1\}$  назива се доминантном функцијом ако и само ако је скуп  $V_1 = \{v \in V \mid f(v) = 1\}$  доминантан скуп.

У даљем разматрању сматраћемо да су сви графови прости.

**Дефиниција 1.4.** Нека је дат граф  $G = (V, E)$  и нека су дата два скупа чворова  $U$  и  $W$ ,  $U, W \subseteq V$ . Кажемо да скуп  $U$  доминира скуп  $W$  уколико је сваки чвор  $w \in W$  или у  $U$  или има суседа у  $U$ .

Проблем доминације представља проблем одређивања доминантног броја  $\gamma(G)$  за произвољан прост граф  $G$  и одговарајуће доминантне функције  $f$  за коју се тај број достиже. Постоји велики број различитих врста проблема доминације на графу, видети [7]. Један од значајнијих проблема доминације је проблем римске доминације који је предмет ове тезе.

**Дефиниција 1.5.** ([8]) Нека су дати граф  $G = (V, E)$  и функција  $f : V \rightarrow \{0, 1, 2\}$ . Обележимо са  $V_i = \{v \in V \mid f(v) = i\}$ ,  $i = 0, 1, 2$ . За функцију  $f$  рећи ћемо да је функција римске доминације ( $f_{RD}$  функција) ако скуп  $V_2$  доминира скуп  $V_0$ . Број  $w(f) = \sum_{v \in V} f(v) = n_1 + 2n_2$  где су  $n_1 = |V_1|$  и  $n_2 = |V_2|$  назива се тежином функције  $f$ . Број  $\gamma_R(G)$ , који представља најмању тежину међу свим  $f_{RD}$  функцијама дефинисаним за граф  $G$ , назива се римским доминантним бројем, док се  $f_{RD}$  функција  $f$ , за коју важи  $w(f) = \gamma_R(G)$ , назива  $\gamma_R$ -функцијом за граф  $G$ .

Проблем римске доминације (енг. Roman domination problem) представља проблем одређивања римског доминантног броја  $\gamma_R(G)$  за произвољан прост граф  $G$  и одговарајуће  $\gamma_R$ -функције.

Сада се проблем из уводног поглавља, који се тицао одређивања оптималне стратегије Римског царства у време цара Константина, може математички представити проблемом римске доминације (у даљем тексту  $RD$  проблемом). Свака провинција биће представљена чвором, а чворови ће бити повезани гранама уколико су провинције имале заједничку границу или, уколико је било познато да се морским путем може брзо прећи из једне провинције у другу. Вредности функције римске доминације ће за сваку провинцију означавати да ли је у тој провинцији стационирана једна, две или ни једна легија. Провинција ће се сматрати *сигурном* од напада уколико има бар једну стационирану легију, односно *заштићеном* од напада уколико има суседа са бар две стациониране легије. На овај начин, једна легија може да брани суседну нападнуту провинцију док ће друга наставити да штити провинцију у којој је стационирана. Описаним поступком проблем са којим се сусрео цар

Константин се сада своди на проблем налажења одговарајуће функције римске доминације за граф који се формира од мапе Римског царства са почетка IV века.

**Дефиниција 1.6** ([9]). Нека је дат граф  $G = (V, E)$ . Функција  $f : V \rightarrow \{0, 1, 2\}$  назива се функцијом ограничене римске доминације, у ознаци  $f_{\mathcal{RRD}}$ , уколико за сваки чвор  $u \in V$  такав да је  $f(u) = 0$  постоје суседни чворови  $v, w \in V$  такви да је  $f(v) = 2$  и  $f(w) = 0$ . Вредност  $f(u)$  назива се тежином чвора  $u$ . Тежина функције  $f$  одређује се по формули  $w(f) = \sum_{u \in V} f(u)$ .

Проблем одређивања  $f_{\mathcal{RRD}}$  функције најмање тежине назива се проблемом *ограничене римске доминације* (енг. *Restrained Roman Domination Problem*, у даљем тексту  $\mathcal{RRD}$  проблем) док се добијена кардиналност назива *ограниченим римским доминантним бројем* и обележава са  $\gamma_{rR}(G)$ .

**Дефиниција 1.7** ([10]). Нека су дати граф  $G = (V, E)$  и функција  $f : V \rightarrow \{0, 1, 2\}$ . Обележимо са  $V_i = \{v \in V \mid f(v) = i\}$ ,  $i = 0, 1, 2$ . Чвор  $v \in V_0$  назива се *незаштићеним* уколико нема суседа у скупу  $V_1 \cup V_2$ . Функција  $f$  назива се функцијом *слабе римске доминације* ( $f_{\mathcal{WRD}}$  функција) ако сваки чвор  $u \in V_0$  има бар једног суседа  $v \in V_1 \cup V_2$  и нема незаштићених чворова у односу на функцију  $f' : V \rightarrow \{0, 1, 2\}$  дефинисану са  $f'(u) = 1$ ,  $f'(v) = f(v) - 1$  и  $f'(w) = f(w)$  за свако  $w \in V \setminus \{u, v\}$  (тј. сваки чвор из  $V_0'$  има суседа у  $V_1' \cup V_2'$  где је  $V_i' = \{v \in V \mid f'(v) = i\}$ ,  $i = 0, 1, 2$ ). Тежина  $w(f)$  функције  $f$  дефинише се као  $\sum_{u \in V} f(u)$ . Слаби римски доминантни број,  $\gamma_r(G)$ , представља најмању тежину међу свим  $f_{\mathcal{WRD}}$  функцијама дефинисаним за граф  $G$ , тј.  $\gamma_r(G) = \min \{w(f) : f \text{ је } f_{\mathcal{WRD}}\}$ . Функција слабе римске доминације тежине  $\gamma_r$  назива се  $\gamma_r$ -функцијом.

Проблем слабе римске доминације (енг. *Weak Roman Domination Problem*, у даљем тексту  $\mathcal{WRD}$  проблем) представља проблем одређивања броја  $\gamma_r(G)$  и одговарајуће  $\gamma_r$ -функције за произвољан прост граф  $G$ . Заправо,  $\mathcal{WRD}$  проблемом се свака провинција Римског царства сматра сигурном од напада уколико поседује бар једну стационирану легију, односно заштићеном од напада уколико има суседа са бар једном стационираном легијом уз услов да, у случају напада, премештање легије из једне провинције у њој суседну не нарушава сигурност и заштићеност осталих провинција. Приметимо да  $\mathcal{WRD}$  проблем представља динамички оптимизациони проблем док су  $\mathcal{RD}$  и  $\mathcal{RRD}$  статички проблеми.

Проблем римске доминације може се применити и на велики број различитих проблема распоређивања ресурса. На пример, посматрајмо проблем одржавања система неких фабрика. Како би фабрика радила без великих прекида, циљ је да сваки производни погон има тим за одржавање. Међутим, како се проблеми не дешавају тако често, у циљу минимизације трошкова фабрике, тимови за одржавање могу се распоредити тако да сваки производни систем или има свој тим или тим за одржавање може да дође до њега у кратком временском року (претпоставимо да се производња извршава на неколико различитих локација). Ако додатно претпоставимо да постоји реална потреба за истовременим сервисирањем два различита производна погона (или постоји потреба да два различита тима за одржавање раде истовремено), тада се овај проблем директно може посматрати као проблем римске доминације. Слично се може решавати проблем распоређивања сервера у коме сваки сервер може да изврши највише један захтев у датом моменту, проблем распоређивања антена или било којих других ресурса. Такође, ако постоји потреба да се ресурси премештају са једне локације на другу, тада се такав проблем може посматрати као проблем слабе римске доминације, или као нека друга варијанта проблема римске доминације. Поред наведених, проблем римске доминације може се искористити и приликом одабира нових локација за изградњу школа, болница, бензинских пумпи, продавница и многих других објеката.

## 2 Преглед релевантне литературе о проблемима римске доминације

### 2.1 Проблем римске доминације

Проблем римске доминације увели су Stewart [11] и ReVelle и Rosing [12] док су формалну математичку дефиницију овог проблема дали Cockayne и сар. у [8]. Иако се проблем римске доминације класификује као NP тежак за графове у општем смислу [13], [14], јединичне диск графове [15] и бипартитне и чордалне графове [16], Pagourtzis и сар. [17] показали су да за решавање  $\mathcal{RD}$  проблема постоји апроксимативни алгоритам сложености  $2 + 2\ln(n)$  али (осим ако је  $P = NP$ ) не постоји апроксимативни алгоритам сложености  $c \log(n)$  за неко  $c > 0$ . Додатно, Pagourtzis и сар. [17] показали су да је  $\mathcal{RD}$  проблем NP тежак и за планарне графове.

Поред основног проблема римске доминације уведено је и неколико његових уопштења. Наиме, релаксацијом постојећих и увођењем нових услова настало је више варијанти  $\mathcal{RD}$  проблема. На пример, Rad и Volkmann [18] су проучавали промену вредности римског доминантног броја насталу када се из графа обрише неки чвор. *Независну функцију римске доминације* (енг. independent Roman dominating function) увео је Adabi у [19], док су Rad и Volkmann [20] проучавали класу графова за коју је  $\gamma_R(G) = i_R(G)$  (где је  $i_R(G)$  независни римски доминантни број). *Римску функцију са јединственим одговором* (енг. unique response Roman dominating function) дефинисали су Rubalcaba и сар. [21] док су Pushpan и Mai [22] увеле функцију *римске доминације по гранама* (енг. edge Roman dominating function). *Проблем двоструке римске доминације* увели су Beeler и сар. [23], док су *проблем Италијанске доминације*, који је још познат под именом *римска {2}-доминација*, проучавали Chellali и сар. [24]. Abdollahzadeh Ahangar и сарадници су 2016 године на списак варијанти проблема римске доминације додали *проблем тоталне римске доминације* [25]. Одмах потом, настале су варијанте и овог проблема, [26], [27], [28], [29], [30], [31]. Поред два прегледна рада, [32] и [33], којим су Chellali и сар. пописали одређени број варијанти проблема римске доминације, још неке његове варијанте могу се наћи у [16], [34], [35] и [36], [37], [38].



## Особине функције римске доминације

Упркос чињеници да је појам римског доминантног броја уведен пре 20 година, вредност  $\gamma_R$  броја позната је само за одређене класе графова, док је за неке друге класе графова позната само горња или доња граница. У наставку су изложене особине римског доминантног броја и функције римске доминације познате у литератури.

**Теорема 2.1.** ([8]) *За сваки граф  $G$  важе следеће релације*

1.  $\gamma(G) \leq \gamma_R(G) \leq 2\gamma(G)$ .
2.  $\gamma_R(G) = \gamma(G)$  ако и само ако је  $G = \overline{K}_n$ .
3. Ако је  $G$  нетривијални повезани граф тада је

$$\gamma_R(G) = \min \{2\gamma(G \setminus S) + |S| : S \text{ је } 2\text{-покривање}\}.$$

Граф је римски уколико важи  $\gamma_R(G) = 2\gamma(G)$  (Соскаупе и сар. [8]). Односно, граф  $G = (V, E)$  је римски граф ако и само ако се може дефинисати  $\gamma_R$ -функција  $f : V \rightarrow \{0, 1, 2\}$  тако да је  $n_1 = |V_1| = 0$ ,  $V_1 = \{v \in V, f(v) = 1\}$ .

Узмимо да је  $\delta$  најмањи степен чвора у графу  $G = (V, E)$ , односно да је са  $\Delta$  означен највећи степен чвора графа  $G$  и нека је  $n = |V|$ . Тада се горња и доња граница за римски доминантни број могу дефинисати у функцијама од  $\delta$ ,  $\Delta$  и  $n$ .

**Теорема 2.2.** 1. ([39]) *Нека за граф  $G = (V, E)$  важи  $\delta > 0$ . Тада је*

$$\gamma_R(G) \leq 2 \left(1 - \frac{2^{\frac{1}{\delta}} \delta}{(1 + \delta)^{1 + \frac{1}{\delta}}}\right) n.$$

2. ([8], [13], [39]) *Нека је  $G$  граф реда  $n$ . Тада је*

$$\gamma_R(G) \leq 2n \frac{1 + \ln((1 + \delta)/2)}{1 + \delta}.$$

3. ([39]) *За довољно велико  $n$  постоји граф  $G$  за који важи*

$$\gamma_R(G) \geq \frac{2 \ln(\delta + 1) - \ln 4 + 2}{\delta + 1} n(1 + o(1)).$$

---

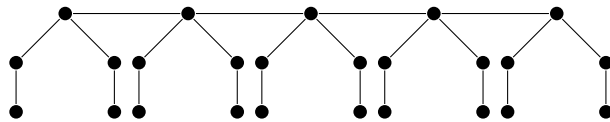
<sup>3</sup>У оригиналном раду [8] је направљена грешка која је исправљена у [39].

4. ([40]) *За сваки граф  $G$  реда  $n$  важи*

$$\gamma_R(G) \geq \frac{2n}{\Delta + 1}.$$

Тврђења 2 и 3 Теореме 2.2 представљају последице резултата добијеног под 1.

За путеве, цикле и комплетне графове реда  $n$  биће коришћене ознаке  $P_n$ ,  $C_n$  и  $K_n$ , док ће за графове који се састоје из  $m$  дисјунктних копија графа  $G$  бити коришћена ознака  $mG$ . На пример, на слици 9 приказан је граф  $5P_5$  сачињен од 5 дисјунктних копија путева  $P_5$  међусобно повезаних путем који пролази кроз централне чворове ових копија.



Слика 9: Граф  $5P_5$

**Теорема 2.3.** ([41])

1. Нека је  $T$  стабло реда  $n$  и нека је  $n \geq 3$ . Тада важи  $\gamma_R(T) \leq 4n/5$ .
2. Нека је  $T$  стабло реда  $n$ . Тада је  $\gamma_R(T) = 4n/5$  ако и само ако се  $V(T)$  може партиционисати на скуп индукованих путева  $P_5$  тако да је подграф индукован централним чворовима ових путева повезан.
3. Нека је  $G$  повезан граф реда  $n$ . Тада је  $\gamma_R(G) \leq 4n/5$ , при чему се једнакост достиже ако и само ако је  $G = C_5$  или је граф  $G$  добијен из  $\frac{n}{5}P_5$  додавањем повезаног подграфа на скуп центара компоненти од  $\frac{n}{5}P_5$ .

**Теорема 2.4.** ([41])

1. Нека је  $G$  реда  $n$ . Тада је  $\gamma_R(G) \leq n - \Delta(G) + 1$ .
2. Нека је  $G$  реда  $n$  и нека је  $n \geq 3$ . Важи релација  $5 \leq \gamma_R(G) + \gamma_R(\overline{G}) \leq n + 3$ , при чему се горња једнакост достиже једино у случајевима када је  $G$  или  $\overline{G}$  једнако или  $C_5$  или  $\frac{n}{2}K_2$ .
3. Нека је  $G$  реда  $n$  и нека важи  $n \geq 160$  и  $\text{diam}(G) = \text{diam}(\overline{G}) = 2$ . Тада је  $\gamma_R(G)\gamma_R(\overline{G}) \leq 16n/5$ .

4. Нека је  $G$  реда  $n$  и нека важи  $n \geq 160$ . За граф  $G$  важи  $\gamma_R(G)\gamma_R(\overline{G}) \leq 16n/5$ , док се једнакост достиже једино у случајевима када је  $G$  или  $\overline{G}$  једнако  $\frac{n}{2}C_5$ .

Поред резултата исказаних теоремом 2.4 Chambers и сар. [41] показали су да за одређене повезане графове са  $n$  чворова важи релација  $\gamma_R(G) \leq 8n/11$ . У [42] је доказано да за сваки повезани граф  $G$  са  $n \geq 3$  чворова важи  $|V_0| \geq n/5 + 1$ ,  $|V_1| \leq 4n/5 - 2$  и  $|V_2| \leq 2n/5$ . Горња граница  $\gamma_R(G) \leq 2n/3$  за граф  $G$  реда  $n$ , минималног степена  $\delta(G) \geq 3$ , дата је у [43].

Мобараку и Shekiholeslami [44] су границу броја римске доминације дефинисали у функцији од дијаметра и опсега (енг. girth). Опсег графа представља дужину најмањег циклa садржаног у том графу и обележава се са  $g(G)$ .

**Теорема 2.5.** ([44])

1. За сваки граф  $G$  са особином  $diam(G) = 2$  важи  $\gamma_R(G) \leq 2\delta$ . Једнакост се достиже за бесконачну фамилију графова.
2. За сваки повезани граф  $G$  важи релација

$$\gamma_R(G) \geq \left\lceil \frac{diam(G) + 2}{2} \right\rceil.$$

При том, дато ограничење достиже се за  $P_3$  и  $P_4$ .

3. За сваки повезани граф  $G$  са  $n$  чворова важи

$$\gamma_R(G) \leq n - \left\lfloor \frac{1 + diam(G)}{3} \right\rfloor.$$

При чему се граница достиже за неке графове.

4. За сваки повезани граф  $G$  реда  $n$ , код кога је  $\delta \geq 3$ , важи

$$\gamma_R(G) \leq n - \left\lfloor \frac{1 + diam(G)}{3} \right\rfloor - (\delta - 2) \left\lfloor \frac{diam(G) + 2}{3} \right\rfloor.$$

За сваки повезани граф  $G$  са особином  $diam(G) \geq 3$  важи  $\gamma_R(\overline{G}) \leq 4$ .

5. За граф  $G$  реда  $n$  код кога је  $g(G) \geq 3$  важи  $\gamma_R(G) \geq \left\lfloor \frac{2g(G)}{3} \right\rfloor$ .

6. Ако за граф  $G$  важи  $g(G) = 4$ , тада је  $\gamma_R(G) \geq 3$ . Једнакост важи ако и само ако је  $G$  бипартитни граф са партицијама  $X$  и  $Y$  при чему је  $|X| = 2$  и  $X$  има један чвор степена  $n - 2$  а други степена најмање 2.

7. Нека је  $G$  прост повезани граф реда  $n$ . Тада важи:

а)  $\gamma_R(G) \leq n - \lfloor \frac{g(G)}{3} \rfloor$  када је  $\delta(G) \geq 2$  и  $g(G) \geq 5$  (граница се достиже за  $C_n$ ,  $n \geq 5$ ).

б)  $\gamma_R(G) \geq 2\delta$  када је  $g(G) \geq 5$  (граница се достиже за  $C_5$  и  $C_6$ ).

в)  $\gamma_R(G) \geq 4(\delta - 1)$  када је  $\delta \geq 2$  и  $g(G) \geq 6$  (граница се достиже за  $C_6$ ).

г)  $\gamma_R(G) \geq 2\Delta$  када је  $\delta \geq 2$  и  $g(G) \geq 7$  (граница се достиже за  $g(G) = 7$ ).

Римски доминантни број за Декартов и кардинални производ два графа проучавали су Соскауне и сар. [40] и Клобуџар и Пулјић [45].

**Теорема 2.6.** ([40]) *За Декартов производ два графа важи*

1.

$$\gamma_R(P_m \square P_k) \leq 2 \left( \left\lceil \frac{mk}{5} \right\rceil + \left\lceil \frac{m}{5} \right\rceil + \left\lceil \frac{k}{5} \right\rceil \right).$$

2.

$$\left\lceil 2 \frac{mk}{5} \right\rceil \leq \gamma_R(C_m \square C_k) \leq \gamma_R(P_m \square P_k) \leq 2 \left( \left\lceil \frac{mk}{5} \right\rceil + \left\lceil \frac{m}{5} \right\rceil + \left\lceil \frac{k}{5} \right\rceil \right).$$

3.

$$\gamma_R(K_m \square K_k) \leq \begin{cases} 2m - 1, & \text{ако је } m = k \\ 2m, & \text{ако је } m < k \end{cases}$$

када је  $2 \leq m \leq k$ .

**Теорема 2.7.** ([45]) *За свака два пута  $P_m$  и  $P_n$ ,  $m, n \geq 2$  важи*

1.

$$\gamma_R(P_m \times P_n) \leq 4 \left\lceil \frac{mn}{10} \right\rceil + 8 \left\lceil \frac{m}{10} \right\rceil + 8 \left\lceil \frac{n}{10} \right\rceil.$$

2.

$$\gamma_R(P_m \times P_n) > 4 \left\lfloor \frac{mn - m - n + 1}{10} \right\rfloor + 4 \left\lfloor \frac{m - 1}{10} \right\rfloor + 4 \left\lfloor \frac{n - 1}{10} \right\rfloor.$$

3.

$$\lim_{m, n \rightarrow \infty} \frac{\gamma_R(P_m \times P_n)}{mn} = \frac{2}{5}.$$

4. Кардинални производ  $C_m \times C_n$ ,  $m, n \equiv 0 \pmod{10}$ , је римски граф, тј.

$$\gamma_R(C_m \times C_n) = 2\gamma(C_m \times C_n) = \frac{2mn}{5}.$$

5. За кардинални производ  $C_m \times C_n$ , када је  $m \equiv 0 \pmod{6}$  и  $n \equiv 0 \pmod{4}$  важи

$$\gamma_R(C_m \times C_n) \leq \frac{mn}{2}.$$

6. За кардинални производ  $C_m \times C_n$  када је  $m$  непарно и  $m \geq 11$  важе релације

$$\gamma_R(C_m \times C_n) \leq \begin{cases} \frac{2m(m-1)}{5} + m, m \equiv 1 \pmod{10} \\ \frac{2m(m-3)}{5} + 3m, m \equiv 3 \pmod{10} \\ \frac{2m(m-k)}{5} + 4m, m \equiv k \pmod{10}, k \in \{7, 9\} \end{cases}$$

и

$$\gamma_R(C_m \times C_n) = \frac{2m^2}{5}, m \equiv 5 \pmod{10}.$$

**Теорема 2.8.** ([46]) За свака два графа  $G$  и  $H$  редова  $m$  и  $n$  важе релације

$$\gamma_R(G \times H) \geq \frac{2mn}{\Delta(G)\delta(H) + 1},$$

$$\gamma_R(G \times H) \leq mn \frac{2 + \ln(1 + \delta(G)\delta(H)/2)}{\delta(G)\delta(H) + 1}.$$

### Тачне вредности римског доминантног броја

Тачне вредности римског доминантног броја за одређене класе графова исказане су наредним теоремама.

**Теорема 2.9.** 1. ([7], [8], [13]) За путеве  $P_n$  и цикле  $C_n$  важи

$$\gamma_R(P_n) = \gamma_R(C_n) = \left\lceil \frac{2n}{3} \right\rceil.$$

2. ([8]) За комплетан граф  $K_n$  важи  $\gamma_R(K_n) = 2$ .

3. ([40]) За комплетан бипартитни граф  $K_{p,q}$ , код кога је  $p \leq q$ , важи

$$\gamma_R(K_{p,q}) = \begin{cases} 2, & p = 1 \\ 3, & p = 2 \\ 4, & p \geq 3. \end{cases}$$

4. ([8]) За комплетан мултипартитни граф,  $G = K_{p_1, p_2, \dots, p_t}$  код кога је  $p_1 \leq p_2 \leq \dots \leq p_t$  и  $t \geq 3$ , важи

$$\gamma_R(G) = \begin{cases} 2, & p_1 = 1 \\ 3, & p_1 = 2 \\ 4, & p_1 \geq 3. \end{cases}$$

5. ([8]) За графове  $G$  реда  $n$ , који садрже чвор степена  $n - 1$ , важи  $\gamma_R(G) = 2$ .

6. ([8], [13]) За гريد графове  $G_{2,n}$  важи  $\gamma_R(G_{2,n}) = n + 1$ .

7. ([13]) За  $k \geq 0$  важи  $\gamma_R(G_{3,4k}) = 6k + 1$ ,  $\gamma_R(G_{3,4k+1}) = 6k + 2$ ,  $\gamma_R(G_{3,4k+2}) = 6k + 4$ ,  $\gamma_R(G_{3,4k+3}) = 6k + 6$ .

8. ([13]) Ако је  $G$  копија од  $G_{3,5}$  код које је обрисан чвор из пете колоне, тада је  $\gamma_R(G) = 8$ .

9. ([13]) За свако  $n \geq 1$  важи  $\gamma_R(G_{4,n}) \geq 2n$ .  $\gamma_R(G_{4,n}) = 2n + 1$  ако је  $n = 1, 2, 3, 4, 5, 6$ , односно  $\gamma_R(G_{4,n}) = 2n$  иначе.

10. ([8], [13]) За графове  $G$  реда  $n$ , који немају изоловане чворове, важи  $\gamma_R(G) = n$  ако и само ако је  $n$  паран и ако је  $G = \frac{n}{2}K_2$ .

**Теорема 2.10.** ([46])

1. За сва стабла  $T$  и све графове  $G$  који немају циклe непарних дужина важе релације

$$\gamma_R(P_2 \times T) = 2\gamma_R(T) = \gamma_R(P_2)\gamma_R(T),$$

$$\gamma_R(P_2 \times G) = 2\gamma_R(G) = \gamma_R(P_2)\gamma_R(G).$$

2. За пут  $P_2$  и било који цикл  $C_{2n+1}$ ,  $n \geq 1$  непарне дужине важи

$$\gamma_R(P_2 \times C_{2n+1}) = \left\lceil \frac{2(4n+2)}{3} \right\rceil = 2 \left\lceil \frac{(4n+2)}{3} \right\rceil = \gamma_R(P_2) \gamma_R(C_{2n+1}).$$

3. Нека је  $n \geq 2$ . Тада важе релације

$$\gamma_R(P_3 \times P_n) = \begin{cases} 6\binom{n}{4}, & n \equiv 0(\text{mod } 4), \\ 6\lfloor \frac{n}{4} \rfloor + 2, & n \equiv 1(\text{mod } 4), \\ 6\lfloor \frac{n}{4} \rfloor + 4, & n \equiv 2(\text{mod } 4), \\ 6\lfloor \frac{n}{4} \rfloor + 5, & n \equiv 3(\text{mod } 4). \end{cases}$$

$$\gamma_R(P_4 \times P_n) = \begin{cases} 2n + 2, & n = 5 \\ 2n, & \text{иначе.} \end{cases}$$

Кловићар и Пуљјић [46] су, поред резултата исказаних Теоремом 2.10, дефинисале границе за римски доминантни број за производе  $P_5 \times P_n$  и  $P_6 \times P_n$ . Још није показано да се границе достижу.

$$\gamma_R(P_5 \times P_n) \leq \begin{cases} 8\binom{n}{3}, & n \equiv 0(\text{mod } 3), \\ 8\lfloor \frac{n}{3} \rfloor + 4, & n \equiv 1(\text{mod } 3), \\ 8\lfloor \frac{n}{3} \rfloor, & n \equiv 2(\text{mod } 3). \end{cases}$$

$$\gamma_R(P_6 \times P_n) \leq \begin{cases} 10\binom{n}{3}, & n \equiv 0(\text{mod } 3), \\ 10\lfloor \frac{n}{3} \rfloor + 2, & n \equiv 1(\text{mod } 3), \\ 10\lfloor \frac{n}{3} \rfloor + 4, & n \equiv 2(\text{mod } 3). \end{cases}$$

$\mathcal{RD}$  проблем на гريد графовима проучавао је и Vincenzo Cunnò [14] који је поред теоријских резултата предложио и хеуристике за одређивање  $\gamma_R$  броја за класу гريد графова, док су Durgun и Торраккаја [47] дефинисали вредности римског доминантног броја за комете, двоструке комете и чешљасте графове (енг. comb graphs)<sup>4</sup>.

Ослањајући се на резултате везане за доминантни број за Петерсен-ове графове  $P(n, 2)$  и  $P(n, 3)$  које су Yuansheng и сар. објавили 2007. го-

<sup>4</sup>Више информација о кометама, двоструким кометама и чешљастим графовима могу се пронаћи у самом раду.

дине у [48] и [49], Wang и сар. долазе до следећег резултата.

**Теорема 2.11.** ([50])  $\gamma_R(P(n, 2)) = \lceil \frac{8n}{7} \rceil, n \geq 5$ .

Фамилију графова за које је  $\gamma_R(G) + \gamma(G)/2 \leq n$  када је  $n \geq 3$  дефинисали су Favron и сар. [42].

**Класа графова са особином  $\gamma_R(G) \leq \gamma(G) + k$**

**А) Класе графова са особином  $\gamma_R(G) = \gamma(G) + 1$**

Соскаупе и сар. [8], као и Дреуер [13], доказали су да за повезан граф  $G$  реда  $n$  важи релација  $\gamma_R(G) = \gamma(G) + 1$  ако и само ако постоји чвор  $v \in V$  степена  $n - \gamma(G)$ . За повезан граф  $G$  важи релација  $\gamma_R(G) = \gamma(G) + 1$  ако је  $1 \leq rad(G) \leq 2$  и  $1 \leq diam(G) \leq 4$ . Посебно,  $\gamma(G) \geq 3$  ако је  $rad(G) = 2$  и  $diam(G) = 4$ .

Дреуер [13] је извршио и карактеризацију класе стабала  $T$  са особином  $\gamma_R(T) = \gamma(T) + 1$ . Наиме, показао је да за стабло  $T$  са два или више чворова важи релација  $\gamma_R(T) = \gamma(T) + 1$  ако и само ако је  $T$  "рањени паук"<sup>5</sup>.

Поред класе рањених паукова, класи графова код које важи  $\gamma_R(G) = \gamma(G) + 1$  припадају: сви графови облика  $G = K_1 + H$  (укључујући комплетне графове  $K_n$  и "волане"  $K_1 + C_n$ ), сви графови облика  $G = \overline{K_2} + H$  (укључујући  $K_{2,n}, n \geq 2$  и "двоструке волане"  $\overline{K_2} + C_n$ ), као и сви графови облика  $G = (K_1 + H) \star K_1$ .

**Б) Класе графова са особином  $\gamma_R(G) = \gamma(G) + 2$**

Соскаупе и сар. [8] и Дреуер [13] показали су да за повезане графове  $G$  реда  $n \geq 2$  важи релација  $\gamma_R(G) = \gamma(G) + 2$  ако и само ако:

- 1)  $G$  нема чвор степена  $n - \gamma(G)$  и
- 2)  $G$  или има чвор степена  $n - \gamma(G) - 1$  или има два чвора  $v$  и  $w$  таква да је  $|N_{[v]} \cup N_{[w]}| = n - \gamma(G) + 2$ .

На крају је закључено да, уколико за повезане графове  $G$  важи релација  $\gamma_R(G) = \gamma(G) + 2$ , да је тада  $2 \leq rad(G) \leq 4$  и  $3 \leq diam(G) \leq 8$ .

<sup>5</sup>Више информација о класи паук графова може се наћи у [51].



## В) Класе графова са особином $\gamma_R(G) = \gamma(G) + k$ , $2 \leq k \leq \gamma(G)$

У [52] је показано да за повезани граф  $G$  реда  $n$  и доминантног броја  $\gamma(G) \geq 2$  важи следеће:

Ако је  $k$  позитиван број такав да је  $2 \leq k \leq \gamma(G)$ , тада је  $\gamma_R(G) = \gamma(G) + k$  ако и само ако:

- а) за сваки природни број  $s$  такав да је  $1 \leq s \leq k - 1$ ,  $G$  не садржи скуп  $U_t$  ( $1 \leq t \leq s$ ) од  $t$  чворова  $|\cup_{v \in U_t} N[v]| = n - \gamma(G) - s + 2t$ .
- б) постоји природни број  $l$  ( $1 \leq l \leq k$ ) такав да  $G$  садржи скуп  $W_l$  од  $l$  чворова таквих да  $|\cup_{v \in W_l} N[v]| = n - \gamma(G) - k + 2l$ .

## Класа графова који су римски графови

Теоремом 2.9 Соскаупе и сар. показали су да класи графова који су римски припадају сви графови облика  $G = K_1 + H$  где је  $\gamma(G) = 1$  и  $\gamma_R(G) = 2$ . Слично, сваки граф реда  $n$  који има чвор степена  $n - 1$ , сви графови облика  $P_{3k}$ ,  $C_{3k}$ ,  $P_{3k+2}$  и  $C_{3k+2}$ , као и да сви комплетни бипартитни графови  $K_{m,n}$ , код којих је  $\min\{m, n\} \neq 2$ , припадају класи римских графова.

Поред особина римског доминантног броја за одређена стабла, Дреуер [13] је показао да је граф  $G$  римски ако и само ако  $\gamma(G) \leq \gamma(G \setminus S) + |S|/2$  за свако 2-покривање  $S \subseteq V$ . У [53] је показано да су Петерсенови графови  $P(n, 2k + 1)$  ( $n \neq 4k + 2$ ,  $n \equiv 0 \pmod{4}$ ) и  $0 \leq k \leq \lfloor \frac{n}{2} \rfloor$ ),  $P(n, 1)$  ( $n \neq 2 \pmod{4}$ ),  $P(n, 3)$  ( $n \geq 7$ ,  $n \not\equiv 3 \pmod{4}$ ) и  $P(11, 3)$  и Декартов производ графова  $C_{5m} \square C_{5n}$  за  $m \geq 1$ ,  $n \geq 1$  такође римски графови.

Класу римских стабала увео је М.А. Henning 2002. године, [54] док су 2006. године Song и Wang [55] дефинисали класу стабала са особином  $\gamma_R(T) = \gamma(T) + 3$ .

Услови под којим је лексикографски производ два графа римски граф дати су у [56]. Поред овог резултата, Shumenjak и сар. одредили су вредности римског броја за лексикографски производ два графа.

## 2.2 Проблем ограничено римске доминације

Нека је  $G = (V, E)$  прост, неоријентисан граф реда  $n$ . Скуп  $S \subseteq V$  назива се *скупом ограничено доминације* ако сваки чвор који није из скупа  $S$  има суседа у  $S$  и суседа у  $V \setminus S$ . Број ограничено доминације, у ознаци  $\gamma_{rst}(G)$ , представља број елемената скупа ограничено доминације најмање кардиналности. Проблем ограничено доминације увели су Telle и Proskurowski [57] 1997. године. Више о овом проблему може се наћи, на пример, у [39] и [58, 59].

Инспирисане појмом ограничено доминације, P.R.L. Pushpan и S. Padma-rgiea [9] дефинисале су функцију ограничено римске доминације на графу тако што су навеле да је провинција Римског царства заштићена уколико има стационирану легију, односно да је провинција која нема стационирану легију заштићена уколико има суседа који има две стациониране легије и суседа који је такође без стациониране легије. За граф  $G$ , проблем одређивања скупа ограничено римске доминације најмање кардиналности назива се проблемом ограничено римске доминације (*RRD* проблем). Доказ да је проблем ограничено римске доминације NP комплетан за графове у општем смислу дат је у [60].

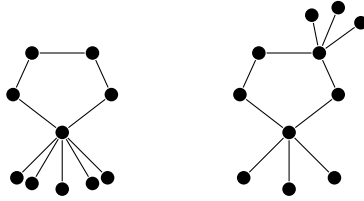
### Особине функције ограничено римске доминације

**Теорема 2.12.** ([9]) *За сваки граф  $G$  важи следећа релација*

$$\gamma_{rst}(G) \leq \gamma_{rR}(G) \leq 2\gamma_{rst}(G).$$

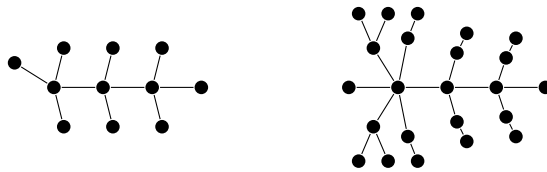
**Став 2.13.** ([9])

1. *Ако граф  $G$  реда  $n$  садржи чвор степена  $n - 1$  и ако је  $\delta(G) > 1$ , тада је  $\gamma_{rR}(G) = 2$  ( $\gamma_{rst}(G) = 1$ ).*
2. *Ако три чвора графа  $G$  формирају цикл дужине три, тада је  $\gamma_{rR} < n$ .*
3. *За сваки граф  $G$ , који није стабло, важи  $\gamma_{rR}(G) = n$  ако и само ако је  $G$  изоморофан са једним од графова:  $C_4$ ,  $C_5$ ,  $G_1$  или  $G_2$  ( $G_1$  и  $G_2$  су приказани на Слици 10).*
4. *За свако стабло  $T$  важи  $\gamma_{rR}(T) = n$  ако и само ако је  $T$  или "гусеница" чија је кичма дужине највише 3 или "јастог" чија је кичма дужине*



Слика 10: Графови  $G_1$  (лево) и  $G_2$  (десно).

највише 3 и чији дијаметар није већи од 5 (на Слици 11 приказани су графови који имају структуру "гусенице" и "јастога", а чија је кичма дужине 3).

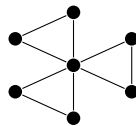


Слика 11: Гусеница (лево) и Јастог (десно)

5. За било који граф  $G$  важи релација  $\gamma_{rR}(G) = \gamma_{rst}(G)$  ако и само ако је свака компонента повезаности од  $G$  или звезда или  $K_1$ .

**Теорема 2.14.** ([60]) Ако је  $G$  граф који има грану која није инцидентна са листом, тада постоји  $\gamma_{rR}$ -функција  $f : V \rightarrow \{0, 1, 2\}$  за коју је  $V_0 = \{v \mid f(v) = 0\} \neq \emptyset$ .

За сваки непаран број  $n \geq 3$  нека  $G_n$  представља граф који се може формирати из  $(n-1)/2$  копија графа  $K_2$  додавањем новог чвора и његовим повезивањем са листовима графа  $K_2$ . Описаним поступком добија се да је  $G_3 = K_3$ . У случају да је  $n > 3$  граф  $G_n$  има један чвор степена  $n-1$ , док су му сви остали чворови степена два. На Слици 12 приказан је граф  $G_7$  формиран описаним поступком. Нека је  $\mathcal{G} = \{G_n : n \geq 3, n \text{ је непарно}\}$ .



Слика 12: Граф  $G_7$ .

**Теорема 2.15.** ([60]) За сваки повезани граф  $G$  реда  $n \geq 3$  са  $t$  грана важи релација  $\gamma_{rR}(G) \geq n + 1 - 2t/3$  при чему се једнакост достиже ако и само ако је  $G \in \mathcal{G}$ .

**Тврђење 2.16.** ([60])

1. За сваки повезан граф  $G$  реда  $n$  важи релација  $\gamma_{rR}(G) \leq n+1 - \lfloor (\text{diam}(G) - 2)/3 \rfloor$ , при чему се граница достиже за неке графове.
2. За сваки повезани граф  $G$  реда  $n$  и опсега  $g(G)$  важи релација  $\gamma_{rR}(G) \leq n + 1 - \lfloor (g(G) - 2)/3 \rfloor$ , код које се граница достиже за неке графове.
3. За сваки повезани граф  $G$  реда  $n$ , са  $t$  грана, важи релација  $\gamma_{rR}(G) \leq 2t - n + 2$ . Једнакост се достиже за граф  $G$  ако и само ако је  $G$  стабло са особиниом  $\gamma_{rR}(G) = n$ .

Аналогно резултатима који се односе на особине функције римске доминације, Rad и Krzywkowski [60] дефинисали су горњу границу за број ограничене римске доминације.

**Теорема 2.17.** ([60])

1. Нека је  $G$  граф реда  $n$ . Уколико је  $\delta > 0$  и  $n < \delta(\delta - 1)/(\ln \delta - \ln 2 + 1)$  тада важи релација

$$\gamma_{rR}(G) \leq n \left( \frac{2 \ln(1 + \delta) - \ln 4 + 2}{\delta + 1} \right).$$

2. Нека је  $G$  чордални граф без артикуларних чворова (чвор чијим брисањем граф постаје неповезан) и нека је  $G$  реда  $n$ . Уколико је  $\delta(G) \geq 2$ , важи релација

$$\gamma_R(G) \geq 2n \left( 1 - \frac{\delta}{(\delta + 1)^{1 + \frac{1}{\delta}}} \right).$$

3. Нека је  $G$  чордални граф без артикуларних чворова и нека је  $G$  реда  $n$ . Уколико је  $\delta(G) \geq 2$  и  $\ln(1 + \delta)/(1 + \delta) \leq 1/2$ , тада је

$$\gamma_{rR}(G) \leq 2n \left( \frac{\ln(1 + \delta) + 1}{\delta + 1} \right).$$

Обележимо са  $\beta(G)$  максимални број спаривања у  $G$ .

**Теорема 2.18.** ([60])

1. За сваки граф  $G$  реда  $n$  важи релација

$$\gamma_{rR}(G) \leq \frac{3 \ln(1 + \delta) + \delta + 4 - \ln 8}{\delta + 1} n - 2\beta(G).$$

2. За сваки граф  $G$  реда  $n$  који има савршено спаривање важи релација

$$\gamma_{rR}(G) \leq \frac{3 \ln(1 + \delta) + 3 - \ln 8}{\delta + 1} n.$$

Rad и Krzywkowski [60] су показали интересантан резултат да релација  $\gamma_{rR}(G) = \gamma_R(G)$  важи за скоро сваки граф.

### Тачне вредности ограниченог римског доминантног броја

Јасно је да је  $\gamma_{rR}(P_2) = 2$  и  $\gamma_{rR}(P_3) = 3$ .

**Теорема 2.19.** ([9])

1. За  $n \geq 4$  важи релација

$$\gamma_{rR}(P_n) = \frac{2n + 3 + r}{3}, \quad n \equiv r \pmod{3}, \quad r \in \{1, 2, 3\}.$$

2. За циклe  $C_n$  важи релација

$$\gamma_{rR}(C_n) = \begin{cases} \frac{2n+3+r}{3} & , n \equiv r \pmod{3}, r \in \{1, 2\} \\ \frac{2n}{3} & , n \equiv 0 \pmod{3} \end{cases}$$

3. За комплетне графове  $K_n$  важи релација  $\gamma_{rR}(K_n) = 2$ .

4. За комплетне бипартитне графове  $K_{m,n}$  важи релација  $\gamma_{rR}(K_{m,n}) = 4$ .

### 1. Класа графова са особином $\gamma_{rR}(G) = \gamma_{rst}(G) + 1$

Карактеризацију класе графова са особином  $\gamma_{rR}(G) = \gamma_{rst}(G) + 1$  дале су Pushram и Radmariea [9] и, при том, доказале следеће резултате:

1. За сваки граф  $G$ , реда  $n$ , важи релација  $\gamma_{rR}(G) = \gamma_{rst}(G) + 1$  ако и само ако постоји чвор  $v \in V$  такав да је број неизолованих чворова у  $N(v)$  једнак  $n - \gamma_{rst}$ .
2. За сваки граф  $G$  са особином  $\gamma_{rR}(G) = \gamma_{rst}(G) + 1$ , важи  $1 \leq \text{diam}(G) \leq 4$  и  $1 \leq \text{rad}(G) \leq 2$ .
3. За било који граф  $G$  са тачно једним циклом  $C_k$  релација  $\gamma_{rR}(G) = \gamma_{rst}(G) + 1$  важи ако и само ако је

- а)  $k = 3$ ,
- б) најмање један чвор у  $C_3$  је степена 2,
- в) сваки чвор који није у  $C_3$  је лист.

**2. Класа графова са особином  $\gamma_{rR}(G) = \gamma_{rst}(G) + 2$**

Pushram и Radmariea [9] су доказале да свако стабло  $T$ , које није звезда, има особину  $\gamma_{rR}(G) \geq \gamma_{rst}(G) + 2$  и дале карактеризацију стабала за које важи  $\gamma_{rR}(T) = \gamma_{rst}(T) + 2$  и графова са тачно једним циклор за које важи  $\gamma_{rR}(G) = \gamma_{rst}(G) + 2$ .

**Теорема 2.20.** ([9])

1. За сваки граф  $G$  реда  $n$  важи  $\gamma_{rR}(G) = \gamma_{rst}(G) + 2$  ако и само ако
  - а) у  $G$  не постоји чвор  $v$  са  $n - \gamma_{rst}$  суседа и
  - б) постоје два чвора  $u, v \in G$ , који заједно имају  $n - \gamma_{rst}$  суседа.
2. За сваки граф  $G$  са особинор  $\gamma_{rR}(G) = \gamma_{rst}(G) + 2$  важи  $2 \leq \text{diam}(G) \leq 7$  и  $1 \leq \text{rad}(G) \leq 4$ .
3. За било које стабло  $T$  које није звезда важи  $\gamma_{rR}(T) \geq \gamma_{rst}(T) + 2$ .
4. За било које стабло  $T$  важи  $\gamma_{rR}(T) = \gamma_{rst}(T) + 2$  ако и само ако је  $T$  "гусеница" са кичмор дужине највише 3 и унутрашњих чворова степена 2.
5. Нека је  $G$  граф са тачно једним циклор  $C_k$ ,  $3 \leq k \leq 6$ . Тада је  $\gamma_{rR}(G) = \gamma_{rst}(G) + 2$  ако и само ако је  $G \cong C_k$ ,  $4 \leq k \leq 6$  или  $G \cong G_i$ ,  $1 \leq i \leq 7$ .

Поред резултата исказаних Теоремор 2.20 у [9] је дефинисан број ограничене римске доминације за подељене графове. Добијени резултати изложени су у наредној теореми.

**Теорема 2.21.** ([9]) Нека је  $G$  граф, дефинисан бипартицијор  $(X, Y)$  где је  $X$  независан скуп а  $G[Y]$  комплетан граф.

1. Нека је  $\text{deg}(x) = 1$  за свако  $x \in X$ .
  - а) Ако је  $\text{deg}(y) = |Y| - 1$  за најмање један чвор  $y \in Y$ , тада је  $\gamma_{rR}(G) = \gamma_{rst}(G) + 1$ .

- б) Ако је  $\deg(y) \geq |Y|$  за сваки чвор  $y \in Y$ , тада је  $\gamma_{rR}(G) = \gamma_{rst}(G) + 2$ .
2. Нека је  $\deg(x) \geq 2$  за најмање један чвор  $x \in X$ . Тада је  $\gamma_{rR}(G) = \gamma_{rst}(G) + k$ ,  $k \geq 1$  ако и само ако постоји тачно  $k$  чворова  $y_i \in Y$ ,  $1 \leq i \leq k$ , такви да је  $\deg(y_1) = \Delta(G)$  и ако се за свако  $i$ ,  $2 \leq i \leq k$ , у скупу  $H_i = N_X(y_i) \setminus \{\bigcup_{j=1}^{i-1} N_X(y_j)\}$  налазе најмање два унутрашња чвора и важи  $|H_{i-1}| \geq |H_i|$ .

## 2.3 Проблем слабе римске доминације

Као алтернативу  $\mathcal{RD}$  проблему, М.А. Henning и S.T. Hedetniemi [10] су 2003. године дефинисали проблем слабе римске доминације ( $\mathcal{WRD}$  проблем). Наиме, Henning и Hedetniemi сматрали су да би се услов, којим свака провинција у којој легија није стационирана мора имати суседа са две стациониране легије, могао ослабити. Тачније, са обзиром да је једна легија довољна да заштити провинцију од напада, уз претпоставку да се само један напад може догодити у датом моменту, провинција која нема стационирану легију може се сматрати заштићеном уколико је суседна са бар једном провинцијом која има бар једну стационирану легију и ако се из те провинције ка њој може послати легија а да се не наруши сигурност свих осталих провинција. Овако дефинисаном стратегијом могао би да се смањи број легија потребних да се заштити царство. Нека је провинција сигурна од напада уколико има стационирану легију, односно нека је провинција без стациониране легије заштићена од напада уколико има сигурног суседа. Проблем слабе римске доминације представља проблем одређивања најмањег броја легија у царству тако да је свака провинција римског царства или сигурна или заштићена од напада уз услов да све провинције остају сигурне или заштићене и уколико се у било коју заштићену провинцију пошаље једна легија из њене суседне сигурне провинције. Прецизна математичка дефиниција овог проблема је дата у [10].

Доказ да је проблем слабе римске доминације NP комплетан за графове у општем смислу, а специјално за бипартитне или чордалне графове, дат је у [10]. Провера једнакости  $\gamma_r(G) = \gamma_R(G)$  за произвољни граф такође се сматра NP тешким проблемом, [61]. Слично  $\mathcal{RD}$  проблему, увођењем додатних услова, дефинисане су верзије  $\mathcal{WRD}$  проблема. На пример,

Pushram и Mai [62] су 2011. године дефинисале функцију *слабе римске доминације на гранама*, док су кроз два рада скоријег датума, Pushram и Kamalam [63] дефинисале функцију *ефикасне слабе римске доминације*, а Abel Cabrera Martinez и сар. [64] проблем *тоталне слабе римске доминације*.

## Особине функције слабе римске доминације

**Став 2.22.** ([10])

1. Свака  $f_{RD}$  функција за  $G$  уједно је и  $f_{WRD}$  за  $G$ .
2. Ако је  $H$  разапитујући подграф графа  $G$  тада је  $\gamma_r(G) \leq \gamma_r(H)$ .
3. Ако је  $G$  граф за који је дефинисна  $\gamma_r$ -функција која 0 додељује суседним чворовима  $u$  и  $v$ , тада је  $\gamma_r(G) = \gamma_r(G \setminus uv)$ .

**Теорема 2.23.** ([10]) За сваки граф  $G$  важи  $\gamma(G) \leq \gamma_r(G) \leq \gamma_R(G) \leq 2\gamma(G)$ .

Однос слабог римског доминантног броја и још неких врста доминантних бројева дат је у [65]. Карактеризација стабала за које је слаби римски доминантни број једнак римском доминантном броју приказана је у [66], а утицај брисања чворова на промену вредности слабог римског доминантног броја у [67].

## Тачне вредности слабог римског доминантног броја

Приметимо прво да се за  $C_3$  лако може закључити да је  $\gamma_r(C_3) = 1$ .

**Теорема 2.24.** 1. ([10]) За путеве  $P_n$  и циклусе  $C_n$  када је  $n \geq 4$  важи

$$\gamma_r(P_n) = \gamma_r(C_n) = \left\lceil \frac{3n}{7} \right\rceil.$$

2. ([40]) За комплетне графове  $K_n$  важи релација  $\gamma_r(K_n) = 1$ .
3. ([68]) За сваки граф  $G$  реда  $n$ ,  $n > 3$ , који није комплетан важи  $\gamma_r(G) = 2$  ако и само ако  $G$  има чвор степена  $n - 1$ .



4. ([40]) За класу комплетних бипартитних графова  $G = K_{p,q}$ , при услову  $p \leq q$  важи релација

$$\gamma_r(G) = \begin{cases} 2, & p = 1, 2, q > 1 \\ 3, & p = 3 \\ 4, & p \geq 4. \end{cases}$$

5. ([40]) За класу комплетних мултипартитних графова,  $G = K_{p_1, p_2, \dots, p_t}$  код којих је  $p_1 \leq p_2 \leq \dots \leq p_t$  и  $t \geq 3$  важи релација

$$\gamma_r(G) = \begin{cases} 2, & p_1 = 1, 2 \\ 3, & p_1 \geq 3. \end{cases}$$

6. ([68]) За класу грид графова  $G_{2,n}$  важи релација

$$\gamma_r(G_{2,n}) = \begin{cases} \left\lfloor \frac{4n}{5} \right\rfloor, & \text{ако је } n \equiv 0 \pmod{5} \\ \left\lfloor \frac{4n}{5} \right\rfloor + 1, & \text{иначе.} \end{cases}$$

7. ([68]) За свако комплетно бинарно стабло  $T$  дубине  $k$  важи  $\gamma_r(T) = 2^m(1 + 2^3 + 2^6 + \dots + 2^{k-1})$  када је  $k \equiv 0 \pmod{3}$ .

Henning и Hedetniemi су у [10] приметили да су разлике у кардиналности доминантних скупова за  $\mathcal{RD}$  и  $\mathcal{WRD}$  проблем за дати граф  $\gamma_r(P_n) - \gamma_r(P_n) = \lfloor 2n/7 \rfloor - \lfloor 3n/7 \rfloor$  једнаке или  $\lfloor 5n/21 \rfloor$  или  $\lfloor 5n/21 \rfloor$ . Pushram је заједно са Mai [68] дала тачне вредности слабог римског доминантног броја за гусенице, док су Valveny и сар. [69] дефинисали вредност слабог римског доминантног броја за још неке типове графова.

**Теорема 2.25.** ([40]) За Декартов производ два графа важе релације

1.  $\gamma_r(P_m \times P_k) \leq \left\lceil \frac{mk}{3} \right\rceil + 2.$
2.  $\gamma_r(C_m \times C_k) \leq \left\lceil \frac{mk}{3} \right\rceil.$

### 1. Класа графова за које је $\gamma_r(G) = \gamma(G)$

Pushram и Mai су у [68] дефинисале стабло  $T$  са особином  $\gamma_r(T) = \gamma(T)$ , док су опште графове за које важи  $\gamma_r(G) = \gamma(G)$  дефинисали Henning и

Hedetniemi [10] наредном теоремом.

**Теорема 2.26.** ([10]) *За сваки граф  $G$  важи  $\gamma_r(G) = \gamma(G)$  ако и само ако постоји доминантни скуп  $S$  такав да је*

- a)  $pn(v, S)$  индикује клику за сваки чвор  $v \in S$ ,*
- b) за сваки чвор  $u \in V(G) \setminus S$ , који није приватни сусед ни једног чвора скупа  $S$ , постоји чвор  $v \in S$  такав да  $pn(v, S) \cup \{u\}$  индикује клику.*

## 2. Класе графова са особином $\gamma_r(G) = 2\gamma(G)$

Карактеризација стабала  $T$  са особином  $\gamma_r(T) = 2\gamma(T)$  дата је у [10]. Наиме, Henning и Hedetniemi су дали карактеризацију шуме  $\mathcal{F}$  код које за свако стабло  $T \in \mathcal{F}$  важи  $\gamma_r(T) = 2\gamma(T)$ .

### 2.4 Преглед алгоритама за решавање проблема римске доминације и неких његових модификација

Будући да се одређени проблеми распоређивања ресурса могу представити као проблеми Римске доминације, постоји потреба за развојем алгоритама којима ће се наведени проблеми решити првенствено за инстанце већих димензија. До сада је развијено свега неколико алгоритама за решавање проблема Римске доминације. Наиме, Римски доминантни број може се одредити у линеарном времену код следећих типова графова

- 1) интервалних, јаких чордалних и кографова, видети [66, 70];<sup>6</sup>
- 2) јединичних диск графова (енг. unit disk graphs), видети [15];<sup>7</sup>
- 3) грид графова, видети [13].

док се у полиномијалном времену може тачно одредити за

- 1) стабла и шуме, видети [13];

---

<sup>6</sup>Граф  $G = (V, E)$  се назива интервалним графом ако постоји скуп  $\{I_v | v \in V\}$  интервала реалне праве са особином да је  $I_u \cap I_v \neq \emptyset$  ако  $u, v \in E$ .

<sup>7</sup>Јединични диск графови представљају колекцију кругова једнаких радијуса са особином да су два круга повезана граном ако један круг садржи центар другог.

- 2) графове који немају АТ подграф<sup>8</sup>, графове који имају подграф облика  $d$ -хоботнице, видети [70];
- 3) графове који немају  $P_4$  подграф, односно графове који немају као подграф пут дужине 4 и немају транспортну мрежу, видети [71].

У [14] је, поред алгоритма полиномијалне сложености за процењивање римског доминантног броја, предложена и хеуристичка метода заснована на генетским алгоритмима, док је у [72] изложена хеуристичка метода заснована на методи променљивих околина. Алгоритми за одређивање римског доминантног броја за графове у општем смислу, графове без  $P_4$  подграфа и графове без  $P_5$  и  $(s, t) - net$  подграфа предложени су у [71]. У истом раду показано је да се време решавања проблема римске доминације може оценити са  $O(mn^2)$  када су у питању графови без  $P_5$  и  $(3, 2) - net$  подграфа. Описани алгоритам са  $O(mn^2)$  бржи је од алгоритма представљеног у [70] чије се време извршавања оцењује са  $O(n^6)$  за графове из исте класе.

Алгоритамске аспекте за решавање проблема римске доминације дали су Liedloff и сар. [70], Padamutham и Palagiri у [73] и Liu и Chang у [16], док је рачунска комплексност одређивања римског доминантног броја проучавана у [74].

Још неки алгоритми за решавање проблема римске доминације могу се наћи у [10, 40, 75, 76, 77, 78, 79].

Алгоритам линеарне комплексности за одређивање слабог римског доминантног броја на блоковским графовима изложен је у [80], међутим једноставним тестирањем се испоставило да предложени алгоритам не даје увек тачне резултате. Исправљени алгоритам су Ивановић и Урошевић [81] представили на Valcoг конференцији у Београду 2018. године. Додатно, Ивановић и Урошевић су у [72] изложили метахеуристичку методу засновану на методи променљивих околина која поред проблема римске доминације решава и проблем слабе римске доминације.

Chapelle и сар. су у [82] представили два алгоритма којима се време решавања проблема слабе римске доминације оцењује са  $O^*(2^n)$  коришћењем експоненцијалног времена и  $O^*(2.2279^n)$  коришћењем полиноми-

---

<sup>8</sup>Три чвора графа формирају АТ подграф уколико за свака два чвора тог подграфа постоји пут који не садржи околину трећег чвора.

јалног времена (нотацијом  $O^*(f(n))$  означава се да су полиномијални фактори  $p(n)$  потиснути, тј.  $O^*(f(n)) = O(p(n)f(n))$ ). У истом раду, Charelle и сар. показали су да се овај проблем може решити у линеарном времену на интервалним графовима.

У литератури не постоје алгоритми за одређивање ограниченог римског доминантног броја познати аутору.

### 3 Математичке формулације проблема римске доминације и неких његових модификација

#### 3.1 Проблем римске доминације на графовима

У литератури су, до момента писања ове тезе, познате две математичке формулације проблема римске доминације на графовима. Прву формулацију су С.С. ReVelle и К.Е. Rosing објавили 2000. године у Mathematical Association of America, [12], док су другу формулацију 2013. године објавили А.Р. Burger, А.Р.Д. Villiers и Ј.Н. van Vuuren у Journal of Combinatorial Mathematics and Combinatorial Computing, [83]. Прва формулација ће у даљем тексту бити означена са **RD1**, док ће се за другу користити ознака **RD2**. Обе формулације биће изложене слично као и у радовима у којима су публиковане.

##### 3.1.1 Математичке формулације проблема римске доминације из релевантне литературе

###### RD1 формулација

Посматрајмо граф  $G = (V, E)$  са скупом чворова  $V = \{1, 2, \dots, n\}$ . Означимо са  $N_i$  скуп свих чворова  $j \in V$  суседних са  $i \in V$ . За функцију  $f$ , дефинисану са  $f : V \rightarrow \{0, 1, 2\}$ , нека су бинарне променљиве  $x_i$  и  $y_i$ ,  $i = 1, 2, \dots, n$  ( $n = |V|$ ) дефинисане тако да за сваки чвор  $i \in V$  важи:

$$x_i = \begin{cases} 1, & f(i) \geq 1 \\ 0, & \text{иначе} \end{cases},$$

$$y_i = \begin{cases} 1, & f(i) = 2 \\ 0, & \text{иначе.} \end{cases}$$

**RD1** формулација може се описати на следећи начин:

$$\min \sum_{i \in V} x_i + \sum_{i \in V} y_i \tag{3.1}$$

$$x_i + \sum_{j \in N_i} y_j \geq 1, \quad i \in V \quad (3.2)$$

$$y_i \leq x_i, \quad i \in V \quad (3.3)$$

$$x_i, y_i \in \{0, 1\}, \quad i \in V \quad (3.4)$$

Вредност функције циља, дефинисане условом (3.1), представља тежину  $f_{RD}$  функције  $f$  за дати граф  $G$ , односно број  $\gamma_R(G)$ . Узимајући да чвор  $i$  представља једну провинцију Римског царства и да је број легија стационараних у тој провинцији изражен тежином чвора  $i$  ( $f(i)$ ), условима (3.2) обезбеђено је да сваки чвор или има позитивну тежину или има суседа тежине 2. Са обзиром да је провинција сигурна од напада уколико има макар једну стационарану легију (важи  $x_i = 1$ ), условима (3.3) обезбеђено је да се сваки чвор тежине два сматра сигурним од напада ( $y_i = 1$  повлачи да је  $x_i = 1$ ). Променљиве  $x_i$  и  $y_i$  су дефинисане тако да идентификују да ли у датим провинцијама има стационараних легија, док је условима (3.4) обезбеђена њихова бинарност.

## RD2 формулација

Посматрајмо поново граф  $G = (V, E)$  са скупом чворова  $V = \{1, \dots, n\}$  дефинисаним тако да сваки чвор представља једну провинцију Римског царства. Нека  $N_i$  представља скуп свих чворова  $j \in V$  суседних са  $i \in V$ . За функцију  $f$ , дефинисану са  $f : V \rightarrow \{0, 1, 2\}$ , нека су променљиве  $x_i$  и  $y_i$ ,  $i = 1, \dots, n$  дефинисане са

$$x_i = \begin{cases} 1, & f(i) = 1 \\ 0, & \text{иначе} \end{cases},$$

$$y_i = \begin{cases} 1, & f(i) = 2 \\ 0, & \text{иначе.} \end{cases}$$

RD2 формулација може се описати на следећи начин:

$$\min \sum_{i \in V} x_i + 2 \sum_{i \in V} y_i \quad (3.5)$$

$$x_i + y_i + \sum_{j \in N_i} y_j \geq 1, \quad i \in V \quad (3.6)$$

$$x_i + y_i \leq 1, \quad i \in V \quad (3.7)$$

$$x_i, y_i \in \{0, 1\}, \quad i \in V \quad (3.8)$$

Тежина  $f_{\mathcal{RD}}$  функције  $f$ , односно вредност броја  $\gamma_R(G)$ , описани су функцијом циља (3.5). Будући да вредност функције  $f$  у чвору  $i \in V$  представља тежину тог чвора, тј. број легија стационираних у провинцији коју тај чвор представља, условима (3.6) дефинисано је да свака провинција која нема своју легију мора бити суседна са бар једном провинцијом код које су две легије стациониране, односно сваки чвор мора имати позитивну тежину и/или мора бити суседан са бар једним чвором тежине 2. У истом чвору истовремено не могу бити смештене и једна и две легије, услови (3.7). Променљиве  $x_i$  и  $y_i$  дефинисане су као бинарне условима (3.8).

**RD1** и **RD2** формулације  $\mathcal{RD}$  проблема припадају проблемима целобројног линеарног програмирања (енг. Integer Linear Programming, у даљем тексту ILP) и састоје се из  $2|V|$  бинарних променљивих и  $2|V|$  ограничења.

### 3.1.2 Побољшање математичких формулација проблема римске доминације из литературе

#### Модификација **RD1** формулације

Приметимо да се променљиве  $x_i$  **RD1** формулације могу релаксирати и уместо из скупа бинарних, узети из скупа позитивних реалних бројева. Нека је зато

$$x_i \in [0, +\infty), \quad y_i \in \{0, 1\}, \quad i \in V. \quad (3.9)$$

Заменимо услове (3.4) **RD1** формулације условима (3.9) и обележимо модификовану формулацију са **RD1'**. Докажимо да се уместо ILP формулације **RD1**,  $\mathcal{RD}$  проблем може представити формулацијом **RD1'** која припада скупу проблема мешовитог целобројног линеарног програмирања (енг. Mixed Integer Linear Programming, у даљем тексту MILP).

**Теорема 3.1.** *Оптимальна вредност функције циља **RD1** формулације (3.1) - (3.4) проблема римске доминације једнака је оптимальној вредности функције циља **RD1'** формулације (3.1) - (3.3) и (3.9).*

*Доказ.* Нека се допустиво решење **RD1'** формулације може представити вектором  $(\bar{x}'', \bar{y}'')$ , где су  $\bar{x}'' = (x_1'', \dots, x_n'')$  и  $\bar{y}'' = (y_1'', \dots, y_n'')$ ,  $n = |V|$ . Такође, нека је вектор  $(\bar{x}', \bar{y}')$  ( $\bar{x}' = (x_1', \dots, x_n')$ ,  $\bar{y}' = (y_1', \dots, y_n')$ ) променљивих  $x'_i$  и  $y'_i$  дефинисан тако да је  $y'_i = y''_i$  за свако  $i \in V$  и

$$x'_i = \begin{cases} 0, & x''_i \in [0, 1) \\ 1, & x''_i \in [1, +\infty). \end{cases}$$

Променљиве  $x'_i$  и  $y'_i$  дефинисане су тако да имају бинарне вредности и стога задовољавају услове (3.4). Као допустиво решење **RD1'** проблема, променљиве  $x''_i$  и  $y''_i$  испуњавају услов (3.3), а са обзиром да је  $y'_i = y''_i$  за свако  $i \in V$ , следи да је  $y'_i = y''_i \leq x''_i$ . Даље се за  $y''_i = 1$  на основу неједнакости  $1 \leq x''_i$  и дефинисаности променљивих  $x'_i$  добија да је  $x'_i = 1$ , тј.  $x'_i \geq 1 = y'_i$ . Слично, када је  $y''_i = 0$  следи да је  $y'_i = 0$ , а имајући у виду да је  $x''_i \geq 0$ , односно да је  $x'_i \in \{0, 1\}$ , лако се може закључити да је  $x'_i \geq 0 = y'_i$ . Комбинујући ове две релације, може се закључити да је  $x'_i \geq y'_i$  за свако  $i \in V$ . Дакле, променљиве  $x'_i$  и  $y'_i$  задовољавају и услове (3.3). Имајући у виду да променљиве  $x''_i$  и  $y''_i$  задовољавају услов (3.2), тј. да је  $x''_i + \sum_{j \in N_i} y''_j \geq 1$ , разликоваћемо два случаја:

- 1)  $(\exists j \in N_i)$  за које је  $y''_j = 1$ ,
- 2)  $(\forall j \in N_i)$   $y''_j = 0$ .

На основу случаја 1) следи да постоји  $j$  за које је  $y'_j = 1$ , тј.  $\sum_{j \in N_i} y'_j \geq 1$  а самим тим и  $x'_i + \sum_{j \in N_i} y'_j \geq 1$ . Из 2) следи да је  $1 \leq x''_i + \sum_{j \in N_i} y''_j = x''_i$ . Сада, са обзиром да је  $x''_i \geq 1$ , следи да је  $x'_i = 1$  а самим тим и да је  $x'_i + \sum_{j \in N_i} y'_j \geq 1$ . Дакле, за свако  $i \in V$ , променљиве  $x'_i$  и  $y'_i$  такође задовољавају услове (3.2).

На основу релација  $y'_i = y''_i$  и  $x'_i \leq x''_i$  и дефиниције функције циља **RD1** формулације  $(\sum_{i \in V} x'_i + \sum_{i \in V} y'_i)$  лако се може закључити да је  $Obj_{RD1} \leq Obj_{RD1'}$ . Како је вредност функције циља сваког релаксираног проблема мања или једнака вредности функције циља одговарајућег оригиналног проблема, следи да је  $Obj_{RD1'} \leq Obj_{RD1}$ . Комбиновањем последње две неједнакости добија се да је  $Obj_{RD1} = Obj_{RD1'}$  чиме је теорема доказана.  $\square$

Из дате теореме може се закључити да **RD1'** формулација такође одговара **RD** проблему. Модификована формулација се састоји из  $|V|$  бинарних,  $|V|$  реалних променљивих и  $2|V|$  услова.



## Модификације RD2 формулације

Означимо формулацију (3.5), (3.6) и (3.8) са **RD2'**, а формулацију дефинисану релацијама (3.5), (3.6) и (3.9) са **RD2''**.

**Теорема 3.2.** *Оптимальна вредност функције циља **RD2'** формулације (3.5), (3.6) и (3.8), проблема римске доминације, једнака је оптимальној вредности функције циља **RD2** формулације (3.5) - (3.8).*

*Доказ.* Нека се допустиво решење **RD2'** формулације може представити вектором  $(\bar{x}'', \bar{y}'')$ ,  $\bar{x}'' = (x''_1, \dots, x''_n)$ ,  $\bar{y}'' = (y''_1, \dots, y''_n)$ ,  $n = |V|$  и нека су  $V_1$  и  $V_2$  два дисјунктна скупа за која важи  $V_1 \cup V_2 = V$  и

- 1)  $x''_i = 0$  или  $y''_i = 0$ , за сваки чвор  $i \in V_1$ ,
- 2)  $x''_i = 1$  и  $y''_i = 1$ , за сваки чвор  $i \in V_2$ .

Дефинишимо вектор  $(\bar{x}', \bar{y}')$  променљивих  $\bar{x}' = (x'_1, \dots, x'_n)$  и  $\bar{y}' = (y'_1, \dots, y'_n)$  на следећи начин. Нека је  $y'_i = y''_i$  за свако  $i \in V$ .

За свако  $i \in V_1$ , дефинишимо  $x'_i$  тако да је  $x'_i = x''_i$ . Са обзиром да вектор  $(\bar{x}'', \bar{y}'')$  представља допустиво решење **RD2'** формулације, и да је за свако  $i \in V_1$   $(x'_i, y'_i) = (x''_i, y''_i)$ , следи да за свако  $i \in V_1$  услови (3.6) и (3.8) важе и за  $(x'_i, y'_i)$ . Услед бинарности променљивих  $x''_i$  и  $y''_i$  за дати скуп  $V_1$  су испуњени и услови (3.7):

$$x'_i + y'_i = x''_i + y''_i \leq \max\{x''_i, y''_i\} \in \{0, 1\} \leq 1.$$

Дефинишимо сада  $x'_i$  за свако  $i \in V_2$ . Са обзиром на дефинисаност променљивих  $y'_i$  и скупа  $V_2$  ( $y'_i = y''_i$  и  $y''_i = 1$ ), нека је  $x'_i = 0$ . Докажимо да су услови (3.6)-(3.8) испуњени за свако  $i \in V_2$ :

$$x'_i + y'_i + \sum_{j \in N_i} y'_j = 0 + 1 + \sum_{j \in N_i} y''_j = 1 + |V_2| \geq 1, \quad (\forall i) \in V_2$$

$$x'_i + y'_i = 0 + 1 \leq 1, \quad (\forall i) \in V_2$$

$$x'_i = 0 \in \{0, 1\}, \quad y'_i = 1 \in \{0, 1\}$$

Дакле, допустиво решење **RD2'** формулације је уједно и допустиво решење **RD2** формулације. Вредности функција циља могу се израчунати

по следећој формули:

$$Obj_{RD2'} = \sum_{i \in V} x_i'' + 2 \sum_{i \in V} y_i'' = \sum_{i \in V_1} x_i'' + 2 \sum_{i \in V_1} y_i'' + \sum_{i \in V_2} x_i'' + 2 \sum_{i \in V_2} y_i'' = \sum_{i \in V_1} x_i'' + 2 \sum_{i \in V_1} y_i'' + 3|V_2|.$$

$$Obj_{RD2} = \sum_{i \in V} x_i' + 2 \sum_{i \in V} y_i' = \sum_{i \in V_1} x_i' + 2 \sum_{i \in V_1} y_i' + \sum_{i \in V_2} x_i' + 2 \sum_{i \in V_2} y_i' = \sum_{i \in V_1} x_i'' + 2 \sum_{i \in V_1} y_i'' + 2|V_2|.$$

Одавде се може закључити да је  $Obj_{RD2} \leq Obj_{RD2'}$ .

Претпоставимо да скуп  $U$  представља скуп допустивих решења **RD2** формулације. Изостављањем услова (3.7) из **RD2** формулације, скуп допустивих решења **RD2'** формулације може се обележити са  $U^\star$ . Јасно је да је  $U \subseteq U^\star$  и да је, самим тим, допустива тачка **RD2** формулације уједно и допустива тачка **RD2'** формулације. На основу дефиниције глобалног и локалног минимума следи да је

$$Obj_{RD2'} = \min_{U^\star} \left( \sum_{i \in V} x_i + 2 \sum_{i \in V} y_i \right) \leq \min_U \left( \sum_{i \in V} x_i + 2 \sum_{i \in V} y_i \right) = Obj_{RD2}$$

Најзад, комбиновањем изведених релација  $Obj_{RD2'} \geq Obj_{RD2}$  и  $Obj_{RD2'} \leq Obj_{RD2}$ , добија се да је  $Obj_{RD2'} = Obj_{RD2}$ .  $\square$

**Теорема 3.3.** *Оптимальна вредност функције циља **RD2''** формулације (3.5), (3.6) и (3.9) једнака је оптимальној вредности функције циља **RD2'** формулације (3.5), (3.6) и (3.8).*

*Доказ.* Аналогно доказу Теореме 3.1, нека је допустиво решење **RD2''** формулације дефинисано вектором  $(\bar{x}'', \bar{y}'')$ , где је  $\bar{x}'' = (x_1'', \dots, x_n'')$ ,  $\bar{y}'' = (y_1'', \dots, y_n'')$  и  $n = |V|$ . Поново, нека је вектор  $(\bar{x}', \bar{y}')$  променљивих  $x_i'$  ( $\bar{x}' = (x_1', \dots, x_n')$ ) и  $y_i'$  ( $\bar{y}' = (y_1', \dots, y_n')$ ) дефинисан са  $y_i' = y_i''$  за свако  $i \in V$  и

$$x_i' = \begin{cases} 0, & x_i'' \in [0, 1), \\ 1, & x_i'' \in [1, +\infty). \end{cases}$$

Променљиве  $x_i'$  и  $y_i'$  су бинарне, стога задовољавају услове (3.8). Прет-

постављајући да је  $x_i'' + y_i'' + \sum_{j \in N_i} y_j'' \geq 1$ , добијају се два случаја:

$$1) (y_i'' = 1) \vee (\exists j \in N_i)(y_j'' = 1),$$

$$2) (y_i'' = 0) \wedge (\forall j \in N_i)(y_j'' = 0).$$

Из 1) следи да је  $y_i'' = 1$  или постоји  $j \in N_i$  за које је  $y_j'' = 1$ , тј. важи  $y_i'' + \sum_{j \in N_i} y_j'' \geq 1$ . Будући да је  $y_i' = y_i''$  и  $x_i' \geq 0$  за свако  $i \in V$ , следи да је  $x_i' + y_i' + \sum_{j \in N_i} y_j' \geq 1$ . Из 2) следи да је  $1 \leq x_i'' + y_i'' + \sum_{j \in N_i} y_j'' = x_i''$ . Сада, са обзиром да је  $x_i'' \geq 1$ , следи да је  $x_i' = 1$ , што даље имплицира да је  $x_i' + y_i' + \sum_{j \in N_i} y_j' \geq 1$  ( $y_i' \in \{0, 1\}$ ). Дакле, променљиве  $x_i'$  и  $y_i'$  задовољавају услове (3.6) за свако  $i \in V$ .

Вредност функције циља **RD2'** формулације одређује се по формули  $\sum_{i \in V} x_i' + 2 \sum_{i \in V} y_i'$ . Из релација  $y_i' = y_i''$  и  $x_i' \leq x_i''$  лако се може закључити да је  $Obj_{RD2'} \leq Obj_{RD2''}$ . Поново, познато је да је вредност функције циља сваког релаксираног проблема минимизације мања или једнака вредности функције циља одговарајућег основног проблема, дакле  $Obj_{RD2''} \leq Obj_{RD2'}$ . Комбиновањем изведених неједнакости добија се да је  $Obj_{RD2'} = Obj_{RD2''}$  чиме је теорема доказана.  $\square$

Приметимо да се формулацијама **RD2'** и **RD2''** број услова смањује на  $|V|$ . Такође, **RD2''** формулација уместо  $2|V|$  бинарних променљивих, користи  $|V|$  бинарних и  $|V|$  реалних променљивих. На основу Теорема 3.2 и 3.3 може се закључити да су **RD2'** и **RD2''** формулације добро дефинисане математичке формулације  $\mathcal{RD}$  проблема.

Модификације **RD1'**, **RD2'** и **RD2''** изложене су слично као у раду [84] у коме су публиковане.

## 3.2 Проблем ограничено римске доминације

Дефиниција проблема ограничено римске доминације дата је у поглављу 1.4. Посматрајмо граф  $G = (V, E)$  са скупом чворова  $V = \{1, \dots, n\}$ , и функцију  $f : V \rightarrow \{0, 1, 2\}$ . Математички модел за проблем ограничено римске доминације може се описати са:

$$\min_f F_1(f) \quad (3.10)$$

при условима:

$$F_1(f) = \sum_{u \in V} f(u) \quad (3.11)$$

$$(\forall u \in V) \quad f(u) = 0 \quad \Rightarrow \quad (\exists v, w \in N_u)(f(v) = 2 \wedge f(w) = 0). \quad (3.12)$$

Тежина  $f_{\mathcal{RRD}}$  функције  $f$ , односно вредност броја  $\gamma_{rR}(G)$  описани су функцијом циља (3.10), док је условом (3.12) обезбеђено да функција  $f$  буде из класе  $f_{\mathcal{RRD}}$  функција.

У релевантној литератури сви радови који се односе на  $\mathcal{RRD}$  проблем најчешће су засновани на анализи броја  $\gamma_{rR}(G)$  за различите класе графова  $G$ . Будући да до момента писања ове тезе у литератури није била позната ни једна математичка формулација  $\mathcal{RRD}$  проблема, у наредној секцији биће изложена прва MILP формулација слично као што је изложена у раду [85].

### 3.2.1 Нова математичка формулација проблема ограничено римске доминације

Нека је за граф  $G = (V, E)$  са скупом чворова  $V = \{1, \dots, n\}$  функција  $f$  дефинисана са  $f : V \rightarrow \{0, 1, 2\}$  и нека су променљиве  $x_i$ ,  $x_i \in [0, \infty)$ ,  $i \in V$ , дефинисане тако да представљају тежину чвора  $i \in V$ . Иако је  $f(i)$  целобројна а  $x_i$  ненегативна реална променљива, за обе вредности биће коришћен израз *тежина*. Такође, биће показано да су вредности за  $x_i$  и  $f(i)$  (иако важи  $f \in \{0, 1, 2\}$  и  $x_i \in [0, \infty)$ ), једнаке у оптималном решењу, али нису обавезно једнаке за свако допустиво решење.

Нека се бинарним променљивим  $y_i$  и  $z_i$  идентификује тежина чвора

$i \in V$  на следећи начин:

$$y_i = \begin{cases} 1, & f(i) = 2 \\ 0, & \text{иначе} \end{cases} \quad \text{и} \quad z_i = \begin{cases} 1, & f(i) = 0 \\ 0, & \text{иначе} \end{cases}.$$

Нова математичка формулација  $\mathcal{RRD}$  проблема сада се може описати на следећи начин:

$$\min \sum_{i \in V} x_i \quad (3.13)$$

при ограничењима

$$x_i + \sum_{j \in N_i} y_j \geq 1, \quad i \in V \quad (3.14)$$

$$x_i + \sum_{j \in N_i} z_j \geq 1, \quad i \in V \quad (3.15)$$

$$x_i \geq 2y_i, \quad i \in V \quad (3.16)$$

$$x_i + 2z_i \leq 2, \quad i \in V \quad (3.17)$$

$$x_i \in [0, +\infty), \quad y_i, z_i \in \{0, 1\}, \quad i \in V. \quad (3.18)$$

Вредност функције циља, дефинисане условом (3.13), представља број  $\gamma_{rR}$ . Условима (3.14) дефинисано је да сваки чвор реалне тежине мање од један мора бити суседан са бар једним чвором тежине 2. Слично, условима (3.15) дефинише се да сваки чвор реалне тежине мање од један мора бити суседан са бар једним чвором тежине нула. Неједнакостима (3.16) и (3.17) дефинисано је да за сваки чвор  $i \in V$  тежине мање од 2, одговарајућа променљива  $y_i$  има вредност нула, односно да за сваки чвор  $i \in V$  позитивне тежине, одговарајућа променљива  $z_i$  има вредност нула. Коначно, променљиве  $x_i$  су ненегативне реалне, док су променљиве  $y_i$  и  $z_i$  ( $i \in V$ ) бинарне због услова (3.18).

Представљена MILP формулација састоји се из  $2|V|$  променљивих које су бинарне и  $|V|$  реалних променљивих док је број ограничења једнак  $4|V|$ .

За  $n$ -торке  $x = (x_1, \dots, x_n)$ ,  $y = (y_1, \dots, y_n)$  и  $z = (z_1, \dots, z_n)$ , који задовољавају услове (3.14) - (3.18), биће коришћена нотација  $F_2(x, y, z) = \sum_{i \in V} x_i$ . Услов (3.13), којим се минимизује број легија потребних да се заштити царство, сада се може написати као  $\min_{(x,y,z)} F_2(x, y, z)$ .

За формулацију (3.13) - (3.18) у даљем тексту биће коришћена ознака **RRD1**.

**Теорема 3.4.** *Оптимальна вредност функције циља математичке формулације (3.10) - (3.12) проблема ограничене римске доминације једнака је оптимальној вредности функције циља **RRD1** формулације.*

*Доказ.* ( $\Rightarrow$ ) Докажимо прво да функција циља формулације (3.10) - (3.12) **RRD** проблема није мања од функције циља **RRD1** формулације, т.ј.  $F_1(f) \geq F_2(x, y, z)$  на допустивом скупу (за проблем (3.10) - (3.12)) и да је одговарајућа тачка  $(x, y, z)$  допустива за **RRD1**.

За фиксиран чвор  $i \in V$  и функцију  $f : V \rightarrow \{0, 1, 2\}$ , нека су променљиве  $x_i$ ,  $y_i$  и  $z_i$  дефинисане са

$$x_i = f(i), \quad y_i = \begin{cases} 1, & f(i) = 2 \\ 0, & \text{иначе} \end{cases} \quad \text{и} \quad z_i = \begin{cases} 1, & f(i) = 0 \\ 0, & \text{иначе} \end{cases}.$$

Како су  $x_i \in [0, +\infty)$ ,  $y_i, z_i \in \{0, 1\}$ , услови (3.16) - (3.18) су задовољени по дефиницији. На пример, услов (3.17) је задовољен са обзиром да је  $z_i = 1$  када је  $x_i = f(i) = 0$  ( $x_i + 2z_i = 2$ ),  $z_i = 0$  када је  $x_i = f(i) = 1$  ( $x_i + 2z_i = 1 < 2$ ), и  $z_i = 0$  када је  $x_i = f(i) = 2$  ( $x_i + 2z_i = 2$ ).

Посматрајмо фиксирано  $i \in V$ . Разликују се два случаја:

1. Нека је  $f(i) \geq 1$ . Са обзиром да је  $x_i = f(i)$ , следи да је  $x_i \geq 1$ . Из последње релације и из услова бинарности променљивих  $y_i$  и  $z_i$  следи да је  $x_i + \sum_{j \in N_i} y_j \geq 1$  и  $x_i + \sum_{j \in N_i} z_j \geq 1$ .
2. Нека је  $f(i) = 0$ . Како је релација (3.12) задовољена,  $(\exists j, k \in N_i)(f(j) = 2 \wedge f(k) = 0)$ , следи да је  $y_j = 1$  и да је  $z_k = 1$ . Дакле,  $x_i + \sum_{j \in N_i} y_j = \sum_{j \in N_i} y_j \geq 1$  и  $x_i + \sum_{k \in N_i} z_k = \sum_{k \in N_i} z_k \geq 1$ .

Тиме су услови (3.14) и (3.15) задовољени. Будући да променљиве  $x_i$ ,  $y_i$  и  $z_i$  задовољавају услове (3.14) - (3.18) за произвољно  $i \in V$ , следи да је  $F_2(x, y, z) = \sum_{i \in V} x_i = \sum_{i \in V} f(i) = F_1(f)$  на допустивом скупу за проблем (3.10) - (3.12) и одговарајућим допустивим тачкама **RRD1** проблема.

( $\Leftarrow$ ) Покажимо да је вредност функције циља **RRD** проблема (3.10) - (3.12) највише једнака функцији циља **RRD1** формулације, тј.  $F_1(f) \leq$

$F_2(x, y, z)$  на допустивом скупу **RRD1** проблема и за одговарајуће тачке (3.10) - (3.12) проблема. За скуп променљивих  $x_i, y_i$  и  $z_i$  које задовољавају услове (3.14) - (3.18), нека је функција  $f$  дефинисана са

$$f(i) = \begin{cases} 0, & x_i \in [0, 1) \\ 1, & x_i \in [1, 2) \\ 2, & x_i \in [2, +\infty) \end{cases} . \quad (3.19)$$

На основу дефиниције функције  $f$  она заиста слика  $V$  у скуп  $\{0, 1, 2\}$ . Будући да је за фиксирани чвор  $i \in V$ ,  $f(i) \in \{0, 1, 2\}$ , разматрају се следећа два случаја:

*Случај 1.* Нека је  $x_i \in [1, +\infty)$ . На основу дефиниције функције  $f$ , следи да је или  $f(i) = 1$  или  $f(i) = 2$ . Сада, будући да је  $\perp \Rightarrow p$  таутологија за сваки логички услов  $p$ , услов (3.12) је такође задовољен.

*Случај 2.* Нека је  $x_i \in [0, 1)$ . На основу дефиниције функције  $f$ ,  $f(i) = 0$ . Из услова (3.14),  $x_i + \sum_{j \in N_i} y_j \geq 1$ , следи да је  $\sum_{j \in N_i} y_j \geq 1 - x_i > 0$ . Обзиром да су променљиве  $y_j$  бинарне, израз  $\sum_{j \in N_i} y_j$  мора имати целобројну вредност одакле даље следи да је  $\sum_{j \in N_i} y_j \geq 1$ . Дакле, постоји чвор  $j \in N_i$  такав да је  $y_j = 1$ . На основу услова (3.16) и релације  $x_j \geq 2y_j = 2$  следи да је  $f(j) = 2$ . Слично, из услова (3.15) следи да је  $\sum_{j \in N_i} z_j \geq 1 - x_i > 0$ . На основу бинарности променљивих  $z_j$  следи да  $\sum_{j \in N_i} z_j$  има целобројну вредност. Сада, са обзиром да је  $\sum_{j \in N_i} z_j \geq 1$ , може се закључити да постоји чвор  $k \in N_i$  за који је  $z_k = 1$ . Коначно, из услова (3.17), тј.  $x_k \leq 2 - 2z_k = 0$ , следи да је  $x_k = 0$ , односно  $f(k) = 0$ , на основу чега се може закључити да је испуњен и услов (3.12).

На основу дефиниције функције  $f$ , јасно је да је  $f(i) \leq x_i$ , за свако  $i \in V$ . Стога се може закључити да је  $F_1(f) = \sum_{i \in V} f(i) \leq \sum_{i \in V} x_i = F_2(x, y, z)$  на допустивом скупу **RRD1** проблема и за одговарајуће тачке (3.10) - (3.12) проблема.

Пошто за свако допустиво решење проблема (3.10) - (3.12) постоји допустиво решење проблема (3.13) - (3.18) које задовољава релацију  $F_2(x, y, z) \leq F_1(f)$  и за свако допустиво решење проблема (3.13) - (3.18) постоји допустиво решење проблема (3.10) - (3.12) које задовољава релацију  $F_1(f) \leq F_2(x, y, z)$ , директно следи да је  $\min_f F_1(f) = \min_{(x,y,z)} F_2(x, y, z)$ .  $\square$

### 3.3 Проблем слабе римске доминације

Иако се у литератури доста писало о броју слабе римске доминације, забележена је само једна математичка формулација  $WRD$  проблема. Заправо, 2013. године су А.Р. Burger, А.Р.Д. Villiers и Ј.Н. van Vuuren у Journal of Combinatorial Mathematics and Combinatorial Computing [83] предложили математички модел  $WRD$  проблема у форми целобројног линеарног програмирања. У наставку ће се за предложени математички модел користити ознака  $WRD_{BVV}$ . Међутим, у оквиру овог поглавља показаћемо да  $WRD_{BVV}$  формулација не одговара дефиницији проблема слабе римске доминације коју су дали Henning и Hedetniemi у [10]. Под претпоставком да је начињена штампарска грешка, предложене су коригована математичка формулација  $WRD$  проблема уз доказ да тако коригована формулација заиста одговара дефиницији проблема слабе римске доминације из [10] и нова математичка формулација, за коју је такође показано да одговара проблему слабе римске доминације дефинисаном у [10].

#### 3.3.1 Математичка формулација проблема слабе римске доминације из релевантне литературе

##### $WRD_{BVV}$ формулација

Нека је за дат граф  $G = (V, E)$  са својом матрицом суседства  $A = [a_{ij}]$ , уз конвенцију да је  $a_{ii} = 1$ ,  $i = 1, \dots, n$ ,  $n = |V|$ , функција  $f$  дефинисана са  $f : V \rightarrow \{0, 1, 2\}$ , тј. нека је са  $f(i)$  дефинисана тежина чвора  $i \in V$ . Даље, нека скуп  $X$  представља скуп чворова тежине 1, односно нека скуп  $Y$  представља скуп чворова тежине 2. Нека су бинарне променљиве  $x_i$  и  $y_i$  дефинисане са:

$$x_i = \begin{cases} 1, & f(i) = 1 \\ 0, & \text{иначе} \end{cases} \quad (3.20)$$

$$y_i = \begin{cases} 1, & f(i) = 2 \\ 0, & \text{иначе} \end{cases} \quad (3.21)$$



за свако  $i \in V$  и нека су бинарне променљиве  $z_{ij}$  дефинисане са

$$z_{ij} = \begin{cases} 1, & (x_i = 0 \text{ и } y_i = 0) \text{ и (или } x_j = 1 \text{ или } y_j = 1) \\ 0, & \text{иначе} \end{cases} \quad (3.22)$$

за свако  $i, j \in V, i \neq j$ .

У литератури се за  $z_{ij} = 1$  користи израз ”чворови  $i$  и  $j$  формирају скуп размене тежина” (енг. vertices  $i$  and  $j$  form a swap set) а за чворове позитивне тежине ( $x_i > 0$  или  $y_i > 0$ ) каже да су ”сигурни од напада”.

**WRD<sub>BVV</sub>** формулација проблема слабе римске доминације дефинисана је на следећи начин:

$$\min \sum_{i=1}^n (x_i + 2y_i) \quad (3.23)$$

при ограничењима:

$$\sum_{j=1}^n a_{ij}(x_j + y_j) \geq 1, \quad i = 1, \dots, n \quad (3.24)$$

$$x_i + y_i \leq 1, \quad i = 1, \dots, n \quad (3.25)$$

$$x_i + y_i + \sum_{\substack{j=1 \\ j \neq i}}^n z_{ij} \geq 1, \quad i = 1, \dots, n \quad (3.26)$$

$$a_{ij}(x_j + y_j - x_i - y_i + 1) \geq 2z_{ij}, \quad i, j = 1, \dots, n, j \neq i \quad (3.27)$$

$$y_j a_{ki} + a_{kj} + \sum_{\substack{l=1 \\ l \neq i \\ l \neq j}}^n a_{kl}(x_l + y_l) \geq z_{ij}, \quad i, j, k = 1, \dots, n, i \neq j \quad (3.28)$$

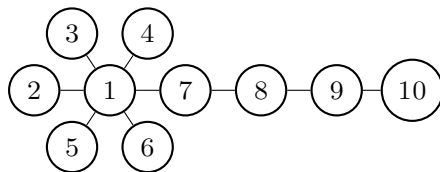
$$x_i, y_i, z_{ij} \in \{0, 1\}, \quad i, j = 1, \dots, n \quad (3.29)$$

Функција циља, чија вредност представља број слабе римске доминације  $\gamma_r$ , дефинисана је условом (3.23). Условима (3.24) дефинисано је да за сваки чвор тежине нула мора постојати суседни чвор позитивне тежине. Условима (3.25) обезбеђује се дисјунктност скупова  $X$  и  $Y$ . Даље, условима (3.26) дефинисано је да за сваки чвор  $i$  такав да је  $x_i = y_i = 0$ , постоји чвор  $j, j \in N_i$  који ће са њим моћи да подели тежину (тј. могућа је прерасподела тежина између чворова  $i$  и  $j$ ), док се ограничењима (3.27) обезбеђује да решење остане допуштиво у односу на прерасподелу тежина између чворова  $i$  и  $j$ . Осим тога, допуштивост

решења након прерасподеле тежина између чворова  $i$  и  $j$  дефинисана је условима (3.28). Ограничења (3.29) чувају бинарност променљивих  $x_i$ ,  $y_i$  и  $z_{ij}$ .  $\text{WRD}_{\text{BVV}}$  формулација састоји се из  $n^2 + 2n$  променљивих и  $n^3 + n^2 + 3n$  ограничења.

### WRD1 формулација

Посматрајмо граф  $G_P$  приказан на слици 13. Директним решавањем  $\text{WRD}_{\text{BVV}}$  формулације (уз помоћ решавача CPLEX) се за посматрани граф добија да је  $\gamma_r(G_P) = 2$ , тј. да би се граф  $G_P$  сматрао заштићеним довољно је да по једна легија буде стационирана у чворовима означеним са 1 и 9. Међутим, једноставном провером може се видети да овакав



Слика 13: Граф  $G_P$

распоред није у складу са дефиницијом проблема слабе римске доминације коју су Henning и Hedetniemi дали у [10] с обзиром на то да, у складу са оваквим распоредом, ни једна стационирана легија не може да се пресели у чвор без стациониране легије а да не наруши заштићеност својих суседа. У складу са дефиницијом проблема слабе римске доминације коју су Henning и Hedetniemi дали у [10] (видети Дефиницију 1.7.) решење проблема слабе римске доминације за посматран граф је  $\gamma_r(G_P) = 4$  (ако се по две легије стационирају у чворове 1 и 9, чворове 2 - 7 штити једна од стационираних легија из чвора 1, док чворове 8 и 10 штити једна од стационираних легија из чвора 9). До проблема је, можда, дошло зато што су Burger и сар. у [83], занемаривши услов да легија може да се пресели само у њој суседну, дефинисали  $\text{WRD}$  проблем на следећи начин:

**Дефиниција 3.1.** *Скуп слабе римске доминације графа  $G$  представља пар  $(X_r, Y_r)$  међусобно дисјунктних подскупова чворова графа  $G$  (са  $X_r$  је означен скуп чворова графа  $G$  који имају једну легију а са  $Y_r$  скуп чворова графа  $G$  који садрже две легије), за који је  $X_r \cup Y_r$  доминантни скуп за  $G$ , са особином да за сваки чвор  $u \notin X_r \cup Y_r$  постоји чвор  $v \in X_r$  за који је скуп размене  $((X_r \cup \{u\}) \setminus \{v\}) \cup Y_r$  такође доминантни скуп у  $G$ , или чвор  $v \in Y_r$  за који је скуп размене  $(Y_r \setminus \{v\}) \cup (X_r \cup \{u, v\})$  доминантни скуп у  $G$ .*

Дакле, у складу са дефиницијом коју су дали Burger и сар. решење проблема слабе римске доминације за посматрани граф је  $\gamma_r(G_P) = 3$  (на пример: две легије су стациониране у 1; једна легија се стационира у 9; чворове без стациониране легије штити једна од стационираних легија из чвора 1). Из свега изложеног може се закључити да је највероватније дошло до штампарске грешке и да услов (3.28) треба да гласи

$$y_j a_{kj} + a_{ki} + \sum_{\substack{l=1 \\ l \neq i \\ l \neq j}}^n a_{kl}(x_l + y_l) \geq z_{ij}, \quad i, j, k = 1, \dots, n, j \neq i. \quad (3.30)$$

Иако дефиниција коју су Burger и сар. изложили занемарује да је пресељење могуће само ка суседним чворовима, директним решавањем формулације (3.23) - (3.27), (3.29) и (3.30) добија се да је оптимално решење за посматран граф  $\gamma_r(G_P) = 4$ . Претпостављајући да је у [83] начињена штампарска грешка и да је начињен превид при запису дефиниције *WRD* проблема, показаћемо да се за прву формулацију слабе римске доминације може узети формулација (3.23) - (3.27), (3.29) и (3.30) (у даљем тексту формулација **WRD1**).

**Теорема 3.5.** *Оптимална вредност функције циља **WRD1** формулације за граф  $G$  једнака је вредности слабог римског доминантног броја,  $\gamma_r(G)$ .*

*Доказ.* ( $\Rightarrow$ ) Нека за граф  $G = (V, E)$  са скупом чворова  $V = \{1, \dots, n\}$ , скуп  $(x_1, \dots, x_n, y_1, \dots, y_n, z_{11}, \dots, z_{nn})$  представља допустиво решење **WRD1** формулације. Дефинишимо функцију  $f : V \rightarrow \{0, 1, 2\}$  са  $f(i) = x_i + 2y_i$  ( $f$  је добро дефинисана будући да је  $x_i + 2y_i \in \{0, 1, 2\}$ ). Показаћемо да функција  $f$  представља функцију слабе римске доминације за граф  $G$  тако што ћемо за сваки чвор  $i = 1, \dots, n$  разликовати два случаја:

- 1)  $x_i + y_i = 1$ . Обзиром да случај  $x_i + y_i = 1$  означава да је  $x_i = 1$  или је  $y_i = 1$ , следи да је  $f(i) \geq 1$  ( $f(i) \in \{1, 2\}$ ).
- 2)  $x_i + y_i = 0$ . Обзиром да случај  $x_i + y_i = 0$  означава да је  $x_i = 0$  и  $y_i = 0$ , следи да је  $f(i) = 0$ . Из услова (3.24) следи да постоји чвор  $j \in N_i$  ( $a_{ij} = 1$ ) такав да је  $x_j + y_j = 1$ , тј. важи  $f(j) \geq 1$ . На основу дефинисаности променљивих  $z_{ij}$  следи да је  $z_{ij} = 1$ . Из услова (3.30) за свако  $k \in V$  важи  $y_j a_{kj} + a_{ki} + \sum_{l=1, l \neq i, j}^n a_{kl}(x_l + y_l) \geq 1$ . Претпоставићемо да је  $k \neq i$  с обзиром да би тада могли да кажемо да је чвор  $k$  заштићен јер има суседа  $j$  позитивне тежине, односно

да је  $k \neq j$  јер би у том случају и сам био позитивне тежине. Дакле, нека је  $k \in V \setminus \{i, j\}$ , због услова (3.30) један од следећа три случаја мора бити испуњен:

- а)  $k$  је сусед са  $i$  ( $a_{ki} = 1$ );
- б)  $k$  је сусед са  $j$  ( $a_{kj} = 1$ ) и важи  $y_j = 1$ ;
- в)  $\sum_{l=1, l \neq i, j}^n a_{kl}(x_l + y_l) \geq 1$ .

Нека је функција  $f' : V \rightarrow \{0, 1, 2\}$  дефинисана са  $f'(i) = f(i) + 1$ ,  $f'(j) = f(j) - 1$  и  $f'(k) = f(k)$ ,  $k \in V \setminus \{i, j\}$ . Функција  $f'$  коректно је дефинисана будући да је  $f'(i) = 1$ ,  $f'(j) \in \{0, 1\}$  и  $f'(k) \in \{0, 1, 2\}$  за свако  $k \in V \setminus \{i, j\}$ . У случају под а) чвор  $k$  је заштићен у односу на функцију  $f'$  јер има суседа  $i$  који је позитивне тежине ( $f'(i) = 1$ ). У случају под б) чвор  $k$  је заштићен у односу функцију  $f'$  зато што је суседан чвору  $j$  који је позитивне тежине ( $f'(j) = 1$ ), док је у случају под в) чвор  $k$  заштићен у односу на функцију  $f'$  као чвор чија је тежина позитивна (ако је  $a_{kk}(x_k + y_k) = 1$ ), односно као чвор који има суседа  $l$ ,  $l \neq i, j$  чија је тежина позитивна (ако је  $\sum_{l=1, l \neq i, j, k}^n a_{kl}(x_l + y_l) \geq 1$ ).

Дакле, за свако  $i$  или важи  $f(i) \geq 1$ , или је  $f(i) = 0$  и постоји сусед  $j$  са особином  $f(j) \geq 1$ , тако да, у односу на функцију  $f' : V \rightarrow \{0, 1, 2\}$  дефинисану са  $f'(i) = f(i) + 1$ ,  $f'(j) = f(j) - 1$  и  $f'(k) = f(k)$ ,  $k \in V \setminus \{i, j\}$  сви чворови су или позитивне тежине или имају суседа позитивне тежине.

Из свега наведеног, може се закључити да је свако допустиво решење **WRD1** формулације уједно допустиво решење **WRD** проблема, тј. да је  $f$  функција слабе римске доминације за граф  $G$ . На основу вредности променљивих  $x_i$  и  $y_i$  следи да је

$$\sum_{i=1}^n x_i + 2y_i = \sum_{i \in V} f(i). \quad (3.31)$$

( $\Leftarrow$ ) Нека за граф  $G$  функција  $f$ ,  $f : V \rightarrow \{0, 1, 2\}$  представља функцију слабе римске доминације. Покажимо, сада, да је решење проблема слабе римске доминације допустиво решење **WRD1** формулације. Нека су променљиве  $x_i$ ,  $y_i$  и  $z_{ij}$ ,  $i, j = 1, \dots, n$  дефинисане са (3.20)-(3.22). За сваки чвор  $i \in V$  разликујемо два случаја:

- 1)  $f(i) = 0$ . На основу дефиниције функције слабе римске доминације постоји чвор  $j \in N_i$  за који важи  $f(j) \geq 1$  и, у односу на функцију  $f' : V \rightarrow \{0, 1, 2\}$ , дефинисану са  $f'(i) = 1$ ,  $f'(j) = f(j) - 1$  и  $f'(k) = f(k)$  за свако  $k \in V \setminus \{i, j\}$ , важи  $\sum_{l=1}^n a_{kl} f'(l) \geq 1$  (за чвор  $k$  важи  $f'(k) \geq 1$  и/или постоји сусед  $l \in N_k$  такав да је  $f'(l) \geq 1$ ). Дакле,  $f(i) = 0$  имплицира да је  $x_i + y_i = 0$ , а  $f(j) \geq 1$  за  $j \in N_i$  имплицира да је  $x_j + y_j = 1$ ,  $a_{ij} = 1$ , тј. испуњени су услови (3.24) и (3.25). На основу вредности променљивих  $x_i$ ,  $y_i$ ,  $x_j$  и  $y_j$ , следи да је  $z_{ij} = 1$ , тј. испуњени су услови (3.26) и (3.27). Проверићемо испуњеност услова (3.30). За  $k \in N_i$  услов (3.30) је испуњен јер је  $a_{ki} = 1$ . За  $k \in N_j$  и  $f'(j) = 1$  услов (3.30) је испуњен јер је  $a_{kj} y_j = 1$ . Међутим, уколико је  $k \notin N_i$  и  $f'(j) = 0$  чвор  $k$  мора бити заштићен у односу на функцију  $f'$ , тј. важи  $\sum_{l=1, l \neq i, j}^n a_{kl} f'(l) \geq 1$ . Из последње релације следи да је  $f'(k) \geq 1$  и/или постоји бар један чвор  $l \in N_k \setminus \{i, j\}$  такав да је  $f'(l) \geq 1$ . Уколико је  $f'(k) \geq 1$  следи да је  $f(k) = f'(k) \geq 1$  а самим тим и да је  $a_{kk}(x_k + y_k) = 1$  чиме је услов (3.30) задовољен. Уколико за  $l \in N_k \setminus \{i, j\}$  важи  $f'(l) \geq 1$ , услов (3.30) је испуњен зато што је  $f(l) = f'(l) \geq 1$  а самим тим и  $a_{kl}(x_l + y_l) = 1$ .
- 2)  $f(i) \geq 1$ . Услови (3.24) - (3.26) су испуњени будући да за променљиве  $x_i$  и  $y_i$  важи  $x_i + y_i = 1$ . На основу вредности  $x_i$  и  $y_i$  и из начина на који су дефинисане променљиве  $z_{ij}$  добија се да је  $z_{ij} = 0$ , тј. испуњени су услови (3.27). Услови (3.30) су тривијално испуњени јер сабирци са леве стране неједнакости имају ненегативне вредности.

Најзад, на основу вредности функције  $f$  и дефинисаности променљивих  $x_i$ ,  $y_i$  и  $z_{ij}$ , може се закључити да решење проблема слабе римске доминације представља допустиво решење **WRD1** формулације. Такође, важи релација

$$\sum_{i \in V} f(i) = \sum_{i \in V, f(i)=1} x_i + \sum_{i \in V, f(i)=2} 2y_i = \sum_{i \in V} (x_i + 2y_i). \quad (3.32)$$

Будући да једнакост важи за сваку функцију слабе римске доминације  $f$  и сваки допустив скуп **WRD1** формулације, следи да је

$$\min_f \sum_{i \in V} f(i) = \min \sum_{i \in V} (x_i + 2y_i). \quad (3.33)$$

□

### 3.3.2 Побољшана математичка формулација WRD проблема

Приметимо да су за проблем слабе римске доминације променљиве  $z_{ij}$  WRD1 формулације дефинисане тако да је  $z_{ij} = 1$  уколико чвор  $i$  има тежину нула и уколико је њему суседан чвор  $j$  позитивне тежине, тј. за гране  $e = (i, j)$  и  $e = (j, i)$  код којих је  $z_{ij} = 1$  важи  $z_{ij} \neq z_{ji}$ . Са тим у вези, посматраћемо оријентисан граф у коме је свака неоријентисана грана замењена са две оријентисане, тј. нека је  $G = (V, E \cup E')$  где је  $E' = \{(j, i) | (i, j) \in E\}$ . Приметимо још да су вредности променљивих  $z_{ij}$  за свака два несуседна чвора  $i$  и  $j$  познате, тј. да је због услова (3.27)  $z_{ij} = 0$  када је  $a_{ij} = 0$  (за свако  $i, j \in V$ ,  $(i, j) \notin E \cup E'$ ). Стога, нека је

$$z_e = \begin{cases} 1, & i \text{ и } j \text{ могу да формирају скуп размене, тј.} \\ & e = (i, j), (x_i = 0 \text{ и } y_i = 0) \text{ и } (x_j = 1 \text{ или } y_j = 1) \\ 0, & \text{иначе} \end{cases} \quad (3.34)$$

дефинисано за свако  $e \in E \cup E'$ . Нека су, даље, бинарне променљиве  $x_i$  и  $y_i$  дефинисане условима (3.20) и (3.21).

Нова математичка формулација WRD проблема на графу  $G$  може се дефинисати на следећи начин:

$$\min \sum_{i \in V} (x_i + 2y_i) \quad (3.35)$$

при ограничењима:

$$x_i + y_i + \sum_{e=(i,j) \in E \cup E'} z_e \geq 1, \quad i \in V \quad (3.36)$$

$$x_j + y_j - x_i - y_i + 1 \geq 2z_e, \quad e = (i, j) \in E \cup E' \quad (3.37)$$

$$y_j + \sum_{\substack{l \in N_k \\ l \neq i, \\ l \neq j}} (x_l + y_l) \geq z_e, \quad e = (i, j) \in E \cup E', k \in N_j \setminus N_i \quad (3.38)$$

$$x_i, y_i, z_e \in \{0, 1\} \quad i \in V, e \in E' \cup E' \quad (3.39)$$

Вредност броја  $\gamma_r(G)$  за дати граф  $G$  дефинисана је функцијом циља (3.35). Условима (3.36) обезбеђује се да сваки чвор буде позитивне тежине или да има бар једног суседа чија је тежина позитивна а који са њим може да прераспоређи своју тежину. Валидност прераспоређивања

тежина између чворова  $i$  и  $j$  дефинисана је условом (3.37). Условима (3.38) дефинише се допустивост решења након што је тежина између чворова  $i$  и  $j$  прераспоређена. Променљиве су бинарне на основу услова (3.39).

У даљем тексту за формулацију (3.35)-(3.39) биће коришћена ознака **WRD2**. **WRD2** формулација састоји се из  $2n + m$  бинарних променљивих и  $3n + 2m$  ограничења.

**Теорема 3.6.** *За сваки прости граф  $G$ , оптимална вредност функције циља **WRD1** формулације једнака је оптималној вредности функције циља **WRD2** формулације.*

*Доказ.* ( $\Rightarrow$ ) Нека је допустиво решење **WRD1** формулације **WRD** проблема дефинисано скупом  $(X', Y', Z')$ ,  $X' = (x'_1, \dots, x'_n)$ ,  $Y' = (y'_1, \dots, y'_n)$ ,  $Z' = (z'_{11}, \dots, z'_{nn})$ ,  $n = |V|$ , и нека је скуп  $(X'', Y'', Z'')$  променљивих  $x''_i$ ,  $y''_i$  и  $z''_e$  дефинисан тако да важи  $x''_i = x'_i$ ,  $y''_i = y'_i$ ,  $i = 1, \dots, n$  и

$$z''_e = \begin{cases} 1, & e = (i, j), z'_{ij} = 1 \\ 0, & e = (i, j), z'_{ij} = 0 \end{cases}, \quad e \in E \cup E'. \quad (3.40)$$

Приметимо да су променљиве  $z'_{ij}$  дефинисане за сваки пар чворова  $(i, j)$ , док су променљиве  $z''_e$  дефинисане само за гране које постоје у графу  $G$  ( $z''_e$  је дефинисано само ако је  $a_{ij} = 1$ , односно за  $e = (i, j) \in E \cup E'$ ). Приметимо још да је, због услова (3.27),  $z'_{ij} = 0$  када је  $a_{ij} = 0$ . Променљиве  $z''_e$  добро су дефинисане с обзиром да код услова (3.26) променљиве за које важи  $z'_{ij} = 0$  не утичу на вредност посматране суме јер је нула неутрал за сабирање. При том, за овако дефинисане променљиве  $z'_{ij}$ , услови (3.27) и (3.30) постају тривијални. Из претходне дискусије следи да је израз  $\sum_{j \in V} z'_{ij}$  једнак изразу  $\sum_{e=(i,j) \in E \cup E'} z''_e$  за свако  $i \in V$  чиме се имплицира да услови (3.36) директно следе из услова (3.26). Приметимо још да су услови (3.27) нетривијални само када је  $a_{ij} = 1$ . Због тога, и из дефиниције променљивих  $z''_e$ , директно следи да услови (3.27) имплицирају (3.37). Да би показали испуњеност услова (3.38) неопходно је да се размотре следећа три случаја:

*Случај 1:*  $k \notin N_j$ .

Како  $k$  и  $j$  нису суседни чворови, следи да је  $a_{kj} = 0$ , односно лева страна (енг. left hand side, у даљем тексту LHS) услова (3.30)

$$y'_j a_{kj} + a_{ki} + \sum_{l=1}^n a_{kl}(x'_l + y'_l) - a_{ki}(x'_i + y'_i) - a_{kj}(x'_j + y'_j),$$

постаје

$$-a_{kj}x'_j + (1 - x'_i - y'_i)a_{ki} + \sum_{l=1}^n a_{kl}(x'_l + y'_l).$$

Даље, релација  $a_{kj}x'_j = 0$  важи на основу почетних претпоставки, док из услова (3.25) следи да је  $1 - x'_i - y'_i \geq 0$ . Најзад, услови (3.24) имплицирају да је

$$-a_{kj}x'_j + (1 - x'_i - y'_i)a_{ki} + \sum_{l=1}^n a_{kl}(x'_l + y'_l) \geq \sum_{l=1}^n a_{kl}(x'_l + y'_l) \geq 1 \geq z'_{ij}.$$

На основу претходне дискусије очигледно је да су у *Случају 1* услови (3.30) редундантни.

*Случај 2:*  $k \in N_i \cap N_j$ .

Под претпоставком да је чвор  $k$  суседан чворовима  $i$  и  $j$  следи да је  $a_{ki} = a_{kj} = 1$ . Будући да су  $a_{kj}y'_j \geq 0$  и  $\sum_{l=1, l \neq i, l \neq j}^n a_{kl}(x'_l + y'_l) \geq 0$  добија се да је LHS услова (3.30) најмање једнака  $a_{ki} = 1$ , одакле следи да је LHS већа или једнака  $z'_{ij}$ , одакле поново следи да су услови (3.30) редундантни.

*Случај 3:*  $k \in N_j \setminus N_i$ .

Под претпоставком да је чвор  $k$  суседан чвору  $j$  и при том му чвор  $i$  није суседан, следи да су  $a_{kj} = 1$  и  $a_{ki} = 0$ . Стога, услови (3.30) постају  $y'_j + \sum_{l=1, l \neq i, l \neq j}^n a_{kl}(x'_l + y'_l) \geq z'_{ij}$ . На основу дефиниције променљивих  $z''_e$  следи да су услови (3.38) испуњени.

Обзиром да су у *Случају 1* и *Случају 2* услови (3.30) редундантни а да су у *Случају 3* услови (3.30) еквивалентни условима (3.38) и да не могу утицати на допустивост решења проблема (3.23) - (3.27), (3.29) и (3.30), следи да се могу изоставити.

Променљиве  $x''_i$ ,  $y''_i$  и  $z''_e$  бинарне су по дефиницији, тј. важе услови (3.39).

Коначно, како су  $x''_i = x'_i$  и  $y''_i = y'_i$  следи да су функције циља једнаких вредности, тј.  $Obj_{\mathbf{WRD1}} = \sum_{i \in V} (x'_i + 2y'_i) = \sum_{i \in V} (x''_i + 2y''_i) = Obj_{\mathbf{WRD2}}$ . При том, из допустивости решења **WRD1** проблема следи допустивост решења **WRD2** формулације.

( $\Leftrightarrow$ ) Нека скуп  $(X'', Y'', Z'')$  представља допустиво решење **WRD2** формулације  $(X'' = (x''_1, \dots, x''_n), Y'' = (y''_1, \dots, y''_n), Z'' = (z''_1, \dots, z''_m))$ ,  $n = |V|$ ,  $m = |E|$  и нека је скуп  $(X', Y', Z')$  променљивих  $x'_i$ ,  $y'_i$  и  $z'_{ij}$ ,  $i, j = 1, \dots, n$



дефинисан на следећи начин:  $y'_i = y''_i$ ,  $i = 1, \dots, n$ ,

$$x'_i = \begin{cases} x''_i, & x''_i + y''_i \leq 1 \\ x''_i - 1, & x''_i + y''_i = 2 \end{cases}$$

и

$$z'_{ij} = \begin{cases} z''_e, & x''_j + y''_j \leq 1, e = (i, j) \in E \cup E' \\ 0, & \text{иначе} \end{cases}.$$

У циљу доказивања услова (3.24) - (3.27) и (3.30) потребно је размотрити два случаја:

(1.)  $x''_i + y''_i \leq 1$  и

(2.)  $x''_i + y''_i = 2$ .

Услови (3.25) тривијално важе с обзиром да је у (1.)  $x'_i + y'_i = x''_i + y''_i \leq 1$  и у (2.)  $x'_i + y'_i = x''_i - 1 + y''_i = 1 \leq 1$ .

Полазећи од услова (3.36), а на основу дефиниције променљивих  $z'_{ij}$ , услови (3.26) такође важе:

$$x'_i + y'_i + \sum_{\substack{j=1 \\ j \neq i}}^n z'_{ij} =$$

$$\left\{ \begin{array}{l} = x''_i + y''_i + \sum_{\substack{e=(i,j) \in E \cup E' \\ x''_j + y''_j \leq 1}} z'_{ij} + \sum_{\substack{e=(i,j) \in E \cup E' \\ x''_j + y''_j = 2}} z'_{ij} + \sum_{e=(i,j) \notin E \cup E'} z'_{ij} = \\ x''_i + y''_i + \sum_{\substack{e=(i,j) \in E \cup E' \\ x''_j + y''_j \leq 1}} z'_{ij} = \\ x''_i + y''_i + \sum_{e=(i,j) \in E \cup E'} z''_e - \sum_{\substack{e=(j,i) \in E \cup E' \\ x''_j + y''_j = 2}} z''_e = \\ x''_i + y''_i + \sum_{e=(i,j) \in E \cup E'} z''_e \geq 1, \end{array} \right. \quad \text{случај (1.)}$$

$$\left\{ \begin{array}{l} = x''_i - 1 + y''_i + \sum_{\substack{j=1 \\ j \neq i}}^n z'_{ij} \geq \\ x''_i - 1 + y''_i = 1, \end{array} \right. \quad \text{случај (2.)}$$

за свако  $i = 1, \dots, n$ .

Дате формуле су коректне,  $\sum_{e=(i,j) \in E \cup E', x''_j + y''_j = 2} z'_{ij} = 0$  и  $\sum_{e=(i,j) \notin E \cup E'} z'_{ij} = 0$  на основу дефиниције променљивих  $z'_{ij}$ . Израз  $\sum_{e=(j,i) \in E \cup E', x''_j + y''_j = 2} z''_e = 0$  је коректан на основу услова (3.37). Услови (3.37) се, за  $e = (j, i)$ , могу интерпретирати на следећи начин  $x''_i + y''_i - x''_j - y''_j + 1 \geq 2z''_e$  одакле директно следи да је  $z''_e = 0$ .

Коректност услова (3.27) се такође може показати: за  $a_{ij} = 0$  услови (3.27) тривијално важе с обзиром да је  $0 \geq 0$ , док је за  $a_{ij} = 1$  потребно размотрити четири случаја:

- услов (1.) важи за свако  $i, j = 1, \dots, n$ .

$$x'_j + y'_j - (x'_i + y'_i) + 1 = x''_j + y''_j - (x''_i + y''_i) + 1 \geq 2z''_e = 2z'_{ij}.$$

- услов (1.) важи за  $i$  док услов (2.) важи за  $j$ , ( $i = 1, \dots, n-1, j = 2, \dots, n, i < j$ ).

$$\begin{aligned} x'_j + y'_j - (x'_i + y'_i) + 1 &= x''_j - 1 + y''_j - (x''_i + y''_i) + 1 = \\ x''_j + y''_j - (x''_i + y''_i) &= 2 - (x''_i + y''_i) \geq 1 \geq 0 = 2z'_{ij} \end{aligned}$$

будући да је, из дефиниције,  $z'_{ij} = 0$  када је  $x''_j + y''_j = 2$ .

- услов (1.) важи за  $j$  док услов (2.) важи за  $i$  ( $j = 1, \dots, n-1, i = 2, \dots, n, j < i$ ).

$$\begin{aligned} x'_j + y'_j - (x'_i + y'_i) + 1 &= x''_j + y''_j - (x''_i - 1 + y''_i) + 1 = \\ x''_j + y''_j - (x''_i + y''_i) + 2 &\geq 2z''_e \geq 2z'_{ij}. \end{aligned}$$

- услов (2.) важи за свако  $i, j = 1, \dots, n$ .

$$\begin{aligned} x'_j + y'_j - (x'_i + y'_i) + 1 &= x''_j - 1 + y''_j - (x''_i - 1 + y''_i) + 1 = \\ x''_j + y''_j - (x''_i + y''_i) + 1 &\geq 2z''_e \geq 2z'_{ij}. \end{aligned}$$

Комбиновањем сва четири услова заједно са условом  $a_{ij} = 0$  следи да услови (3.27) важе. Услови (3.24), који се еквивалентно могу записати

$$x'_i + y'_i + \sum_{j \in N_i} (x'_j + y'_j) \geq 1,$$

важе у случају (2.), с обзиром да је

$$x'_i + y'_i + \sum_{j \in N_i} (x'_j + y'_j) = x''_i - 1 + y''_i + \sum_{j \in N_i} (x'_j + y'_j) \geq 1 + \sum_{j \in N_i} (x'_j + y'_j) \geq 1.$$

Уколико случај (1.) важи за  $i$  и услови  $x''_j + y''_j \leq 1$  важе за свако  $j$ , услови (3.24) се еквивалентно могу записати као  $x''_i + y''_i + \sum_{j \in N_i} (x''_j + y''_j) \geq 1$ . Последња релација важи уколико је бар једна од следећих релација задовољена:  $x''_i + y''_i \geq 1$  или је  $\sum_{j \in N_i} (x''_j + y''_j) \geq 1$ . Претпостављајући да услови (3.36)-(3.39) важе, из услова (3.36) следи да бар једна од релација  $x''_i + y''_i \geq 1$  и  $\sum_{e=(i,j) \in EU \cup E'} z''_e \geq 1$  важи. Уколико прва релација

важи, услови (3.24) су задовољени. Уколико друга релација важи, тада постоји грана  $\bar{e}$  таква да је  $z_{\bar{e}}'' = 1$ . Сада, из услова (3.37) следи да је  $x_j'' + y_j'' \geq 1$  за  $\bar{e} = (i, j)$  показујући да су услови (3.24) такође задовољени.

Уколико услов (1.) важи за  $i$  и постоји  $\bar{j}$  такво да је  $x_{\bar{j}}'' + y_{\bar{j}}'' = 2$ , тада се услови (3.24) могу записати као  $x_i' + y_i' + \sum_{j \in N_i, j \neq \bar{j}} (x_j' + y_j') + (x_{\bar{j}}' + y_{\bar{j}}') \geq x_{\bar{j}}' + y_{\bar{j}}' = x_{\bar{j}}'' - 1 + y_{\bar{j}}'' = 1$ .

Да бисмо показали валидност услова (3.30), потребно је размотрити наредна три случаја.

1.  $a_{kj} = a_{ki} = 0$ .

Ако узмемо да услови (3.24) важе, услови (3.30) се еквивалентно могу записати као

$$a_{kj}y_j' + a_{ki} + \sum_{l=1, l \neq i, l \neq j}^n a_{kl}(x_l' + y_l') = \sum_{l=1}^n a_{kl}(x_l' + y_l') - a_{ki}(x_i' + y_i') - a_{kj}(x_j' + y_j') = \sum_{l=1}^n a_{kl}(x_l' + y_l') \geq 1 \geq z_{ij}'$$

одакле се може закључити да су испуњени.

2.  $a_{kj} = 1, a_{ki} = 0$ .

Ако претпоставимо да су услови (3.38) такође задовољени, услови (3.30) су поново испуњени,

$$a_{kj}y_j' + a_{ki} + \sum_{l=1, l \neq i, l \neq j}^n a_{kl}(x_l' + y_l') = y_j' + \sum_{l \in N_k, l \neq i, l \neq j} (x_l' + y_l') =$$

случај  $((\forall l)(x_l'' + y_l'' \leq 1))$   $y_j'' + \sum_{l \in N_k, l \neq i, l \neq j} (x_l'' + y_l'') \geq z_e'' \geq z_{ij}'$

случај  $((\exists \bar{l})(x_{\bar{l}}'' + y_{\bar{l}}'' = 2))$   $y_j' + \sum_{\substack{l \in N_k, l \neq i, \\ l \neq j, l \neq \bar{l}}} (x_l' + y_l') + (x_{\bar{l}}' + y_{\bar{l}}') \geq x_{\bar{l}}' + y_{\bar{l}}' = 1 \geq z_{ij}'$

3.  $a_{kj} = 0 \wedge a_{ki} = 1$  или  $a_{kj} = a_{ki} = 1$ .

Будући да су  $a_{kj}y_i' \geq 0$  и  $a_{kl}(x_l' + y_l') \geq 0$  за свако  $k, l = 1, \dots, n, l \neq i$ , услови (3.30) су поново задовољени,

$$a_{kj}y_j' + a_{ki} + \sum_{\substack{l=1 \\ l \neq i \\ l \neq j}}^n a_{kl}(x_l' + y_l') = a_{kj}y_i' + 1 + \sum_{\substack{l=1 \\ l \neq i \\ l \neq j}}^n a_{kl}(x_l' + y_l') \geq 1 \geq z_{ij}'.$$

Из претходног се може закључити да су услови (3.30) испуњени.

Коначно, с обзиром да су  $x'_k \leq x''_k$  и  $y'_k = y''_k$  следи да је функција циља, која одговара **WRD1** формулацији, мања или једнака функцији циља која одговара **WRD2** формулацији, тј.  $Obj_{WRD1} \leq Obj_{WRD2}$ . Уједно, допуство решење **WRD2** формулације је допустиво решење **WRD1** формулације.

Комбновањем датих неједнакости теорема је доказана, тј.  $Obj_{WRD2} = Obj_{WRD1}$ . □

### 3.4 Поређења математичких формулација

У поглављу 3.1.1. представљене су две математичке формулације проблема римске доминације, **RD1** и **RD2** формулација. Обе формулације дефинисане су коришћењем  $2n$  (где је  $n$  број чворова графа) целобројних променљивих и  $2n$  ограничења. У оквиру поглавља 3.1.2. показано је да се **RD** проблем може описати и са  $n$  целобројних и  $n$  реалних променљивих, односно да је за описивање **RD** проблема довољно користити  $n$  ограничења (**RD1'**, **RD2'** и **RD2''** формулације). Слично, за проблем слабе римске доминације, у поглављу 3.3. описане су две математичке формулације. **WRD1** формулацијом **WRD** проблем може се описати коришћењем  $n^2 + 2n$  целобројних променљивих и  $n^3 + n^2 + 3n$  ограничења, док се модификованом формулацијом (**WRD2** формулација) исти проблем може описати коришћењем  $2n + 3m$  променљивих ( $m$  представља број грана графа) и  $3n + 2m$  ограничења. Поређења ради, у најгорем случају, односно када се ради о комплентим графовима, побољшан модел има  $O(n^2)$  ограничења док оригинални модел има  $O(n^3)$  ограничења.

Ради илустрације квалитета и ефикасности унапређених формулација за решавање **RD** и **WRD** проблема у односу на формулације познате из релевантне литературе, формулације описане у претходним секцијама кодирани су у C++ програмском језику тако да се инстанце на којима ће се вршити тестирање могу решити познатим решавачима за решавање проблема комбинаторне оптимизације, CPLEX 12.6 и Gurobi 5.6. Тестирања су извршена на Intel(R) Core(TM) i7-4700MQ CPU @ 2.40GHz рачунару са 8GB RAM меморије под Windows 10 оперативним системом. Поређења су извршена за четири различите класе графова: грид (енг. grid graphs), планарне (енг. planar graphs), нет (енг. net graphs) и произвољно генер-

исане графове (енг. random generated graphs). Прве три класе графова су познате из литературе док је произвољно генерисане графове генерисао и обезбедио Vincenzo Cuttò [14]. Класу грид графова чини 171 инстанца при чему је најмања инстанца са 9 чворова док је највећа са 600 чворова. Класу планарних графова чини 12 инстанци са 10, 20, 30, 50, 100, 150, 200, 250, 300, 400, 500 и 600 чворова. Класу нет графова представљају 4 инстанце са 100, 200, 400 и 600 чворова. Класу произвољно генерисаних графова чини укупно 108 инстанци, односно по 18 инстанци са по 50, 100, 150, 200, 250 и 300 чворва.

Детаљни резултати тестирања приказани су у Табелама 1 - 6 и 8 - 9, које су формиране на следећи начин. У прве три колоне приказани су име инстанце, број чворова и број грана графа коју дата инстанца представља. За инстанце за које су оптимизациони решавачи применом бар једне математичке формулације добили решење и доказали оптималност добијеног решења, такво решење приказано је у четвртој колони означеној са *opt*. У табелама у којима су приказани резултати тестирања инстанци које се не могу решити оптимално, колона *opt* замењена је колоном *best*. У колони *best* приказане су најниже вредности добијене за ту инстанцу независно од формулације и решавача. Потом следе колоне са резултатима за сваку формулацију посебно. У табелама у којима су приказане оптимално решене инстанце, у колонама  $t_{cpl}$  и  $t_{gur}$ , су за сваку формулацију посебно дата времена потребна да се одговарајућа инстанца оптимално реши применом CPLEX и Gurobi решавача (тим редом). Колоне које означавају решење инстанце су изостављане будући да су оптимално решене и да је решење таквих инстанци већ приказано у колони *opt*. У табелама, у којима је CPLEX (Gurobi) прекинуо процес решавања услед недостатка меморије или временског ограничења, у колони  $v_{cpl}$  ( $v_{gur}$ ) приказана је вредност коју је CPLEX (Gurobi) добио пре тренутка прекида, док у колони  $t_{cpl}$  ( $t_{gur}$ ) стоји "нм" уколико је до прекида дошло услед недостатка меморије, односно задато временско ограничење уколико је оно достигнуто. Ради боље прегледности у табелама су подебљане вредности које представљају најбоље резултате у односу на одговарајућа поређења.

## Поређење RD1, RD2, RD1', RD2' и RD2'' формулација

Резултати поређења математичких формулација дефинисаних за  $\mathcal{RD}$  проблем, приказани су Табелама 1 - 6. Због великог броја података, тестирања **RD1** и **RD1'** формулације приказани су засебно у односу на резултате тестирања **RD2** формулације и њених модификација. Будући да тестиране инстанце припадају различитим врстама графова, у табелама су прво приказани резултати тестирања инстанци грид типа, потом инстанци нет и планарног типа, док су последњи приказани резултати тестирања произвољно генерисаних инстанци. За сваку врсту графа инстанце су сортиране по броју чворова.

Резултати тестирања инстанци које су оптимално решене од стране оба решавача и применом свих пет формулација приказани су у Табелама 1 и 2.

Табела 1: Поређења **RD1** и **RD1'** формулација над инстанцама које су оптимално решене од стране оба решавача применом обе формулације

Инстанце				RD1		RD1'	
Име	$ V $	$ E $	$opt$	$t_{cpl}$	$t_{gur}$	$t_{cpl}$	$t_{gur}$
грид03×03	9	12	6	0.105	<b>0.01</b>	<b>0.059</b>	0.39
грид04×03	12	17	7	<b>0.128</b>	< <b>0.01</b>	0.161	0.37
грид03×04	12	17	7	<b>0.098</b>	<b>0.02</b>	1.180	0.33
грид05×03	15	22	8	<b>0.119</b>	<b>0.01</b>	0.173	0.68
грид03×05	15	22	8	<b>0.116</b>	<b>0.03</b>	2.189	0.28
грид04×04	16	24	8	<b>0.287</b>	0.03	2.452	<b>0.02</b>
грид06×03	18	27	10	<b>0.087</b>	<b>0.01</b>	2.063	1.29
грид03×06	18	27	10	<b>0.110</b>	<b>0.03</b>	1.695	0.06
грид04×05	20	31	11	<b>0.208</b>	<b>0.02</b>	0.547	0.55
грид05×04	20	31	11	0.140	<b>0.02</b>	<b>0.097</b>	2.22
грид07×03	21	32	12	<b>0.063</b>	<b>0.02</b>	0.289	1.05
грид03×07	21	32	12	<b>0.461</b>	<b>0.04</b>	1.785	0.98
грид03×08	24	37	13	<b>0.197</b>	<b>0.05</b>	0.240	1.40
грид08×03	24	37	13	<b>0.395</b>	<b>0.02</b>	1.248	1.66
грид04×06	24	38	13	<b>0.226</b>	<b>0.04</b>	0.510	0.29
грид06×04	24	38	13	<b>0.136</b>	<b>0.03</b>	0.492	0.28
грид05×05	25	40	14	1.488	<b>0.03</b>	<b>1.235</b>	1.67
грид03×09	27	42	14	<b>0.239</b>	<b>0.03</b>	1.333	0.45
грид09×03	27	42	14	<b>0.204</b>	<b>0.02</b>	1.994	0.03
грид07×04	28	45	14	<b>0.072</b>	<b>0.01</b>	1.842	0.02
грид04×07	28	45	14	<b>0.141</b>	<b>0.02</b>	0.365	0.19
грид03×10	30	47	16	<b>0.298</b>	<b>0.03</b>	0.333	1.33
грид06×05	30	49	16	<b>0.207</b>	<b>0.02</b>	1.211	1.53
грид10×03	30	47	16	<b>0.133</b>	<b>0.03</b>	1.348	2.10
грид05×06	30	49	16	<b>0.810</b>	<b>0.02</b>	2.698	1.60
грид04×08	32	52	16	0.751	<b>0.02</b>	<b>0.680</b>	0.07
грид08×04	32	52	16	<b>0.269</b>	<b>0.02</b>	0.444	0.60
грид11×03	33	52	18	3.834	<b>0.03</b>	<b>2.561</b>	0.11
грид03×11	33	52	18	<b>0.213</b>	<b>0.02</b>	1.384	1.71

Наставак Табеле 1 ...

Инстанце				RD1		RD1'	
Име	$ V $	$ E $	$opt$	$t_{cpl}$	$t_{gur}$	$t_{cpl}$	$t_{gur}$
грид07×05	35	58	18	<b>0.777</b>	<b>0.02</b>	2.802	1.01
грид05×07	35	58	18	<b>0.735</b>	<b>0.05</b>	3.871	1.40
грид09×04	36	59	18	<b>0.095</b>	<b>0.03</b>	2.517	0.28
грид04×09	36	59	18	<b>0.215</b>	<b>0.02</b>	3.489	1.60
грид06×06	36	60	19	<b>0.689</b>	<b>0.20</b>	2.487	0.27
грид12×03	36	57	19	3.044	<b>0.05</b>	<b>1.079</b>	0.66
грид03×12	36	57	19	<b>0.589</b>	<b>0.05</b>	2.865	0.92
грид03×13	39	62	20	<b>0.812</b>	<b>0.03</b>	1.909	0.82
грид13×03	39	62	20	1.451	<b>0.02</b>	<b>0.874</b>	0.18
грид10×04	40	66	20	<b>0.376</b>	<b>0.04</b>	2.018	0.61
грид04×10	40	66	20	<b>0.504</b>	<b>0.04</b>	1.485	0.99
грид05×08	40	67	21	<b>0.723</b>	<b>0.01</b>	1.732	1.40
грид08×05	40	67	21	<b>1.247</b>	<b>0.06</b>	3.528	2.09
грид03×14	42	67	22	<b>0.648</b>	<b>0.02</b>	3.106	0.60
грид14×03	42	67	22	0.891	<b>0.01</b>	<b>0.653</b>	0.11
грид07×06	42	71	22	<b>0.864</b>	<b>0.07</b>	3.311	1.32
грид06×07	42	71	22	<b>0.993</b>	<b>0.08</b>	2.321	0.90
грид11×04	44	73	22	<b>3.131</b>	<b>0.03</b>	3.694	1.06
грид04×11	44	73	22	<b>0.672</b>	<b>0.05</b>	3.091	1.42
грид15×03	45	72	24	1.766	<b>0.02</b>	<b>0.258</b>	0.43
грид03×15	45	72	24	<b>0.775</b>	<b>0.04</b>	1.576	0.75
грид09×05	45	76	23	<b>0.398</b>	<b>0.13</b>	3.357	1.67
грид05×09	45	76	23	<b>0.903</b>	<b>0.09</b>	2.968	0.30
грид04×12	48	80	24	<b>0.599</b>	<b>0.04</b>	3.040	0.61
грид12×04	48	80	24	<b>3.591</b>	<b>0.03</b>	3.829	0.67
грид08×06	48	82	24	<b>1.212</b>	<b>0.80</b>	3.538	1.24
грид06×08	48	82	24	<b>1.189</b>	<b>0.13</b>	3.435	0.89
грид07×07	49	84	24	<b>1.211</b>	<b>0.07</b>	2.810	1.14
грид10×05	50	85	26	<b>0.416</b>	<b>0.14</b>	1.875	0.60
грид05×10	50	85	26	<b>1.377</b>	<b>0.11</b>	4.406	0.56
грид13×04	52	87	26	4.385	<b>0.08</b>	<b>2.436</b>	0.17
грид04×13	52	87	26	<b>0.867</b>	<b>0.11</b>	2.072	1.06
грид06×09	54	93	27	<b>0.897</b>	<b>0.12</b>	2.391	<b>0.12</b>
грид09×06	54	93	27	<b>0.319</b>	<b>0.13</b>	2.084	0.72
грид05×11	55	94	28	<b>0.469</b>	<b>0.12</b>	4.047	0.46
грид11×05	55	94	28	<b>2.021</b>	<b>0.10</b>	4.582	0.67
грид04×14	56	94	28	<b>1.267</b>	<b>0.04</b>	3.233	1.41
грид14×04	56	94	28	<b>0.618</b>	<b>0.07</b>	2.884	0.83
грид07×08	56	97	28	<b>1.506</b>	<b>0.19</b>	4.746	1.21
грид08×07	56	97	28	<b>1.382</b>	<b>0.90</b>	3.343	1.00
грид04×15	60	101	30	<b>1.079</b>	<b>0.05</b>	4.138	0.36
грид15×04	60	101	30	2.163	<b>0.07</b>	<b>1.533</b>	0.12
грид05×12	60	103	30	<b>0.264</b>	<b>0.11</b>	2.146	0.29
грид06×10	60	104	30	<b>2.212</b>	<b>0.10</b>	4.344	1.39
грид10×06	60	104	30	<b>0.552</b>	<b>0.12</b>	1.046	1.03
грид12×05	60	103	30	<b>1.793</b>	<b>0.12</b>	3.483	2.05
грид09×07	63	110	31	<b>0.429</b>	<b>0.15</b>	2.708	0.57
грид07×09	63	110	31	<b>1.278</b>	<b>0.21</b>	3.435	1.69
грид08×08	64	112	32	<b>0.261</b>	<b>0.27</b>	3.828	0.55
грид05×13	65	112	33	<b>1.094</b>	<b>0.17</b>	3.332	1.01
грид13×05	65	112	33	4.563	<b>0.15</b>	<b>2.379</b>	0.49
грид11×06	66	115	33	<b>0.789</b>	<b>0.18</b>	3.436	1.63
грид06×11	66	115	33	<b>0.521</b>	<b>0.19</b>	3.139	1.88
грид10×07	70	123	34	0.625	<b>0.21</b>	<b>0.571</b>	1.79

Наставак Табеле 1 ...

Инстанце				RD1		RD1'	
Име	$ V $	$ E $	$opt$	$t_{cpl}$	$t_{gur}$	$t_{cpl}$	$t_{gur}$
грид07×10	70	123	34	<b>1.401</b>	<b>0.21</b>	1.843	1.89
грид14×05	70	121	35	<b>1.682</b>	<b>0.21</b>	3.770	0.84
грид05×14	70	121	35	<b>1.037</b>	<b>0.16</b>	3.941	0.73
грид06×12	72	126	36	<b>0.977</b>	<b>0.20</b>	1.834	2.03
грид09×08	72	127	35	<b>0.460</b>	<b>0.16</b>	3.180	0.59
грид08×09	72	127	35	<b>0.359</b>	<b>0.21</b>	3.051	1.66
грид12×06	72	126	36	<b>0.712</b>	<b>0.18</b>	1.119	1.43
грид05×15	75	130	38	<b>1.356</b>	<b>0.25</b>	4.727	1.90
грид15×05	75	130	38	3.652	<b>0.27</b>	<b>1.648</b>	1.64
грид07×11	77	136	38	<b>0.899</b>	<b>0.21</b>	3.639	1.67
грид11×07	77	136	38	<b>2.924</b>	<b>0.23</b>	3.843	1.56
грид06×13	78	137	38	<b>1.160</b>	<b>0.19</b>	4.419	0.75
грид13×06	78	137	38	3.110	<b>0.20</b>	<b>2.816</b>	0.56
грид10×08	80	142	39	<b>0.207</b>	<b>0.22</b>	1.696	1.55
грид08×10	80	142	39	<b>0.189</b>	<b>0.24</b>	1.869	1.74
грид09×09	81	144	38	<b>0.621</b>	<b>0.11</b>	5.194	0.96
грид06×14	84	148	41	2.068	<b>0.31</b>	<b>1.441</b>	0.40
грид07×12	84	149	41	<b>1.282</b>	<b>0.24</b>	4.372	1.98
грид12×07	84	149	41	0.976	<b>0.20</b>	<b>0.909</b>	0.78
грид14×06	84	148	41	1.757	<b>0.22</b>	<b>0.443</b>	1.11
грид11×08	88	157	42	<b>1.460</b>	6.90	2.746	<b>1.32</b>
грид08×11	88	157	42	<b>0.397</b>	<b>0.37</b>	2.886	1.39
грид09×10	90	161	43	<b>0.579</b>	<b>0.27</b>	3.192	2.28
грид15×06	90	159	44	3.046	<b>0.29</b>	<b>0.615</b>	1.20
грид10×09	90	161	43	<b>0.548</b>	<b>0.34</b>	2.063	0.85
грид06×15	90	159	44	<b>1.077</b>	<b>0.33</b>	3.038	1.29
грид07×13	91	162	44	<b>0.685</b>	<b>0.26</b>	4.182	1.69
грид13×07	91	162	44	3.033	<b>0.29</b>	<b>2.737</b>	0.37
грид12×08	96	172	46	<b>1.288</b>	<b>0.34</b>	2.988	1.91
грид08×12	96	172	46	<b>0.484</b>	<b>0.24</b>	0.863	2.49
грид14×07	98	175	47	3.381	<b>0.31</b>	<b>0.647</b>	0.84
грид07×14	98	175	47	<b>1.311</b>	<b>0.25</b>	3.492	2.61
грид09×11	99	178	47	<b>0.721</b>	<b>0.32</b>	2.800	1.69
грид11×09	99	178	47	<b>0.998</b>	<b>0.25</b>	2.673	1.57
грид10×10	100	180	48	<b>0.470</b>	<b>0.32</b>	4.603	2.86
грид08×13	104	187	50	<b>0.466</b>	<b>0.33</b>	1.129	2.66
грид13×08	104	187	50	<b>3.350</b>	<b>0.36</b>	3.512	1.49
грид15×07	105	188	50	2.270	<b>0.34</b>	<b>0.834</b>	0.82
грид07×15	105	188	50	<b>1.000</b>	<b>0.31</b>	3.516	2.00
грид12×09	108	195	51	<b>0.864</b>	<b>0.35</b>	2.683	2.71
грид09×12	108	195	51	<b>0.459</b>	<b>0.41</b>	1.819	1.90
грид11×10	110	199	52	2.269	<b>0.29</b>	<b>1.699</b>	2.07
грид10×11	110	199	52	<b>0.445</b>	<b>0.31</b>	0.781	1.91
грид14×08	112	202	53	1.992	2.30	<b>0.558</b>	<b>0.82</b>
грид08×14	112	202	53	<b>0.576</b>	<b>0.29</b>	2.229	2.40
грид09×13	117	212	55	<b>0.795</b>	<b>0.42</b>	1.683	2.49
грид13×09	117	212	55	<b>1.491</b>	<b>0.34</b>	3.873	1.46
грид15×08	120	217	57	2.454	<b>0.58</b>	<b>1.655</b>	3.90
грид10×12	120	218	56	<b>1.186</b>	<b>0.46</b>	3.693	2.46
грид08×15	120	217	57	<b>0.655</b>	<b>0.37</b>	4.149	5.05
грид12×10	120	218	56	<b>1.265</b>	<b>0.44</b>	2.007	3.40
грид11×11	121	220	57	<b>4.229</b>	<b>0.42</b>	4.723	5.28
грид09×14	126	229	58	<b>1.238</b>	<b>0.38</b>	2.998	1.41
грид14×09	126	229	58	<b>0.772</b>	<b>0.37</b>	0.981	0.89



Наставак Табеле 1 ...

Инстанце				RD1		RD1'	
Име	$ V $	$ E $	$opt$	$t_{cpl}$	$t_{gur}$	$t_{cpl}$	$t_{gur}$
грид10×13	130	237	61	<b>1.534</b>	<b>0.48</b>	2.907	7.08
грид13×10	130	237	61	<b>2.743</b>	<b>0.70</b>	3.555	2.23
грид11×12	132	241	62	2.016	<b>0.65</b>	<b>1.557</b>	5.67
грид12×11	132	241	62	<b>1.230</b>	<b>0.72</b>	1.506	6.00
грид09×15	135	246	63	<b>0.793</b>	<b>0.52</b>	1.901	4.34
грид15×09	135	246	63	2.589	<b>0.64</b>	<b>1.532</b>	2.47
грид14×10	140	256	65	3.845	<b>0.72</b>	<b>1.525</b>	3.64
грид10×14	140	256	65	<b>2.867</b>	<b>0.49</b>	4.933	3.91
грид13×11	143	262	66	<b>2.166</b>	<b>0.52</b>	3.369	2.30
грид11×13	143	262	66	4.242	<b>0.59</b>	<b>3.825</b>	5.28
грид12×12	144	264	67	<b>1.865</b>	<b>0.66</b>	2.815	5.16
грид10×15	150	275	70	<b>3.670</b>	<b>1.35</b>	3.949	8.96
грид15×10	150	275	70	2.645	<b>1.31</b>	<b>1.244</b>	9.91
грид11×14	154	283	71	<b>2.395</b>	<b>0.84</b>	5.150	9.01
грид14×11	154	283	71	4.085	<b>0.99</b>	<b>1.837</b>	7.64
грид12×13	156	287	72	3.586	<b>0.87</b>	6.483	<b>4.57</b>
грид13×12	156	287	72	<b>2.947</b>	<b>1.05</b>	6.596	5.35
грид11×15	165	304	76	<b>2.676</b>	<b>1.29</b>	6.145	10.05
грид15×11	165	304	76	2.831	<b>1.22</b>	<b>1.270</b>	9.56
грид14×12	168	310	77	5.220	<b>0.93</b>	<b>1.326</b>	6.49
грид12×14	168	310	77	<b>4.318</b>	<b>1.08</b>	5.922	5.11
грид13×13	169	312	78	6.437	<b>1.55</b>	<b>4.753</b>	7.81
грид12×15	180	333	82	7.050	<b>1.59</b>	<b>3.400</b>	9.86
грид15×12	180	333	82	6.229	<b>0.80</b>	<b>2.618</b>	9.52
грид14×13	182	337	83	5.849	<b>1.43</b>	<b>2.433</b>	9.07
грид13×14	182	337	83	6.186	<b>1.41</b>	<b>5.768</b>	8.99
грид15×13	195	362	89	3.644	<b>2.70</b>	<b>2.730</b>	14.54
грид13×15	195	362	89	12.179	<b>3.05</b>	<b>8.886</b>	12.03
грид14×14	196	364	88	4.878	<b>0.67</b>	<b>1.924</b>	2.51
грид14×15	210	391	95	3.225	<b>2.37</b>	<b>1.720</b>	17.42
грид15×14	210	391	95	3.888	<b>2.05</b>	<b>3.265</b>	17.38
грид15×15	225	420	102	<b>2.477</b>	<b>9.75</b>	3.364	37.14
грид20×20	400	760	176	176.900	<b>446.08</b>	<b>161.508</b>	1095.8
Нет-10-10	100	342	28	<b>0.034</b>	0.09	0.046	<b>0.03</b>
Нет-10-20	200	712	56	<b>0.058</b>	<b>0.07</b>	0.079	0.38
Нет-20-20	400	1482	98	<b>0.114</b>	<b>0.09</b>	0.134	0.50
Нет-30-20	600	2252	140	<b>0.154</b>	0.23	0.18	<b>0.10</b>
план10	10	27	3	<b>0.030</b>	<b>0.01</b>	0.034	0.21
план20	20	105	5	<b>0.041</b>	<b>0.02</b>	0.057	0.83
план30	30	182	5	<b>0.045</b>	<b>0.03</b>	0.060	0.50
план50	50	465	6	<b>0.073</b>	<b>0.02</b>	0.086	0.56
план100	100	1540	10	<b>0.284</b>	<b>0.53</b>	0.368	2.06
план150	150	2867	12	<b>1.064</b>	<b>9.27</b>	1.413	23.81
план200	200	4475	16	<b>191.312</b>	<b>287.68</b>	196.477	1120.5
Рандом-50-1	50	49	32	<b>0.041</b>	0.02	0.043	<b>0.01</b>
Рандом-50-2	50	49	33	<b>0.029</b>	0.02	0.037	<b>0.01</b>
Рандом-50-3	50	58	28	<b>0.030</b>	0.08	0.061	<b>0.02</b>
Рандом-50-4	50	54	30	<b>0.030</b>	0.06	0.050	<b>0.02</b>
Рандом-50-5	50	67	28	<b>0.042</b>	0.21	0.046	<b>0.03</b>
Рандом-50-6	50	86	25	<b>0.078</b>	<b>0.03</b>	0.094	0.07
Рандом-50-7	50	84	26	<b>0.063</b>	0.17	0.070	<b>0.06</b>
Рандом-50-8	50	95	23	<b>0.084</b>	0.23	0.091	<b>0.05</b>
Рандом-50-9	50	108	23	0.086	0.62	<b>0.073</b>	<b>0.07</b>
Рандом-50-10	50	112	22	0.107	0.36	<b>0.078</b>	<b>0.07</b>

Наставак Табеле 1 ...

Инстанце				RD1		RD1'	
Име	$ V $	$ E $	$opt$	$t_{cpl}$	$t_{gur}$	$t_{cpl}$	$t_{gur}$
Рандом-50-20	50	248	12	0.200	0.62	<b>0.169</b>	<b>0.09</b>
Рандом-50-30	50	373	9	0.169	0.34	<b>0.157</b>	<b>0.09</b>
Рандом-50-40	50	475	8	0.164	0.50	<b>0.153</b>	<b>0.16</b>
Рандом-50-50	50	597	6	0.167	1.05	<b>0.157</b>	<b>0.12</b>
Рандом-50-60	50	739	4	0.114	0.15	<b>0.108</b>	<b>0.12</b>
Рандом-50-70	50	860	4	0.061	0.28	<b>0.051</b>	<b>0.10</b>
Рандом-50-80	50	980	4	0.181	1.24	<b>0.138</b>	<b>0.09</b>
Рандом-50-90	50	1103	3	0.065	0.27	<b>0.050</b>	<b>0.11</b>
Рандом-100-1	100	100	61	<b>0.049</b>	0.12	0.066	<b>0.01</b>
Рандом-100-2	100	109	59	<b>0.049</b>	0.18	<b>0.049</b>	<b>0.01</b>
Рандом-100-3	100	181	48	<b>0.060</b>	0.11	0.062	<b>0.05</b>
Рандом-100-4	100	206	45	<b>0.231</b>	0.30	0.232	<b>0.11</b>
Рандом-100-5	100	231	39	<b>0.236</b>	0.55	0.240	<b>0.10</b>
Рандом-100-6	100	321	34	<b>0.230</b>	1.24	0.269	<b>0.32</b>
Рандом-100-7	100	317	32	0.236	0.60	<b>0.195</b>	<b>0.20</b>
Рандом-100-8	100	398	29	<b>0.234</b>	1.42	0.257	<b>0.56</b>
Рандом-100-9	100	430	27	<b>0.353</b>	1.41	0.387	<b>0.87</b>
Рандом-100-10	100	498	24	<b>0.871</b>	<b>1.47</b>	0.983	2.45
Рандом-100-20	100	981	14	<b>0.417</b>	1.40	0.539	<b>0.95</b>
Рандом-100-30	100	1477	11	3.661	10.61	<b>3.649</b>	<b>6.11</b>
Рандом-100-40	100	1945	8	0.701	3.41	<b>0.635</b>	<b>2.39</b>
Рандом-100-50	100	2483	7	0.811	3.56	<b>0.744</b>	<b>3.10</b>
Рандом-100-60	100	2985	6	0.523	4.77	<b>0.439</b>	<b>2.55</b>
Рандом-100-70	100	3435	5	0.454	2.88	<b>0.372</b>	<b>2.22</b>
Рандом-100-80	100	3935	4	<b>0.352</b>	4.30	0.362	<b>0.40</b>
Рандом-100-90	100	4446	4	<b>0.317</b>	2.40	0.351	<b>0.38</b>
Рандом-150-1	150	157	94	<b>0.050</b>	0.10	0.063	<b>0.01</b>
Рандом-150-2	150	243	78	0.191	1.11	<b>0.178</b>	<b>0.17</b>
Рандом-150-3	150	322	65	0.447	0.87	<b>0.438</b>	<b>0.24</b>
Рандом-150-4	150	437	53	<b>0.358</b>	1.57	0.433	<b>1.10</b>
Рандом-150-5	150	557	46	<b>0.976</b>	4.99	1.115	<b>4.34</b>
Рандом-150-6	150	705	38	<b>5.579</b>	13.78	5.734	<b>12.59</b>
Рандом-150-7	150	778	34	<b>2.492</b>	<b>13.41</b>	2.505	14.62
Рандом-150-8	150	906	31	<b>12.07</b>	<b>28.34</b>	12.401	30.61
Рандом-150-9	150	965	30	16.002	<b>32.42</b>	<b>15.577</b>	62.4
Рандом-150-10	150	1152	27	<b>56.194</b>	<b>80.14</b>	56.24	267.44
Рандом-150-20	150	2228	16	<b>29.974</b>	<b>41.58</b>	30.05	166.23
Рандом-150-30	150	3318	12	<b>28.619</b>	<b>75.28</b>	28.766	396.54
Рандом-150-40	150	4476	9	6.592	<b>40.34</b>	<b>6.499</b>	201.13
Рандом-150-50	150	5550	8	<b>8.74</b>	<b>51.5</b>	8.799	91.68
Рандом-150-60	150	6734	6	1.701	<b>8.30</b>	<b>1.569</b>	28.30
Рандом-150-70	150	7807	6	5.822	33.82	<b>5.652</b>	<b>29.64</b>
Рандом-150-80	150	8924	4	1.128	10.31	<b>1.029</b>	<b>4.93</b>
Рандом-150-90	150	10043	4	1.149	7.64	<b>1.045</b>	<b>4.43</b>
Рандом-200-1	200	229	116	<b>0.059</b>	0.19	0.082	<b>0.03</b>
Рандом-200-2	200	390	92	<b>0.355</b>	<b>2.15</b>	0.417	2.19
Рандом-200-3	200	581	69	<b>0.885</b>	<b>4.72</b>	1.003	15.96
Рандом-200-4	200	737	60	<b>10.192</b>	<b>43.36</b>	10.529	90.31
Рандом-200-5	200	1010	47	<b>50.231</b>	<b>84.78</b>	50.448	422.88
Рандом-200-6	200	1180	42	<b>332.77</b>	<b>675.14</b>	332.851	984.33
Рандом-200-7	200	1453	36	168.558	<b>308.36</b>	<b>168.053</b>	1468.5
Рандом-200-40	200	7907	10	141.298	<b>194.33</b>	<b>65.582</b>	907.59
Рандом-200-50	200	9895	8	43.664	<b>116.57</b>	<b>14.347</b>	512.45
Рандом-200-60	200	11971	6	5.39	<b>16.79</b>	<b>2.098</b>	56.80

Наставак Табеле 1 ...

Инстанце				RD1		RD1'	
Име	$ V $	$ E $	$opt$	$t_{cpl}$	$t_{gur}$	$t_{cpl}$	$t_{gur}$
Рандом-200-70	200	14059	6	25.798	129.18	<b>8.497</b>	<b>93.97</b>
Рандом-200-80	200	15918	4	4.043	21.28	<b>1.594</b>	<b>12.03</b>
Рандом-200-90	200	17821	4	5.437	23.61	<b>1.867</b>	<b>20.80</b>
Рандом-250-1	250	345	136	0.16	<b>0.23</b>	<b>0.049</b>	1.81
Рандом-250-2	250	633	97	15.633	<b>28.03</b>	<b>2.318</b>	61.62
Рандом-250-3	250	956	73	499.135	<b>1170.17</b>	<b>175.521</b>	3033.2
Рандом-250-40	250	12500	10	956.294	<b>1915.69</b>	<b>234.285</b>	2467.9
Рандом-250-50	250	15605	8	194.38	<b>364.33</b>	<b>36.864</b>	550.55
Рандом-250-60	250	18660	8	580.454	<b>1259.29</b>	<b>153.671</b>	3643.3
Рандом-250-70	250	21741	6	154.645	<b>139.9</b>	<b>32.326</b>	199.11
Рандом-250-80	250	24836	4	29.746	<b>38.51</b>	<b>5.127</b>	54.83
Рандом-250-90	250	27974	4	17.334	<b>17.87</b>	<b>5.376</b>	22.83
Рандом-300-1	300	481	145	4.548	3	<b>0.216</b>	<b>1.16</b>
Рандом-300-2	300	876	103	411.938	<b>82.46</b>	<b>80.454</b>	1049.9
Рандом-300-50	300	22520	8	625.574	<b>3227.83</b>	<b>308.016</b>	3258
Рандом-300-60	300	26952	8	1514.056	<b>3751.24</b>	<b>576.214</b>	7062.3
Рандом-300-70	300	31390	6	136.638	<b>334.07</b>	<b>54.132</b>	449.16
Рандом-300-80	300	35871	5	77.434	<b>36.27</b>	<b>15.064</b>	229.45
Рандом-300-90	300	40412	4	20.195	41.84	<b>9.159</b>	<b>29.95</b>

Табела 2: Поређења **RD2**, **RD2'** и **RD2''** формулација над инстанцама које су оптимално решене од стране оба решавача применом све три формулације

Инстанце				RD2		RD2'		RD2''	
Име	$ V $	$ E $	$opt$	$t_{cpl}$	$t_{gur}$	$t_{cpl}$	$t_{gur}$	$t_{cpl}$	$t_{gur}$
грид03×03	9	12	6	<b>0.035</b>	<b>0.01</b>	0.049	0.13	0.055	0.02
грид04×03	12	17	7	0.066	0.13	0.050	0.12	<b>0.040</b>	<b>0.01</b>
грид03×04	12	17	7	<b>0.066</b>	<b>0.02</b>	0.071	0.20	0.069	0.03
грид05×03	15	22	8	<b>0.030</b>	0.02	0.041	0.09	0.037	< <b>0.01</b>
грид03×05	15	22	8	0.097	0.02	0.098	0.06	<b>0.069</b>	<b>0.01</b>
грид04×04	16	24	8	<b>0.019</b>	0.15	0.048	<b>0.01</b>	0.034	<b>0.01</b>
грид06×03	18	27	10	<b>0.031</b>	<b>0.01</b>	0.040	<b>0.01</b>	0.045	0.03
грид03×06	18	27	10	<b>0.066</b>	<b>0.02</b>	0.077	0.03	0.115	0.03
грид04×05	20	31	11	0.044	<b>0.02</b>	0.061	0.30	<b>0.055</b>	0.04
грид05×04	20	31	11	<b>0.034</b>	0.06	0.074	0.06	0.052	<b>0.04</b>
грид07×03	21	32	12	0.061	0.02	<b>0.041</b>	<b>0.01</b>	0.067	0.02
грид03×07	21	32	12	0.08	<b>0.20</b>	<b>0.075</b>	0.21	0.093	0.03
грид03×08	24	37	13	0.049	<b>0.02</b>	0.050	0.11	<b>0.046</b>	0.03
грид08×03	24	37	13	<b>0.045</b>	<b>0.02</b>	0.054	<b>0.02</b>	0.052	0.04
грид04×06	24	38	13	<b>0.052</b>	<b>0.02</b>	0.063	<b>0.02</b>	0.056	<b>0.02</b>
грид06×04	24	38	13	0.063	<b>0.02</b>	0.065	<b>0.02</b>	<b>0.049</b>	<b>0.02</b>
грид05×05	25	40	14	<b>0.044</b>	0.03	0.055	0.07	0.083	<b>0.02</b>
грид03×09	27	42	14	0.044	0.23	<b>0.043</b>	0.04	0.045	<b>0.02</b>
грид09×03	27	42	14	<b>0.032</b>	0.23	0.129	0.02	0.037	<b>0.01</b>
грид07×04	28	45	14	0.058	0.02	<b>0.050</b>	<b>0.01</b>	0.052	0.02
грид04×07	28	45	14	<b>0.037</b>	0.02	0.059	0.19	0.054	<b>0.01</b>
грид03×10	30	47	16	0.062	0.16	<b>0.028</b>	0.10	0.044	<b>0.03</b>
грид06×05	30	49	16	0.065	<b>0.02</b>	<b>0.053</b>	<b>0.02</b>	0.057	0.04
грид10×03	30	47	16	<b>0.039</b>	0.04	0.053	<b>0.01</b>	0.046	0.03
грид05×06	30	49	16	0.063	0.04	0.068	<b>0.01</b>	<b>0.050</b>	0.03

Наставак Табеле 2 ...

Инстанце				RD2		RD2'		RD2''	
Име	V	E	opt	$t_{cpl}$	$t_{gur}$	$t_{cpl}$	$t_{gur}$	$t_{cpl}$	$t_{gur}$
грид04×08	32	52	16	0.047	0.08	0.054	0.22	<b>0.045</b>	<b>0.05</b>
грид08×04	32	52	16	<b>0.035</b>	0.21	0.066	<b>0.02</b>	0.046	<b>0.02</b>
грид11×03	33	52	18	<b>0.048</b>	0.18	0.048	<b>0.04</b>	0.062	0.05
грид03×11	33	52	18	0.052	0.19	0.058	0.05	<b>0.047</b>	<b>0.03</b>
грид07×05	35	58	18	<b>0.031</b>	0.04	0.103	<b>0.01</b>	0.099	0.02
грид05×07	35	58	18	0.078	<b>0.02</b>	0.173	<b>0.02</b>	<b>0.054</b>	0.03
грид09×04	36	59	18	0.061	0.04	0.060	<b>0.02</b>	<b>0.046</b>	0.03
грид04×09	36	59	18	<b>0.058</b>	0.09	0.071	0.03	6.986	<b>0.02</b>
грид06×06	36	60	19	0.147	0.19	0.076	<b>0.04</b>	<b>0.045</b>	0.05
грид12×03	36	57	19	<b>0.082</b>	0.04	0.090	0.05	0.121	<b>0.03</b>
грид03×12	36	57	19	0.096	0.13	<b>0.092</b>	0.50	0.100	<b>0.05</b>
грид03×13	39	62	20	<b>0.050</b>	0.19	0.064	0.26	0.052	<b>0.05</b>
грид13×03	39	62	20	0.084	<b>0.03</b>	0.070	<b>0.03</b>	<b>0.066</b>	0.05
грид10×04	40	66	20	0.082	0.65	0.088	<b>0.05</b>	<b>0.077</b>	<b>0.05</b>
грид04×10	40	66	20	<b>0.081</b>	0.22	0.098	0.36	0.084	<b>0.05</b>
грид05×08	40	67	21	<b>0.081</b>	0.13	0.099	0.08	0.084	<b>0.04</b>
грид08×05	40	67	21	0.112	0.34	0.151	<b>0.05</b>	<b>0.062</b>	<b>0.05</b>
грид03×14	42	67	22	0.044	<b>0.02</b>	0.055	0.30	<b>0.042</b>	0.07
грид14×03	42	67	22	<b>0.062</b>	0.03	<b>0.062</b>	<b>0.02</b>	0.084	0.06
грид07×06	42	71	22	<b>0.115</b>	0.44	0.137	0.21	0.130	<b>0.03</b>
грид06×07	42	71	22	<b>0.119</b>	0.33	0.185	0.18	0.132	<b>0.05</b>
грид11×04	44	73	22	<b>0.046</b>	0.13	0.071	<b>0.02</b>	0.080	0.05
грид04×11	44	73	22	<b>0.057</b>	0.27	0.103	0.18	0.100	<b>0.05</b>
грид15×03	45	72	24	<b>0.061</b>	0.44	0.134	<b>0.04</b>	0.146	<b>0.04</b>
грид03×15	45	72	24	<b>0.046</b>	0.15	0.154	0.35	0.112	<b>0.06</b>
грид09×05	45	76	23	<b>0.148</b>	0.56	0.175	0.10	0.153	<b>0.07</b>
грид05×09	45	76	23	0.200	0.18	0.224	0.10	<b>0.168</b>	<b>0.06</b>
грид04×12	48	80	24	<b>0.058</b>	0.09	0.078	0.27	0.100	<b>0.05</b>
грид12×04	48	80	24	<b>0.076</b>	0.11	0.136	<b>0.04</b>	0.272	0.06
грид08×06	48	82	24	<b>0.098</b>	0.26	0.176	<b>0.06</b>	0.137	0.08
грид06×08	48	82	24	<b>0.050</b>	0.10	0.205	0.08	0.188	<b>0.06</b>
грид07×07	49	84	24	<b>0.098</b>	0.51	0.176	<b>0.06</b>	0.137	<b>0.06</b>
грид10×05	50	85	26	<b>0.166</b>	0.42	0.194	0.12	0.170	<b>0.07</b>
грид05×10	50	85	26	0.291	0.14	0.292	0.10	<b>0.147</b>	<b>0.08</b>
грид13×04	52	87	26	<b>0.099</b>	0.39	0.142	0.12	0.138	<b>0.05</b>
грид04×13	52	87	26	0.164	0.35	0.194	0.17	<b>0.162</b>	<b>0.06</b>
грид06×09	54	93	27	<b>0.111</b>	0.84	0.144	0.11	0.155	<b>0.08</b>
грид09×06	54	93	27	<b>0.179</b>	0.57	0.297	0.11	0.184	<b>0.07</b>
грид05×11	55	94	28	<b>0.153</b>	0.12	0.182	0.14	0.200	<b>0.08</b>
грид11×05	55	94	28	0.239	0.53	0.216	0.13	<b>0.184</b>	<b>0.08</b>
грид04×14	56	94	28	<b>0.059</b>	0.33	0.104	0.13	0.097	<b>0.08</b>
грид14×04	56	94	28	<b>0.060</b>	0.05	0.126	<b>0.02</b>	0.124	0.07
грид07×08	56	97	28	0.175	0.50	0.173	0.09	<b>0.131</b>	<b>0.08</b>
грид08×07	56	97	28	<b>0.153</b>	0.54	0.204	0.12	0.200	<b>0.08</b>
грид04×15	60	101	30	0.150	0.11	0.095	0.14	<b>0.092</b>	<b>0.07</b>
грид15×04	60	101	30	<b>0.075</b>	0.15	0.121	<b>0.06</b>	0.097	0.07
грид05×12	60	103	30	0.225	0.19	0.156	0.17	<b>0.130</b>	<b>0.14</b>
грид06×10	60	104	30	<b>0.092</b>	0.42	0.158	0.14	0.185	<b>0.07</b>
грид10×06	60	104	30	0.157	0.30	0.183	0.17	<b>0.152</b>	<b>0.09</b>
грид12×05	60	103	30	<b>0.177</b>	0.23	0.198	0.14	0.184	<b>0.08</b>
грид09×07	63	110	31	0.208	0.48	0.172	0.17	<b>0.162</b>	<b>0.1</b>
грид07×09	63	110	31	<b>0.066</b>	0.35	0.295	0.16	0.137	<b>0.1</b>
грид08×08	64	112	32	0.119	0.64	<b>0.118</b>	0.16	0.121	<b>0.11</b>
грид05×13	65	112	33	0.176	0.20	0.208	0.18	<b>0.173</b>	<b>0.11</b>

Наставак Табеле 2 ...

Инстанце				RD2		RD2'		RD2''	
Име	V	E	opt	$t_{cpl}$	$t_{gur}$	$t_{cpl}$	$t_{gur}$	$t_{cpl}$	$t_{gur}$
грид13×05	65	112	33	0.229	0.45	0.211	<b>0.17</b>	<b>0.204</b>	0.12
грид11×06	66	115	33	<b>0.169</b>	0.23	0.195	0.2	0.174	<b>0.1</b>
грид06×11	66	115	33	<b>0.137</b>	0.87	0.213	0.16	0.168	<b>0.1</b>
грид10×07	70	123	34	<b>0.119</b>	0.49	0.137	0.22	0.12	<b>0.08</b>
грид07×10	70	123	34	0.149	0.65	0.196	0.2	<b>0.146</b>	<b>0.11</b>
грид14×05	70	121	35	0.225	1.01	0.202	0.16	<b>0.191</b>	<b>0.11</b>
грид05×14	70	121	35	0.216	0.23	0.208	0.18	<b>0.207</b>	<b>0.12</b>
грид06×12	72	126	36	0.236	0.45	0.18	0.18	<b>0.169</b>	<b>0.09</b>
грид09×08	72	127	35	0.17	0.35	0.185	0.23	<b>0.125</b>	<b>0.1</b>
грид08×09	72	127	35	<b>0.153</b>	0.39	0.21	0.21	0.182	<b>0.09</b>
грид12×06	72	126	36	<b>0.186</b>	0.52	0.272	0.2	0.315	<b>0.11</b>
грид05×15	75	130	38	0.269	0.27	0.225	0.19	<b>0.214</b>	<b>0.16</b>
грид15×05	75	130	38	<b>0.247</b>	0.61	0.316	0.27	0.56	<b>0.16</b>
грид07×11	77	136	38	0.182	0.74	0.174	0.23	<b>0.169</b>	<b>0.14</b>
грид11×07	77	136	38	<b>0.186</b>	0.27	0.238	0.2	0.304	<b>0.12</b>
грид06×13	78	137	38	<b>0.148</b>	0.87	0.26	0.26	0.213	<b>0.18</b>
грид13×06	78	137	38	<b>0.209</b>	0.4	0.32	0.23	0.263	<b>0.2</b>
грид10×08	80	142	39	<b>0.142</b>	0.55	0.156	0.25	0.147	<b>0.15</b>
грид08×10	80	142	39	<b>0.128</b>	0.34	0.167	0.27	0.153	<b>0.17</b>
грид09×09	81	144	38	<b>0.073</b>	0.63	0.121	0.24	0.184	<b>0.12</b>
грид06×14	84	148	41	<b>0.134</b>	0.63	0.197	0.24	0.163	<b>0.18</b>
грид07×12	84	149	41	0.193	0.6	0.205	0.33	<b>0.168</b>	<b>0.17</b>
грид12×07	84	149	41	<b>0.192</b>	0.84	0.266	0.25	0.23	<b>0.18</b>
грид14×06	84	148	41	<b>0.231</b>	0.6	0.288	0.31	0.24	<b>0.17</b>
грид11×08	88	157	42	0.184	0.73	0.144	0.32	<b>0.141</b>	<b>0.15</b>
грид08×11	88	157	42	0.233	1.08	<b>0.192</b>	0.29	0.206	<b>0.21</b>
грид09×10	90	161	43	<b>0.223</b>	0.6	0.307	0.33	0.245	<b>0.2</b>
грид15×06	90	159	44	<b>0.264</b>	0.76	0.348	0.33	0.356	<b>0.25</b>
грид10×09	90	161	43	0.282	0.45	0.361	0.34	<b>0.237</b>	<b>0.19</b>
грид06×15	90	159	44	0.284	0.75	0.376	0.36	<b>0.247</b>	<b>0.24</b>
грид07×13	91	162	44	0.252	0.68	0.181	0.33	<b>0.178</b>	<b>0.2</b>
грид13×07	91	162	44	<b>0.178</b>	0.58	0.201	0.31	0.251	<b>0.17</b>
грид12×08	96	172	46	0.269	0.75	<b>0.191</b>	0.28	0.573	<b>0.2</b>
грид08×12	96	172	46	<b>0.21</b>	0.52	0.441	<b>0.21</b>	0.231	<b>0.21</b>
грид14×07	98	175	47	0.315	0.6	<b>0.214</b>	<b>0.38</b>	0.353	1.18
грид07×14	98	175	47	<b>0.247</b>	0.62	0.315	0.27	0.3	<b>0.24</b>
грид09×11	99	178	47	0.259	0.52	0.203	0.32	<b>0.194</b>	<b>0.21</b>
грид11×09	99	178	47	0.374	0.52	<b>0.287</b>	0.22	0.409	<b>0.19</b>
грид10×10	100	180	48	<b>0.22</b>	0.94	0.239	<b>0.34</b>	0.244	0.52
грид08×13	104	187	50	0.457	0.92	0.351	0.39	<b>0.262</b>	<b>0.31</b>
грид13×08	104	187	50	0.438	1.03	<b>0.401</b>	0.37	0.647	<b>0.29</b>
грид15×07	105	188	50	<b>0.278</b>	0.53	0.494	0.29	0.879	<b>0.28</b>
грид07×15	105	188	50	<b>0.348</b>	0.64	0.504	0.34	0.354	<b>0.3</b>
грид12×09	108	195	51	0.3	0.65	<b>0.29</b>	0.27	0.617	<b>0.18</b>
грид09×12	108	195	51	0.303	0.72	0.305	0.29	<b>0.268</b>	<b>0.26</b>
грид11×10	110	199	52	0.318	0.69	0.27	0.31	<b>0.256</b>	<b>0.26</b>
грид10×11	110	199	52	0.328	0.97	0.364	0.29	<b>0.306</b>	<b>0.26</b>
грид14×08	112	202	53	<b>0.284</b>	0.69	0.32	<b>0.3</b>	0.414	0.36
грид08×14	112	202	53	0.296	0.56	0.365	<b>0.38</b>	<b>0.289</b>	0.66
грид09×13	117	212	55	0.376	0.61	0.323	0.53	<b>0.232</b>	<b>0.29</b>
грид13×09	117	212	55	0.484	1.08	0.444	0.35	<b>0.439</b>	<b>0.26</b>
грид15×08	120	217	57	0.636	0.91	<b>0.414</b>	<b>0.38</b>	0.777	0.81
грид10×12	120	218	56	<b>0.236</b>	0.68	0.483	0.52	0.409	<b>0.32</b>
грид08×15	120	217	57	0.451	1.41	0.488	<b>0.48</b>	<b>0.404</b>	0.96

Наставак Табеле 2 ...

Инстанце				RD2		RD2'		RD2''	
Име	V	E	opt	$t_{cpl}$	$t_{gur}$	$t_{cpl}$	$t_{gur}$	$t_{cpl}$	$t_{gur}$
грид12×10	120	218	56	<b>0.326</b>	1.67	0.506	<b>0.66</b>	0.83	0.82
грид11×11	121	220	57	0.531	1.25	0.494	<b>0.35</b>	<b>0.443</b>	0.61
грид09×14	126	229	58	0.363	0.37	0.299	0.38	<b>0.25</b>	<b>0.3</b>
грид14×09	126	229	58	0.363	0.85	<b>0.334</b>	0.29	0.469	<b>0.25</b>
грид10×13	130	237	61	0.592	0.73	<b>0.519</b>	<b>0.5</b>	0.568	0.96
грид13×10	130	237	61	<b>0.529</b>	1.33	0.551	<b>0.49</b>	0.788	0.9
грид11×12	132	241	62	0.571	1.63	<b>0.453</b>	<b>0.64</b>	0.917	1.04
грид12×11	132	241	62	0.483	1.24	<b>0.464</b>	<b>0.59</b>	1.124	1.18
грид09×15	135	246	63	<b>0.535</b>	0.98	0.724	0.57	0.662	<b>0.52</b>
грид15×09	135	246	63	0.753	1.09	<b>0.733</b>	<b>0.41</b>	0.887	0.99
грид14×10	140	256	65	<b>0.432</b>	1.52	0.47	<b>0.55</b>	0.522	0.58
грид10×14	140	256	65	0.481	2.35	0.484	<b>0.66</b>	<b>0.478</b>	0.71
грид13×11	143	262	66	0.77	2.11	0.52	0.95	<b>0.503</b>	<b>0.5</b>
грид11×13	143	262	66	<b>0.463</b>	0.98	0.677	<b>0.74</b>	0.622	1.38
грид12×12	144	264	67	<b>0.516</b>	1.52	0.803	<b>0.5</b>	1.026	1.3
грид10×15	150	275	70	<b>0.715</b>	2.12	0.8	<b>1.6</b>	0.977	1.84
грид15×10	150	275	70	1.282	2.42	<b>0.951</b>	<b>1.08</b>	1.426	1.91
грид11×14	154	283	71	<b>0.483</b>	1.49	0.515	7.1	0.579	<b>0.97</b>
грид14×11	154	283	71	<b>0.677</b>	2.32	0.67	1.4	0.871	<b>0.68</b>
грид12×13	156	287	72	<b>0.715</b>	1.4	0.806	<b>1.12</b>	1.014	1.15
грид13×12	156	287	72	0.786	1.58	0.903	1.28	<b>0.783</b>	<b>1.09</b>
грид11×15	165	304	76	0.917	1.9	<b>0.77</b>	<b>1.84</b>	1.463	2.04
грид15×11	165	304	76	1.255	2.74	<b>0.918</b>	<b>1.44</b>	1.006	2.08
грид14×12	168	310	77	<b>0.721</b>	2.88	0.804	<b>0.72</b>	1.185	1.37
грид12×14	168	310	77	<b>0.614</b>	2.7	1.079	2.01	1.554	<b>1.37</b>
грид13×13	169	312	78	<b>0.77</b>	2.88	1.084	<b>1.71</b>	1.042	2.9
грид12×15	180	333	82	0.957	6.4	<b>0.94</b>	1.23	1.013	<b>0.77</b>
грид15×12	180	333	82	<b>1.3</b>	2.36	1.368	2.31	2.09	<b>1.9</b>
грид14×13	182	337	83	0.978	2.77	<b>0.776</b>	<b>1.05</b>	1.278	2.29
грид13×14	182	337	83	1.001	4.3	<b>0.777</b>	<b>2.06</b>	1.209	2.18
грид15×13	195	362	89	2.033	5.38	<b>1.309</b>	<b>3.08</b>	1.657	3.97
грид13×15	195	362	89	2.228	8.36	<b>1.73</b>	<b>2.83</b>	1.894	3.89
грид14×14	196	364	88	<b>0.739</b>	6.2	0.879	2.66	0.969	<b>1.93</b>
грид14×15	210	391	95	1.486	4.87	<b>1.198</b>	<b>1.39</b>	6.194	4.66
грид15×14	210	391	95	<b>1.159</b>	9.7	1.24	5.54	2.749	<b>3.72</b>
грид15×15	225	420	102	3.152	13.59	1.357	<b>6.9</b>	<b>1.608</b>	8.31
грид20×20	400	760	176	91.63	263.84	<b>37.579</b>	<b>204.96</b>	44.775	479.33
Нет-10-10	100	342	28	0.073	0.69	0.054	0.17	<b>0.046</b>	<b>0.02</b>
Нет-10-20	200	712	56	0.081	0.52	0.049	0.06	<b>0.046</b>	<b>0.02</b>
Нет-20-20	400	1482	98	0.128	0.18	<b>0.06</b>	0.07	0.068	<b>0.02</b>
Нет-30-20	600	2252	140	0.168	0.7	0.094	0.03	<b>0.083</b>	<b>0.02</b>
план10	10	27	3	0.162	0.29	0.051	<b>0.01</b>	<b>0.03</b>	<b>0.01</b>
план20	20	105	5	0.058	1.26	0.054	<b>0.01</b>	<b>0.051</b>	0.09
план30	30	182	5	0.062	1.91	0.046	<b>0.02</b>	<b>0.043</b>	0.17
план50	50	465	6	0.202	0.47	0.051	<b>0.01</b>	<b>0.033</b>	0.23
план100	100	1540	10	0.559	3.4	0.653	<b>0.72</b>	<b>0.483</b>	0.77
план150	150	2867	12	2.035	38.83	1.39	<b>13.5</b>	<b>1.035</b>	25.92
план200	200	4475	16	155.969	176.01	124.34	<b>91.8</b>	<b>77.613</b>	562.1
Рандом-50-1	50	49	32	0.069	0.02	0.032	< <b>0.01</b>	<b>0.031</b>	0.09
Рандом-50-2	50	49	33	0.039	0.15	0.023	< <b>0.01</b>	<b>0.019</b>	0.09
Рандом-50-3	50	58	28	<b>0.046</b>	0.11	0.152	< <b>0.01</b>	0.047	0.08
Рандом-50-4	50	54	30	<b>0.041</b>	0.01	0.052	< <b>0.01</b>	0.042	0.02
Рандом-50-5	50	67	28	<b>0.051</b>	<b>0.02</b>	0.12	0.02	0.068	0.27
Рандом-50-6	50	86	25	0.076	0.03	0.088	<b>0.02</b>	<b>0.072</b>	0.24

Наставак Табеле 2 ...

Инстанце				RD2		RD2'		RD2''	
Име	V	E	opt	$t_{cpl}$	$t_{gur}$	$t_{cpl}$	$t_{gur}$	$t_{cpl}$	$t_{gur}$
Рандом-50-7	50	84	26	0.056	<b>0.02</b>	<b>0.053</b>	<b>0.02</b>	0.057	0.05
Рандом-50-8	50	95	23	0.065	0.22	0.076	<b>0.01</b>	<b>0.055</b>	0.33
Рандом-50-9	50	108	23	0.063	0.07	<b>0.058</b>	<b>0.04</b>	0.070	0.64
Рандом-50-10	50	112	22	<b>0.076</b>	0.34	0.095	<b>0.03</b>	0.085	0.35
Рандом-50-20	50	248	12	0.189	0.51	<b>0.308</b>	0.08	<b>0.170</b>	0.38
Рандом-50-30	50	373	9	<b>0.080</b>	0.23	0.276	<b>0.08</b>	0.207	0.63
Рандом-50-40	50	475	8	0.205	0.49	0.452	<b>0.17</b>	<b>0.178</b>	0.48
Рандом-50-50	50	597	6	0.141	0.38	0.357	<b>0.10</b>	<b>0.115</b>	0.64
Рандом-50-60	50	739	4	<b>0.058</b>	0.06	0.081	<b>0.01</b>	0.059	0.44
Рандом-50-70	50	860	4	0.062	0.09	0.061	<b>0.02</b>	<b>0.059</b>	0.59
Рандом-50-80	50	980	4	<b>0.060</b>	0.06	0.134	<b>0.02</b>	0.069	0.50
Рандом-50-90	50	1103	3	<b>0.054</b>	<b>0.02</b>	0.076	<b>0.02</b>	0.065	0.52
Рандом-100-1	100	100	61	0.049	0.09	0.040	<b>0.01</b>	<b>0.028</b>	0.04
Рандом-100-2	100	109	59	<b>0.040</b>	0.02	0.564	<b>0.01</b>	0.057	0.03
Рандом-100-3	100	181	48	0.064	0.52	<b>0.058</b>	<b>0.08</b>	0.072	<b>0.08</b>
Рандом-100-4	100	206	45	<b>0.202</b>	0.49	0.561	<b>0.17</b>	0.219	0.67
Рандом-100-5	100	231	39	0.222	0.24	<b>0.213</b>	<b>0.05</b>	0.350	0.38
Рандом-100-6	100	321	34	<b>0.236</b>	0.79	0.398	<b>0.23</b>	0.315	0.46
Рандом-100-7	100	317	32	<b>0.189</b>	0.54	1.010	<b>0.17</b>	0.265	0.60
Рандом-100-8	100	398	29	0.406	0.97	1.043	<b>0.41</b>	<b>0.380</b>	1.30
Рандом-100-9	100	430	27	<b>0.344</b>	1.33	0.836	0.44	0.391	<b>0.68</b>
Рандом-100-10	100	498	24	<b>0.564</b>	3.39	1.484	<b>1.65</b>	0.852	2.85
Рандом-100-20	100	981	14	0.478	1.88	0.776	<b>1.05</b>	<b>0.451</b>	1.62
Рандом-100-30	100	1477	11	<b>1.304</b>	9.80	2.480	<b>3.20</b>	5.596	11.79
Рандом-100-40	100	1945	8	0.735	<b>3.28</b>	1.509	3.69	<b>0.601</b>	3.52
Рандом-100-50	100	2483	7	0.765	10.69	0.948	8.04	<b>0.540</b>	<b>7.47</b>
Рандом-100-60	100	2985	6	<b>0.458</b>	2.75	1.515	<b>0.92</b>	0.467	6.19
Рандом-100-70	100	3435	5	0.512	3.63	0.981	<b>0.58</b>	<b>0.425</b>	4.63
Рандом-100-80	100	3935	4	<b>0.105</b>	0.41	0.213	<b>0.10</b>	0.197	1.32
Рандом-100-90	100	4446	4	0.200	0.23	0.138	<b>0.06</b>	<b>0.118</b>	1.53
Рандом-150-1	150	157	94	<b>0.053</b>	0.17	0.287	<b>0.01</b>	0.056	0.20
Рандом-150-2	150	243	78	0.203	0.57	0.160	<b>0.17</b>	<b>0.150</b>	0.40
Рандом-150-3	150	322	65	0.373	0.71	1.103	<b>0.17</b>	<b>0.325</b>	0.34
Рандом-150-4	150	437	53	0.834	1.54	0.717	<b>0.59</b>	<b>0.486</b>	1.35
Рандом-150-5	150	557	46	<b>1.623</b>	3.13	3.365	<b>0.85</b>	2.674	5.26
Рандом-150-6	150	705	38	<b>4.797</b>	8.89	7.355	<b>3.19</b>	4.956	10.16
Рандом-150-7	150	778	34	<b>4.903</b>	11.31	10.312	<b>3.13</b>	6.143	11.60
Рандом-150-8	150	906	31	<b>11.375</b>	14.99	19.434	<b>6.24</b>	11.431	26.40
Рандом-150-9	150	965	30	16.995	23.02	20.391	<b>8.94</b>	<b>11.673</b>	59.77
Рандом-150-10	150	1152	27	55.064	43.91	71.839	<b>17.23</b>	<b>45.863</b>	129.33
Рандом-150-20	150	2228	16	23.012	27.14	32.993	<b>11.12</b>	<b>20.728</b>	80.96
Рандом-150-30	150	3318	12	23.82	28.43	34.696	<b>20.21</b>	<b>21.173</b>	90.9
Рандом-150-40	150	4476	9	<b>7.274</b>	33.94	11.458	<b>19.25</b>	7.445	132.78
Рандом-150-50	150	5550	8	8.427	53.02	13.059	<b>24.94</b>	<b>7.541</b>	60.77
Рандом-150-60	150	6734	6	<b>0.775</b>	9.19	1.388	<b>3.45</b>	0.904	23.34
Рандом-150-70	150	7807	6	<b>3.894</b>	18.11	9.994	<b>8.65</b>	3.896	29.76
Рандом-150-80	150	8924	4	<b>0.192</b>	0.52	1.440	<b>0.22</b>	0.379	4.31
Рандом-150-90	150	10043	4	<b>0.350</b>	0.49	1.178	<b>0.15</b>	0.408	3.44
Рандом-200-1	200	229	116	<b>0.059</b>	0.02	0.084	<b>0.01</b>	0.072	0.45
Рандом-200-2	200	390	92	0.874	1.07	1.705	<b>0.37</b>	<b>0.541</b>	<b>0.37</b>
Рандом-200-3	200	581	69	5.454	3.81	4.254	<b>0.98</b>	<b>1.520</b>	5.20
Рандом-200-4	200	737	60	<b>8.797</b>	33.32	16.470	<b>5.22</b>	12.252	41.48
Рандом-200-5	200	1010	47	45.992	46.07	49.756	<b>20.58</b>	<b>31.180</b>	115.36
Рандом-200-7	200	1453	36	214.723	179.66	233.565	<b>75.96</b>	<b>158.090</b>	604.32

Наставак Табеле 2 ...

Инстанце				RD2		RD2'		RD2''	
Име	V	E	opt	$t_{cpl}$	$t_{gur}$	$t_{cpl}$	$t_{gur}$	$t_{cpl}$	$t_{gur}$
Рандом-200-30	200	5876	12	<b>75.831</b>	61.73	136.495	<b>60.21</b>	131.625	515.73
Рандом-200-40	200	7907	10	<b>60.885</b>	163.54	73.242	<b>153.54</b>	115.944	878.61
Рандом-200-50	200	9895	8	<b>19.946</b>	<b>80.72</b>	25.542	118.45	41.740	332.18
Рандом-200-60	200	11971	6	<b>2.936</b>	10.81	3.615	<b>8.01</b>	5.954	62.56
Рандом-200-70	200	14059	6	<b>8.075</b>	24.36	8.870	<b>23.39</b>	16.253	81.44
Рандом-200-80	200	15918	4	<b>0.367</b>	<b>0.53</b>	0.760	0.73	1.339	9.01
Рандом-200-90	200	17821	4	0.976	0.32	<b>0.392</b>	<b>0.31</b>	0.958	8.2
Рандом-250-1	250	345	136	0.146	<b>0.03</b>	<b>0.112</b>	<b>0.03</b>	0.518	0.85
Рандом-250-2	250	633	97	12.43	<b>6.02</b>	<b>3.670</b>	7.66	7.837	36.44
Рандом-250-3	250	956	73	514.863	<b>231.4</b>	<b>362.720</b>	368.38	459.505	2158.29
Рандом-250-40	250	12500	10	389.729	832.91	<b>182.286</b>	<b>815.15</b>	373.433	4520.82
Рандом-250-50	250	15605	8	68.395	<b>403.21</b>	<b>34.325</b>	434.1	75.471	1388.73
Рандом-250-60	250	18660	8	234.492	<b>948.72</b>	<b>95.914</b>	1110.85	242.71	5947.35
Рандом-250-70	250	21741	6	29.293	137.00	<b>12.263</b>	<b>128.68</b>	33.471	341.01
Рандом-250-80	250	24836	4	2.340	<b>36.00</b>	<b>0.981</b>	39.08	3.650	263.42
Рандом-250-90	250	27974	4	0.906	<b>1.25</b>	<b>0.486</b>	1.36	1.386	29.36
Рандом-300-1	300	481	145	0.516	<b>0.61</b>	<b>0.106</b>	0.68	0.853	0.73
Рандом-300-2	300	876	103	271.706	434.25	<b>103.488</b>	<b>226.88</b>	376.396	664.32
Рандом-300-50	300	22520	8	471.28	6102.06	<b>106.509</b>	1319.78	220.62	<b>1195.86</b>
Рандом-300-60	300	26952	8	287.351	1353.2	<b>193.105</b>	<b>885.15</b>	417.262	6392.44
Рандом-300-70	300	31390	6	25.146	222.67	<b>23.656</b>	<b>90.73</b>	51.194	372.42
Рандом-300-80	300	35871	5	<b>11.201</b>	307.91	26.392	<b>110.80</b>	23.203	173.29
Рандом-300-90	300	40412	4	1.757	2.46	<b>0.724</b>	<b>0.66</b>	1.612	13.52

Као што се из Табеле 1 може приметити, у поређењу са модификацијом, време решавања грид инстанци мање је када је коришћена **RD1** формулација (код 126 од 170 тестираних инстанци применом CPLEX решавача, односно код 167 инстанци применом Gurobi решавача). Инстанце нет типа се CPLEX решавачем брже решавају када се користи **RD1** формулација, док су Gurobi решавачем обе формулације једнако успешне. Све инстанце планарног типа брже се решавају **RD1** формулацијом, док је већи број инстанци произвољно генерисаног типа брже решен применом њене модификације **RD1'** (48 од 83 тестираних инстанци CPLEX, односно 49 инстанци Gurobi решавачем). Ако сада погледамо резултате тестирања **RD2** формулације и њених модификација (Табела 2) видећемо да се CPLEX решавачем инстанце грид типа најбрже решавају **RD2** формулацијом (87 инстанци), док се Gurobi решавачем највећи број инстанци најбрже решава **RD2''** формулацијом (114 инстанци). Инстанце нет типа најбрже су решене **RD2''** формулацијом независно од примењеног решавача. Инстанце планарног типа су, међутим, CPLEX решавачем најбрже решене **RD2''** модификацијом, док су Gurobi решавачем најбрже решене у случају када је коришћена **RD2'** формулација. Највише инстанци произвољно генерисаног типа је CPLEX решавачем најбрже реш-



ено коришћењем **RD2** формулације (36 од 83 тестиране инстанце), док је Gurobi решавачем најбрже решено коришћењем **RD2'** формулације (71 од 83 тестиране инстанце).

Резултати тестирања инстанци које су оптимално решене али чија оптимална решења нису достигнута применом оба оптимизациона решавача и коришћењем свих пет формулација приказани су у Табелама 3 и 4.

Табела 3: Поређења **RD1** и **RD1'** формулација над инстанцама које су оптимално решене али оптималност добијеног решења није доказана од стране оба оптимизациона решавача коришћењем бар једне формулације

Инстанце				RD1				RD1'			
Име	$ V $	$ E $	$opt$	$v_{cpl}$	$t_{cpl}$	$v_{gur}$	$t_{gur}$	$v_{cpl}$	$t_{cpl}$	$v_{gur}$	$t_{gur}$
грид30×20	600	1100	260	<i>onm</i>	3758.364	<i>onm</i>	<b>2123.72</b>	<i>onm</i>	<b>3620.132</b>	260	7200
план250	250	6495	17	19	нм	<i>onm</i>	<b>2187.83</b>	19	нм	18	7200
Рандом-200-8	200	1580	34	36	нм	<i>onm</i>	<b>502.11</b>	36	нм	35	7200
Рандом-200-9	200	1781	31	32	нм	<i>onm</i>	<b>478.69</b>	32	нм	32	7200
Рандом-200-30	200	5876	12	13	нм	<i>onm</i>	<b>1633.8</b>	<i>onm</i>	<b>139.829</b>	<i>onm</i>	3322.3
Рандом-250-4	250	1194	62	<i>onm</i>	2705.189	<i>onm</i>	<b>3989.29</b>	<i>onm</i>	<b>1043.172</b>	62	7200
Рандом-250-30	250	9347	13	<i>onm</i>	5799.799	14	7200	<i>onm</i>	<b>1018.686</b>	14	7200
Рандом-300-3	300	1363	76	77	нм	<i>onm</i>	<b>6595.36</b>	77	нм	76	7200
Рандом-300-20	300	8925	20	20	нм	20	7200	<i>onm</i>	<b>38335.6</b>	21	7200
Рандом-300-40	300	17934	10	<i>onm</i>	721.239	<i>onm</i>	<b>2497.33</b>	<i>onm</i>	<b>386.037</b>	11	7200

Табела 4: Поређења **RD2**, **RD2'** и **RD2''** формулација над инстанцама које су оптимално решене али оптималност добијеног решења није доказана од стране оба оптимизациона решавача коришћењем бар једне формулације

Инстанце				RD2				RD2'				RD2''			
Име	$ V $	$ E $	$opt$	$v_{cpl}$	$t_{cpl}$	$v_{gur}$	$t_{gur}$	$v_{cpl}$	$t_{cpl}$	$v_{gur}$	$t_{gur}$	$v_{cpl}$	$t_{cpl}$	$v_{gur}$	$t_{gur}$
грид30×20	600	1100	260	261	нм	<i>onm</i>	1590.86	<i>onm</i>	<b>1279.438</b>	<i>onm</i>	<b>1432.62</b>	<i>onm</i>	1466.063	260	7200
план250	250	6495	17	18	нм	<i>onm</i>	3769.46	18	нм	<i>onm</i>	<b>1841.74</b>	18	нм	18	7200
Рандом-200-6	200	1180	42	45	нм	<i>onm</i>	163.17	<i>onm</i>	<b>151.979</b>	<i>onm</i>	<b>88.87</b>	<i>onm</i>	202.475	<i>onm</i>	272.64
Рандом-200-8	200	1580	34	36	нм	<i>onm</i>	1116.58	36	нм	<i>onm</i>	<b>665.56</b>	<i>onm</i>	<b>590.152</b>	<i>onm</i>	1350.94
Рандом-200-9	200	1781	31	32	нм	<i>onm</i>	1024.11	<i>onm</i>	<b>1170.18</b>	<i>onm</i>	<b>581.37</b>	32	нм	<i>onm</i>	3020.64
Рандом-200-10	200	1984	30	30	нм	<i>onm</i>	3936.76	30	нм	<i>onm</i>	<b>2979.21</b>	30	нм	30	7200
Рандом-200-20	200	4116	18	18	нм	<i>onm</i>	4702.01	18	нм	<i>onm</i>	<b>2890.08</b>	18	нм	18	7200
Рандом-250-4	250	1194	62	62	нм	<i>onm</i>	<b>1311.99</b>	<i>onm</i>	<b>1277.281</b>	<i>onm</i>	3306.03	<i>onm</i>	2448.703	62	7200
Рандом-250-5	250	1540	51	53	нм	<i>onm</i>	<b>3733.52</b>	52	нм	52	7200	52	нм	52	7200
Рандом-250-6	250	1869	44	46	нм	<i>onm</i>	<b>4183.86</b>	46	нм	44	7200	46	нм	44	7200
Рандом-300-40	300	17934	10	11	нм	<i>onm</i>	2565.14	<i>onm</i>	<b>395.403</b>	<i>onm</i>	<b>1154.37</b>	<i>onm</i>	528.868	<i>onm</i>	4944.92

Инстанца ”грид30×20” (Табеле 3 и 4) оптимално је решена применом оба решавача једино у случајевима када су коришћене **RD1** и **RD2'**

формулација, с тим да је брже време решавања остварено коришћењем **RD2'** формулације. Инстанцу "план250" оптимално је решио само Gurobi решавач (коришћењем **RD1**, **RD2** и **RD2'** формулација), а најкраће време решавања остварено је применом **RD2'** формулације. Од 8 инстанци произвољно генерисаног типа (Табела 3), CPLEX решавач је већи број инстанци оптимално решио коришћењем **RD1'** формулације (5 инстанци), док је Gurobi решавач већи број инстанци оптимално решио коришћењем **RD1** формулације (6 инстанци). Ако поредимо времена решавања инстанци које су решене применом обе формулације (Табела 3), приметимо да модификована метода има краће време решавања уколико су инстанце решаване CPLEX решавачем, односно краће време решавања Gurobi решавачем када се користи **RD1** формулација. Када су резултати из Табеле 4 у питању, Gurobi решавач је коришћењем **RD2** формулације оптимално решио свих 9 приказаних инстанци произвољно генерисаног типа, односно 7 инстанци коришћењем **RD2'** формулације и 4 инстанце коришћењем **RD2''** формулације, док је CPLEX имао успеха у оптималном решавању само са модификацијама **RD2** формулације (по 4 инстанце су оптимално решене коришћењем **RD2'** и **RD2''** формулације). Међу инстанцама које су оптимално решене од стране **RD2** формулације и обе њене модификације, време решавања је мање када се користи **RD2'** формулација.

Међу инстанцама које нису оптимално решене (Табеле 5 и 6) применом **RD1** формулације добијене су мање вредности у поређењу са вредностима добијеним њеном **RD1'** модификацијом како за инстанце планарног типа тако и за инстанце произвољно генерисаног типа. Међутим, модификације **RD2** формулације имају боље резултате у односу на ту формулацију. Ради лакше прегледности, за сваку инстанцу посебно, најмање вредности у Табелама 5 и 6 су подебљане.

Табела 5: Поређења **RD1** и **RD1'** формулација над инстанцама за које оптималност добијеног решења није доказана

Инстанце				<b>RD1</b>				<b>RD1'</b>			
Име	$ V $	$ E $	<i>best</i>	$v_{cpl}$	$t_{cpl}$	$v_{gur}$	$t_{gur}$	$v_{cpl}$	$t_{cpl}$	$v_{gur}$	$t_{gur}$
план300	300	8199	20	<b>20</b>	нм	<b>20</b>	7200	<b>20</b>	нм	21	7200
план400	400	13750	24	<b>24</b>	нм	<b>24</b>	7200	<b>24</b>	нм	25	7200
план500	500	19352	27	<b>27</b>	нм	28	7200	<b>27</b>	нм	28	7200
план600	600	24353	32	33	нм	33	7200	33	нм	34	7200
Рандом-200-10	200	1984	30	<b>30</b>	нм	<b>30</b>	7200	<b>30</b>	нм	<b>30</b>	7200
Рандом-200-20	200	4116	18	<b>18</b>	нм	<b>18</b>	7200	<b>18</b>	нм	<b>18</b>	7200
Рандом-250-5	250	1540	51	54	нм	<b>51</b>	7200	54	нм	52	7200

Наставак Табеле 5 ...

Инстанце				RD1				RD1'			
Име	$ V $	$ E $	$best$	$v_{cpl}$	$t_{cpl}$	$v_{gur}$	$t_{gur}$	$v_{cpl}$	$t_{cpl}$	$v_{gur}$	$t_{gur}$
Рандом-250-6	250	1869	44	46	нм	<b>44</b>	7200	46	нм	46	7200
Рандом-250-7	250	2159	41	43	нм	<b>41</b>	7200	43	нм	43	7200
Рандом-250-8	250	2420	37	38	нм	38	7200	38	нм	38	7200
Рандом-250-9	250	2880	34	35	нм	<b>34</b>	7200	35	нм	35	7200
Рандом-250-10	250	3167	30	31	нм	<b>30</b>	7200	31	нм	33	7200
Рандом-250-20	250	6280	19	<b>19</b>	нм	20	7200	<b>19</b>	нм	20	7200
Рандом-300-4	300	1765	64	66	нм	<b>64</b>	7200	65	нм	65	7200
Рандом-300-5	300	2259	54	56	нм	55	7200	56	нм	56	7200
Рандом-300-6	300	2657	49	51	нм	<b>49</b>	7200	50	нм	50	7200
Рандом-300-7	300	3101	44	46	нм	45	7200	46	нм	<b>44</b>	7200
Рандом-300-8	300	3620	39	40	нм	40	7200	40	нм	<b>39</b>	7200
Рандом-300-9	300	4008	36	37	нм	<b>36</b>	7200	37	нм	37	7200
Рандом-300-10	300	4413	34	35	нм	35	7200	35	нм	37	7200
Рандом-300-30	300	13648	14	<b>14</b>	нм	<b>14</b>	7200	<b>14</b>	нм	<b>14</b>	7200

Табела 6: Поређења  $RD2$ ,  $RD2'$  и  $RD2''$  формулација над инстанцама за које оптималност добијеног решења није доказана

Инстанце				RD2				RD2'				RD2''			
Име	$ V $	$ E $	$best$	$v_{cpl}$	$t_{cpl}$	$v_{gur}$	$t_{gur}$	$v_{cpl}$	$t_{cpl}$	$v_{gur}$	$t_{gur}$	$v_{cpl}$	$t_{cpl}$	$v_{gur}$	$t_{gur}$
план300	300	8199	20	<b>20</b>	нм	<b>20</b>	7200	21	нм	<b>20</b>	7200	<b>20</b>	нм	21	7200
план400	400	13750	24	<b>24</b>	нм	<b>24</b>	7200	<b>24</b>	нм	<b>24</b>	7200	<b>24</b>	нм	<b>24</b>	7200
план500	500	19352	28	<b>28</b>	нм	<b>28</b>	7200	<b>28</b>	нм	<b>28</b>	7200	<b>28</b>	нм	29	7200
план600	600	24353	32	<b>32</b>	нм	34	7200	<b>32</b>	нм	34	7200	33	нм	<b>32</b>	7200
Рандом-250-7	250	2159	41	42	нм	42	7200	42	нм	43	7200	42	нм	42	7200
Рандом-250-8	250	2420	37	39	нм	38	7200	39	нм	38	7200	39	нм	<b>37</b>	7200
Рандом-250-9	250	2880	34	35	нм	<b>34</b>	7200	35	нм	36	7200	35	нм	<b>34</b>	7200
Рандом-250-10	250	3167	30	32	нм	32	7200	32	нм	32	7200	32	нм	31	7200
Рандом-250-20	250	6280	19	<b>19</b>	нм	<b>19</b>	7200	20	нм	<b>19</b>	7200	<b>19</b>	нм	<b>19</b>	7200
Рандом-250-30	250	9347	13	14	нм	14	7200	14	нм	14	7200	14	нм	14	7200
Рандом-300-4	300	1765	64	66	нм	65	7200	66	нм	66	7200	66	нм	65	7200
Рандом-300-5	300	2259	54	56	нм	56	7200	55	нм	56	7200	<b>54</b>	нм	57	7200
Рандом-300-6	300	2657	49	50	нм	51	7200	50	нм	<b>49</b>	7200	51	нм	51	7200
Рандом-300-7	300	3101	44	<b>44</b>	нм	46	7200	<b>44</b>	нм	46	7200	<b>44</b>	нм	46	7200
Рандом-300-8	300	3620	39	40	нм	42	7200	40	нм	40	7200	40	нм	43	7200
Рандом-300-9	300	4008	36	38	нм	<b>36</b>	7200	37	нм	<b>36</b>	7200	<b>36</b>	нм	37	7200
Рандом-300-10	300	4413	34	35	нм	36	7200	35	нм	36	7200	35	нм	<b>34</b>	7200
Рандом-300-20	300	8925	20	<b>20</b>	нм	<b>20</b>	7200	<b>20</b>	нм	<b>20</b>	7200	<b>20</b>	нм	<b>20</b>	7200
Рандом-300-30	300	13648	14	<b>14</b>	нм	<b>14</b>	7200	<b>14</b>	нм	<b>14</b>	7200	<b>14</b>	нм	<b>14</b>	7200

Ако истовремено упоредимо свих пет формулација приметимо да је коришћењем  $RD2''$  формулације оптимално решење добијено за најкраће време за највећи број инстанци грид (93 инстанце) и нет типа (4 инстанце). Слично, инстанце из класе планарних графова су у више случајева (6 инстанци) за краће време оптимално решене коришћењем унапређених формулација. Најзад, од 108 инстанци произвољно генерисаног типа већина (69 инстанци) је најбрже оптимално решена при-

меном  $RD2'$  формулације.

У Табели 7 приказани су сумарни резултати поређења формулација које су познате из литературе и њихових модификација над свим тестираним инстанцама (295 инстанци) по следећим критеријумима:

1. Број оптимално решених инстанци CPLEX решавачем.
2. Број оптимално решених инстанци Gurobi решавачем.
3. Број најбрже оптимално решених инстанци CPLEX решавачем.
4. Број најбрже оптимално решених инстанци Gurobi решавачем.
5. Број инстанци за које је CPLEX решавачем добијена вредност из колоне *best*.
6. Број инстанци за које је Gurobi решавачем добијена вредност из колоне *best*.

Табела 7: Сумарни резултати поређења формулација из литературе са њиховим модификацијама

Критеријум	$RD1$ и $RD2$	$RD1'$ , $RD2'$ и $RD2''$
1. CPLEX: бр. опт. решених	268	270
2. Gurobi: бр. опт. решених	276	274
3. CPLEX: бр. најбрже решених	111	76
4. Gurobi: бр. најбрже решених	64	114
5. CPLEX: бр. најбоље решених	6	7
6. Gurobi: бр. најбоље решених	9	7

### Поређење формулација WRD1 и WRD2

Слично као и за  $RD$  проблем, формулације описане у поглављу 3.3. детаљно су тестиране у оквиру овог поглавља. Тестирање  $WRD1$  и  $WRD2$  формулације извршено је за 105 различитих инстанци (101 инстанца са највише 100 чворова и 4 инстанце са више од 100 чворова). Резултати тестирања одређених инстанци са више од 100 чворова су изостављени зато што су за те инстанце комерцијални решавачи прекидали процес решавања услед недостатка меморије или су их решавали, али нису могли да пронађу и докажу оптималност добијеног решења у оквиру временског

ограничења. Сумарни преглед резултата дат је у наставку, док су детаљне информације приказане у Табелама 8 и 9.

У Табели 8 приказане су 84 оптимално решене инстанце (54 GRID, 4 планар, 1 нет и 25 произвољно генерисаних инстанци). Коришћењем **WRD1** формулације оптимално решење достигнуто је за 69 инстанци (CPLEX решавачем пронађена су оптимална решења за 43 GRID, 4 планарне и 22 произвољно генерисане инстанце, а Gurobi решавачем за 30 GRID, 3 планарне и 11 произвољно генерисаних инстанци), док је коришћењем **WRD2** формулације оптимално решење достигнуто за 84 инстанце (CPLEX решавачем оптимална решења пронађена су за 53 GRID, 4 планарне, 1 нет и 24 произвољно генерисане инстанце, а Gurobi решавачем за 53 GRID, 4 планарне, 1 нет и 25 произвољно генерисаних инстанци). Када су времена решавања инстанци у питању, примена **WRD2** формулације далеко је успешнија у односу на **WRD1** формулацију: CPLEX решавачем најбрже је решено 59 инстанци, а Gurobi решавачем 83 (CPLEX је **WRD1** формулацијом најбрже решио 22 инстанце, а Gurobi ни једну).

Табела 8: Поређење **WRD1** и **WRD2** формулације над оптимално решеним инстанцама

Инстанце				<b>WRD1</b>				<b>WRD2</b>			
Име	$ V $	$ E $	$opt$	$v_{cpl}$	$t_{cpl}$	$v_{gur}$	$t_{gur}$	$v_{cpl}$	$t_{cpl}$	$v_{gur}$	$t_{gur}$
GRID03×03	9	12	4	<i>onm</i>	0.113	<i>onm</i>	0.19	<i>onm</i>	<b>0.093</b>	<i>onm</i>	<b>0.06</b>
GRID03×04	12	17	5	<i>onm</i>	<b>0.14</b>	<i>onm</i>	0.3	<i>onm</i>	0.249	<i>onm</i>	<b>0.12</b>
GRID03×05	15	22	6	<i>onm</i>	<b>0.226</b>	<i>onm</i>	0.69	<i>onm</i>	0.296	<i>onm</i>	<b>0.17</b>
GRID04×04	16	24	7	<i>onm</i>	<b>0.407</b>	<i>onm</i>	1.48	<i>onm</i>	0.453	<i>onm</i>	<b>0.27</b>
GRID03×06	18	27	7	<i>onm</i>	<b>0.466</b>	<i>onm</i>	1.6	<i>onm</i>	0.515	<i>onm</i>	<b>0.25</b>
GRID04×05	20	31	8	<i>onm</i>	<b>0.438</b>	<i>onm</i>	3.06	<i>onm</i>	0.953	<i>onm</i>	<b>0.49</b>
GRID03×07	21	32	8	<i>onm</i>	0.846	<i>onm</i>	3.31	<i>onm</i>	<b>0.781</b>	<i>onm</i>	<b>0.44</b>
GRID03×08	24	37	10	<i>onm</i>	<b>1.121</b>	<i>onm</i>	11.21	<i>onm</i>	1.281	<i>onm</i>	<b>1.48</b>
GRID04×06	24	38	9	<i>onm</i>	<b>0.593</b>	<i>onm</i>	8.02	<i>onm</i>	1.124	<i>onm</i>	<b>0.8</b>
GRID05×05	25	40	9	<i>onm</i>	<b>0.656</b>	<i>onm</i>	10.23	<i>onm</i>	1	<i>onm</i>	<b>0.77</b>
GRID03×09	27	42	10	<i>onm</i>	<b>1.469</b>	<i>onm</i>	15.68	<i>onm</i>	1.687	<i>onm</i>	<b>0.72</b>
GRID04×07	28	45	11	<i>onm</i>	1.843	<i>onm</i>	44.2	<i>onm</i>	<b>1.546</b>	<i>onm</i>	<b>5.11</b>
GRID03×10	30	47	12	<i>onm</i>	<b>1.843</b>	<i>onm</i>	41.48	<i>onm</i>	3.218	<i>onm</i>	<b>5.94</b>
GRID05×06	30	49	11	<i>onm</i>	2.143	<i>onm</i>	39.69	<i>onm</i>	<b>2.14</b>	<i>onm</i>	<b>1.55</b>
GRID04×08	32	52	12	<i>onm</i>	1.891	<i>onm</i>	116.13	<i>onm</i>	<b>1.718</b>	<i>onm</i>	5.55
GRID03×11	33	52	13	<i>onm</i>	<b>2.594</b>	<i>onm</i>	83.49	<i>onm</i>	2.687	<i>onm</i>	<b>3.98</b>
GRID05×07	35	58	13	<i>onm</i>	<b>3.531</b>	<i>onm</i>	180.58	<i>onm</i>	3.546	<i>onm</i>	<b>8.47</b>
GRID03×12	36	57	14	<i>onm</i>	4	<i>onm</i>	166.91	<i>onm</i>	<b>3.984</b>	<i>onm</i>	<b>10.7</b>
GRID04×09	36	59	13	<i>onm</i>	<b>2.86</b>	<i>onm</i>	153.66	<i>onm</i>	2.984	<i>onm</i>	<b>7.93</b>
GRID06×06	36	60	13	<i>onm</i>	4.687	<i>onm</i>	215.47	<i>onm</i>	<b>2.843</b>	<i>onm</i>	<b>6.47</b>
GRID03×13	39	62	15	<i>onm</i>	4.781	<i>onm</i>	271.44	<i>onm</i>	<b>4.558</b>	<i>onm</i>	<b>9.73</b>
GRID04×10	40	66	15	<i>onm</i>	4.954	<i>onm</i>	497.37	<i>onm</i>	<b>4.109</b>	<i>onm</i>	<b>9.16</b>
GRID05×08	40	67	14	<i>onm</i>	5.191	<i>onm</i>	551.75	<i>onm</i>	<b>4.64</b>	<i>onm</i>	<b>7.56</b>
GRID03×14	42	67	16	<i>onm</i>	<b>4.829</b>	<i>onm</i>	314.72	<i>onm</i>	5.372	<i>onm</i>	<b>13.64</b>
GRID06×07	42	71	15	<i>onm</i>	6.927	<i>onm</i>	454.75	<i>onm</i>	<b>5.801</b>	<i>onm</i>	<b>12.28</b>
GRID04×11	44	73	16	<i>onm</i>	6.13	<i>onm</i>	912.76	<i>onm</i>	<b>5.5</b>	<i>onm</i>	<b>14.1</b>

Наставак Табеле 8 ...

Инстанце				WRD1				WRD2			
Име	$ V $	$ E $	$opt$	$v_{cpl}$	$t_{cpl}$	$v_{gur}$	$t_{gur}$	$v_{cpl}$	$t_{cpl}$	$v_{gur}$	$t_{gur}$
грид03×15	45	72	17	<i>onm</i>	8.36	<i>onm</i>	712.63	<i>onm</i>	<b>7.789</b>	<i>onm</i>	<b>19.69</b>
грид05×09	45	76	16	<i>onm</i>	<b>7.908</b>	<i>onm</i>	942.68	<i>onm</i>	10.781	<i>onm</i>	<b>19.48</b>
грид04×12	48	80	17	<i>onm</i>	14.364	<i>onm</i>	891.93	<i>onm</i>	<b>12.84</b>	<i>onm</i>	<b>14.91</b>
грид06×08	48	82	18	<i>onm</i>	26.662	18	7200	<i>onm</i>	<b>25.499</b>	<i>onm</i>	<b>108.8</b>
грид07×07	49	84	18	<i>onm</i>	<b>9.845</b>	18	7200	<i>onm</i>	12.844	<i>onm</i>	<b>80.37</b>
грид05×10	50	85	18	<i>onm</i>	11.469	<i>onm</i>	5953.67	<i>onm</i>	<b>10.61</b>	<i>onm</i>	<b>71.83</b>
грид04×13	52	87	19	<i>onm</i>	13.16	22	HM	<i>onm</i>	<b>11.813</b>	<i>onm</i>	<b>69.13</b>
грид06×09	54	93	19	<i>onm</i>	26.988	55	HM	<i>onm</i>	<b>25.539</b>	<i>onm</i>	<b>79.64</b>
грид05×11	55	94	19	<i>onm</i>	14.365	59	HM	<i>onm</i>	<b>11.424</b>	<i>onm</i>	<b>107.34</b>
грид04×14	56	94	20	<i>onm</i>	35.6	25	HM	<i>onm</i>	<b>35.326</b>	<i>onm</i>	<b>52.53</b>
грид07×08	56	97	20	<i>onm</i>	<b>21.882</b>	23	HM	<i>onm</i>	42.48	<i>onm</i>	<b>107.63</b>
грид04×15	60	101	22	<i>onm</i>	<b>40.256</b>	27	HM	<i>onm</i>	51.518	<i>onm</i>	<b>172.62</b>
грид05×12	60	103	21	<i>onm</i>	41.364	28	HM	<i>onm</i>	<b>14.88</b>	<i>onm</i>	<b>127.57</b>
грид06×10	60	104	21	<i>onm</i>	51.458	26	HM	<i>onm</i>	<b>35.713</b>	<i>onm</i>	<b>58.41</b>
грид07×09	63	110	22	<i>onm</i>	995.684	27	HM	<i>onm</i>	<b>70.259</b>	<i>onm</i>	<b>101.89</b>
грид08×08	64	112	23	23	7200	26	HM	<i>onm</i>	<b>171.925</b>	<i>onm</i>	1664.53
грид05×13	65	112	23	<i>onm</i>	1825.01	28	HM	<i>onm</i>	<b>67.007</b>	<i>onm</i>	<b>195.49</b>
грид06×11	66	115	24	24	7200	29	HM	<i>onm</i>	<b>381.771</b>	<i>onm</i>	<b>1001.9</b>
грид05×14	70	121	24	<i>onm</i>	5368.89	28	HM	<i>onm</i>	<b>73.489</b>	<i>onm</i>	<b>677.63</b>
грид07×10	70	123	25	25	7200	30	HM	<i>onm</i>	<b>618.089</b>	<i>onm</i>	<b>829.21</b>
грид06×12	72	126	26	27	HM	31	HM	<i>onm</i>	<b>1166.41</b>	<i>onm</i>	<b>3254.34</b>
грид08×09	72	127	25	25	7200	32	HM	<i>onm</i>	<b>435.146</b>	<i>onm</i>	<b>4195.49</b>
грид05×15	75	130	26	26	7200	31	HM	<i>onm</i>	<b>465.006</b>	<i>onm</i>	<b>288.06</b>
грид07×11	77	136	27	28	7200	32	HM	<i>onm</i>	<b>988.596</b>	<i>onm</i>	<b>1543.24</b>
грид06×13	78	137	27	28	7200	33	HM	<i>onm</i>	<b>1005.13</b>	<i>onm</i>	<b>6657.33</b>
грид08×10	80	142	28	28	7200	33	HM	<i>onm</i>	<b>2162.81</b>	<i>onm</i>	<b>3693.43</b>
грид09×09	81	144	28	29	7200	35	HM	<i>onm</i>	<b>737.579</b>	28	7200
грид07×12	84	149	29	30	7200	35	HM	30	HM	<i>onm</i>	<b>4637.38</b>
план10	10	27	3	<i>onm</i>	<b>0.156</b>	<i>onm</i>	0.31	<i>onm</i>	0.206	<i>onm</i>	<b>0.17</b>
план20	20	105	3	<i>onm</i>	<b>1.363</b>	<i>onm</i>	2.75	<i>onm</i>	1.423	<i>onm</i>	<b>1.36</b>
план30	30	182	5	<i>onm</i>	8.724	<i>onm</i>	26.73	<i>onm</i>	<b>7.49</b>	<i>onm</i>	<b>11.45</b>
план50	50	465	6	<i>onm</i>	302.815	6	HM	<i>onm</i>	<b>153.215</b>	<i>onm</i>	<b>98.49</b>
Нет-10-10	100	342	20	90	HM	21	HM	<i>onm</i>	<b>598.98</b>	<i>onm</i>	<b>148.213</b>
Рандом-50-1	50	49	24	<i>onm</i>	1.125	28	HM	<i>onm</i>	<b>0.281</b>	<i>onm</i>	<b>0.72</b>
Рандом-50-2	50	49	23	<i>onm</i>	1.156	<i>onm</i>	85.66	<i>onm</i>	<b>0.343</b>	<i>onm</i>	<b>1.48</b>
Рандом-50-3	50	58	24	<i>onm</i>	1.265	<i>onm</i>	53.95	<i>onm</i>	<b>0.39</b>	<i>onm</i>	<b>0.66</b>
Рандом-50-4	50	54	24	<i>onm</i>	1.239	<i>onm</i>	66.97	<i>onm</i>	<b>0.484</b>	<i>onm</i>	<b>0.58</b>
Рандом-50-5	50	67	22	<i>onm</i>	1.635	<i>onm</i>	79.59	<i>onm</i>	<b>0.968</b>	<i>onm</i>	<b>1.41</b>
Рандом-50-6	50	86	19	<i>onm</i>	4.915	<i>onm</i>	118.34	<i>onm</i>	<b>2.053</b>	<i>onm</i>	<b>12.31</b>
Рандом-50-7	50	84	19	<i>onm</i>	4.677	<i>onm</i>	248.14	<i>onm</i>	<b>3.171</b>	<i>onm</i>	<b>9.33</b>
Рандом-50-8	50	95	17	<i>onm</i>	15.308	<i>onm</i>	226.28	<i>onm</i>	<b>3.093</b>	<i>onm</i>	<b>4.11</b>
Рандом-50-9	50	108	17	<i>onm</i>	90.188	<i>onm</i>	865.54	<i>onm</i>	<b>26.373</b>	<i>onm</i>	<b>39.31</b>
Рандом-50-10	50	112	16	<i>onm</i>	43.673	<i>onm</i>	716.16	<i>onm</i>	<b>6.781</b>	<i>onm</i>	<b>34.2</b>
Рандом-50-20	50	248	9	<i>onm</i>	1304.74	13	HM	<i>onm</i>	<b>346.264</b>	<i>onm</i>	<b>792.33</b>
Рандом-50-30	50	373	7	<i>onm</i>	1143.9	9	HM	<i>onm</i>	<b>476.278</b>	<i>onm</i>	<b>943.77</b>
Рандом-50-40	50	475	6	<i>onm</i>	1767.91	9	HM	<i>onm</i>	<b>1447.32</b>	<i>onm</i>	<b>3734.27</b>
Рандом-50-50	50	597	5	<i>onm</i>	1856.43	6	HM	<i>onm</i>	<b>1779.27</b>	<i>onm</i>	<b>1545.06</b>
Рандом-50-60	50	739	4	<i>onm</i>	509.084	6	HM	<i>onm</i>	<b>260.999</b>	<i>onm</i>	<b>210.71</b>
Рандом-50-70	50	860	3	<i>onm</i>	203.182	3	HM	<i>onm</i>	<b>168.483</b>	<i>onm</i>	<b>156.14</b>
Рандом-50-80	50	980	3	<i>onm</i>	<b>90.813</b>	<i>onm</i>	381.44	<i>onm</i>	235.731	<i>onm</i>	<b>172.01</b>
Рандом-50-90	50	1103	2	<i>onm</i>	120.87	<i>onm</i>	248.67	<i>onm</i>	<b>38.206</b>	<i>onm</i>	<b>36.53</b>
Рандом-100-1	100	100	46	<i>onm</i>	8.769	123	HM	<i>onm</i>	<b>0.64</b>	<i>onm</i>	<b>2.61</b>
Рандом-100-2	100	109	46	<i>onm</i>	11.385	125	HM	<i>onm</i>	<b>0.843</b>	<i>onm</i>	<b>5.3</b>
Рандом-100-3	100	181	37	<i>onm</i>	704.88	109	HM	<i>onm</i>	<b>7.421</b>	<i>onm</i>	<b>23.17</b>

Наставак Табеле 8 ...

Инстанце				WRD1				WRD2			
Име	$ V $	$ E $	$opt$	$v_{cpl}$	$t_{cpl}$	$v_{gur}$	$t_{gur}$	$v_{cpl}$	$t_{cpl}$	$v_{gur}$	$t_{gur}$
Рандом-100-4	100	206	34	<i>onm</i>	2533.22	99	HM	<i>onm</i>	<b>61.702</b>	<i>onm</i>	<b>66.7</b>
Рандом-100-5	100	231	32	35	HM	101	HM	<i>onm</i>	<b>164.502</b>	<i>onm</i>	<b>562.64</b>
Рандом-100-6	100	321	26	28	7200	90	HM	26	HM	<i>onm</i>	<b>5806.74</b>
Рандом-100-7	100	317	25	26	7200	87	HM	<i>onm</i>	<b>4009.38</b>	<i>onm</i>	<b>4493.7</b>

У Табели 9 приказани су резултати тестирања 21 инстанце код којих оптимална решења нису позната (9 грид, 2 планарне, 3 нет и 7 произвољно генерисаних инстанци). Као што се из приказаних резултата може видети, коришћењем **WRD1** формулације CPLEX решавач је дао нека решења пре прекидања програма за свега 12 инстанци, а Gurobi за 13. Коришћењем **WRD2** формулације CPLEX је за 18 инстанци дао неке вредности пре прекидања програма, док је Gurobi решавач дао решења за све приказане инстанце. Поред тога што су Gurobi решавачем за све приказане инстанце добијене неке вредности, добијене вредности су уједно и најмање за већину њих. За само 3 инстанце Gurobi решавач није пронашао вредности из колоне *best* када је коришћена **WRD2** формулација. За те три инстанце вредности из колоне *best* пронађене су коришћењем исте формулације CPLEX решавачем. Ради лакше прегледности најниже достигнуте вредности су подебљане.

Табела 9: Поређења **WRD1** и **WRD2** формулација над инстанцама за које оптимална решења нису позната

Инстанце				WRD1				WRD2			
Име	$ V $	$ E $	<i>best</i>	$v_{cpl}$	$t_{cpl}$	$v_{gur}$	$t_{gur}$	$v_{cpl}$	$t_{cpl}$	$v_{gur}$	$t_{gur}$
грид06×14	84	148	30	<b>30</b>	7200	34	HM	<b>30</b>	HM	<b>30</b>	7200
грид08×11	88	157	31	<b>31</b>	7200	36	HM	<b>31</b>	HM	<b>31</b>	7200
грид06×15	90	159	32	<b>32</b>	7200	95	HM	33	HM	<b>32</b>	7200
грид09×10	90	161	31	32	7200	90	HM	32	HM	<b>31</b>	7200
грид07×13	91	162	32	<b>32</b>	7200	109	HM	33	HM	<b>32</b>	7200
грид08×12	96	172	33	38	HM	106	HM	34	HM	<b>33</b>	7200
грид07×14	98	175	34	<b>34</b>	7200	115	HM	<b>34</b>	HM	35	7200
грид09×11	99	178	35	<b>35</b>	7200	110	HM	<b>35</b>	HM	<b>35</b>	7200
грид10×10	100	180	35	36	HM	100	HM	36	HM	35	7200
план100	100	1540	9		HM		HM	<b>9</b>	HM	10	7200
план150	150	2867	13		HM		HM		HM	<b>13</b>	7200
Нет-10-20	200	712	40		HM		HM	<b>40</b>	HM	42	7200
Нет-20-20	400	1482	83		HM		HM	87	HM	<b>83</b>	7200
Нет-30-20	600	2252	122		HM		HM	142	HM	<b>122</b>	7200
Рандом-100-8	100	317	23	25	7200	91	HM	24	7200	<b>23</b>	7200
Рандом-100-9	100	430	21	23	7200	81	HM	22	7200	<b>21</b>	7200
Рандом-100-10	100	498	19	22	HM	76	HM	20	7200	<b>19</b>	7200
Рандом-100-20	100	981	12		HM	64	HM	<b>12</b>	7200	<b>12</b>	7200
Рандом-100-30	100	1477	11		HM		HM	11	HM	<b>11</b>	HM

Наставак Табеле 9 ...

Инстанце				WRD1				WRD2			
Име	$ V $	$ E $	$best$	$v_{cpl}$	$t_{cpl}$	$v_{gur}$	$t_{gur}$	$v_{cpl}$	$t_{cpl}$	$v_{gur}$	$t_{gur}$
Рандом-100-40	100	1945	9		нм		нм	нм		<b>9</b>	нм
Рандом-100-50	100	2483	7		нм		нм	нм		<b>7</b>	нм

Слично као и за  $\mathcal{RD}$  проблем, сумарно поређење резултата тестирања **WRD1** и **WRD2** формулација на основу критеријума:

1. Број оптимално решених инстанци CPLEX решавачем.
2. Број оптимално решених инстанци Gurobi решавачем.
3. Број најбрже оптимално решених инстанци CPLEX решавачем.
4. Број најбрже оптимално решених инстанци Gurobi решавачем.
5. Број инстанци за које је CPLEX решавачем добијена вредност из колоне  $best$ .
6. Број инстанци за које је Gurobi решавачем добијена вредност из колоне  $best$ .

приказано је у Табели 10.

Табела 10: Сумарно поређење резултата тестирања **WRD1** и **WRD2** формулација на основу 6 критеријума

Критеријум	WRD1	WRD2
1. CPLEX: бр. опт. решених	70	83
2. Gurobi: бр. опт. решених	44	83
3. CPLEX: бр. најбрже опт. решених	22	61
4. Gurobi: бр. најбрже опт. решених	0	83
5. CPLEX: бр. најбоље решених	6	8
6. Gurobi: бр. најбоље решених	0	18



## 4 Алгоритми за решавање проблема римске и слабе римске доминације на графу

Методe за решавање проблема комбинаторне оптимизације најчешће се деле на егзактне и приближне методе. Егзактне методе се, даље, могу класификовати на програмирање са ограничењима, динамичко програмирање и оптимизацију засновану на гранању, док се под приближним методама подразумевају детерминистичке и недетерминистичке методе. За разлику од егзактних метода, које гарантују оптималност добијеног решења, приближне методе не само што не могу да гарантују оптималност добијеног решења, већ не могу ни да процене колико се добијено решење разликује од оптималног. Међутим, будући да постоји велики број проблема комбинаторне оптимизације код којих примена егзактних метода није увек ефикасна, што због временског ограничења, што због недостатка меморије или неког другог разлога, потпуно је оправдана жеља да се развију приближне методе које би за разумно време налазиле решење задовољавајућег квалитета. Управо се из тог разлога последњих година доста ради на развоју и усавршавању метахеуристичких метода као једне подврсте недетерминистичких алгоритама. Сам назив ”метахеуристика” потиче од грчке речи *heuriskein* која у преводу значи ”уметност проналажења новог правила у решавању проблема” и префикса *meta* који је такође грчког порекла и значи ”методологија вишег нивоа”. Дакле, метахеуристичке методе се могу дефинисати као хеуристичке методе вишег нивоа које се могу применити на већем броју оптимизационих проблема. Међутим, иако се метахеуристичке методе могу применити на широј класи оптимизационих проблема, не постоји јасна дефиниција када је употреба метахеуристика ефикаснија у односу на егзактне методе. Наиме, код неких проблема из класе NP-тешких проблема малих димензија егзактним методама се може доћи до оптималног решења, док неки проблеми из класе полиномијално решивих проблема могу имати неефикасне егзактне методе које услед великог броја локалних оптимума користе превише времена за извршавање. Стога се идеја метахеуристичких метода састоји управо у томе да се нађу ефикасне и брзе процедуре које доводе до неког довољно квалитетног решења у кратком року.

Данас су метахеуристике опште прихваћене методе за решавање проблема комбинаторне оптимизације, о чему сведочи велики број научних

скупова посвећених њиховим применама и развоју. У зависности од броја решења која користе у процесу претраге, метахеуристике се деле на метахеуристике вођене једним решењем и метахеуристике вођене популацијом могућих решења (тзв. популационе метахеуристике), а неке од најпознатијих метахеуристика су, на пример, симулирано каљење (енг. Simulated annealing, SA), вишестартно локално претраживање (енг. Multi-start Local Search, MLS), метода променљивих околина (енг. Variable Neighborhood Search, VNS), табу претраживање (енг. Tabu Search, TS), Greedy Randomized Adaptive Search Procedure (GRASP)), генетски алгоритми (енг. Genetic Algorithm, GA), еволутивни алгоритми (енг. Evolutionary algorithm, EA), мравље колоније (енг. Ant Colony Optimization, ACO), пчелиње колоније (енг. Bee colony Optimization, BCO), неуронске мреже (енг. Neural networks, NN). Поред наведених развијено је још пуно других метода заснованих на понашањима из природе, међутим, како детаљни приказ свих метахеуристика превазилази оквир овог рада, за више информација о метахеуристикима и њиховим применама видети [86].

У оквиру тезе конструисани су егзактна метода за решавање проблема слабе римске доминације на једној специјалној класи графова и нови алгоритам заснован на методи променљивих околина за решавање проблема римске и проблема слабе римске доминације на свим врстама графова.

#### **4.1 Егзактна метода за решавање проблема римске доминације**

Као што је раније наведено, до сада је развијено свега неколико алгоритама за решавање проблема римске доминације, док су за проблем слабе римске доминације позната два, а за проблем ограничене римске доминације ни један. Заправо, ако се изузме метахеуристика заснована на методи променљивих околина која решава проблеме римске и слабе римске доминације представљена у [72], за проблем римске доминације развијени алгоритми су углавном теоријског карактера и без икаквих експерименталних резултата. Од два алгоритма који решавају проблем слабе римске доминације, један је управо поменути хеуристика а други је конструисан тако да проблем слабе римске доминације решава оптимално. Међутим, други поменути алгоритам ипак не решава проблем

слабе римске доминације оптимално будући да не покрива све случајеве. Стога је тај алгоритам модификован, а резултати добијени модификованим алгоритмом представљени на Valcoг конференцији у Београду 2018. године, [81]. У наставку је изложен модификован алгоритам уз доказ да тако модификовани алгоритам заиста решава проблем слабе римске доминације оптимално.

#### 4.1.1 Алгоритам линеарне комплексности за решавање проблема слабе римске доминације на блок-графовима

##### Уводни појмови и нотације

Нека је  $G = (V, E)$  неусмерени граф са скупом чворова  $V$  и скупом грана  $E$ . Следећи појмови су делимично или у потпуности преузети из [87].

- Граф  $U$  назива се *подграфом* графа  $G$  уколико је скуп његових чворова подскуп скупа  $V$  и ако се све његове гране могу наћи и у  $E$ . Подграф  $H = (V', E')$  графа  $G$  назива се *индукованим подграфом* графа  $G$  ако важи  $V' \subseteq V$  и  $E' = E \cap V'^2$ .
- *Клика* је подграф неоријентисаног графа код кога су свака два различита чвора међусобно суседна, тј. клика је индуковани подграф који је комплетан.
- Граф  $G$  је *повезан* уколико за свака два чвора тог графа постоји пут који их повезује. Граф  $G$  је *неповезан* уколико постоје два чвора у  $G$  која нису крајеви ни једног пута у  $G$ .
- Максимални повезани подскуп  $H = (V_H, E_H)$  графа  $G$  назива се *компонентом повезаности* графа  $G$  ако:
  - 1) за свака два чвора  $v_1, v_2 \in V_H$  грана  $(v_1, v_2) \in E_H$  ако  $(v_1, v_2) \in E$  и
  - 2) не постоји пут чији су крајеви у  $V_H$  и у  $V \setminus V_H$ .
- Чвор графа назива се *везивним чвором* или *артикулационим чвором* (енг. articulation point или cut-vertex) уколико се његовим уклањањем (заједно са гранама које су њему инцидентне) из графа број компоненти повезаности тог графа повећава.

- Повезани граф  $G$  назива се *двоструко-повезаним графом* (енг. *biconnected graph*) ако уклањање било ког његовог чвора (заједно са гранама које су инцидентне са њим) не нарушава његову повезаност.
- *Двоструко-повезана компонента* (познатија као *блок*) представља максимални двоструко повезани подграф.
- *Бипартитни граф* (или *биграф*)  $G = (V, E)$  представља граф чији се чворови могу поделити у два скупа чворова  $U$  и  $W$ ,  $V = U \cup W$  и  $U \cap W = \emptyset$ , таква да свака грана тог графа повезује један чвор из  $U$  са једним чвором из  $W$ .
- *Дрво*  $T$  је повезан граф који нема циклe.
- Неусмерени граф  $G$  сачињен од фамилије скупова чворова  $S_i$ ,  $i = 0, 1, 2, \dots$ , назива се *пресечним графом* (енг. *intersection graph*) уколико се сваки скуп чворова  $S_i$  може представити неким одабраним чвором  $v_i$  и ако су два одабрана чвора  $v_i$  и  $v_j$  повезана граном када год одговарајуће фамилије скупова имају непразан пресек, тј.  $E(G) = \{\{v_i, v_j\} | S_i \cap S_j \neq \emptyset\}$ .

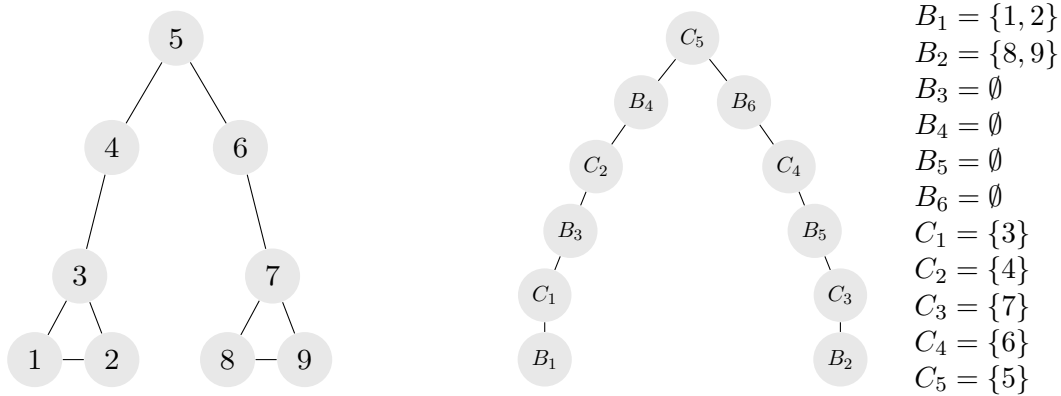
Дефиниције које се односе на коренско стабло, а које су преузете из [80], дате су у наставку.

- Растојање између два чвора  $u$  и  $v$ , означено са  $d(u, v)$ , одређује се као најмањи број грана потребних да се прошета од чвора  $u$  до чвора  $v$ ; растојање између два суседна чвора је 1.
- Дрво  $T$  назива се *коренским стаблом* уколико има један издвојен чвор  $r$  који се назива *кореном* стабла, чворове степена 1 (који се називају *листовима*) и остале чворове (који се називају *унутрашњим чворовима*).
- За два суседна чвора  $u$  и  $v$  коренског стабла  $T$ , чвор  $u$  назива се *родитељем* од  $v$  ако се налази на путу од  $r$  до  $v$  и ако је  $d(r, u) < d(r, v)$  у односу на задато  $r$ . Чвор  $v$  назива се *дететом чвора*  $u$ . Дете чвора  $v$  назива се *унуком* чвора  $u$ .
- Сваки чвор (осим коренског) може имати само једног родитеља и више деце.

Сада се *блок граф* (енг. *block graph*) може представити као неусмерени граф у коме је свака двоструко повезана компонента клика. Блок граф

који је настао од графа  $G$ , најчешће у ознаци  $B(G)$ , може се дефинисати на следећи начин: за свака два блока из  $G$  њима одговарајући чворови из  $B(G)$  су суседни у  $B(G)$  уколико одговарајући блокови имају заједнички везивни чвор у  $G$ . За више информација о блок-графовима видети [88].

За повезани блок граф  $G$ , стабло са везивним чворовима блок графа (мост-стабло скраћено) графа  $G$ , означава се као  $bc(G)$ , и дефинише на следећи начин: свака компонента повезаности графа  $G$  представљена је једним чвором мост-стабла  $bc(G)$ , сваки везивни чвор графа  $G$  такође је представљен једним чвором мост-стабла  $bc(G)$ ; када год су компоненте повезаности графа  $G$  директно повезане граном са неким везивним чвором њима одговарајући чворови мост-стабла  $bc(G)$  такође су повезани граном; за свака два суседна везивна чвора која имају непразан пресек, мост-стаблу се додаје још по један чвор (који се назива празним блоком) тако да, уместо да се чворови мост-стабла који представљају те везивне чворове графа директно повежу граном, оба чвора мост-стабла директно се повезују са тим празним блоком. Блок граф и њему одговарајуће мост-стабло илустровани су на Слици 14.



Слика 14: Граф  $G$  (лево) и њему одговарајуће стабло  $T_G$  (десно)

На слици 14 (десно), листови  $B_1$  и  $B_2$  представљају двокомпонентне блокове, чворови  $C_j$ ,  $j = 1, \dots, 5$  представљају једнокомпонентне блокове (тј.  $C_j$  представља блок који садржи само по један везивни чвор из графа  $G$ ), док унутрашњи чворови  $B_i$ ,  $i = 3, \dots, 6$ , представљају празне блокове (будући да су одговарајући везивни чворови графа  $G$  суседни и имају непразни пресек, пресек одговарајућих  $C_j$  чворова стабла  $T_G$  ће бити празни скупови, тј. празни блокови).

## Алгоритам за решавање проблема слабе римске доминације на блок-графовима из литературе

У [80] Liu и сарадници су, уз претпоставку да само везивни чворови могу бити позитивне тежине, развили алгоритам који као улазни аргумент користи мост-стабло  $T_G = (V', E')$  одговарајућег блок графа  $G = (V, E)$ . Они су функцију  $f : V' \rightarrow \{0^-, 0^+, 1, 2\}$  дефинисали у нешто измењеном облику у односу на стандардни запис функције слабе римске доминације. Наиме, уместо тежине 0 (вредност функције  $f$  у том чвору је једнака нули), уведене су тежине  $0^-$  и  $0^+$ . Тежину  $0^-$  добија чвор који је доминиран (нема суседа са позитивном тежином), док тежину  $0^+$  чвор који је био доминиран до тог момента (једини сусед са тежином 1 је ту тежину разменио са другим својим суседом тежине нула). На почетку алгоритма сваком листу додељена је тежина  $0^-$ , док се тежине унутрашњих чворова одређују пратећи следећи поступак:

Нека је  $u$  чвор коме треба да се одреди тежина.

Ако је  $u$  везивни чвор,

- i.  $f(u) = 2$  уколико скуп чворова сачињен од деце или унучад од  $u$  садржи више од два незаштићена чвора, односно садржи више од два чвора таква да су тежине тих чворова и тежине њихове деце једнаке нули.
- ii.  $f(u) = 1$  уколико у скупу чворова сачињеног од деце и унучади од  $u$  постоји само један чвор који је незаштићен.
- iii.  $f(u) = 0^-$  уколико су сва деца и унучад од  $u$  или тежине  $0^+$  или тежине 1.
- iv.  $f(u) = 0^+$  уколико у скупу чворова који се састоји из деце и унучади од  $u$  постоји чвор  $v$  код кога је  $f(v) = 2$ .

иначе

- i.  $f(u) = 0^+$  уколико чвор  $u$  представља празан блок.
- ii.  $f(u) = 0^+$  уколико једно дете од  $u$  има тежину 2.
- iii.  $f(u) = 0^-$  иначе.

Иако су Liu и сар. у [80] тврдили да њихов алгоритам решава проблем слабе римске доминације оптимално, функција  $f$  конструисана описаним поступком није увек  $\gamma_r$ -функција. Прво, према описаном алгоритму није прецизно дефинисана ситуација у којој везивни чвор  $u$  има тачно два

незаштићена чвора у скупу чворова који му представљају децу и унучад. Ако претпоставимо да случај i. подразумева ситуацију у којој чвор  $u$  има најмање два незаштићена чвора у скупу чворова који му представљају децу и унучад, тада би блокови  $C_1$  и  $C_3$  били тежине 2, чвор  $C_5$  тежине 1, док би сви остали чворови били тежине 0, чиме се добија да је минимална тежина мост-стабла  $T_G$  приказаног на Слици 14 (десно) 5. Слично, ако претпоставимо да случај ii. подразумева двоје незаштићене деце, онда је минимална тежина мост-стабла  $T_G$  једнака 4: чвор  $C_5$  је тежине 2, чворови  $C_1$  и  $C_3$  су тежине 1, док су сви остали чворови мост-стабла тежине 0. Међутим, оптимално решење овог графа је 3: тежина 1 треба да се додели чворовима  $C_1$ ,  $C_3$  и  $C_5$ , док су сви остали чворови тежине 0. Заиста, ако предложено оптимално решење применимо на одговарајући блок-граф, односно ако чворовима који су означени бројевима 3, 5 и 7, доделимо тежину 1, а свим осталим чворовима доделимо тежину 0, тада ће сви чворови бити заштићени од напада као чворови који имају стационирану легију, односно као чворови који имају суседа који има стационирану легију (суседа који може да им пошаље легију у случају да су нападнути). У складу са описаним распоредом легија, у случају напада, чворове 1 и 2 брани легија која је стационирана у 3; чворове 4 и 6 легија стационирана у 5; чворове 8 и 9 легија стационирана у 7, док су чворови 3, 5 и 7 заштићени јер имају стационирану легију чије померање неће нарушити њихову сигурност.

### **Модификован алгоритам за решавање проблема слабе римске доминације на блок-графовима линеарне комплексности**

Имајући у виду горе наведено, претпоставићемо да је  $G$  повезани блок-граф за који се, претрагом у дубину за  $O(|V| + |E|)$  времена, може формирати одговарајуће мост-стабло  $T_G = (V_{T_G}, E_{T_G})$ . Алгоритам користи формирано мост-стабло  $T_G$  као улазну инстанцу. За формирано мост-стабло  $T_G$  алгоритам на излазу враћа  $\gamma_r$ -функцију, тј. излаз је функција  $F$ ,  $F : V_{T_G} \rightarrow \{0, 1, 2\}$  која сваком чвору  $i \in V_{T_G}$  мост-стабла  $T_G$  додељује вредност (тежину)  $F(i)$ ,  $i = 1, \dots, |V_{T_G}|$ . Чворовима мост-стабла тежина се додељује постфиксним распоредом (комплексност постфиксног одређивања је  $O(|V_{T_G}|)$ ) на следећи начин: тежина се прво додељује листовима, потом њиховим родитељима и на крају корену стабла, с тим да се тежина додаје родитељу тек након што је додељена свој његовој деци. Доби-

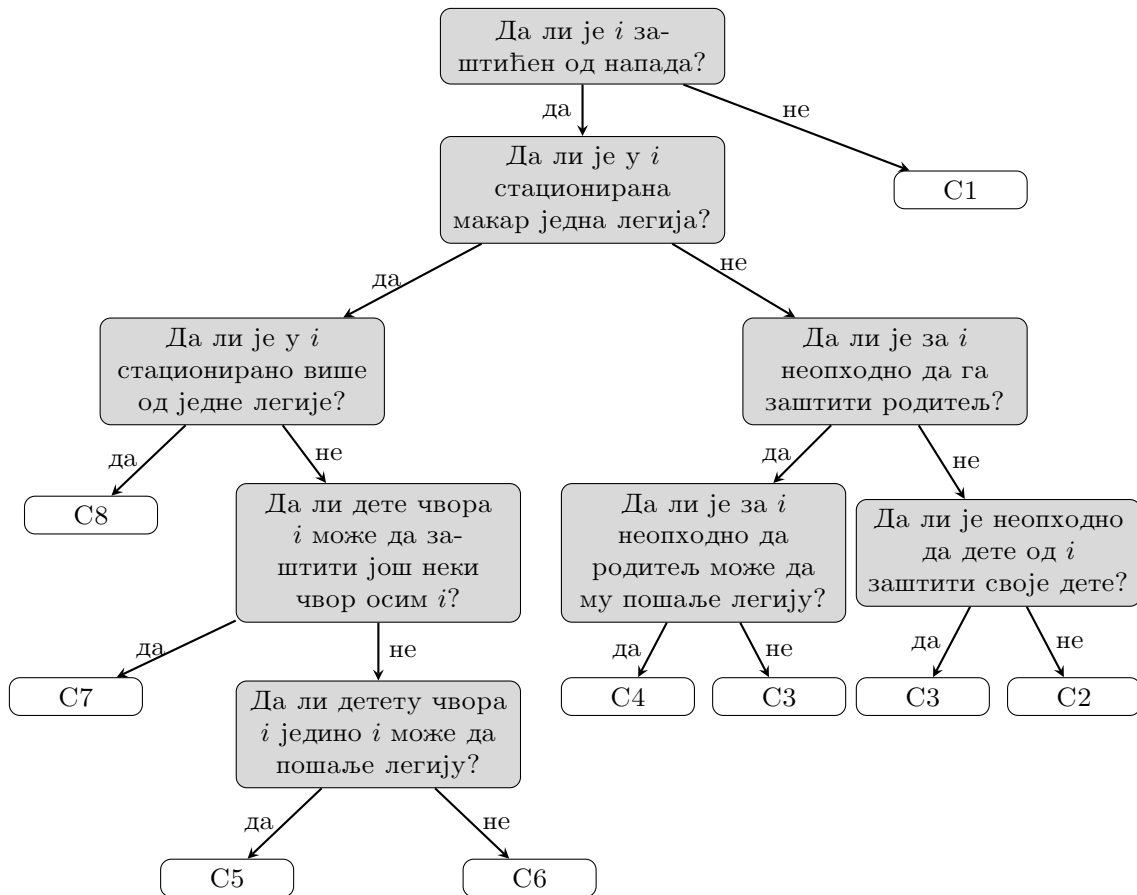
јено решење лако се може применити на граф  $G$  тако што се везивним чворовима графа  $G$  додељује иста тежина коју су имали у мост-стаблу а свим осталим чворовима графа  $G$  додељује тежина 0.

Будући да је  $WRD$  динамички проблем и да се односи на распоред легија по провинцијама Римског царства пре и након напада, поново ћемо претпоставити да су провинције Римског царства представљене чворовима графа (који има структуру блок-графа), да су чворови међусобно повезани ако је постојао пут (водени или копнени) између њима суседних провинција и да је број легија стационираних у провинцијама представљен тежинама њима одговарајућих чворова. Сигурност чворова пре и након напада дефинишемо на следећи начин: за чвор  $i$  кажемо да је тренутно заштићен (ако има позитивну тежину или суседа  $j$  чија је тежина позитивна) и да може постати незаштићен ако је његова тежина једнака нули и једини сусед  $j$ , чија је тежина једнака 1, мора да пошаље своју легију у одбрану суседа  $k$  ( $i, k$  су суседи са  $j$ ,  $k \neq i$ ), чија је тежина такође једнака нули. Дакле, означавамо чвор  $i$  као "заштићен од стране стациониране легије", "заштићен од стране суседове легије", или као "незаштићен". Будући да чвор позитивне тежине може да пошаље своју легију ка свом родитељу или детету, статус сваког чвора тежине нула директно зависи од статуса чвора који му представља родитеља и статуса чворова који му представљају децу. Другим речима, статус чвора  $i$  може се одредити на основу дијаграма приказаног као Слика 15.

Као што се може видети из дијаграма, за сваки чвор разликујемо један од следећих 8 статуса:

- C1: чвор  $i$  и сва његова деца (уколико  $i$  има деце), немају стационирану легију, због чега родитељ чвора  $i$  мора имати макар једну стационирану легију коју може да му пошаље, а коју неће слати ни једном другом свом детету, осим у случају у коме родитељ има две стациониране легије;
- C2: чвор  $i$  нема стационирану легију, али дете од  $i$ , чвор  $j$ , има стационирану легију коју може послати свом родитељу, а коју није дужан да шаље свом детету  $k$  у случају да је оно нападнуто;
- C3: чвор  $i$  нема стационирану легију, али дете од  $i$ , чвор  $j$ , има стационирану легију коју може послати свом родитељу, с тим да је дужан да ту легију пошаље и свом детету, чвору  $k$ , у случају да





Слика 15: Дијаграм могућих статуса чворова

је оно нападнуто, због чега је неопходно да и родитељ чвора  $i$  има стационирану легију како би штитио чвор  $i$  док  $j$  брани  $k$ ;

- C4: чвор  $i$  нема стационирану легију, али је зато имају његов родитељ, чвор  $m$ , и његово дете, чвор  $j$ , с тим да  $m$  може да му пошаље легију у случају напада, док  $j$  поред чвора  $i$  штити и своје дете, чвор  $k$ , коме је дужан да пошаље легију у случају да је нападнуто;
- C5: чвор  $i$  има стационирану легију коју може да пошаље свом родитељу у случају да је нападнут, нема незаштићене деце и није дужан да шаље своју легију нити једном детету у случају да је неко од њих нападнуто;
- C6: чвор  $i$  има стационирану легију коју може да пошаље свом родитељу у случају напада и има незаштићено дете, али није обавезан да шаље легију свом детету у случају да је оно нападнуто;
- C7: чвор  $i$  има стационирану легију којом штити себе и незаштићено

дете, чвор  $j$ , коме је дужан да пошаље легију у случају напада. Истовремено чвор  $i$  штити и родитеља али му не може послати своју легију;

C8: чвор  $i$  има две стациониране легије од којих једну може послати незаштићеном детету или незаштићеном родитељу у случају да је неко од њих нападнут.

Тежине чворова мост-стабла  $T_G$  додељују се постфиксним редоследом пратећи следећи поступак. Нека  $i_1$  представља први посматрани чвор, а  $i_n$ ,  $n = |V_{T_G}|$ , последњи коме ће се доделити тежина. Тежина чвора  $i_m$ ,  $m = 1, \dots, n$ , одређује се на следећи начин:

- Ако  $i_m$  представља лист, тада је његов статус C1, важи  $F(i_m) = 0$ .
- Ако  $i_m$  представља празан блок, тада је његов статус C2, важи  $F(i_m) = 0$ .
- Нека је  $i_m$  непразан блок или везивни чвор. Статус непразног блока зависи од статуса његове деце. Статус везивног чвора такође зависи од статуса његове деце, с тим да у случајевима када је дете празан блок, онда и у зависности од статуса деце празног блока.

Нека међу чворовима који утичу на статус чвора  $i_m$ ,  $num_1$  представља број чворова означених са C1,  $num_2$  број чворова означених са C2, и тако редом ( $num_k$  представља број чворова означених са  $C_k$ ,  $k = 1, \dots, 8$ ). У зависности од вредности  $num_k$ ,  $k = 1, \dots, 8$ , статус чвора  $i_m$  одређује се на следећи начин:

- Непразном блоку  $i_m$  код кога важи  $num_2 + num_3 + num_4 = 0$  и  $num_5 + num_6 + num_7 + num_8 > 0$ , додељује се следећи статус:

1. C3 ако су  $num_7 > 0$  и  $num_5 + num_6 + num_8 = 0$ ,
2. C2 ако је  $num_5 + num_6 + num_8 > 0$ ,

иначе чвор  $i_m$  добија статус C1. У сваком случају важи  $F(i_m) = 0$ .

- Уколико је  $i_m$  везивни чвор и

1. важи један од следећих услова:

- 1)  $num_1 > 0$  и  $num_4 > 0$ ,
- 2)  $num_1 \geq 2$ ,

$i_m$  добија статус С8, важи  $F(i_m) = 2$ .

2. важи један од наредних услова

1)  $num_1 = 1$  и  $num_4 = 0$ ,  $i_m$  добија статус С7.

2)  $num_1 = 0$  и  $num_4 > 0$ ,  $i_m$  добија статус С6.

3)  $num_1 = 0$ ,  $num_4 = 0$  и  $num_3 > 0$ ,  $i_m$  добија статус С5.

4)  $num_1 + num_3 + num_4 = 0$  заједно са једним од наредна два услова:

i)  $num_5 + num_8 > 0$ ,

ii)  $num_6 + num_7 > 1$  и  $num_6 > 0$ ,

$i_m$  добија статус С2.

5)  $num_1 + num_3 + num_4 + num_5 + num_6 + num_8 = 0$  и  $num_7 > 0$ ,  
 $i_m$  добија статус С4.

6)  $num_1 + num_3 + num_4 + num_5 + num_7 + num_8 = 0$  и  $num_6 = 1$ ,  
 $i_m$  добија статус С3.

7)  $num_1 + num_3 + num_4 + num_5 + num_6 + num_7 + num_8 = 0$ ,  $i_m$   
добија статус С1.

У случајевима 2.1)-2.3)  $F(i_m) = 1$ , док је за 2.4)-2.7)  $F(i_m) = 0$ .

Будући да за сваки чвор  $i_m$  мост-стабла  $T_G$  важи  $F(i_m) \in \{0, 1, 2\}$  и да су тежине одређене тако да сваки чвор има позитивну тежину, или има суседа позитивне тежине, и да у случају да је чвор који има тежину 0 нападнут, постоји сусед који са њим може да прераспореде тежину (тежина суседа се може умањити за један а тежина нападнутог чвора увећати за један) не нарушавајући безбедност било ког чвора графа, конструисана функција  $F$  представља  $f_{WRD}$  функцију за то мост-стабло  $T_G$ . Показаћемо да је функција  $F$  уједно и  $f_{WRD}$  функција минималне тежине.

Претпоставимо супротно, нека функција  $F'$  представља  $f_{WRD}$  функцију дефинисану за мост-стабло  $T_G$  која има мању укупну тежину од функције  $F$ . Довољно је да се покаже да је укупна тежина функције  $F'$  мања од укупне тежине функције  $F$  за један. Стога, нека је  $F'(i) = F(i) - 1$  за неки чвор  $i$  и нека је  $F'(j) = F(j)$  за све остале чворове  $T_G$ . Пошто је  $F'(i) \in \{0, 1, 2\}$ , могућа су само два случаја за чвор  $i$ : 1)  $F(i) = 2$ , и 2)  $F(i) = 1$ .

Размотримо прво случај 1). На основу конструкције функције  $F$ ,  $F(i) = 2$  уколико чвор  $i$  има најмање двоје незаштићене деце којима ће

послати једну своју стационирану легију у случају да их неко напада. На основу претпоставке,  $F'(i) = F(i) - 1 = 1$  и  $F'(j) = F(j)$  за све чворове мост-стабла  $T_G$ . Сада, ако је дете од  $i$ , коме  $i$  мора да пошаље легију у случају напада, нападнуто, тада  $i$  мора да му пошаље легију како би га одбранио. На овај начин  $i$  остаје без и једне стациониране легије (имаће тежину 0) и више неће моћи да штити осталу своју децу, тј. деца коју је штитила легија из  $i$  постају незаштићена. Лако се може закључити да овако конструисана функција  $F'$  не може да буде  $f_{WRD}$  функција за мост-стабло  $T_G$ . Дакле, овај случај није могућ.

Размотримо сада случај 2). На основу конструкције функције  $F$ , следи да је  $F(i) = 1$  када

i)  $i$  има само једно незаштићено дете коме је он једини сусед позитивне тежине а коме ће послати своју легију у случају напада, док су остала деца чвора  $i$  заштићена (а ако је неко дете празан блок, онда су и деца тог празног блока такође заштићена) и чвор  $i$  не мора да шаље своју легију нити једном од њих.

ii)  $i$  има само једно дете (или унуче ако је дете празан блок) коме ће слати легију у случају да је нападнуто, и нема незаштићене деце нити унучади (ако је дете празан блок),

iii)  $i$  нема незаштићене деце (нити унучад уколико су деца празни блокови), али има најмање једно дете (унуче ако је дете празан блок) коме је неопходно да он, као родитељ, буде позитивне тежине.

Сада је, на основу претпоставке,  $F'(i) = F(i) - 1 = 0$  и  $F'(j) = F(j)$  за све остале чворове. У случају i) неопходно је да чвор  $i$  има стационирану легију коју ће послати свом детету уколико оно буде нападнуто јер је то једина легија у његовом окружењу. Будући да је  $F'(i) = 0$ , следи да овако конструисана функција  $F'$  не може да буде  $f_{WRD}$  функција за мост-стабло  $T_G$  зато што би у том случају дете чвора  $i$  било незаштићено. У случају ii) неопходно је да чвор  $i$  може да пошаље легију свом детету, а како је  $F'(i) = 0$  поново се може закључити да  $F'$  није  $f_{WRD}$  функција за мост-стабло  $T_G$ . Најзад, у случају iii) тежина чвора  $i$  је једнака нули, што је у супротности са условом да дете чвора  $i$  мора имати родитеља позитивне тежине, па функција  $F'$  ни у овом случају не може бити  $f_{WRD}$  функција за мост-стабло  $T_G$ .

Дакле, како се тежина ни једног чвора (позитивне тежине) не може

умањити за један, следи да се ни укупна тежина свих чворова не може умањити за један. Стога је конструисана функција  $F$  заиста функција минималне тежине, што значи да се  $\gamma_r$  за дати граф  $T_G$  може рачунати као  $\sum_{i \in V_{T_G}} F(i)$ .

Сада, са обзиром да је тежина свих чворова мост-стабла  $T_G$  позната, добијено решење може се применити на почетни блок граф на следећи начин: за сваки везивни чвор стабла  $T_G$ , одговарајући везивни чвор блок графа биће исте тежине. Сви остали чворови су тежине нула. Стога се  $\gamma_r$  одговарајућег блок графа такође може одредити као  $\sum_{i \in V_{T_G}} F(i)$ .

Представљени алгоритам захтева линеарно време: трансформација блок графа у мост-стабло захтева  $O(|V| + |E|)$ , сортирање чворова мост-стабла у складу са постфиксним индексом захтева  $O(|V_{T_G}|)$ , док алгоритам пролази кроз сваки чвор мост-стабла  $T_G$  једном, тј. захтева  $O(|V_{T_G}|)$ .

## Рачунски резултати

Пошто су у [80] представљени само теоријски резултати без изворног кода, односно фајла који се може користити за тестирање, за потребе поређења алгоритам из [80] и његова модификација испрограмирани су у MATLAB2016b програму и тестирани на Intel(R) Core(TM) i7-4700MQ CPU @ 2.40GHz рачунару са 8GB RAM, под оперативним системом Windows 10. Тестирања су извршена на скупу инстанци које се могу преузети са <http://pallini.di.uniroma1.it/Graphs.html>. Резултати тестирања представљени су у Табели 11 на следећи начин. У првих пет колона дати су подаци везани за сваку инстанцу: име инстанце, број чворова, број грана, као и оптимално решење добијено коришћењем оптимизационог решавача CPLEX и целобројне формулације **WRD2** заједно са временом решавања  $t$ . У наредне две колоне приказани су вредност функције циља ( $val$ ) и време рада ( $t$ ) алгоритма представљеног у [80], док су у последње две колоне ( $val$  и  $t$ ) приказани резултати добијени коришћењем модификованог алгоритма. Подвучене вредности колоне  $val$  (потколоне Алг. из [80]) означавају да су те вредности веће од оптималних за ту инстанцу.

Као што се може видети из Табеле 11, модификовани алгоритам одређује решење проблема које је једнако оптималном (из колоне  $opt$ ) за сваку тестирану инстанцу, док алгоритам који је предложен у [80] одређује

Табела 11: Поређење егзактних алгоритама за решавање проблема слабе римске доминације на блок-графовима

Инстанце				Алг. из [80]		Модиф. алг.		
Име	$ V $	$ E $	$opt$	$t$	$val$	$t$	$val$	$t$
рантрее-10	10	9	5	0.0310	5	0.1037	5	<b>0.0103</b>
рантрее-20	20	19	10	0.0460	10	0.1511	10	<b>0.0041</b>
рантрее-50	50	49	25	0.1560	25	0.3022	25	<b>0.0062</b>
рантрее-100	100	99	48	0.2970	<u>49</u>	0.4254	48	<b>0.0084</b>
рантрее-200	200	199	103	0.4220	<u>105</u>	0.5338	103	<b>0.0132</b>
рантрее-300	300	299	145	0.6880	<u>153</u>	0.7563	145	<b>0.0184</b>
рантрее-400	400	399	201	0.9690	<u>208</u>	1.0028	201	<b>0.0255</b>
рантрее-500	500	499	249	1.2970	<u>255</u>	1.2237	249	<b>0.0296</b>
рантрее-600	600	599	299	1.3280	<u>313</u>	1.4610	299	<b>0.0381</b>
рантрее-700	700	699	346	1.6570	<u>360</u>	1.7541	346	<b>0.0401</b>
рантрее-800	800	799	391	2.2070	<u>403</u>	2.5813	391	<b>0.0469</b>
рантрее-900	900	899	452	2.1720	<u>463</u>	2.1692	452	<b>0.0541</b>
рантрее-1000	1000	999	490	3.3280	<u>510</u>	2.4234	490	<b>0.0590</b>

решење које је једнако оптималном (из колоне  $opt$ ) за само прве три инстанце. Уз то, време решавања модификованог алгорита је значајно мање у односу на време решавања алгорита из [80].

## 4.2 Недетерминистичка метода за решавање проблема римске доминације

Имајући у виду да се математички модел за решавање проблема римске доминације може брзо решити за инстанце различитих типова и димензија, а да се проблем слабе римске доминације изузетно споро решава чак и за инстанце мањих димензија, Ивановић и Урошевић су у [72] представили методу засновану на методи променљивих околина за решавање оба проблема. Представљена метода је, уједно, једина позната хеуристичка метода за решавање ова два проблема. Иако је рачунским резултатима показано да представљени алгоритама доста добро решава оба проблема, примећено је да се неке његове функције могу другачије имплементирати, односно профинити одређеним условима и да се на тај начин, потенцијално, могу добити још бољи резултати. С тим у вези, у оквиру овог поглавља представљен је још један алгоритама заснован на методи променљивих околина, због чега ће у наставку бити осврта само на ту метахеуристику. Пошто је имплементација ове методе заснована на појму допустивог решења и дефиницији његове околине, уведе се следеће дефиниције.

Означимо са  $S$  скуп решења неког проблема. Околина (енг. neighborhood) решења  $x$ , у ознаци  $\mathcal{N}(x)$  ( $\mathcal{N}(x) \subseteq S$ ), представља скуп решења придружених решењу  $x$  по неком правилу. Ако претпоставимо да се над решењем  $x$  могу извршити неке елементарне трансформације (које углавном зависе од оптимизационе методе и њене имплементације), тада се са  $\mathcal{N}(x)$  може означити скуп решења добијених применом тих трансформација над решењем  $x$ . Скуп свих решења добијених применом једне елементарне трансформације на решење  $x$  обележава се са  $\mathcal{N}_1(x)$ , скуп свих решења добијених применом две узастопне елементарне трансформације на решење  $x$  обележава се са  $\mathcal{N}_2(x)$  итд.

Будући да су локални минимуми обично субоптимална решења, идеја приближних метода састоји се у проналажењу што бољег локалног минимума за што краће време. Нека је, зато,  $S$  скуп свих допустивих решења и нека је  $x \in S$  произвољно или по неком правилу изабрано почетно решење проблема. Процес претраживања унапред дефинисане околине решења проблема назива се *локално претраживање* (енг. Local search, LS) и као такво представља основу многих метахеуристичких метода. Локално претраживање подразумева да се за свако  $x' \in \mathcal{N}(x)$  из околине неког почетног решења  $x$  израчуна вредност  $f(x')$  и, уколико је  $f(x') < f_{min}$  (на почетку је  $f_{min} = f(x)$ ), пронађено боље решење памти као најбоље решење,  $x_{min} = x'$  ( $f_{min} = f(x')$ ). Након што се сва решења околине  $\mathcal{N}(x)$  провере, претрага се наставља у околини  $\mathcal{N}(x_{min})$  и, најчешће, зауставља уколико околина  $\mathcal{N}(x_{min})$  не садржи боље решење. С обзиром да за проблеме великих димензија процес претраге читаве околине може бити јако дуг, један од основних захтева приликом конструкција метахеуристика представља пажљив одабир околине.

#### 4.2.1 Метода променљивих околина

Младеновић и Хансен предложили су методу променљивих околина (енг. Variable Neighborhood Search, VNS) 1997. год. у [89] док је идеја саме методе изложена 1995. године у [90]. Иако се основна идеја методе своди на систематској замени околина у којима се врши локална претрага, метода је заснована на три чињенице:

- 1) Локални минимум једне околине није обавезно локални минимум

неке друге околине.

- 2) Глобални минимум представља локални минимум у свим околинама које га садрже.
- 3) Локални минимума различитих околина могу бити релативно близу један другом.

Водећи се првом чињеницом, суштинска идеја VNS методе састоји се у промени околина како би се избегла могућност заглављивања у локалном минимуму. Друга чињеница заправо говори да ако неко решење није локални минимум своје околине онда сигурно није ни глобални минимум. Последња чињеница указује на то да се до глобалног минимума може доћи преко низа локалних минимума блиских једни другима. Метода променљивих околина показала се као врло успешна метода за решавање проблема комбинаторне и континуалне оптимизације и, у зависности од тога да ли се ове три чињенице користе детерминистички, стохастички или комбиновано, развијено је више различитих верзија VNS-а.

### Метода променљивог спуста

Метода променљивог спуста (енг. Variable Neighborhood Descent, VND) заснована је на чињеници да различите околине могу имати различите локалне минимуме.

Означимо са  $S$  скуп свих допустивих решења неког проблема и са  $\mathcal{N}_k$  ( $k = 1, \dots, k_{max}$ ) коначну колекцију унапред дефинисаних структура околина. VND метода се може описати на следећи начин. На почетку методе се на случајан начин или применом неке хеуристике из скупа решења  $S$  бира инцијално решење  $x$  и проглашава за тренутно најбоље решење  $x_{min} = x$ ,  $f_{min} = f(x)$ . Затим се врши локално претраживање околине  $\mathcal{N}_1(x_{min})$  тог решења како би се, као резултат претраживања, добио локални минимум  $x'$ ,  $x' \in \mathcal{N}_1(x_{min})$ . Уколико  $x'$  не представља побољшање тренутно најбољег решења, поступак локалне претраге се наставља у наредним околинама  $\mathcal{N}_k(x_{min})$ ,  $k = 2, 3, \dots, k_{max}$ . Сваки пут када се у некој околини  $\mathcal{N}_k(x_{min})$  нађе боље решење  $x'$ , тј. решење за које је  $f(x') < f_{min}$ , тренутно најбоље решење се ажурира,  $x_{min} = x'$  ( $f_{min} = f(x')$ ), а претрага се наставља у околини  $\mathcal{N}_1(x_{min})$ . Подразумева се да се, уколико се у



оквиру околине  $\mathcal{N}_1(x_{min})$  не налази побољшање тренутно најбољег решења, претрага наставља у наредној околини  $\mathcal{N}_2(x_{min})$  и тако редом. Поступак се завршава када се  $x_{min}$  не може поправити ни у једној околини, што значи да је  $x_{min}$  локални минимум у односу на све његове дефинисане околине.

Постоје два приступа локалног претраживања околине решења. Са првим приступом се у свакој итерацији тражи најбоље решење (енг. Best improvement strategy, BI) у које се прелази тек након што је читава околина претражена. Други приступ представља решење прве поправке (енг. First improvement strategy, FI) којом се прелази у прво боље пронађено решење у оквиру тренутне околине, након чега се претрага наставља у околини  $\mathcal{N}_1$  новог решења.

### **Редукована метода променљивих околина**

Редукована метода променљивих околина (енг. Reduced Variable Neighborhood Search, RVNS) заснива се на систематској промени околина и избору једног случајног решења у свакој од тих околина. Наиме, као одговор на питање како да се "побегне" из достигнутог локалног минимума и његове околине у циљу налажења неког другог минимума, метода је дефинисана тако да се на случајан начин бира решење из тренутно обрађиване околине. Дакле, прво се изабере скуп околина  $\mathcal{N}_k$ ,  $k = 1, \dots, k_{max}$  и почетно решење  $x$  (које се аутоматски проглашава за најбоље  $x_{min} = x$ ,  $f_{min} = f(x)$ ). Затим се, на случајан начин, бира тачка  $x'$ ,  $x' \in \mathcal{N}_1(x_{min})$ . Уколико за изабрану тачку важи  $f(x') \geq f_{min}$  претрага се наставља у наредној околини. Корак у ком се на случајан начин бира тачка  $x' \in \mathcal{N}_k(x_{min})$  ( $k = 1, 2, \dots, k_{max}$ ) назива се *размрдавање* (енг. shaking). Након што се испитају све околине без успешног побољшања, поступак претраге се циклично наставља почевши од прве околине, све док неки од критеријума заустављања не буде испуњен. Критеријуми заустављања дефинишу се зависно од конкретне примене методе а неки од најчешћих критеријума заустављања су максимално време извршавања и максимални број итерација. За разлику од VND методе, може се приметити да код RVNS методе не постоји процес локалне претраге чиме се избегава сложено и дуготрајно извршавање у оквиру једне VNS итерације. Постоје модификације RVNS методе у којима се у сваком кораку размрдавања уместо

једне бира више случајних тачака. Овакве модификације могу дати бржу конвергенцију ка бољем решењу код неких класа проблема. RVNS метода се најчешће користи за добијање почетног решења за неку другу методу.

### Основна метода променљивих околина

Комбинацијом два описана принципа настала је *основна метода променљивих околина* (енг. (Basic) Variable Neighborhood Search, (B)VNS). (B)VNS метода конструисана је тако да у оквиру једне околине, на случајан начин, бира тачку те околине а затим врши локалну претрагу околине те тачке. Стога се, за унапред дефинисан скуп околина  $\mathcal{N}_k(x)$  ( $k = 1, \dots, k_{max}$ ) решења  $x$ , (B)VNS метода може дефинисати на следећи начин. Најпре се из скупа допустивих решења бира почетно решење  $x$  (које се проглашава за тренутно најбоље решење  $x^*$ ), затим се, у оквиру петље која мења индекс  $k$  ( $k \in \{1, \dots, k_{max}\}$ ), који се односи на тренутну околину  $\mathcal{N}_k(x)$ , размрдавањем бира неко решење  $x' \in \mathcal{N}_k(x)$  над којим се, даље, процедуром локалног претраживања тражи потенцијално боље решење  $x''$ . Добијено решење  $x''$  пореди се са тренутно најбољим решењем  $x^*$  и уколико је боље, најбоље решење се ажурира, индекс  $k$  поставља на 1 и процес претраге наставља у околини  $\mathcal{N}_1$  текућег најбољег решења. У супротном се, заједно са размрдавањем, процес претраге наставља у широј околини  $\mathcal{N}_{k+1}(x^*)$ . Када индекс  $k$  достигне свој максимум  $k_{max}$ , процес претраге се наставља почевши од околине  $\mathcal{N}_1(x)$  текућег решења  $x$  и прекида тек када се задовољи неки критеријум заустављања. У току процеса локалне претраге, (B)VNS метода увек прелази у боље решење и то у прво на које наиђе (FI). Околине се најчешће бирају тако да важи  $|\mathcal{N}_1(x)| < |\mathcal{N}_2(x)| < \dots < |\mathcal{N}_{k_{max}}(x)|$ <sup>9</sup>. У неким модификацијама (B)VNS методе индекс  $k$  узима вредности од  $k_{min}$  до  $k_{max}$  са кораком  $k_{step}$ . Корак  $k_{step}$  може имати негативну вредност имплицирајући да се може кретати од највеће ка најмањој околини решења. Алгоритам (B)VNS методе илустрован је псеудокодом Алгоритам 1.

Ако се, уместо процедуре локалног претраживања у оквиру (B)VNS методе искористи VND процедура, добија се *општа метода променљивих околина* (енг. General Variable Neighborhood Search, GVNS). Основни параметар (B)VNS методе је  $k_{max}$  који представља максимални број околина,

<sup>9</sup>Израз  $|A|$  означава кардиналност скупа  $A$ .

---

**Алгоритам 1** Метода променљивих околина, (B)VNS

---

```
1:  $x^* \leftarrow pocetnoResenje()$ ;  
2:  $f_{min} \leftarrow f(x^*)$ ;  
3: while (kriterijumi zaustavljanja nisu ispunjeni)  
4:    $k \leftarrow k_{min}$ ;  
5:   while ( $k \leq k_{max}$ )  
6:      $x \leftarrow x^*$ ;  
7:      $x' \leftarrow razmrdavanje(x, k)$ ;  
8:      $x'' \leftarrow lokalnoPretrazivanje(x')$ ;  
9:     if ( $f(x'') < f_{min}$ ) then  
10:        $f_{min} \leftarrow f(x'')$ ;  
11:        $x^* \leftarrow x''$ ;  
12:        $k \leftarrow k_{min}$ ;  
13:     else  
14:        $k \leftarrow k + k_{step}$ ;  
15:     end  
16:   end  
17: end
```

---

[89], [91]. У зависности од проблема који се решава VNS методом, критеријуми заустављања могу бити различити. На пример, максимално дозвољено време извршавања, максимални број итерација, тј. колико пута се достигне  $k_{max}$  или број итерација између две поправке локално најбољег решења. Поред поменуте модификације, постоје разне додатне модификације VNS методе и на њиховом развоју најчешће се ради како би се решили сложенији или проблеми великих димензија.

Неке од познатих модификација а које ће бити коришћене при имплементацији ове методе су:

- 1) прихватање решења истог квалитета за тренутно најбоље решење са неком задатом вероватноћом,
- 2) разматрање читаве околине пре него што се донесе одлука у које решење прећи, тј. LS метода са VI стратегијом претраживања,
- 3) уместо једног почетног решења у околини  $k$ , генерише се  $b$  случајних решења за неки унапред задат параметар  $b$ .

Поред наведених, неке од познатих модификација основних верзија VNS-а су: (а) Метода променљивих околина са декомпозицијом (енг. Variable Neighborhood Decomposition Search, VNDS), [92] (б) Укошена метода променљивих околина (енг. Skewed VNS, SVNS) и друге. За више детаља видети [91].

Пошто је оквиру тезе конструисана нова метода за решавање проблема римске и слабе римске доминације заснована на методи променљивих околина, у наставку ће се метода коју су Ивановић и Урошевић представили у [72] означавати као VNS-CAI. Поређења нове и VNS-CAI методе биће извршена након описа нове методе, а у оквиру дела у коме ће нова метода бити тестирана над истим скупом инстанци над којим је тестирана VNS-CAI метода, док ће се на сличности и разлике у кодирањима указати током и након описа нове методе.

#### 4.2.2 Нова метода за решавање проблема римске доминације заснована на методи променљивих околина

##### Простор решења и структура допустивих решења проблема

Претпоставимо да се Римско царство може представити графом  $G = (V, E)$  на следећи начин: сваки чвор графа представља једну провинцију Римског царства; између два чвора постоји грана уколико између њима одговарајућих провинција постоји копнена граница или директан морски пут. Нека се, на пример, Римско царство састоји из  $n$  провинција које се могу обележити бројевима од 1 до  $n$ . Тада се скуп  $V$  може записати као  $V = \{1, \dots, n\}$  а скуп грана као  $E = \{(i, j) | i, j \in V, \text{ постоји копнена граница или директан морски пут између провинција } i \text{ и } j\}$ . Граф  $G$  је повезан<sup>10</sup>, неоријентисан и без паралелних грана.

**Простор решења**  $\mathcal{RD}$  проблема представља скуп свих  $n$ -торки  $X = (x_1, \dots, x_n)$ , за које важи  $x_i \in \{0, 1, 2\}$ ,  $i = 1, \dots, n$ . Вредност  $i$ -те координате решења  $X$  ( $x_i$ ), представља број стационираних легија у  $i$ -тој провинцији, односно тежину чвора  $i$ . Вредности координата решења  $X$  могу се увећати или умањити за 1 (односно 2) уколико се таквим поступком поново добију вредности из скупа  $\{0, 1, 2\}$ . Дакле, разликујемо следеће **трансформације** вредности координата решења  $X$ :

- T1. Вредност  $x_i$  решења  $X$  може се умањити за један уколико је  $x_i > 0$ , тј. за  $x_i > 0$  решења  $X$  дозвољена је трансформација  $x_i \leftarrow x_i - 1$ .

<sup>10</sup>Граф  $G$  може бити неповезан с тим да се у таквом случају проблем римске доминације посматра за сваку компоненту повезаности тог графа посебно а коначни резултат добија сабирањем добијених резултата.

- T2. Над вредностима  $x_i$  и  $x_j$  решења  $X$ , код којих је  $x_i > 0$  и  $x_j < 2$ , дозвољене су трансформације  $x_i \leftarrow x_i - 1$  и  $x_j \leftarrow x_j + 1$ .
- T3. Над вредностима  $x_i$  и  $x_j$  решења  $X$ , код којих је  $x_i = 2$  и  $x_j = 0$ , дозвољене су трансформације  $x_i \leftarrow 0$  и  $x_j \leftarrow 2$ .
- T4. Над вредностима  $x_i$ ,  $x_j$  и  $x_s$  решења  $X$ , за које важи  $x_i = 2$ ,  $x_j < 2$  и  $x_s < 2$ , дозвољене су трансформације  $x_i \leftarrow 0$ ,  $x_j \leftarrow x_j + 1$  и  $x_s \leftarrow x_s + 1$ .
- T5. Вредност  $x_i$  решења  $X$  може се увећати за један уколико је  $x_i < 2$ , тј. за  $x_i < 2$  решења  $X$  дозвољена је трансформација  $x_i \leftarrow x_i + 1$ .
- T6. Вредност  $x_i$  решења  $X$  може се увећати за два уколико је  $x_i = 0$ , тј. за  $x_i = 0$  решења  $X$  дозвољена је трансформација  $x_i \leftarrow x_i + 2$ .

За два решења  $X$  и  $X'$  кажемо да имају *разлику реда 1* уколико се применом трансформације T1, T2 или T5 на решење  $X$  може добити решење  $X'$ . Два решења  $X$  и  $X'$  имају *разлику реда 2* уколико се двоструком применом трансформације T2 на координате  $x_i$  и  $x_j$  решења  $X$  за које важи  $x_i = 2$  и  $x_j = 0$ , добије решење  $X'$ ; ако се на те исте координате уместо двоструке примене трансформације T2 једанпут примени трансформација T3 или ако се на  $x_i$  једанпут примени трансформација T6. Такође, за два решења  $X$  и  $X'$  кажемо да су разлике реда 2 уколико се трансформација T2 прво примени на координате  $x_i$  и  $x_j$  ( $x_i = 2$  и  $x_j < 2$ ) а затим на  $x_i$  и  $x_s$  ( $x_s < 2$ ,  $s \neq j$ ) решења  $X$  како би се добило решење  $X'$ , односно ако се на координате  $x_i$ ,  $x_j$  и  $x_s$  ( $x_i = 2$ ,  $x_j < 2$  и  $x_s < 2$ ,  $s \neq j$ ) једанпут примени трансформација T4. Уколико се поступак којим се на решење  $X$  примене трансформације T1, T2 или T5 понови  $k$  пута кажемо да су трансформисано и почетно решење *разлике реда  $k$* . Слично, два решења,  $X$  и  $X'$ , могу да буду разлике реда  $k$  уколико се на решење  $X$  одређени број пута примени и нека од трансформација T3, T4 или T6, с тим да се, за разлику од примена трансформација T1, T2 и T5, код којих се са сваком применом ред разлике увећава за један, применом трансформација T3, T4 и T6 ред разлике увећава за два. На пример, решења  $X$  и  $X'$  су разлике реда пет уколико се решење  $X'$  може добити тако што се на решење  $X$  по једанпут примене трансформације T1, T3 и T4.

У даљем тексту израз "отпуст легије" подразумеваће трансформацију T1, израз "премештање једне легије" трансформацију T2, док ће

”пресељење две легије у исту провинцију” и ”пресељење две легије у две различите провинције” подразумевати трансформације Т3 и Т4 (тим редом). Слично, израз ”доделиће се нова легија” подразумеваће трансформацију Т5, док ће израз ”доделиће се две нове легије” подразумевати трансформацију Т6.

Скуп свих решења  $X'$  чија је разлика од решења  $X$  реда  $k$  ( $k = k_{min}, \dots, k_{max}$ ) назива се  $k$ -том *околином решења*  $X$  и обележава са  $\mathcal{N}_k(X)$ .

Решење  $X = (x_1, \dots, x_n)$  назива се **допустивим решењем**  $\mathcal{RD}$  проблема ако постоји функција  $f$ ,  $f : V \rightarrow \{0, 1, 2\}$  дефинисана са

$$f(i) = x_i, \quad i = 1, \dots, n, \quad (4.1)$$

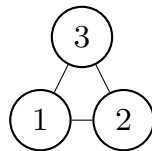
таква да је  $f$  функција римске доминације тј.  $X$  је допустиво решење  $\mathcal{RD}$  проблема уколико за свако  $x_i$  ( $i = 1, \dots, n$ ) важи један од услова:

- 1)  $x_i > 0$  или
- 2)  $x_i = 0$  и постоји најмање једно  $x_j = 2$  такво да је  $(i, j) \in E$ .

Тежина функције  $f$  рачуна се по формули  $F(X) = \sum_{i=1}^n x_i$ . Број  $F(X)$  назива се *вредност решења*  $X$  (*тежина решења*  $X$ ).

*Скуп допустивих решења*  $\mathcal{RD}$  проблема представљају све  $n$ -торке  $X = (x_1, \dots, x_n)$  код којих за свако  $i = 1, \dots, n$  важи  $x_i \in \{1, 2\}$  или је испуњен услов 2).

На пример, за граф  $\mathcal{G}_1 = (V_{\mathcal{G}_1}, E_{\mathcal{G}_1})$ , где је  $V_{\mathcal{G}_1} = \{1, 2, 3\}$  и  $E_{\mathcal{G}_1} = \{(1, 2), (1, 3), (2, 3)\}$  (Слика 16),



Слика 16: Граф  $\mathcal{G}_1$

скуп допустивих решења је следећег облика:  $X_{\mathcal{RD}} = \{(2, 2, 2), (2, 2, 1), (2, 1, 2), (1, 2, 2), (2, 1, 1), (1, 2, 1), (1, 1, 2), (2, 2, 0), (2, 0, 2), (0, 2, 2), (1, 1, 1), (2, 1, 0), (2, 0, 1), (1, 2, 0), (1, 0, 2), (0, 2, 1), (0, 1, 2), (2, 0, 0), (0, 2, 0), (0, 0, 2)\}$ .

## Имплементација нове методе за решавање проблема римске доминације

Служећи се истом идејом као код VNS-CAI методе, нова метода проблем римске доминације решава на следећи начин. Најпре се за задати граф (улазну инстанцу) формира решење  $X = (0, \dots, 0)$  (решење  $X$  је дужине  $n$ , где  $n$  представља број чворова посматраног графа). Потом се произвољно изабраним чворовима графа додељује једна по једна легија (вредности произвољно изабраних координата решења  $X$  увећавају се за један), с тим да се ни једном чвору не доделе више од две легије, све док решење  $X$  не постане допустиво решење посматраног проблема за тај граф. Након што се пронађе допустиво решење посматраног проблема за тај граф, такво решење се запамти као најбоље допустиво решење ( $X^*$ ). Надаље се, пошто проблем римске доминације подразумева да се пронађе допустиво решење најмање тежине, проверава да ли би отпуст неке од стационарираних легија могао да се изврши, а да након тог отпуста решење остане допустиво. Сваки пут када се отпусти нека легија а решење остане допустиво, такво решење се запамти као најбоље допустиво решење, односно вредност  $X^*$  се ажурира. Уколико отпуст легије не даје допустиво решење, легија се поново додељује чвору са ког је отпущена. Дакле, за један по један чвор графа прво се проверава да ли би отпуст стационариране легије (уколико чвор има стационарирану легију) дао допустиво решење, затим се врши отпуст те легије (уколико решење остаје допустиво након отпуста) и на крају се добијено решење памти као најбоље (ажурира се решење  $X^*$ ) а описани поступак наставља са наредним чвором графа. Најзад, након што је испитивање извршено за све чворове који имају стационарирану легију, поступак тражења допустивог решења још мање тежине се наставља на следећи начин. У оквиру *while* петље прво се допустиво решење најмање тежине ( $X^*$ ) запамти као текуће ( $X'$ ). Затим се  $k$  (на почетку је  $k = k_{min}$ ) произвољно изабраних легија решења  $X'$  премести са чворова на којима су стационариране на других  $k$  произвољно изабраних чворова (водећи рачуна да премештање легије задовољава трансформацију T2), а потом се, додатно, једна произвољно изабрана легија решења  $X'$  отпусти (водећи рачуна да је трансформација T1 задовољена). Поступак којим се  $k$  легија прераспореди, а затим још једна легија отпусти, назива се *размрдавањем* решења  $X'$ . Размрдано решење  $X'$  је, очигледно, мање тежине у односу на решење

$X^*$ . Стога се, надаље, претраживањем  $\mathcal{N}_1(X')$  околине текућег решења  $X'$  тражи ново ( $X''$ ) решење које ће бити допустиво а, ако се претраживањем  $\mathcal{N}_1(X')$  околине текућег решења  $X'$  не пронађе допустиво решење, онда се оно тражи у околини  $\mathcal{N}_2(X')$ . Поступак којим се тражи допустиво решење  $X''$  назива се поступком *локалног претраживања*. У случају да се поступцима размрдавања и локалног претраживања не пронађе допустиво решење, они се примењују испочетка, с тим да се, у оквиру поступка размрдавања, трансформација Т2 извршава  $k$  пута, при чему је број  $k$  претходно увећан за вредност  $k_{step}$ , док се поступак локалног претраживања и даље изводи над околинама  $\mathcal{N}_1(X')$  и  $\mathcal{N}_2(X')$  текућих решења. Аналогно, уколико размрдавање и локално претраживање поново не буду успешни у проналажењу допустивог решења, при наредној примени размрдавања се број извршавања трансформација Т2 поново увећава за  $k_{step}$  водећи рачуна да максимални број извршавања Т2 трансформација не прелази  $k_{max}$ . Сваки пут када  $k$  достигне своју максималну вредност  $k_{max}$ ,  $k$  се поставља на  $k_{min}$ , бројач  $brIter$  увећава за један (на почетку је  $brIter = 0$ ), а описани поступци *while* петље понављају испочетка. Иначе, уколико се поступком размрдавања и локалног претраживања пронађе решење  $X''$  које је допустиво, до тада најбоље допустиво решење се ажурира ( $X^* \leftarrow X''$ )<sup>11</sup>, вредност параметра  $k$  поставља на  $k_{min}$ , након чега се читав поступак размрдавања и локалног претраживања понавља испочетка, али над ажурираним решењем  $X^*$ . Увећавање вредности бројача  $brIter$ , односно поступци *while* петље се понављају све док бројач  $brIter$  не достигне своју максималну вредност ( $brIter_{max}$ ) или док време извршавања кода не буде једнако  $t_{max}$  када се *while* петља прекида.

Као што се може приметити, начин на који се решење  $X^*$  размрдава, односно избор околина које се користе приликом размрдавања решења није карактеристичан за класичан VNS стога се и сама метода не може назвати класичном VNS методом. Међутим, како је принцип претраживања околина и рада самог алгорита заснован на методи променљивих околина, описана метода се може сматрати једном његовом модификацијом.

Модификована VNS метода за решавање проблема римске доминације детаљно је илустрована Алгоритмом 2.

<sup>11</sup>Како се у размрдавању отпушта легија, тежина допустивог решења  $X''$  биће мања од тежине до тада најбољег допустивог решења  $X^*$ , тј.  $F(X'') < F^*$ .



---

**Алгоритам 2** Модификована VNS метода за решавање  $\mathcal{RD}$  проблема

---

```
1:  $k \leftarrow k_{min}$ ;
2:  $X^* \leftarrow pocetnoResenje()$ ;
3:  $F^* \leftarrow F(X^*)$ ;
4:  $brIter \leftarrow 0$ ;
5: while (nije ispunjen kriterijum zaustavljanja)
6:    $X' \leftarrow razmrdavanje(X^*, k)$ ;
7:    $X'' \leftarrow lokalnoPretrazivanje(X')$ ;
8:   if ( $X''$  je dopustivo rešenje ) then
9:      $X^* \leftarrow X''$  (у размрдавању је смањена тежина решења  $X''$ );
10:     $k \leftarrow k_{min}$ ;
11:   else
12:      $k \leftarrow k + k_{step}$ ;
13:   end
14:   if ( $k == k_{max}$ ) then
15:      $k \leftarrow k_{min}$ ;
16:      $brIter \leftarrow brIter + 1$ ;
17:   end
18: end
```

---

Метода користи помоћне функције  $brNedopustivih()$ ,  $ispitajDopustivost()$ ,  $smanjiBazu()$  и  $kvariBazu()$ , и главне функције  $pocetnoResenje()$ ,  $razmrdavanje()$  и  $lokalnoPretrazivanje()$ . Наведене функције описане су у наставку по редоследу по ком се користе.

Провера да ли је решење  $X$  допустиво решење  $\mathcal{RD}$  проблема врши се позивањем функције  $brNedopustivih(X)$ . Функција  $brNedopustivih(X)$  испитује да ли је улазни аргумент  $X$  допустиво решење  $\mathcal{RD}$  проблема за дату инстанцу, односно колико координата решења  $X$  не испуњава услове допустивости (услови дефинисани у поглављу 4.2.1.). Уколико је  $X$  допустиво решење датог проблема, функција  $brNedopustivih(X)$  враћа нулу, иначе враћа број координата решења  $X$  ( $nd$ ) који не испуњавају оба услова. Функција  $brNedopustivih(X)$  имплементирана је на следећи начин. На почетку се узима да је  $nd = 0$ . Потом се за једно по једно  $i$ ,  $i = 1, \dots, n$ , испитују вредности координата решења  $X$ . Уколико је  $x_i > 0$  задовољен је први услов допустивости решења, проверава се вредност  $x_i$  за наредно  $i$ . У случају да је  $x_i = 0$  проверава се да ли је испуњен други услов допустивости решења, након чега се испитивање вредности координата решења наставља за наредно  $i$ . Испитивање другог услова допустивости врши се на следећи начин. Узимајући да чвор  $i$  има  $n_i$  суседа и да је скуп чворова суседних чвору  $i$  скуп  $V_i = \{i_1, \dots, i_{n_i}\}$ , проверава се да ли за неко  $j \in V_i$  важи  $x_j = 2$ . Уколико је одговор потврдан, други услов је испуњен,  $nd$  остаје непромењен. Иначе, ако је за свако

$j \in V_i$   $x_j < 2$ , други услов није испуњен,  $nd$  се увећава за 1. Функција  $brNedopustivih(X)$  враћа вредност  $nd$  након што су испитане вредности свих координата решења  $X$ . Сложеност функције  $brNedopustivih(X)$  може се оценити са  $O(n^2)$ .

Узмимо граф  $\mathcal{G}_1$  за пример. Функција  $brNedopustivih(X)$  враћа нулу за сваку тројку  $X \in X_{\mathcal{RD}}$ . Заиста, почевши од  $nd = 0$ , за  $X = (2, 1, 0)$ ,  $brNedopustivih(X)$  проверава, редом, вредности  $x_1$ ,  $x_2$  и  $x_3$ . Будући да је  $x_1 > 0$  број  $nd$  остаје непромењен. Прелази се на проверу вредности  $x_2$ . Како је  $x_2 > 0$  број  $nd$  поново остаје непромењен док се провера наставља али са  $x_3$ . Како је  $x_3 = 0$ , испитивање се врши над вредностима координата које одговарају чворовима суседним са 3 ( $V_3 = \{1, 2\}$ ). Узимајући да је  $j = 1, 2$  проверавају се вредности  $x_1$  и  $x_2$  и, уколико је бар једна од те две вредности једнака 2, број  $nd$  остаје непромењен. Са обзиром да је  $x_1 = 2$ ,  $nd$  остаје непромењен. Како су вредности свих координата решења  $X$  испитане и  $nd$  није промењен, функција  $brNedopustivih(X)$  враћа нулу. Проверимо сада допустивост решења  $X$  које није из скупа  $X_{\mathcal{RD}}$ . Нека је, на пример,  $X = (1, 1, 0)$ . За овако дефинисано решење вредност функције  $brNedopustivih(X)$  је  $nd = 1$ . Заиста, будући да су  $x_1 > 0$  и  $x_2 > 0$  други услов би се, поново, проверавао само за  $x_3$ . Чвор 3 има два суседа,  $V_3 = \{1, 2\}$ . Са обзиром да је  $x_1 < 2$  и  $x_2 < 2$  следи да  $x_3$  не задовољава ниједан од два услова допустивости. Слично, за  $X = (1, 0, 0)$  функција  $brNedopustivih(X)$  враћа вредност 2 ( $x_2$  и  $x_3$  не задовољавају услове допустивости), односно за  $X = (0, 0, 0)$  враћа вредност 3 (вредности све 3 координате не задовољавају услове допустивости).

Како би се избегло често сумирање вредности координата решења  $X$ , тежина решења ( $F = F(X)$ ) ажурира се заједно са променом тежина координата решења. Подразумева се следеће: увећање вредности једне координате решења  $X$  за 1 (на једну координату решења  $X$  примени се трансформација Т5) имплицира увећање тежине решења за 1; уколико се вредност једне координате увећа за 2 (на једну координату решења  $X$  примени се трансформација Т6) или се вредност две различите координате увећа за 1, без промена вредности осталих координата решења  $X$  (трансформација Т5 се примени на две различите координате), тежина решења ће се увећати за 2; ако се вредност неке координате решења  $X$  умањи за један или два, без промена вредности осталих координата (трансформација Т1 се примени једном или два пута на ту координату),

тада ће се тежина решења умањити за исту вредност. Међутим, уколико се вредност једне координате умањи за 1 (односно 2), а вредност друге координате увећа за исту вредност (на те две координате се примени трансформација Т2 или трансформација Т3), тада се тежина решења неће мењати. Сваки пут када се ажурира најбоље допустиво решење  $X^*$ , ажурира се и његова тежина,  $F^* \leftarrow F(X^*)$ .

Функција  $smanjiBazu(X)$  проверава да ли би отпуст неке легије аргумента  $X$  такође резултовао допустивим решењем. Функција  $smanjiBazu(X)$  имплементирана је на следећи начин. Прво се провери да ли је решење  $X$  допустиво ( $nd \leftarrow brNedopustivih(X)$ ), а затим се, за једну по једну вредност бројача  $i = 1, \dots, n$  проверава да ли је  $x_i > 0$ . Уколико је  $x_i > 0$  даље се врши трансформација  $x_i \leftarrow x_i - 1$  и поново одређује вредност  $nd$  ( $nd \leftarrow brNedopustivih(X)$ ). Ако је  $nd = 0$ , добијено решење памти се као најбоље ( $X^* \leftarrow X, F^* \leftarrow F(X^*)$ ). Иначе се вредност координате  $x_i$  решења  $X$  враћа на првобитну ( $x_i \leftarrow x_i + 1$ ). Провера се наставља за наредну вредност бројача  $i$ . Након што су све вредности бројача  $i$  испитане, уколико је  $nd = 0$ , описани поступак се понавља. Функција  $smanjiBazu(X)$  не враћа ни једну вредност као свој излазни аргумент, већ само врши трансформације над улазним аргументом  $X$  и тренутно најбољим допустивим решењем  $X^*$  у смислу да се, сваки пут када је пронађено допустиво решење, такво решење се запамти као  $X^*$ . Уколико је на почетку функције аргумент  $X$  недопустиво решење, излази се из функције.

### Почетно решње.

Идеја функције  $rocetnoResenje()$  (енг. Initial solution) састоји се у томе да се по посматраном графу (улазној инстанци) произвољно распореде легије тако да су све провинције сигурне или заштићене од напада у односу на задати проблем. Да би се ова идеја реализовала, прво се формира вектор  $X = (0, \dots, 0)$  дужине  $n$ , где  $n$  представља број чворова посматраног графа, а вредности координата  $x_i, i = 1, \dots, n$ , представљају број легија (тежину чвора  $i$ ) стационираних у провинцији  $i$ . Затим се, пошто на сваки чвор могу да се стационирају по највише две легије, формира помоћни вектор  $rom$  тако да се сваки чвор у том вектору спомене по два пута, тј. формира се помоћни вектор  $rom$  облика  $rom = (1, 1, 2, 2, \dots, n, n)$ . Даље се одговарајућим позивом уграђене C++ функција  $radnom_shuffle()$  вредности координата вектора  $rom$  произвољно прераспореде. Потом се за једно по једно  $i = 1, 2, \dots, 2n$ , вредности координата  $x_{rom(i)}$  реше-

ња  $X$  увећавају за 1 (за једно по једно  $i$  се чвору  $pot(i)$  додељује по једна нова легија) све док вектор  $X$  не постане допустиво решење посматраног проблема. Уколико се описаним поступком добије допустиво решење (функција  $brNedopustivih(X)$  врати нулу за решење  $X$ ), поступак увећавања вредности координата решења  $X$  се прекида. Пошто је потпуно могуће да је распоред по коме су се легије стационирале такав да је распоређено више легија него што је неопходно, проверава се да ли би и након отпуста неких легија решење остало допустиво, тј. позива се функција  $smanjiBazu(X)$ , након чега се текуће решење  $X$  враћа као излазни аргумент функције. Функција  $pocetnoResenje()$  детаљно је описана псеудокодом Алгоритам 3.

Поређења ради, код VNS-CAI методе се, приликом формирања почетног решења, у оквиру *while* петље вредности произвољно изабраних координата решења  $X = (0, \dots, 0)$  увећавају за један или два, све док  $X$  не постане допустиво решење. Потенцијално, пошто се изнова може бирати једна те иста координата или се изнова може бирати мањи број различитих координата, овај поступак се може извршавати бесконачно дуго, због чега се, у оквиру нове методе, поступак формирања допустивог решења извршава тако да сигурно даје допустиво решење у коначном времену.

---

### Алгоритам 3 $pocetnoResenje()$

---

```

1:  $X \leftarrow (0, \dots, 0)$ ;
2:  $pot \leftarrow (1, 1, 2, 2, \dots, n, n)$ ;
3:  $pot \leftarrow random\_shuffle(pot.begin(), pot.end());$ 
4: for ( $i = 1, \dots, 2n$ ) do
5:    $x_{pot(i)} \leftarrow x_{pot(i)} + 1$ ;
6:   if ( $brNedopustivih(X) == 0$ ) then
7:      $smanjiBazu(X)$ ;
8:     return  $X$ ;
9:   end
10: end

```

---

Функција  $kvaziBazu(X)$  је, слично функцији  $smanjiBazu(X)$ , дефинисна тако да врши отпуст легије, али са циљем да након отпуста легија аргумент  $X$  постане недопустиво решење. Као што је раније наведено, основна идеја методе за решавање  $RD$  проблема састоји се у томе да се пронађе неко допустиво решење а да се затим, полазећи од тог решења, пронађе допустиво решење мање тежине. Из тог разлога ова функција на произвољан начин смањује тежину аргумента  $X$  све док оно не постане недопустиво. Функција је имплементирана на следећи начин. Прво се

одговарајућим позивом уграђене C++ функције *GetRandom()* произвољно изабере број  $i$  из скупа  $\{1, \dots, n\}$ . Уколико је  $x_i == 0$  бира се нови број. Поступак се понавља све док се не изабере број  $i$  за који важи  $x_i > 0$ <sup>12</sup>. Затим се врши отпуст легије са чвора  $i$  ( $x_i \leftarrow x_i - 1$ ) и одређује број недопустивих вредности ажурираног решења  $X$  ( $nd \leftarrow brNedopustivih(X)$ ). У случају да је  $nd == 0$  понављати следеће:

- добијено решење се запамти као најбоље допустиво решење ( $X^* \leftarrow X$ );
- примени се функција *manjiBazu(X)*;
- произвољно се изабере нови број  $i \in \{1, \dots, n\}$  све док се не добије број  $i$  за који важи  $x_i > 0$ ;
- врши се отпуст легије са чвора  $i$  ( $x_i \leftarrow x_i - 1$ );
- одређује се број недопустивих вредности ажурираног решења  $X$  ( $nd \leftarrow brNedopustivih()$ ).

Описани поступак се понавља све док  $nd$  не постане позитивно. Слично као и код функције *manjiBazu(X)*, ова функција врши трансформације над аргументом  $X$  и тренутно најбољим допустивим решењем и нема излазних аргумената.

### Размрдавање.

Функција *razmrdavanje(X\*, k)* (енг. Shake function) дефинисана је са циљем да се на почетку  $k$  произвољно изабраних легија тренутно најбољег допустивог решења прераспореди, а да се потом једна произвољно изабрана легија отпусти. Псеудокод функције *razmrdavanje(X\*, k)* илустрован је Алгоритмом 4. Функција је имплементирана на следећи начин. Прво се најбоље допустиво решење  $X^*$  памти као текуће ( $X' \leftarrow X^*$ ). Затим се, с обзиром да вредност корака  $k$  може бити већа од тренутног броја легија, односно од тежине  $F^*$ , уводи помоћни параметар  $k_1 \leftarrow \min\{k, F^*\}$ . Помоћни параметар се уводи како би се, у случају да је  $k > k_1$ , прераспоредило највише  $k_1$  легија. Поступак којим се  $k_1$  произвољно изабраних

<sup>12</sup>Теоретски, исти број  $i$  може да се бира бесконачно пута, тј. поступак бирања новог броја  $i$  све док се не пронађе број  $i$  за који важи  $x_i > 0$  може да се понавља бесконачно пута али је вероватноћа да се тако нешто догоди изузетно мала.

легија прераспоређује, односно поступак којим се на решење  $X'$  трансформација T2 примењује  $k_1$  пута изводи се тако што се на  $k_1$  произвољно изабраних координата примењује трансформација T1, а на других  $k_1$  произвољно изабраних координата примењује трансформација T5. С тим у вези, пошто се трансформације T1 и T5 врше искључиво над вредностима координата над којима је то дозвољено, поред параметра  $k_1$ , уводе се бројачи,  $razlika_p$  и  $razlika_m$ , и помоћни вектор  $pot$ . Бројачи се уводе како би се пратило колико пута се применила трансформација T5, а колико пута трансформација T1, односно колико се легија доделило а колико отпустило са текућег решења. Сматра се да је процес прераспоређивања легија завршен уколико су број отпуштених и број додељених легија једнаки  $k_1$ , односно ако је  $razlika_p = razlika_m = k_1$ . На почетку су вредности бројача  $razlika_p$  и  $razlika_m$  постављени на нулу (редови 1-3 Алг. 4). Помоћни вектор  $pot$  уводи се са циљем да се унапред дефинише редослед по коме се легије отпуштају са чворова и додељују чворовима, односно редослед по коме се легије селе са једног чвора на други. Наиме, идеја је да елементи вектора  $pot$  буду индекси чворова са којих се легије отпуштају, односно индекси чворова на које се легије додају. Стога, с обзиром да тежина одређених чворова може бити највише 2, помоћни вектор  $pot$  дефинише се тако да се сваки чвор спомене по два пута ( $pot = (1, 1, 2, 2, \dots, n, n)$ ). Вредности овог вектора се, затим, произвољно прераспоређују одговарајућим позивом уграђене C++ функције  $random\_shuffle()$  (ред 5 Алг. 4). Након што су вредности вектора  $pot$  произвољно прераспоређене, функција  $razmrdavanje(X^*, k)$  размрдава елементе вектора  $X'$  на следећи начин. За једно по једно  $i = 1, \dots, 2n$  проверава се да ли је број отпуштених легија у *размрдавању* мањи од  $k_1$  ( $razlika_m < k_1$ ) и, додатно, да ли важи  $x'_{pot(i)} \geq 1$  и  $x^*_{pot(i)} \geq 1$  (пошто се прераспоређивањем легије премештају са једног чвора на други, условима  $x'_{pot(i)} \geq 1$  и  $x^*_{pot(i)} \geq 1$  се проверава да ли је чвор  $pot(i)$  имао стационираних легија и пре примене функције  $razmrdavanje(X^*, k)$ , односно да ли и даље са њега може да се отпусти нека легија). Уколико је број отпуштених легија мањи од  $k_1$  и легија може да се отпусти са чвора  $pot(i)$ , врши се отпуст легије са чвора  $pot(i)$  ( $x'_{pot(i)} \leftarrow x'_{pot(i)} - 1$ ) и увећава вредност бројача  $razlika_m$  за 1, редови 8-11 Алг.4. Надаље се за исту вредност бројача  $i$ , а након што се провери да ли је  $razlika_m < k_1$ , проверава да ли је број додатих легија мањи од  $k_1$ , односно да ли је  $razlika_p < k_1$  и, додатно, да ли чвору  $pot(i)$  може да се додели једна нова легија. Чвору  $pot(i)$  може да

се додели нова легија уколико важи услов 1)  $x'_{pom(i)} \leq 1$  и  $x^*_{pom(i)} = 0$  (чвор  $pom(i)$  раније није имао ни једну легију и може да му се додели једна нова легија) или услов 2)  $x'_{pom(i)} = 1$  и  $x^*_{pom(i)} \leq 1$  (чвор  $pom(i)$  раније није имао више од једне легије и може да му се додели још једна нова легија). С тим у вези, уколико је број додељених легија мањи од  $k_1$  и чвору  $pom(i)$  може да се додели још једна легија, вредности координате  $x'_{pom(i)}$  и параметра  $razlika_p$  увећавају се за један (редови 14-17 Алг. 4). Испитивања се настављају за наредну вредност бројача  $i$ , осим у случају када је број пресељених легија једнак  $k_1$  ( $razlika_p = razlika_m = k_1$ ). Уколико је број пресељених легија једнак  $k_1$  испитивања за једну по једну вредност бројача  $i$  се прекидају. Као што се може приметити, истовремено се проверавају вредности текућег и тренутно најбољег решења. Оваква провера вршена је како би се избегла ситуација у којој би се умањење/увећање вредности одређене координате а затим увећање/умањење вредности исте координате за исту вредност рачунало као премештање легије/легија, видети пример ниже. Најзад, након што су испитивања за све вредности бројача  $i$  извршена и није дошло до прекида функције, проверава се да ли је отпуштено више легија него што је додељено (редови 23-32 Алг. 4), односно да ли је додељено више легија него што је отпуштено (редови 34-44 Алг. 4). Стога се прво поставља питање да ли је  $razlika_m > razlika_p$  и, уколико је одговор потврдан, пролази кроз нову петљу у којој се за једно по једно  $i = 1, \dots, n$  проверава да ли је  $x_i^* > x_i'$  и  $x_i' < 2$  како би се вредност координате  $x_i'$  увећала за један (подразумева се да би се у том случају и вредност бројача  $razlika_p$  увећала за 1) све док није испуњен услов  $razlika_p = razlika_m$ , када се излази из *for* петље. Иначе, ако није испуњен услов  $razlika_m > razlika_p$ , проверава се да ли важи  $razlika_m < razlika_p$ , односно да ли је додељено више легија него што је отпуштено. Уколико је додељено више легија него што је отпуштено, пролази се кроз нову петљу, односно за једно по једно  $i = 1, \dots, n$  проверава да ли су испуњени услови  $x_i^* < x_i'$  и  $x_i' > 0$ . Ако су наведени услови испуњени, вредности координате  $x_i'$  умањују се за један, док се вредност бројача  $razlika_m$  увећава за један. Слично као и код претходне петље, пролазак кроз вредности једне по једне координате решења  $X'$  се прекида уколико је  $razlika_m = razlika_p$ . Треба имати у виду да код одређених графова, односно код графова одређених тежина,  $razlika_m$  и  $razlika_p$  не морају нужно да буду једнаки  $k_1$ , али свакако нису већи од броја  $k_1$ . Примера ради, ако имамо граф који се састоји из само два

чвора повезана граном и ако је тежина тог графа 3, тада има смисла преселити највише једну легију. С тим у вези, у зависности од размрдавања помоћног вектора  $pot$  вредности параметара  $razlika_m$  и  $razlika_p$  могу бити мање од или једнаке  $k_1$  након што се изврше сви кораци Алгоритма 4 представљени редовима 6-22.

Описаним поступком је пронађено решење из  $\mathcal{N}_{k_1}(X^*)$  околине најмањег допустивог решења. Међутим, пошто је циљ методе да пронађе допустиво решење најмање тежине, на решење  $X'$  се примењује функција  $kvariBazu(X')$  којом се, уместо решења из околине  $\mathcal{N}_{k_1}(X^*)$  добија решење из околине  $\mathcal{N}_{k_2}(X^*)$ , где је  $k_2 > k_1$  и  $k_2$  је цео број. Као свој излазни аргумент, функција  $razmrdavanje(X^*, k)$  враћа решење  $X'$ .

---

**Алгоритам 4**  $razmrdavanje(X^*, k)$

---

```

1:  $X' \leftarrow X^*$ ;
2:  $k_1 \leftarrow \min\{k, F^*\}$ ;
3: вредности бројача који означавају број дodeljenih и број отпуштенih легија постави на нулу;
4:  $pot \leftarrow \{1, 1, 2, 2, \dots, n, n\}$ ;
5:  $pot \leftarrow \text{random\_shuffle}()$ ;
6: for ( $i = 1, \dots, 2n$ ) do
7:   if (број отпуштенih легија је мањи од  $k_1$ ) then
8:     if ( $x'_{pot(i)} \geq 1$ )  $\wedge$  ( $x^*_{pot(i)} \geq 1$ ) then
9:       отпусти легију са чвора  $pot(i)$  ( $x'_{pot(i)} \leftarrow x'_{pot(i)} - 1$ );
10:      увећај број отпуштенih легија за 1 ( $razlika_m \leftarrow razlika_m + 1$ );
11:     end
12:   end
13:   if (број dodeljenih легија је мањи од  $k_1$ ) then
14:     if ( $(x'_{pot(i)} \leq 1) \wedge (x^*_{pot(i)} == 0)$ ) или ( $(x'_{pot(i)} == 1) \wedge (x^*_{pot(i)} \leq 1)$ ) then
15:       dodeli легију чвору  $pot(i)$  ( $x'_{pot(i)} \leftarrow x'_{pot(i)} + 1$ );
16:       увећај број dodeljenih легија за 1 ( $razlika_p \leftarrow razlika_p + 1$ );
17:     end
18:   end
19:   if (број preseljenih легија је једнак  $k_1$ ) then
20:     break;
21:   end
22: end
23: if (отпуштено је више легија него што је dodeljeno) then
24:   for ( $i = 1, \dots, n$ ) do
25:     if ( $x_i^* > x_i'$ )  $\wedge$  ( $x_i' < 2$ ) then
26:       dodeli легију чвору  $i$  ( $x_i' \leftarrow x_i' + 1$ );
27:       увећај број dodeljenih легија за 1 ( $razlika_p \leftarrow razlika_p + 1$ );
28:       if (број dodeljenih легија је једнак броју отпуштенih легија) then
29:         return  $X'$ ;
30:       end
31:     end
32:   end
33: else

```

---



---

```

34:   if (dodeljeno je više legija nego što je otpušteno) then
35:     for ( $i = 1, \dots, n$ ) do
36:       if ( $x_i^* < x_i'$ )  $\wedge$  ( $x_i' > 0$ ) then
37:         otpusti legiju sa čvora  $i$  ( $x'_{pom(i)} \leftarrow x'_{pom(i)-1}$ );
38:         uvećaj broj otpuštenih legija za 1 ( $razlika_m \leftarrow razlika_m + 1$ );
39:         if (broj dodeljenih legija je jednak broju otpuštenih legija) then
40:           return  $X'$ ;
41:         end
42:       end
43:     end
44:   end
45: end
46: kvariBazu( $X'$ );
47: vрати  $X'$ ;

```

---

Упркос чињеници да се описаном методом проналази решење  $X'$  из веће околине у односу на решења која би се добила позивањем функције  $razmrdavanje(X^*, k)$  написане за класичан VNS, сличан приступ показао се као доста успешан код VNS-CAI методе. Заправо, суштинска разлика између функција којима се *размрдава* решење код VNS-CAI и нове методе састоји се у начину на који се прераспоређује  $k$  легија. Док је код VNS-CAI методе премештање могуће једино са чвора који има две легије на чвор који нема ни једну стационирану легију, код нове методе премештање легије могуће је између било која два чвора докле год је оно допуштиво. Додатно, код VNS-CAI методе се прераспоређује највише  $k$  легија, док се код нове методе прераспоређује највише  $k_1 = \min\{k, F^*\}$  легија. Ово не утиче на број  $k$  у оквиру саме VNS методе, већ само гарантује да ће се преместити или  $k$  легија, или ако их нема толико, онда  $F^*$ , односно највише онолико легија колико тренутни распоред дозвољава.

У наставку је рад функције  $razmrdavanje(X^*, k)$  написане за модификовану методу детаљно илустрован на конкретном примеру.

**Пример.** Посматрајмо поступно како се функција  $razmrdavanje(X^*, k)$  извршава за  $X^* = (2, 1, 0, 0)$  и  $k = 2$ . На почетку функције вектори  $X'$  и  $pot$  и променљиве  $razlika_m$ ,  $razlika_p$  и  $k_1$  добијају иницијалне вредности:  $X' \leftarrow X^*$ ,  $pot \leftarrow (1, 1, 2, 2, 3, 3, 4, 4)$ ,  $razlika_m \leftarrow 0$ ,  $razlika_p \leftarrow 0$  и  $k_1 \leftarrow 2$ . Потом се вектор  $pot$  произвољно прераспоређује одговарајућим позивом функције  $random\_shuffle()$ . На пример, нека је прераспоређен вектор  $pot$  облика  $pot = (4, 2, 1, 1, 3, 2, 4, 3)$ . Надаље се за једну по једну вредност бројача  $i = 1, \dots, 8$ , вредности координата вектора  $X'$  и променљивих  $razlika_m$  и  $razlika_p$  мењају на следећи начин:

$i = 1$ , ( $pot(1) = 4$ ): иако је услов  $razlika_m < 2$  испуњен, због услова  $x'_4 \geq 1$  и  $x_4^* \geq 1$  се прелази на део у коме се испитује да ли је  $razlika_p < 2$ ; пошто је  $razlika_p < 2$ , додатно се проверава да ли су  $x'_4 \leq 1$  и  $x_4^* == 0$ ; пошто су последња три услова испуњена, врше се трансформације  $x'_4 \leftarrow x'_4 + 1$  и  $razlika_p \leftarrow razlika_p + 1$ ; ажуриране вредности су следећег облика:  $X' = (2, 1, 0, 1)$ ,  $razlika_p = 1$ ;

$i = 2$  ( $pot(2) = 2$ ): пошто су услови  $razlika_m < 2$ ,  $x'_2 \geq 1$  и  $x_2^* \geq 1$  испуњени, врше се трансформације  $razlika_m \leftarrow razlika_m + 1$  и  $x'_2 \leftarrow x'_2 - 1$ ; ажуриране вредности су следећег облика:  $X' \leftarrow (2, 0, 0, 1)$ ,  $razlika_m \leftarrow 1$ ;

$i = 3$  ( $pot(3) = 1$ ): услов  $razlika_m < 2$  је и даље испуњен; такође су испуњени и услови  $x'_1 \geq 1$  и  $x_1^* \geq 1$ ; врше се трансформације  $razlika_m \leftarrow razlika_m + 1$  и  $x'_1 \leftarrow x'_1 - 1$ ; ажуриране вредности су облика:  $X' \leftarrow (1, 0, 0, 1)$ ,  $razlika_m \leftarrow 2$ ;

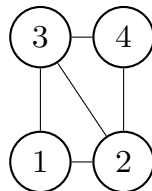
$i = 4$  ( $pot(4) = 1$ ): услов  $razlika_m < 2$  није испуњен; прелази се на услов  $razlika_p < 2$ ; пошто је други услов испуњен, даље се проверавају услови  $x'_1 \leq 1$  и  $x_1^* == 0$ ; с обзиром да услов  $x_1^* == 0$  није испуњен надаље се проверавају услови  $x'_1 == 1$  и  $x_1^* \leq 1$ ; како услов  $x_1^* \leq 1$  није испуњен, прелази се на наредну вредност бројача  $i$ . Приметимо да би се, у случају да услов  $x_1^* == 0$  не постоји (односно да не постоји услов  $x_1^* \leq 1$ ), вредност координате  $x'_1$  увећала за 1 и, самим тим, вратила на првобитну ( $x_1^* = 2$ ) што би могло да се интерпретира као да ни једна легија са чвора 1 није ни одлазила, док би се број прераспоређених легија само за ову ситуацију рачунао као 2 (једна легија се отпустила са чвора 1 а друга легија се стационира на чвор 1), што није тачно;

$i = 5$  ( $pot(5) = 3$ ): пошто је  $razlika_m = 2$  прелази се на проверу испуњености услова  $razlika_p < 2$ ; други услов је испуњен а испуњени су и услови  $x'_3 \leq 1$  и  $x_3^* == 0$ ; врше се трансформације  $x'_3 \leftarrow x'_3 + 1$ ,  $razlika_p \leftarrow razlika_p + 1$ ; ажуриране вредности су  $X' \leftarrow (1, 0, 1, 1)$ ,  $razlika_p \leftarrow 2$ ;

Како је  $razlika_p = razlika_m = 2$ , функција  $razmrdavanje(X^*, 2)$  завршава рад *for* петље. На пронађено решење  $X' = (1, 0, 1, 1)$  примењује се функција  $kvariBazu(X')$ , након чега функција  $razmrdavanje(X^*, 2)$  као свој излазни аргумент враћа решење  $X' \in \{(0, 0, 1, 1), (1, 0, 0, 1), (1, 0, 1, 0)\}$ . Излазни аргумент  $X'$  је разлике реда 3 у односу на решење  $X^*$ : по једна легија са чворова 1 и 2 премештена је на чворове 3 и 4 формирајући

решење које је реда 2 у односу на  $X^*$ ; на решење које је реда 2 у односу на  $X^*$  је, потом, примењена трансформација T1, тј. извршен је отпуст легије са једног од чорова 1, 3 и 4, чиме се добило решење које је разлике реда 3 у односу на решење  $X^*$ .

Пре описа функције *lokalnoPretrazivanje*( $X'$ ) упознаћемо се са помоћном функцијом *ispitajDopustivost*( $X, i$ ). Укратко, функција *ispitajDopustivost*( $X, i$ ) дефинисана је тако да проверава допустивост решења улазног вектора  $X$  у зависности од вредности улазног параметра  $i$  на следећи начин. Прво се, уколико је  $i > 0$ , отпусти легија са чвора  $i$ , а затим се проверава да ли отпуст даје допустиво решење или не (одређује се број координата измењеног решења  $X$  које не задовољавају услове допустивости проблема,  $nd \leftarrow brNedopustivih(X)$ ). Уколико отпуст даје допустиво решење ( $nd \leftarrow 0$ ), такво решење се памти. Уколико отпуст не даје допустиво решење, проверава се да ли би се сеобом легије са чвора  $i$  на неки други чвор добило допустиво решење или не. Уколико би се сеобом добило допустиво решење, функција памти такво решење. Међутим, уколико сеобом легије са чвора  $i$  на неки други чвор не може да се добије допустиво решење, функција тражи решење које ће бити најближе допустивом у смислу да ће имати најмање недопустивих координата. Након што је пронашла допустиво решење или решење које је најближе допустивом, за  $i > 0$  функција *ispitajDopustivost*( $X, i$ ) враћа отпуштену легију на чвор  $i$ . За  $i \leq 0$  функција *ispitajDopustivost*( $X, i$ ) не врши отпуст легије већ одмах прелази на део кода у коме се за једну по једну координату вектора  $X$  проверава да ли би се увећањем њене вредности добило решење са мање недопустивих координата, односно проверава да ли би се добило допустиво решење. Ради лакшег описа саме функције посматрајмо граф  $\mathcal{G}_{ID} = (V_{\mathcal{G}_{ID}}, E_{\mathcal{G}_{ID}})$  приказан на слици 17 и примере који су дати у наставку. Детаљни опис саме функције дат је након примера.



Слика 17: Граф  $\mathcal{G}_{ID}$

**Пример 1.** За улазни вектор  $X = (1, 0, 2, 1)$  и параметар  $i = 1$  функција *ispitajDopustivost*( $X, i$ ) проверава да ли је решење  $X = (0, 0, 2, 1)$  такође

допустиво решење посматраног проблема за граф  $\mathcal{G}_{ID}$ . Пошто вектор  $X = (0, 0, 2, 1)$  јесте допустиво решење посматраног проблема, излазном аргументу функције  $ispitajDopustivost(X, i)$ ,  $nd$ , додељује се вредност 0.

**Пример 2.** За улазни вектор  $X = (1, 0, 2, 1)$  и параметар  $i = 3$  функција  $ispitajDopustivost(X, i)$  променљивој  $nd$  додељује вредност 1 зато што једна координата вектора  $X = (1, 0, 1, 1)$  не задовољава услове допустивости посматраног проблема (отпустом легије са чвора 3 чвор 2 постаје незаштићен). Пошто отпуст легије даје недопустиво решење, функција  $ispitajDopustivost(X, i)$  даље проверава да ли би неком другом чвору могла да се додели та легија и да се на тај начин поново формира допустиво решење (у Примеру 1 додавање легије није било потребно обзиром да је пронађено допустиво решење). Испитујући вредност једног по једног чвора графа, функција  $ispitajDopustivost(X, i)$  проверава да ли би, уместо отпуста легије са чвора 3, та легија могла да се пресели на неки други чвор и да се на тај начин формира допустиво решење. Пошто би се премештањем легије са чвора 3 на чвор 1 формирало допустиво решење, функција  $ispitajDopustivost(X, i)$  памти индексе тих чворова као  $ls$  и  $ln$ ,  $ls \leftarrow 3$ ,  $ln \leftarrow 1$  (у Примеру 1. параметрима  $ls$  и  $ln$  додељује се вредност  $i$ ) и број недопустивих вредности тако формираног решења као  $ndID_{min}$  ( $ndID_{min} \leftarrow 0$ ).

**Пример 3.** За улазни вектор  $X = (1, 0, 0, 1)$  и параметар  $i = 1$  број недопустивих координата вектора  $X = (0, 0, 0, 1)$  је 3. Пошто сеобом легије са чвора  $i$  на било који други чвор не може да формира допустиво решење, функција тражи решење са најмањим бројем недопустивих координата. С обзиром да би се сеобом легије са чвора 1 на чвор 4 добило решење које има само једну недопустиву координату и да је такво решење најближе допустивом, функција  $ispitajDopustivost(X, i)$  памти следеће  $ls \leftarrow 1$ ,  $ln \leftarrow 4$  и  $ndID_{min} \leftarrow 1$  (променљива  $ndID_{min}$  памти број недопустивих координата пронађеног решења).

Дакле, као што је раније напоменуто, функција  $ispitajDopustivost(X, i)$  умањује вредност  $i$ -те координате за 1 уколико је  $i > 0$  (при позиву функције  $ispitajDopustivost(X, i)$  води се рачуна о томе да је  $x_i > 0$ , односно да је трансформација  $x_i \leftarrow x_i - 1$  допустива) а затим се вредност параметра  $nd$  одређује одговарајућим позивом функције  $brNedopustivih(X)$  ( $nd \leftarrow brNedopustivih(X)$ ). Уколико је  $nd = 0$ , функција враћа ту вредност и

памти вредност параметра  $i$  као  $ls$  и  $ln$  ( $ls = ln = i$ ). Међутим, уколико је  $nd > 0$ , даље се за једно по једно  $j = 1, \dots, n, j \neq i$ , проверава да ли би увећање вредности координате  $x_j$  ( $x_j < 2$ ) улазног аргумента  $X$  за један формирало допустиво решење, односно решење са мањим бројем недопустивих вредности од  $nd$ . Пролазећи кроз једну по једну вредност бројача  $j$ , функција  $ispitajDopustivost(X, i)$  памти оно померање легије са  $x_i$  на  $x_j$  ( $j = 1, \dots, n$ ) за које је број недопустивих вредности најмањи као  $ls \leftarrow i$  и  $ln \leftarrow j$ , а најмањи број недопустивих вредности памти као  $ndID_{min}$ . Након што је вредност  $ndID_{min}$  одређена, функција  $ispitajDopustivost(X, i)$  поставља вредност координате  $x_i$  решења  $X$  на почетну и као излазни аргумент враћа вредност параметра  $nd$ . Уколико је  $i \leq 0$  функција  $ispitajDopustivost(X, i)$  извршава само део кода у коме се за једно по једно  $j = 1, \dots, n$  проверава да ли би се увећањем вредности координате  $x_j$  улазног аргумента  $X$  добило решење са мањим бројем недопустивих вредности од  $nd$ , тј. одмах извршава део кода у коме се одређују вредности  $ls, ln$  и  $ndID_{min}$  и враћа вредност излазног аргумента  $nd$  (на почетку  $ls$  и  $ln$  добијају вредност  $i$  без обзира што је  $i < 0$  а затим се, сваки пут када се пронађе број  $j$  такав да увећање вредности координате  $x_j$  даје решење са мање недопустивих вредности, вредност параметра  $ln$  поставља на  $j$ ; истовремено се ажурира и вредност параметра  $ndID_{min}$  тако да одговара решењу са мањим бројем недопустивих вредности).

### Локално претраживање (енг. Local Search).

Полазећи од решења  $X'$ , функција  $lokalnoPretrazivanje(X')$ <sup>13</sup> тражи допустиво решење  $X''$  на следећи начин. Прво се решење  $X'$  запамти као  $X''$ . У даљем тексту за решење  $X''$  користиће се израз *текуће* решење, док ће се за решење  $X^*$  користити израз *најбоље допустиво решење*. Идеја функције  $lokalnoPretrazivanje(X')$  је да у околини текућег решења пронађе ново допустиво решење мање тежине од  $X^*$ . Како је тежина решења  $X''$  мања у односу на тежину најбољег допустивог решења, ако би  $X''$  било допустиво решење, онда би оно уједно било и допустиво решење најмање тежине пронађено до тог момента. С тим у вези, сваки пут када се пронађе ново допустиво решење, такво решење постаје најбоље допустиво решење (вредност вектора  $X^*$  се ажурира). Решење  $X''$  се на почетку функције сматра и решење са најмањим бројем недопустивих вредности (*најбоље пронађено решење*), па се број коо-

<sup>13</sup>Функција је детаљно илустрована Алгоритмом 5.

рдината решења  $X''$  који не испуњавају услов допустивости памти као  $nd_{min}$ ,  $nd_{min} \leftarrow brNedopustivih(X'')$ . Функција  $lokalnoPretrazivanje(X')$  дефинисана је тако да поступком локалног претраживања, поставивши у  $X''$  улазно  $X'$ , у околини решења  $X''$  пронађе решење са мањим бројем недопустивих вредности од  $nd_{min}$ , односно решење које ће имати нула недопустивих вредности. Поступак локалног претраживања врши се тако што се за један по један чвор графа (односно, за једну по једну координату решења  $X''$ ) проверава да ли су на њему стациониране легије (да ли је позитивне тежине) и да ли би пресељење једне стациониране легије (умањење тежине те координате за један) на други чвор (увећање тежине неке друге координате за један) могло да формира допустиво решење или решење са мање недопустивих вредности од  $nd_{min}$ . Заправо, прво се за једну по једну координату позитивне тежине (за једно по једно  $i = 1, \dots, n$  за које важи  $x_i'' \neq 0$ ) проверава да ли би отпуст једне њене легије дао допустиво решење (позива се функција  $nd \leftarrow ispitajDopustivost(X'', i)$  и проверава да ли је  $nd == 0$ , редови 8-9 Алг. 5). Уколико би отпуст легије дао допустиво решење, примењује се отпуст легије са те координате ( $x_i'' \leftarrow x_i'' - 1$ ) а пронађено допустиво решење памти као најбоље допустиво решење ( $X^* \leftarrow X''$ ), редови 10-11 Алг. 5. Додатно, са циљем да се пронађе допустиво решење још мање тежине, на решење  $X''$  се примењује функција  $manjiBazu(X'')$  након чега функција  $lokalnoPretrazivanje(X')$  враћа вектор  $X''$  као свој излазни аргумент (редови 12-13 Алг. 5). Иначе, ако отпуст легије не даје допустиво решење, али се, зато, допустиво решење може добити ако се легија пресели са чвора  $i$  на чвор  $ln$  (позивом функције  $ispitajDopustivost(X'', i)$  добије се да је  $nd \neq 0$ , али да је зато  $ndID_{min} = 0$ ), на чворове  $i$  и  $ln$  примењује се трансформација Т2 (редови 15-16 Алг.5). Слично као и у претходном случају, најбоље допустиво решење се ажурира, на вектор  $X''$  додатно примењује функција  $manjiBazu(X'')$ , након чега функција  $lokalnoPretrazivanje(X')$  прекида свој рад и враћа вектор  $X''$  као излазни аргумент (редови 17-19 Алг. 5). Међутим, уколико ни отпуст а ни трансформација Т2 не резултују допустивим решењем ( $nd \neq 0$  и  $ndID_{min} \neq 0$ ), надаље се проверава да ли би се трансформацијом Т2 добило решење са мањим бројем недопустивих вредности од  $nd_{min}$  (решење са мањим бројем недопустивих вредности називамо још и *решењем бољег квалитета*), тј. проверава се да ли је  $ndID_{min} < nd_{min}$ , ред 21 Алг.5. Уколико важи  $ndID_{min} < nd_{min}$ , решење  $X''$  се памти као  $X_b''$  ( $X_b'' \leftarrow X''$ ), након чега се на

координате  $i$  и  $ln$  решења  $X_b''$  примењује трансформација T2 (у наставку ћемо само рећи да се решење бољег квалитета памти као  $X_b''$ ), док се вредност  $nd_{min}$  ажурира тако да одговара броју недопустивих вредности решења  $X_b''$  ( $nd_{min} \leftarrow ndID_{min}$ ). Међутим, ако се T2 трансформацијом пронађе решење са  $nd_{min}$  недопустивих вредности (решење за које важи  $ndID_{min} == nd_{min}$ ), сматра се да је пронађено решење истог квалитета. Решење истог квалитета памти се као  $X_i''$  (прво се решење  $X''$  запамти као  $X_i''$  а затим се на  $i$  и  $ln$  координате решења  $X_i''$  примени T2 трансформација) уколико је то прво такво решење. Пошто је потпуно могуће да се током процеса претраге пронађе још неко решење са  $nd_{min}$  недопустивих вредности, такво решење ће се са вероватноћом  $p$  запамтити као  $X_i''$  (редови 24-30 Алг. 5). Најзад, након што су испитане вредности свих координата вектора  $X''$ , односно, након што су испитане све вредности бројача  $i$ , пронађено квалитетније решење постаје текуће уколико се описаним поступком пронашло решење бољег квалитета (редови 36-38 Алг. 5), односно пронађено решење истог квалитета постаје текуће уколико се описаним поступком није пронашло решење бољег али се пронашло решење истог квалитета и ако важи  $GetRandom(10) > p$  (редови 40-42 Алг. 5). У оба случаја променљива *napredak* добија вредност 1. Међутим, уколико ни решење бољег квалитета нити решење истог квалитета није постало текуће, одређује се број недопустивих вредности текућег решења  $X''$  и памти као  $nd_{min}$ , а затим се приступа делу кода који је представљен као помоћна функција *lokalnoPretrazivanje2*( $X''$ )<sup>14</sup> (ред 45 Алг. 5). Иако функција *lokalnoPretrazivanje2*( $X''$ ) не мора нужно да се користи као засебна функција, она се у оквиру функције *lokalnoPretrazivanje*( $X'$ ) користи као засебна функција искључиво ради боље прегледности кода. Слично функцији *lokalnoPretrazivanje*( $X'$ ), ова функција претражује околине текућег решења  $X''$  с циљем да пронађе допустиво решење мање тежине од тренутно најбољег допустивог решења али користећи трансформације T1, T2, T3, T4, T5 и T6. Након детаљне претраге, она враћа број недопустивих вредности ( $nd$ ) решења  $X''$ . Потом, ако је  $nd = 0$ , функција *lokalnoPretrazivanje*( $X'$ ) враћа текуће решење као излазни аргумент, ред 47 Алг. 5. Иначе, ако је  $nd < nd_{min}$ , то значи да се помоћу трансформација T1 - T6 пронашло решење бољег квалитета, па параметар *napredak* добија вредност 1. У супротном параметар *napredak*

<sup>14</sup>Функција *lokalnoPretrazivanje2*( $X''$ ) детаљно је објашњена након описа функције *lokalnoPretrazivanje*( $X'$ ).

добија вредност 0. Уколико параметар *napredak* има позитивну вредност описани поступак се понавља све док нису испуњени унапред дефинисани критеријуми заустављања, односно све док није достигнуто максимално време извршавања кода. Уколико су критеријуми заустављања испуњени, функција *lokalnoPretrazivanje(X')* враћа текуће решење као свој излазни аргумент.

---

**Алгоритам 5** *lokalnoPretrazivanje(X')*

---

```

1:  $X'' \leftarrow X'$ ;
2:  $nd_{min} \leftarrow brNedopustivih(X'')$ ;
3:  $napredak \leftarrow 1$ ;
4: while ( $napredak$ ) i (kriterijum zaustavljanja nije zadovoljen) do
5:    $napredak \leftarrow 0$ ;
6:   for ( $i = 1, \dots, n$ ) do
7:     if ( $x_i'' \neq 0$ ) then
8:        $nd \leftarrow ispitajDopustivost(X'', i)$ ;
9:       if (rešenje je dopustivo) then
10:        otпусти legiju sa čvora i;
11:        ažuriraj najbolje rešenje;
12:         $smanjiBazu(X'')$ ;
13:        return  $X''$ ;
14:      else
15:        if (rešenje je dopustivo kada se legija preseli sa čvora i na čvor ln) then
16:          preseli legiju sa čvora i na čvor ln;
17:          ažuriraj najbolje rešenje;
18:           $smanjiBazu(X'')$ ;
19:          return  $X''$ ;
20:        else
21:          if (pronađeno je rešenje boljeg kvaliteta) then
22:            zapamti rešenje boljeg kvaliteta kao  $X_b''$ ;
23:          else
24:            if (pronađeno je rešenje istog kvaliteta) then
25:              if (prvi put je pronađeno rešenje istog kvaliteta) then
26:                pamti rešenje istog kvaliteta kao  $X_i''$ ;
27:              else
28:                pamti rešenje istog kvaliteta kao  $X_i''$  sa verovanoćom p;
29:              end
30:            end
31:          end
32:        end
33:      end
34:    end
35:  end

```

---



---

```

36:   if (pronađeno je rešenje boljeg kvaliteta) then
37:     rešenje boljeg kvaliteta postaje tekuće,  $X'' \leftarrow X_b''$ ;
38:     napredak  $\leftarrow$  1;
39:   else
40:     if (pronađeno je rešenje istog kvaliteta) i (GetRandom(10) > p) then
41:       napredak  $\leftarrow$  1;
42:       rešenje istog kvaliteta postaje tekuće,  $X'' \leftarrow X_i''$ ;
43:     else
44:        $nd_{min} \leftarrow brNedopustivih(X'')$ ;
45:        $nd \leftarrow lokalnoPretrazivanje2(X'')$ ;
46:       if (pronađeno je dopustivo rešenje) then
47:         return  $X''$ ;
48:       end
49:       if ( $nd < nd_{min}$ ) then
50:          $nd_{min} \leftarrow nd$ ;
51:         napredak  $\leftarrow$  1;
52:       else
53:         napredak  $\leftarrow$  0;
54:       end
55:     end
56:   end
57: end
58: return  $X''$ ;

```

---

Функција  $lokalnoPretrazivanje2(X'')$  детаљно је представљена алгоритмом 6. Слично функцији  $lokalnoPretrazivanje(X')$ , ова функција поступком локалног претраживања тражи допустиво или решење са мањим бројем недопустивих вредности од текућег али користећи трансформације T1, T2, T3, T4, T5 и T6. Другим речима, она проверава да ли две легије могу да се преселе са једног чвора на неки други чвор, односно да ли са једног чвора две легије могу да се преселе на нека друга два различита чвора и да се на тај начин пронађе допустиво или решење са мањим бројем недопустивих вредности у односу на оно са којим је та претрага започета. Зато ће се решење  $X''$  поново сматрати *текућим* а број његових недопустивих вредности на почетку запамтити као  $nd_{min}$  ( $nd_{min} \leftarrow brNedopustivih(X'')$ ). Водећи се истом идејом као код функције  $lokalnoPretrazivanje(X')$ , ако се током процеса локалног претраживања пронађе допустиво решење, најбоље допустиво решење ће се ажурирати, затим ће се на пронађено допустиво решење применити функција  $manjiBazu(X'')$ , након чега ће функција  $lokalnoPretrazivanje2(X'')$  вратити нулу као излазни аргумент. Међутим, ако се током процеса локалног претраживања не пронађе допустиво, али се пронађе решење које има мање недопустивих вредности од  $nd_{min}$ , такво решење сматраће се решењем бољег квалитета и памтити као  $X_b''$  (док ће се вредност

$nd_{min}$  ажурирати тако да одговара броју недопустивих вредности решења  $X_b''$ ), односно сваки пут када се пронађе решење које има исти број недопустивих вредности као  $nd_{min}$ , такво решење сматраће се решењем истог квалитета и, уз одређене услове, памтити као вектор  $X_i''$ . Затим, након што су испитивања завршена без пронађеног допустивог решења, али са пронађеним решењем бољег квалитета, решење бољег квалитета постаће текуће. Иначе, ако се не пронађе допустиво, нити решење бољег квалитета, али се пронађе решење истог квалитета, решење истог квалитета постаће текуће са вероватноћом  $p$ . У оба случаја, процес локалног претраживања ће се применити из почетка над трансформисаним текућим решењем, уз услов да решења истог квалитета могу да постану текућа највише  $k_{max}$  узастопних пута. Услов да се у решења истог квалитета прелази највише  $k_{max}$  узастопних пута уведен је с обзиром да је потпуно могуће да се наизменично прелази из једног у друго решење, што би се, без овог услова, понављало све док се не испуни критеријум за прекид рада методе. Дакле, функција  $lokalnoPretrazivanje2(X'')$  дефинисана је тако да се у оквиру *while* петље врше испитивања једне по једне координате текућег решења све док се не пронађе допустиво решење, односно док постоји неки напредак (решење бољег квалитета је постало текуће или је решење истог квалитета постало текуће). У случају да до напретка није дошло, или да је број прелазака у решења истог квалитета достигао свој максимум, функција  $lokalnoPretrazivanje2(X'')$  прекида свој процес и враћа број недопустивих вредности текућег решења као излазни аргумент.

Тело *while* петље детаљно се може описати на следећи начин. Прво се за једну по једну координату решења  $X''$ , односно за једно по једно  $i = 1, \dots, n$  проверава да ли је  $x_i'' = 2$ . Уколико је одговор потврдан, даље се за једно по једно  $j = 1, \dots, n$  проверава да ли је  $x_j'' = 0$ . Ако је  $x_j'' = 0$ , две легије се селе са чвора  $i$  на чвор  $j$ , односно над вредностима координата  $x_i''$  и  $x_j''$  примењује се трансформација ТЗ ( $x_i'' \leftarrow 0, x_j'' \leftarrow 2$ ), ред 10 Алг. 6. Потом се одређује број недопустивих вредности ( $nd_1$ ) тако формираног текућег решења  $X''$  ( $nd_1 \leftarrow ispitajDopustivost(X'', j)$ ), ред 11 Алг. 6. У случају да функција  $ispitajDopustivost(X'', j)$  врати нулу, односно важи  $nd_1 == 0$ , то значи да је решење  $X''$  допустиво и да ће остати допустиво и након што се са чвора  $j$  отпусти једна легија. Будући да је пронађено допустиво решење мање тежине, прво се једна легија отпушта са чвора  $j$  ( $x_j'' \leftarrow x_j'' - 1$ ) а затим се трансформисано

решење памти као најбоље допустиво решење (најбоље допустиво решење се ажурира,  $X^* \leftarrow X''$ ). Иако је функција  $lokalnoPretrazivanje2(X'')$  пронашла ново допустиво решење, додатно се проверава да ли би отпуст још неке легије резултовао допустиво решење још мање тежине (позива се функција  $smanjiBazu(X'')$ ), након чега се, као излазни аргумент функције  $lokalnoPretrazivanje2(X'')$ , враћа нула, редови 13-16 Алг.6. Међутим, ако функција  $ispitajDopustivost(X'', j)$  не врати нулу, али пронађе чвор  $ln$  за који важи да је  $ndID_{min} == 0$ , следи да ће се допустиво решење добити уколико се једна легија пресели са чвора  $j$  на чвор  $ln$ . Стога се на координате  $x''_j$  и  $x''_{ln}$  текућег решења примењује T2 трансформација, а резултујуће решење памти као најбоље допустиво решење ( $X^* \leftarrow X''$ ). Функција  $lokalnoPretrazivanje2(X'')$  и у овом случају примењује функцију  $smanjiBazu(X'')$  на текуће решење након чега враћа нулу као излазни аргумент (редови 18-22 Алг. 6). Међутим, уколико је  $nd_1 \neq 0$  и  $ndID_{min} \neq 0$ , даље се проверава да ли би се трансформацијом T2 пронашло решење бољег квалитета, тј. проверава се да ли важе услови  $ndID_{min} < nd_1$  и  $ndID_{min} < nd_{min}$  (ред 24 Алг.6). Ако се трансформацијом T2 може формирати решење бољег квалитета, решење бољег квалитета се памти тако што се решење  $X''$  запамти као  $X''_b$  а затим се на  $j$  и  $ln$  координате решења  $X''_b$  примењује T2 трансформација. Иначе, ако квалитетније решење није пронађено, проверава се да ли је пронађено решење истог квалитета (редови 27-33 Алг.6). Код испитивања да ли је пронађено решење истог квалитета води се рачуна о томе да ли је  $ndID_{min} \leq nd_1$  и  $ndID_{min} == nd_{min}$  и, додатно, да ли је такво решење први пут пронађено или не. Ако је решење истог квалитета први пут пронађено, онда се оно памти као  $X''_i$  (прво се  $X''$  запамти као  $X''_i$  а затим се на  $j$  и  $ln$  координате решења  $X''_i$  примењује трансформација T2) међутим, ако ово није прво пронађено решење истог квалитета, онда се оно памти са вероватноћом  $p$ . Најзад, пошто су за почетне вредности координата  $x''_i$  и  $x''_j$  решења  $X''$  испитане све могућности, и функција  $lokalnoPretrazivanje2(X'')$  није вратила нулу као свој излазни аргумент, вредности координата  $x''_i$  и  $x''_j$  постављају се на првобитне  $x''_i \leftarrow 2$ ,  $x''_j \leftarrow 0$  (трансформација T3 се сада примењује на  $j$  и  $i$ ). Слична испитивања се врше уколико за чвор  $j$  важи  $x''_j == 1$  (редови 39-69 Алг.6). Пошто је чвор  $i$  тежине 2, а чвор  $j$  тежине 1, прво се једна легија сели са чвора  $i$  на чвор  $j$  а друга легија са чвора  $i$  отпушта ( $x''_i \leftarrow 0$ ,  $x''_j \leftarrow 2$ ), а затим се проверава да ли би једна легија могла да се додели неком чвору и да се на тај начин формира

допустиво решење (позива се функција  $nd \leftarrow \text{ispitajDopustivost}(X'', -1)$ ). У случају да је  $nd == 0$ , следи да једна легија не мора да се додаје ни једном чвору јер је решење  $X''$  допустиво и без ње. Ако је решење  $X''$  допустиво, онда се оно памти као најбоље допустиво решење ( $X^* \leftarrow X''$ ). Међутим, ако се позивањем функције  $\text{ispitajDopustivost}(X'', -1)$  добије да је  $nd \neq 0$  али да је  $ndID_{min} == 0$ , то значи да би се додавањем легије на чвор  $ln$  могло формирати допустиво решење, па се вредност координате  $x''_{ln}$  решења  $X''$  увећава за један (примењује се трансформација T5) а тако формирано решење памти као најбоље допустиво решење. У оба случаја, након што је вредност најбољег допустивог решења ажурирана, на решење  $X''$  се примењује функција  $\text{ smanjiBazu}(X'')$ , а нула враћа као излазни аргумент функције  $\text{ lokalnoPretrazivanje2}(X'')$ . Међутим, ако се описаним поступком није пронашло допустиво, али се пронашло квалитетније решење (важи  $ndID_{min} \leq nd_{min}$ ), пронађено квалитетније решење се памти (редови 53-54 Алг.6). Слично, ако се пронашло решење истог квалитета (важи  $ndID_{min} == nd_{min}$ ), такво решење се памти уколико је први пут пронађено решење истог квалитета, односно памти са вероватноћом  $p$  ако је раније већ постојало решење истог квалитета (редови 56-63 Алг. 6). По завршетку свих испитивања, а унутар питања да ли је  $x''_j == 1$  се, уколико у међувремену рад функције  $\text{ lokalnoPretrazivanje2}(X'')$  није прекинут, вредности координата  $x''_i$  и  $x''_j$  постављају на почетне тако што се чвору  $i$  доделе две легије а са чвора  $j$  отпусти једна легија ( $x''_i \leftarrow 2, x''_j \leftarrow 1$ ), (редови 67-68 Алг.6). Испитивања се настављају за наредну вредност бројача  $j$  све док се испитивања не изврше за све координате решења  $X''$ . Након што су испитивања извршена за све координате решења  $X''$ , односно након што је проверено да ли се вредност једне координате може умањити за 2, а вредност друге увећати за 2 или вредности друге две различите координате може увећати за по 1, решење бољег квалитета ( $X''_b$ ) постаје текуће уколико је такво решење пронађено (редови 74-78 Алг.6). Иначе, ако није пронађено решење бољег квалитета, али је пронађено решење истог квалитета ( $X''_i$ ), решење истог квалитета постаје текуће са вероватноћом  $p$ , уз услов да број прелазака у решења истог квалитета не може да буде већи од  $k_{max}$ . Поред услова да се у решење истог квалитета не прелази више од  $k_{max}$  пута, води се рачуна да се прелазак у решење истог квалитета не врши уколико би се таквим поступком наизменично прелазило из једног решења у друго (сваки пут када се за текуће решење прогласи решење бољег или истог квалитета, такво решење се запамти

као  $X_p''$  и касније пореди са решењем истог квалитета које треба да се прогласи са текуће). Заправо, пошто је потпуно могуће да се током једног извршавања *while* петље за ново текуће решење прогласи решење које се добило селидбом две легије са једног чвора на други, а током наредног извршавања *while* петље пронађе решење истог квалитета којим би се, ако постане текуће, две легије вратиле са чвора коме су се доделиле на чвор са ког су се раније отпустиле, решење истог квалитета се пореди са решењем  $X_p''$  (уколико оно постоји) пре него што се проглашава за текуће. У случају да би се проглашавањем решења истог квалитета за текуће решење прогласило решење које је то непосредно пре њега било, функција *lokalnoPretrazivanje2*( $X''$ ) прекида рад *while* петље и враћа број недопустивих вредности текућег решења као излазни аргумент.

---

#### Алгоритам 6 *lokalnoPretrazivanje2*( $X''$ )

---

```

1:  $nd_{min} \leftarrow brNedopustivih(X')$ ;
2:  $brPonavljanja \leftarrow 0$ ;
3:  $napredak \leftarrow 1$ ;
4: while ( $napredak$ ) и ( $brPonavljanja < k_{max}$ ) do
5:    $napredak \leftarrow 0$ ;
6:   for ( $i = 1, \dots, n$ ) do
7:     if ( $x_i'' == 2$ ) then
8:       for ( $j = 1, \dots, n$ ) do
9:         if ( $x_j'' == 0$ ) then
10:          preseli dve legije sa čvora i na čvor j;
11:           $nd_1 \leftarrow ispitajDopustivost(X'', j)$ ;
12:          if ( $nd_1 == 0$ ) then
13:            otpusti legiju sa čvora j;
14:            ažuriraj najbolje rešenje;
15:            smanjiBazu( $X''$ );
16:            return 0;
17:          else
18:            if ( $ndID_{min} == 0$ ) then
19:              preseli legiju sa čvora j na čvor ln;
20:              ažuriraj najbolje rešenje;
21:              smanjiBazu( $X''$ );
22:              return 0;
23:            else
24:              if (pronađeno je rešenje boljeg kvaliteta) then
25:                zapamti rešenje boljeg kvaliteta kao  $X_b''$ ;
26:              else
27:                if (pronađeno je rešenje istog kvaliteta) then
28:                  if (prvi put je pronašao rešenje istog kvaliteta) then
29:                    pamti rešenje istog kvaliteta kao  $X_i''$ ;
30:                  else
31:                    pamti rešenje istog kvaliteta kao  $X_i''$  sa verovatnoćom p;
32:                  end
33:                end
34:              end
35:            end

```

---

---

```

36:         end
37:         preseli dve legije sa čvora j na čvor i;
38:     else
39:         if ( $x_j'' == 1$ ) then
40:             preseli jednu legiju sa čvora i na čvor j, a drugu otpusti sa čvora i;
41:              $nd \leftarrow \text{ispitajDopustivost}(X'', -1)$ ;
42:             if (pronađeno je dopustivo rešenje) then
43:                 ažuriraj najbolje rešenje;
44:                 smanjiBazu( $X''$ );
45:                 return 0;
46:             else
47:                 if (rešenje je dopustivo ako se jedna legija dodeli čvoru ln) then
48:                     dodeli legiju na čvor ln;
49:                     ažuriraj najbolje rešenje;
50:                     smanjiBazu( $X''$ );
51:                     return 0;
52:                 else
53:                     if (pronađeno je rešenje boljeg kvaliteta) then
54:                         zapamti rešenje boljeg kvaliteta kao  $X_b''$ ;
55:                     else
56:                         if (pronađeno je rešenje istog kvaliteta) then
57:                             if (prvi put je pronašao rešenje istog kvaliteta) then
58:                                 pamti rešenje istog kvaliteta kao  $X_i''$ ;
59:                             else
60:                                 sa verovatnoćom p pamti rešenje istog kvaliteta
61:                                 kao  $X_i''$ ;
62:                             end
63:                         end
64:                     end
65:                 end
66:             end
67:             dodeli čvoru i dve legije;
68:             otpusti jednu legiju sa čvora j;
69:         end
70:     end
71: end
72: end
73: end
74: if (pronađeno je rešenje boljeg kvaliteta) then
75:     brPonavljanja  $\leftarrow 0$ ;
76:     napredak  $\leftarrow 1$ ;
77:     rešenje boljeg kvaliteta postaje tekuće,  $X'' \leftarrow X_b''$ ;
78:     zapamti rešenje koje je postalo tekuće kao  $X_p''$ ;
79: else
80:     if (pronađeno je rešenje istog kvaliteta) then
81:         brPonavljanja  $\leftarrow \text{brPonavljanja} + 1$ ;
82:         if (ranije se prelazilo u rešenje istog kvaliteta) then
83:             if (rešenja  $X_i''$  i  $X_p''$  su dva naizmenična rešenja) then
84:                 break;
85:             else
86:                 napredak  $\leftarrow 1$ ;
87:                 rešenje istog kvaliteta postaje tekuće,  $X'' \leftarrow X_i''$ ;
88:                 zapamti rešenje koje je postalo tekuće kao  $X_p''$ ;
89:             end
90:         else

```

---

---

```

91:         napredak ← 1;
92:         rešenje istog kvaliteta postaje tekuće,  $X'' \leftarrow X''_i$ ;
93:         zapamti rešenje koje je postalo tekuće kao  $X''_p$ ;
94:     end
95:     else
96:         return brNedopustivih( $X''$ );
97:     end
98: end
99: return brNedopustivih( $X''$ );
100: end

```

---

Поређења ради, код VNS-CAI методе се током процеса локалног претраживања сваки пут, када се пронађе ново допустиво решење, прво ажурира вредност најбољег допустивог решења, затим се вредност текућег решења *квари* отпуштањем једне по једне произвољно изабране легије све док текуће решење не постане недопустиво, а потом се, када текуће решење постане недопустиво, процес локалног претраживања примењује испочетка над ажурираним текућим решењем. На овај начин се, процесом локалног претраживања, допустиво решење најмање тежине тражи докле год је то могуће. Процес локалног претраживања се прекида ако се претраживањем  $\mathcal{N}_1$  и  $\mathcal{N}_2$  околина текућег решења није пронашло квалитетније решење, односно ако се није пронашло решење истог квалитета и ако пронађено решење истог квалитета није постало текуће са одређеном вероватноћом.

Улазни подаци модификоване VNS методе су најмањи ( $k_{min}$ ), највећи ( $k_{max}$ ) ред околине, корак  $k$  ( $k_{step}$ ), максимални број итерација ( $brIter_{max}$ ), вероватноћа преласка у решења истог квалитета ( $p$ ) изражена у процентима и максимално време извршавања кода ( $t_{max}$ ) изражено у секундама.

## Резултати тестирања модификоване VNS методе за решавање $\mathcal{RD}$ проблема

Резултати тестирања модификоване VNS методе за решавање  $\mathcal{RD}$  проблема изложени су у оквиру овог поглавља. Са циљем да се изврши детаљно поређење модификоване VNS и VNS-CAI методе, прво је извршено тестирање модификоване VNS методе на истом скупу инстанци на ком је VNS-CAI метода тестирана у [72], а затим су обе методе додатно

тестиране на инстанцама чија решења, до сада, у литератури нису позната.

Слично као што је то био случај са VNS-CAI методом, алгоритам модификоване VNS методе такође не може да гарантује да је добијено решење глобални оптимум због своје недетерминистичке природе. Имајући у виду да се случајност по којој модификована VNS метода тражи почетно и размрдава текућа решења може унапред дефинисати семеном (енг. seed), односно да поступак решавања проблема зависи од генератора псеудослучајних бројева, свака инстанца тестирана је са по 20 међусобно различитих семена, тј. тестирана је по 20 пута. Инстанце су тестиране коришћењем 20 међусобно различитих семена како би добили најбоље решење. Конкретно, као брз начин решавања VNS омогућава већи број тестирања а самим тим и одабир најбољег. Како би се избегло приказивање појединачних резултата, резултати тестирања су обједињени за сваку инстанцу посебно и приказани у наставку.

Резултати тестирања модификоване VNS методе за  $\mathcal{RD}$  проблем на истом скупу инстанци на ком је VNS-CAI метода тестирана у [72] приказани су у Табелама 12-13, док су резултати додатног тестирања обе методе приказани у Табелама 14-15. Табеле 12-13 организоване су на следећи начин. У прве три колоне (Име,  $|V|$  и  $|E|$ ) приказани су име, број грана и број чворова одговарајуће инстанце. Потом следи колона у којој је приказано оптимално решење проблема добијено у поглављу 3 (колона је означена са *opt*). Најмање време за које су оптимизациони решавачи решили дату инстанцу и доказали оптималност добијеног решења налази се у потколони *t* колоне *Cpl/Gur*. У наредне четири колоне (*sol*, *t*, *err* и  $\sigma$ ) приказани су резултати тестирања VNS-CAI алгоритма, док последњих пет колона (*sol*, *t*, *t<sub>tot</sub>*, *err* и  $\sigma$ ) садрже резултате тестирања модификованог VNS алгоритма након што је свака инстанца тестирана са по 20 међусобно различитих семена. Вредности које су записане у потколони *sol*, *t*, *err* и  $\sigma$  колони VNS-CAI и модификовани VNS формиране су на исти начин код обе методе. Најбоље решење добијено након 20 међусобно различитих тестирања приказано је у колони *sol* на следећи начин: ако је достигнуто решење једнако оптималном (из колоне *opt*), у колони *sol* стоји ознака *opt*, међутим, ако достигнуто решење није једнако оптималном, приказано је најбоље достигнуто решење. Слично, након што је свака инстанца тестирана са по 20 различитих семена, најкраће време



потребно да се први пут достигне решење из колоне *sol* приказано је у колони *t*, док је максимално време извршавања кода (најдуже време потребно да описани код реши задату инстанцу за задато семе) приказано у колони  $t_{tot}$ . Дакле, у потколони *t* колоне *Cpl/Gur* приказано је најмање време потребно да се дата инстанца реши коришћењем оптимизационих решавача, док је у потколони  $t_{tot}$  колоне *модификовани VNS* приказано време решавања дате инстанце у најгорем случају. Пошто се у [72] за потребе тестирања користио рачунар истих перформанси, резултати у потколони *sol* и *t* колоне *VNS-CAI* су једноставно преписани из тог рада, док је потколони  $t_{tot}$  изостављена будући да ти подаци нису публиковани. Код обе методе, потколоне *err* и  $\sigma$  садрже информације о квалитету добијеног решења након извршених 20 међусобно различитих тестирања. Обележавајући *VNS*-ово решење проблема добијено у *i*-том тестирању са  $VNS_i$ , релативно одступање од најбољег добијеног решења *sol* са  $err_i = \frac{|VNS_i - sol|}{|sol|}$  ( $i = 1, \dots, 20$ ), у колони *err* приказано је просечно релативно одступање од најбољег решења одређено формулом  $err = \frac{1}{20} \sum_{i=1}^{20} err_i$ , док је стандардно одступање  $\sigma$  од најбољег решења одређено формулом  $\sigma = \sqrt{\frac{1}{20} \sum_{i=1}^{20} (err_i - err)^2}$ . Вредности *err* и  $\sigma$ , приказане у табели, заокружене су на четири децимале<sup>15</sup>.

Табеле 14 и 15 организоване су слично Табелама 12 и 13 с тим да је колона *opt* замењена колоном *best* будући да су у Табелама 14-15 углавном приказане инстанце код којих оптимизациони решавачи нису успели да докажу оптималност пронађених вредности до момента прекида програма, док су потколоне *t* и *val* колоне *Cpl/Gur* груписане у потколону  $val(t)$ . Колона *best* садржи најбоље познато решење за ту инстанцу (достигнуто применом оптимизационих решавача и/или модификованом *VNS* методом и/или *VNS-CAI* методом). У потколони  $val(t)$  приказано је најбоље решење добијено применом оптимизационих решавача у односу на све математичке формулације, на следећи начин: уколико оптимизациони решавачи нису могли да реше дату инстанцу до прекида програма (достигнуто је временско ограничење од 7200 секунди или је достигнут "out of memory" статус) у колони  $val(t)$ , поред вредности које су оптимизаци-

<sup>15</sup>За имплементацију модификоване *VNS* методе коришћени су следећи параметри  $k_{min} = 1$ ,  $k_{max} = 20$ ,  $k_{step} = 1$ ,  $brIter_{max} = 500$ ,  $t_{max} = 7200$  и  $p = 5$ , док су за имплементацију *VNS-CAI* методе коришћени нешто другачији параметри:  $k_{min} = 1$ ,  $k_{max} = 30$ ,  $k_{step} = 1$ ,  $t_{max} = 7200$  и  $p = 0.5$ . Разлика у коришћеним параметрима је ненамерна и без значајног утицаја на крајње резултате а примећена тек након што су све инстанце тестиране.

циони решавачи приказали пре прекидања програма, приказана је ознака "(7200/нм)" (скраћено од "достигнуто је време од 7200 секунди или не може да се добије решење"), иначе су приказане вредности које су оптимizacionи решавачи пронашли и времена решавања тих инстанци. У случајевима када је решавање програма прекинуто пре временског ограничења од 7200 секунди, а без поруке којом је објашњен разлог прекида, уместо времена решавања стоји ознака "-". Слично, ако ни једна вредност дате инстанце није добијена пре прекида програма, уместо вредности инстанце приказана је ознака "-".

Ради боље прегледности за све четири табеле важи следеће: код инстанци које су оптимално решене, а код којих се применом VNS-CAI и/или применом модификованог VNS алгоритма није пронашло решење из колоне *opt*, пронађено решење је подвучено; код инстанци код којих није познато оптимално али је познато неко најбоље решење, пронађено решење је подебљано уколико је једнако најбољем познатом решењу, односно подвучено уколико се разликује од најбољег познатог решења. Инстанце су поређане по класи којој припадају, затим по броју чворова и на крају по броју грана. С тим у вези, Табела 12 садржи резултате тестирања 133 инстанце-grid типа и резултате тестирања 7 инстанци планарног типа, док Табела 13 садржи резултате тестирања 87 инстанци произвољно генерисаног типа и резултате тестирања 4 инстанце-net типа. Као што је раније напоменуто резултати приказани у Табелама 12 и 13 су преузети из [72] и проширени резултатима тестирања модификоване VNS методе. Додатна тестирања извршена су за 54 инстанце произвољно генерисаног типа и за 5 инстанци планарног типа (пошто су у [72] приказани резултати тестирања планарних инстанци са највише 200 чворова, у оквиру Табеле 12 приказани су резултати тестирања само тих инстанци, док су инстанце са 250, 300, 400, 500, 600 и 700 чворова први пут тестиране). Све додатно тестиране инстанце први пут су тестиране како од стране оптимizacionих решавача, тако и од стране VNS-CAI и модификоване VNS методе. Ради боље прегледности и резултати додатних тестирања подељени су у две табеле. У Табели 14 приказани су резултати тестирања 6 инстанци планарног и 25 инстанци произвољно генерисаног типа, а у Табели 15 резултати тестирања 29 инстанци произвољно генерисаног типа. Све четири Табеле приказане су у *landscape* формату због своје ширине, односно због већег броја колона. С тим у вези, детаљни опис и сумарни преглед сваке табеле дат је непосредно испред или након

сваке табеле.

У наставку је приказана Табела 12. Као што се из приказаних резултата може видети, оптимизациони решавачи доказали су оптималност пронађеног решења за све тестиране инстанце изузетно брзо (време решавања већег броја инстанци мање је од 1 секунде). Другим речима, оптимизациони решавачи решавају ове инстанце са таквом лакоћом да се намеће утисак да ни VNS-CAI, а ни модификовани VNS алгоритам нису ни намењени њиховом решавању. Ипак, иако не могу да парирају оптимизационим решавачима по брзини решавања овог типа инстанци, обе методе су биле доста успешне у налажењу решења која су једнака оптималним. Применом VNS-CAI алгоритма вредности из колоне *opt* пронађене су код 128 инстанци (код преосталих 5 инстанци пронађене су вредности код којих релативне грешке у односу на вредности из колоне *opt* приближно износе 1.39%, 1.2%, 1.2%, 5.11% и 10%, тим редом). При том, пошто су све инстанце тестиране са по 20 међусобно различитих семена, одступања од најбоље пронађене вредности, односно вредности потколоне *err* и  $\sigma$  (колоне VNS-CAI) нису веће од 5.8% и 3.1% (вредности потколоне *err* су код 26 инстанци једнаке нули, код 74 инстанци из сегмента  $(0, 3\%]$ , код 13 инстанци из сегмента  $(3\%, 4\%]$ , а код преосталих 20 инстанци из сегмента  $(4\%, 5.8\%)$ ). Применом модификованог VNS алгоритма вредности једнаке оптималним пронађене су за 127 инстанци (код преосталих 6 инстанци пронађене су вредности код којих релативне грешке у односу на вредности из колоне *opt* приближно износе 1.3%, 1.3%, 1.2%, 1.1%, 3.98% и 10.77%, тим редом). Када је квалитет пронађених вредности у питању, може се закључити да су вредности пронађене применом модификоване VNS методе бољег квалитета у односу на вредности добијене применом VNS-CAI методе будући да су вредности потколоне *err* и  $\sigma$  модификоване VNS методе једнаке нули код 53 инстанце, из сегмента  $(0, 3\%]$  код 65 инстанци, за 13 инстанци из сегмента  $(3\%, 4\%]$  и да су код само две инстанце из сегмента  $(4, 6.1\%)$ . Дакле, применом модификоване VNS методе решење једнако оптималном пронађено је за једну инстанцу мање у односу на VNS-CAI методу али је зато за преостале инстанце модификована VNS метода чешће долазила до вредности које су једнаке оптималним у односу на VNS-CAI методу.

Оптимизациони решавачи брзо долазе до решења и код инстанци планарног типа. Заправо, код инстанци са највише 100 чворова време ре-

шавања мање је од 1 секунде, док је код инстанци са 100 и 150 чворова време решавања једнако 1.035 и 77.613 секунди (тим редом). Са друге стране, иако је укупно време решавања инстанци планарног типа применом модификоване VNS методе знатно дуже у односу на време решавања оптимизационих решавача, модификована VNS метода проналази вредности из колоне *opt* за мање од 1 секунде (вредности у потколони *t* су мање од 1 секунде). До најбољих познатих вредности брзо се долази и VNS-CAI методом (вредности потколоне *t* су за инстанце са највише 50 чворова мање од 1 секунде, а за инстанце са 150 и 200 чворова једнаке 1.166 и 2.466 секунди, тим редом). Додатно, обе VNS методе вредности из колоне *opt* пронашле су за сваку планарну инстанцу и за свако семе указујући на квалитет који се добија при решавању овог типа инстанци.

Након Табеле 12 приказана је Табела 13 која садржи резултате тестирања свих инстанци произвољно генерисаног типа са 50, 100 и 150 чворова и резултате тестирања само одређених инстанци произвољно генерисаног типа са 200, 250 и 300 чворова. Изостављене инстанце са 200, 250 и 300 чворова нису тестиране из истог разлога због ког резултати њиховог тестирања нису публиковани у [72], односно зато што нису сви оптимизациони решавачи успели да докажу оптималност пронађеног решења у року од 7200 секунди. Будући да се скуп тестираних инстанци састоји из 87 инстанци, изостављене инстанце нису тестиране ни овом приликом. Применом VNS-CAI методе вредности из колоне *opt* пронађене су за 80 а применом модификованог VNS алгоритма за 82 од 87 тестираних инстанци произвољно генерисаног типа. Код инстанци код којих нису пронађене вредности једнаке оптималним, релативна одступања пронађених вредности од оптималних углавном нису већа од 1% (релативна одступања од вредности из колоне *opt* мања су од 1% код шест инстанци и приближно једнака 7.7% код једне инстанце када су тестирања извршена применом VNS-CAI методе, односно мање су од 1% код шест инстанци и приближно једнака 1.5% код једне инстанце када су тестирања извршена применом модификоване VNS методе). Када је квалитет пронађених вредности у питању, лако се може приметити да су решења добијена применом модификоване VNS методе бољег квалитета у односу на решења добијена VNS-CAI методом: модификованом VNS методом вредности у потколони *err* и  $\sigma$  нису веће од 6.2% и 2.8% (тим редом), а једнаке су нули код 62 инстанце, док вредности у потколони *err* и  $\sigma$  колоне VNS-CAI нису веће од 8% и 9.8% (тим редом) и једнаке су

нули код 50 инстанци.

Резултати тестирања инстанци нет типа приказани су у наставку Табеле 13. Иако оптимизациони решавачи и овај тип инстанци решавају са лакоћом (време решавања сваке појединачне инстанце мање је од 1 секунде), све четири инстанце тестиране су и применом обе VNS методе. Обе методе пронашле су вредности из колоне *opt* за инстанце са 100, 200 и 400 чворова. VNS-CAI метода брже је дошла до најбољег решења за инстанцу *Het-10-10*, док је модификована VNS метода брже и квалитетније (мање вредности у колонама *err* и  $\sigma$ ) пронашла најбоље решење за *Het-10-20* и *Het-20-20*. Са друге стране, за краће време решавања применом VNS-CAI методе пронађена је мања вредност за инстанцу *Het-30-20* у односу на вредност коју је пронашла модификована VNS метода.

Табела 12: Резултати тестирања модификоване VNS методе на оптимално решеним GRID и планарним инстанцама

Инстанце				Cpl/Gur	VNS-CAI				модификовани VNS				
Име	V	E	opt	t	sol	t	err	$\sigma$	sol	t	t <sub>tot</sub>	err	$\sigma$
GRID04×10	40	66	20	0.040	onm	0.010	0	0	onm	0.079	22.439	0	0
GRID05×08	40	67	21	0.010	onm	<0.01	0	0	onm	0.030	21.942	0	0
GRID08×05	40	67	21	0.050	onm	<0.01	0	0	onm	<0.01	22.229	0	0
GRID10×04	40	66	20	0.040	onm	0.013	0	0	onm	0.016	22.535	0	0
GRID03×14	42	67	22	0.020	onm	<0.01	0	0	onm	<0.01	16.302	0	0
GRID06×07	42	71	22	0.050	onm	<0.01	0	0	onm	0.014	26.939	0	0
GRID07×06	42	71	22	0.030	onm	<0.01	0	0	onm	<0.01	22.354	0	0
GRID14×03	42	67	22	0.010	onm	0.031	0	0	onm	<0.01	19.331	0	0
GRID04×11	44	73	22	0.050	onm	<0.01	0	0	onm	0.020	27.231	0	0
GRID11×04	44	73	22	0.020	onm	<0.01	0	0	onm	<0.01	28.420	0	0
GRID03×15	45	72	24	0.040	onm	0.012	0	0	onm	<0.01	30.282	0	0
GRID05×09	45	76	23	0.060	onm	0.004	0	0	onm	0.073	27.581	0	0
GRID09×05	45	76	23	0.070	onm	0.013	0	0	onm	<0.01	26.624	0	0
GRID15×03	45	72	24	0.020	onm	0.022	0	0	onm	0.061	26.819	0	0
GRID04×12	48	80	24	0.040	onm	0.034	0	0	onm	0.119	31.796	0	0
GRID06×08	48	82	24	0.050	onm	0.033	0.0042	0.0125	onm	0.157	30.479	0	0
GRID08×06	48	82	24	0.060	onm	0.012	0.0042	0.0125	onm	0.029	26.757	0	0
GRID12×04	48	80	24	0.030	onm	0.019	0	0	onm	0.079	37.189	0	0
GRID07×07	49	84	24	0.060	onm	0.029	0.0063	0.0149	onm	0.565	189.890	0	0
GRID05×10	50	85	26	0.080	onm	<0.01	0	0	onm	0.026	43.754	0	0
GRID10×05	50	85	26	0.070	onm	<0.01	0	0	onm	0.028	43.211	0	0
GRID04×13	52	87	26	0.060	onm	0.104	0	0	onm	0.087	45.364	0	0
GRID13×04	52	87	26	0.050	onm	0.017	0.0019	0.0084	onm	0.122	46.852	0	0
GRID06×09	54	93	27	0.080	onm	0.436	0.0148	0.0181	onm	0.095	42.162	0	0
GRID09×06	54	93	27	0.070	onm	<0.01	0.0074	0.0148	onm	0.495	40.612	0	0
GRID05×11	55	94	28	0.080	onm	0.013	0	0	onm	0.241	53.295	0	0
GRID11×05	55	94	28	0.080	onm	<0.01	0	0	onm	0.150	60.797	0	0
GRID04×14	56	94	28	0.040	onm	0.035	0.0018	0.0078	onm	<0.01	43.542	0	0
GRID07×08	56	97	28	0.080	onm	<0.01	0	0	onm	0.027	51.047	0	0
GRID08×07	56	97	28	0.080	onm	0.033	0	0	onm	0.101	60.362	0	0
GRID14×04	56	94	28	0.020	onm	0.438	0.0370	0.0117	onm	0.020	54.633	0	0
GRID04×15	60	101	30	0.050	onm	<0.01	0.0017	0.0073	onm	0.137	61.743	0	0

Табела 12: Резултати тестирања модификоване VNS методе на оптимално решеним GRID и планарним инстанцама

Инстанце				Cpl/Gur <i>t</i>	VNS-CAI				модификовани VNS				
Име	V	E	<i>opt</i>		<i>sol</i>	<i>t</i>	<i>err</i>	$\sigma$	<i>sol</i>	<i>t</i>	<i>t<sub>tot</sub></i>	<i>err</i>	$\sigma$
GRID05×12	60	103	30	0.110	<i>opt</i>	0.036	0.0200	0.0163	<i>opt</i>	0.150	221.516	0.0017	0.0073
GRID06×10	60	104	30	0.070	<i>opt</i>	0.041	0.0050	0.0119	<i>opt</i>	0.338	70.810	0	0
GRID10×06	60	104	30	0.090	<i>opt</i>	0.163	0.0100	0.0153	<i>opt</i>	0.253	212.034	0	0
GRID12×05	60	103	30	0.080	<i>opt</i>	0.078	0.0133	0.0163	<i>opt</i>	1.234	264.579	0.0017	0.0073
GRID15×04	60	101	30	0.060	<i>opt</i>	0.040	0	0	<i>opt</i>	0.409	60.832	0	0
GRID07×09	63	110	31	0.066	<i>opt</i>	0.135	0.0097	0.0148	<i>opt</i>	0.475	301.590	0	0
GRID09×07	63	110	31	0.100	<i>opt</i>	0.082	0	0	<i>opt</i>	1.000	74.687	0	0
GRID08×08	64	112	32	0.110	<i>opt</i>	0.031	0.0016	0.0068	<i>opt</i>	0.013	81.975	0	0
GRID05×13	65	112	33	0.110	<i>opt</i>	0.171	0.0030	0.0091	<i>opt</i>	0.195	84.133	0	0
GRID13×05	65	112	33	0.120	<i>opt</i>	0.054	0.0045	0.0108	<i>opt</i>	0.102	70.006	0	0
GRID06×11	66	115	33	0.100	<i>opt</i>	0.045	0.0061	0.0121	<i>opt</i>	0.332	88.966	0	0
GRID11×06	66	115	33	0.100	<i>opt</i>	0.246	0.0030	0.0091	<i>opt</i>	0.148	92.406	0	0
GRID05×14	70	121	35	0.120	<i>opt</i>	0.264	0.0086	0.0131	<i>opt</i>	1.950	352.061	0.0029	0.0086
GRID07×10	70	123	34	0.110	<i>opt</i>	0.164	0.0176	0.0144	<i>opt</i>	1.379	441.171	0.0103	0.0140
GRID10×07	70	123	34	0.080	<i>opt</i>	0.699	0.0176	0.0171	<i>opt</i>	1.471	452.294	0.0029	0.0088
GRID14×05	70	121	35	0.110	<i>opt</i>	0.190	0.0086	0.0131	<i>opt</i>	0.673	346.826	0	0
GRID06×12	72	126	36	0.090	<i>opt</i>	0.198	0.0056	0.0111	<i>opt</i>	0.512	106.656	0	0
GRID08×09	72	127	35	0.090	<i>opt</i>	0.017	0.0071	0.0124	<i>opt</i>	0.928	381.007	0.0014	0.0062
GRID09×08	72	127	35	0.100	<i>opt</i>	0.037	0.0114	0.0190	<i>opt</i>	1.157	364.936	0.0057	0.0114
GRID12×06	72	126	36	0.110	<i>opt</i>	0.161	0.0014	0.0061	<i>opt</i>	0.380	115.703	0	0
GRID05×15	75	130	38	0.160	<i>opt</i>	0.352	0.0026	0.0079	<i>opt</i>	0.437	107.449	0	0
GRID15×05	75	130	38	0.160	<i>opt</i>	0.101	0	0	<i>opt</i>	0.630	105.887	0	0
GRID07×11	77	136	38	0.140	<i>opt</i>	0.094	0.0013	0.0057	<i>opt</i>	0.465	105.275	0	0
GRID11×07	77	136	38	0.120	<i>opt</i>	0.102	0.0026	0.0079	<i>opt</i>	0.618	114.540	0	0
GRID06×13	78	137	38	0.148	<i>opt</i>	1.553	0.0250	0.0155	<i>opt</i>	48.845	580.554	0.0197	0.0114
GRID13×06	78	137	38	0.200	<i>opt</i>	3.840	0.0263	0.0083	<i>opt</i>	2.214	543.290	0.0053	0.0105
GRID08×10	80	142	39	0.128	<i>opt</i>	0.132	0.0115	0.0128	<i>opt</i>	0.723	555.907	0	0
GRID10×08	80	142	39	0.142	<i>opt</i>	0.039	0.0077	0.0118	<i>opt</i>	1.350	673.813	0.0115	0.0128
GRID09×09	81	144	38	0.073	<i>opt</i>	2.937	0.0237	0.0248	<i>opt</i>	2.254	647.906	0.0053	0.0158
GRID06×14	84	148	41	0.134	<i>opt</i>	22.542	0.0317	0.0136	<i>opt</i>	1.770	847.340	0.0110	0.0121
GRID07×12	84	149	41	0.168	<i>opt</i>	1.727	0.0085	0.0116	<i>opt</i>	2.410	116.784	0	0
GRID12×07	84	149	41	0.180	<i>opt</i>	1.062	0.0024	0.0073	<i>opt</i>	1.388	505.451	0	0

Табела 12: Резултати тестирања модификоване VNS методе на оптимално решеним GRID и планарним инстанцама

Инстанце				Cpl/Gur <i>t</i>	VNS-CAI				модификовани VNS				
Име	V	E	<i>opt</i>		<i>sol</i>	<i>t</i>	<i>err</i>	$\sigma$	<i>sol</i>	<i>t</i>	<i>t<sub>tot</sub></i>	<i>err</i>	$\sigma$
GRID14x06	84	148	41	0.170	<i>onm</i>	7.766	0.0232	0.0121	<i>onm</i>	2.167	929.397	0.0134	0.0121
GRID08x11	88	157	42	0.192	<i>onm</i>	11.391	0.0286	0.0161	<i>onm</i>	98.913	810.672	0.0214	0.0071
GRID11x08	88	157	42	0.141	<i>onm</i>	0.778	0.0214	0.0198	<i>onm</i>	6.289	626.039	0.0060	0.0103
GRID06x15	90	159	44	0.240	<i>onm</i>	5.733	0.0193	0.0130	<i>onm</i>	7.713	970.296	0.0045	0.0091
GRID09x10	90	161	43	0.200	<i>onm</i>	1.224	0.0291	0.0193	<i>onm</i>	3.295	858.060	0.0116	0.0116
GRID10x09	90	161	43	0.190	<i>onm</i>	0.672	0.0105	0.0137	<i>onm</i>	3.126	675.473	0.0128	0.0116
GRID15x06	90	159	44	0.250	<i>onm</i>	3.141	0.0136	0.0111	<i>onm</i>	3.957	610.552	0.0080	0.0108
GRID07x13	91	162	44	0.178	<i>onm</i>	0.801	0.0182	0.0116	<i>onm</i>	1.622	873.542	0.0057	0.0098
GRID13x07	91	162	44	0.170	<i>onm</i>	0.882	0.0182	0.0136	<i>onm</i>	0.387	813.986	0.0034	0.0081
GRID08x12	96	172	46	0.210	<i>onm</i>	1.527	0.0185	0.0185	<i>onm</i>	9.441	1148.497	0.0185	0.0078
GRID12x08	96	172	46	0.191	<i>onm</i>	5.175	0.0163	0.0117	<i>onm</i>	9.360	828.634	0.0043	0.0087
GRID07x14	98	175	47	0.240	<i>onm</i>	1.621	0.0255	0.0159	<i>onm</i>	4.260	941.659	0.0053	0.0092
GRID14x07	98	175	47	0.214	<i>onm</i>	2.929	0.0202	0.0142	<i>onm</i>	12.148	1516.416	0.0149	0.0118
GRID09x11	99	178	47	0.194	<i>onm</i>	3.737	0.0128	0.0141	<i>onm</i>	17.767	991.515	0.0170	0.0085
GRID11x09	99	178	47	0.190	<i>onm</i>	4.522	0.0255	0.0197	<i>onm</i>	2.539	1176.165	0.0032	0.0101
GRID10x10	100	180	48	0.220	<i>onm</i>	0.199	0.0052	0.0090	<i>onm</i>	1.886	1487.900	0.0031	0.0074
GRID08x13	104	187	50	0.262	<i>onm</i>	0.274	0.0100	0.0134	<i>onm</i>	5.907	1376.838	0.0040	0.0080
GRID13x08	104	187	50	0.29	<i>onm</i>	9.993	0.0150	0.0140	<i>onm</i>	5.63	1000.012	0	0
GRID07x15	105	188	50	0.3	<i>onm</i>	20.739	0.0260	0.0128	<i>onm</i>	11.085	1617.688	0.0160	0.0102
GRID15x07	105	188	50	0.278	<i>onm</i>	22.274	0.0250	0.0107	<i>onm</i>	10.052	1302.618	0.0170	0.0095
GRID09x12	108	195	51	0.26	<i>onm</i>	9.665	0.0255	0.0216	<i>onm</i>	23.117	1347.883	0.0167	0.0128
GRID12x09	108	195	51	0.18	<i>onm</i>	22.530	0.0216	0.0195	<i>onm</i>	26.098	993.785	0.0137	0.0109
GRID10x11	110	199	52	0.26	<i>onm</i>	2.545	0.0192	0.0202	<i>onm</i>	1.268	1968.171	0.0038	0.0098
GRID11x10	110	199	52	0.256	<i>onm</i>	12.061	0.0231	0.0198	<i>onm</i>	16.247	1538.147	0.0019	0.0058
GRID08x14	112	202	53	0.289	<i>onm</i>	6.864	0.0208	0.0145	<i>onm</i>	9.975	1202.457	0.0057	0.0135
GRID14x08	112	202	53	0.284	<i>onm</i>	1.213	0.0236	0.0167	<i>onm</i>	38.356	1025.130	0.0179	0.0072
GRID09x13	117	212	55	0.232	<i>onm</i>	10.045	0.0273	0.0241	<i>onm</i>	25.155	1728.727	0.0118	0.0087
GRID13x09	117	212	55	0.26	<i>onm</i>	4.409	0.0273	0.0195	<i>onm</i>	11.592	1500.007	0.0145	0.0159
GRID08x15	120	217	57	0.37	<i>onm</i>	5.050	0.0158	0.0123	<i>onm</i>	45.153	1587.274	0.0184	0.0180
GRID10x12	120	218	56	0.236	<i>onm</i>	29.077	0.0402	0.0245	<i>onm</i>	1.333	1105.241	0.0018	0.0078
GRID12x10	120	218	56	0.326	<i>onm</i>	27.666	0.0357	0.0160	<i>onm</i>	48.974	1352.985	0.0196	0.0125
GRID15x08	120	217	57	0.38	<i>onm</i>	18.631	0.0167	0.0162	<i>onm</i>	28.137	2083.260	0.0088	0.0130



Табела 12: Резултати тестирања модификоване VNS методе на оптимално решеним GRID и планарним инстанцама

Инстанце				Cpl/Gur <i>t</i>	VNS-CAI				модификовани VNS				
Име	$ V $	$ E $	<i>opt</i>		<i>sol</i>	<i>t</i>	<i>err</i>	$\sigma$	<i>sol</i>	<i>t</i>	<i>t<sub>tot</sub></i>	<i>err</i>	$\sigma$
GRID11x11	121	220	57	0.35	<i>onm</i>	2.414	0.0228	0.0208	<i>onm</i>	9.468	2188.278	0.0114	0.0160
GRID09x14	126	229	58	0.25	<i>onm</i>	0.518	0.0422	0.0301	<i>onm</i>	2.221	2345.109	0.0250	0.0247
GRID14x09	126	229	58	0.25	<i>onm</i>	46.688	0.0483	0.0186	<i>onm</i>	35.751	2287.817	0.0310	0.0178
GRID10x13	130	237	61	0.48	<i>onm</i>	12.797	0.0180	0.0186	<i>onm</i>	15.502	3531.853	0.0148	0.0155
GRID13x10	130	237	61	0.49	<i>onm</i>	1.850	0.0197	0.0241	<i>onm</i>	16.887	4052.998	0.0090	0.0132
GRID11x12	132	241	62	0.453	<i>onm</i>	26.008	0.0258	0.0194	<i>onm</i>	55.151	2771.248	0.0153	0.0148
GRID12x11	132	241	62	0.464	<i>onm</i>	31.964	0.0210	0.0136	<i>onm</i>	11.391	2321.298	0.0056	0.0117
GRID09x15	135	246	63	0.52	<i>onm</i>	31.738	0.0262	0.0196	<i>onm</i>	35.077	2563.680	0.0246	0.0204
GRID15x09	135	246	63	0.41	<i>onm</i>	23.271	0.0325	0.0177	<i>onm</i>	30.500	2832.502	0.0190	0.0095
GRID10x14	140	256	65	0.478	<i>onm</i>	78.302	0.0354	0.0176	<i>onm</i>	189.140	4101.128	0.0308	0.0169
GRID14x10	140	256	65	0.432	<i>onm</i>	10.337	0.0377	0.0225	<i>onm</i>	457.098	2618.129	0.0285	0.0202
GRID11x13	143	262	66	0.463	<i>onm</i>	70.571	0.0318	0.0239	<i>onm</i>	1868.076	3809.333	0.0295	0.0200
GRID13x11	143	262	66	0.5	<i>onm</i>	21.158	0.0394	0.0273	<i>onm</i>	478.621	2811.693	0.0303	0.0209
GRID12x12	144	264	67	0.5	<i>onm</i>	36.922	0.0366	0.0197	<i>onm</i>	38.508	3402.836	0.0172	0.0207
GRID10x15	150	275	70	0.715	<i>onm</i>	126.053	0.0279	0.0237	<i>onm</i>	37.072	3785.110	0.0157	0.0168
GRID15x10	150	275	70	0.951	<i>onm</i>	24.143	0.0314	0.0200	<i>onm</i>	55.187	3548.597	0.0057	0.0153
GRID11x14	154	283	71	0.483	<i>onm</i>	59.802	0.0465	0.0268	<i>onm</i>	1540.949	4178.963	0.0359	0.0169
GRID14x11	154	283	71	0.67	<i>onm</i>	62.236	0.0401	0.0219	<i>onm</i>	202.497	4696.985	0.0303	0.0143
GRID12x13	156	287	72	0.715	<u>73</u>	115.106	0.0240	0.0167	<i>onm</i>	277.667	3871.311	0.0375	0.0220
GRID13x12	156	287	72	0.783	<i>onm</i>	62.928	0.0403	0.0181	<i>onm</i>	115.678	3385.030	0.0326	0.0242
GRID11x15	165	304	76	0.77	<i>onm</i>	117.803	0.0513	0.0216	<i>onm</i>	2753.136	4068.059	0.0230	0.0131
GRID15x11	165	304	76	0.918	<i>onm</i>	52.315	0.0428	0.0208	<i>onm</i>	326.494	5545.898	0.0342	0.0178
GRID12x14	168	310	77	0.614	<i>onm</i>	181.880	0.0409	0.0222	<u>78</u>	192.454	5535.127	0.0218	0.0182
GRID14x12	168	310	77	0.72	<i>onm</i>	155.635	0.0448	0.0241	<u>78</u>	172.758	5743.454	0.0333	0.0169
GRID13x13	169	312	78	0.77	<i>onm</i>	68.571	0.0340	0.0204	<i>onm</i>	125.798	3736.695	0.0607	0.0265
GRID12x15	180	333	82	0.77	<u>83</u>	164.384	0.0380	0.0206	<i>onm</i>	256.902	7048.402	0.0238	0.0161
GRID15x12	180	333	82	0.8	<u>83</u>	130.710	0.0464	0.0223	<u>83</u>	309.631	4226.957	0.0295	0.0200
GRID13x14	182	337	83	0.777	<i>onm</i>	75.472	0.0518	0.0278	<i>onm</i>	220.737	5282.061	0.0175	0.0155
GRID14x13	182	337	83	0.776	<i>onm</i>	201.980	0.0470	0.0295	<i>onm</i>	486.727	6843.241	0.0361	0.0216
GRID13x15	195	362	89	1.73	<i>onm</i>	407.358	0.0517	0.0306	<i>onm</i>	67.051	7058.387	0.0197	0.0201
GRID15x13	195	362	89	1.309	<i>onm</i>	139.451	0.0489	0.0259	<i>onm</i>	309.240	4926.696	0.0264	0.0196
GRID14x14	196	364	88	0.67	<i>onm</i>	353.878	0.0551	0.0279	<i>onm</i>	301.461	7200.045	0.0347	0.0277

Табела 12: Резултати тестирања модификоване VNS методе на оптимално решеним GRID и планарним инстанцама

Инстанце				Cpl/Gur <i>t</i>	VNS-CAI				модификовани VNS				
Име	$ V $	$ E $	<i>opt</i>		<i>sol</i>	<i>t</i>	<i>err</i>	$\sigma$	<i>sol</i>	<i>t</i>	$t_{tot}$	<i>err</i>	$\sigma$
GRID14×15	210	391	95	1.198	<i>onm</i>	282.147	0.0542	0.0273	<u>96</u>	2255.024	7200.023	0.0370	0.0196
GRID15×14	210	391	95	1.159	<i>onm</i>	92.424	0.0579	0.0222	<i>onm</i>	527.097	5372.785	0.0237	0.0199
GRID15×15	225	420	102	1.357	<i>onm</i>	697.859	0.0574	0.0265	<i>onm</i>	876.249	7200.328	0.0402	0.0201
GRID20×20	400	760	176	37.579	<u>185</u>	676.713	0.0408	0.0145	<u>183</u>	4495.196	7200.254	0.0380	0.0166
GRID30×20	600	1150	260	1279.438	<u>286</u>	5114.624	0.0344	0.0171	<u>288</u>	6510.623	7200.903	0.0293	0.0172
план10	10	27	3	0.01	<i>onm</i>	<0.01	0	0	<i>onm</i>	<0.01	0.795	0	0
план20	20	105	5	0.01	<i>onm</i>	<0.01	0	0	<i>onm</i>	<0.01	21.852	0	0
план30	30	182	5	0.02	<i>onm</i>	<0.01	0	0	<i>onm</i>	<0.01	8.1040	0	0
план50	50	465	6	0.01	<i>onm</i>	<0.01	0	0	<i>onm</i>	<0.01	46.547	0	0
план100	100	1540	10	0.284	<i>onm</i>	0.054	0	0	<i>onm</i>	0.084	1667.590	0	0
план150	150	2867	12	1.035	<i>onm</i>	1.166	0	0	<i>onm</i>	0.170	3189.191	0	0
план200	200	4475	16	77.613	<i>onm</i>	2.466	0	0	<i>onm</i>	0.718	7200.037	0	0

Табела 13: Резултати тестирања модификоване VNS методе на оптимално решеним инстанцама произвољно генерисаног и нет типа

Инстанце				Cpl/Gur <i>t</i>	VNS-CAI				модификовани VNS				
Име	$ V $	$ E $	<i>opt</i>		<i>sol</i>	<i>t</i>	<i>err</i>	$\sigma$	<i>sol</i>	<i>t</i>	$t_{tot}$	<i>err</i>	$\sigma$
Рандом-50-1	50	49	32	<0.01	<i>onm</i>	0.031	0	0	<i>onm</i>	0.142	29.289	0	0
Рандом-50-2	50	49	33	<0.01	<i>onm</i>	0.069	0	0	<i>onm</i>	0.016	34.199	0	0
Рандом-50-3	50	58	28	<0.01	<i>onm</i>	0.029	0	0	<i>onm</i>	0.092	31.373	0	0
Рандом-50-4	50	54	30	<0.01	<i>onm</i>	<0.01	0	0	<i>onm</i>	0.031	28.885	0	0
Рандом-50-5	50	67	28	0.020	<i>onm</i>	<0.01	0	0	<i>onm</i>	0.052	25.339	0	0
Рандом-50-6	50	86	25	0.020	<i>onm</i>	0.041	0	0	<i>onm</i>	0.077	35.325	0	0
Рандом-50-7	50	84	26	0.020	<i>onm</i>	<0.01	0	0	<i>onm</i>	<0.01	32.645	0	0
Рандом-50-8	50	95	23	0.010	<i>onm</i>	<0.01	0	0	<i>onm</i>	0.105	33.160	0	0
Рандом-50-9	50	108	23	0.040	<i>onm</i>	0.011	0	0	<i>onm</i>	0.068	35.158	0	0
Рандом-50-10	50	112	22	0.030	<i>onm</i>	0.021	0	0	<i>onm</i>	<0.01	35.077	0	0
Рандом-50-20	50	248	12	0.080	<i>onm</i>	<0.01	0	0	<i>onm</i>	<0.01	31.166	0	0
Рандом-50-30	50	373	9	0.080	<i>onm</i>	<0.01	0	0	<i>onm</i>	<0.01	51.004	0	0

Табела 13: Резултати тестирања модификоване VNS методе на оптимално решеним инстанцама произвољно генерисаног и нет типа

Инстанце				Cpl/Gur <i>t</i>	VNS-CAI				модификовани VNS				
Име	$ V $	$ E $	<i>opt</i>		<i>sol</i>	<i>t</i>	<i>err</i>	$\sigma$	<i>sol</i>	<i>t</i>	<i>t<sub>tot</sub></i>	<i>err</i>	$\sigma$
Рандом-50-40	50	475	8	0.153	<i>onm</i>	<0.01	0	0	<i>onm</i>	<0.01	37.882	0	0
Рандом-50-50	50	597	6	0.100	<i>onm</i>	<0.01	0	0	<i>onm</i>	<0.01	28.000	0	0
Рандом-50-60	50	739	4	0.010	<i>onm</i>	<0.01	0	0	<i>onm</i>	<0.01	3.846	0	0
Рандом-50-70	50	860	4	0.020	<i>onm</i>	<0.01	0	0	<i>onm</i>	<0.01	22.168	0	0
Рандом-50-80	50	980	4	0.020	<i>onm</i>	<0.01	0	0	<i>onm</i>	<0.01	5.763	0	0
Рандом-50-90	50	1103	3	0.020	<i>onm</i>	<0.01	0	0	<i>onm</i>	<0.01	5.709	0	0
Рандом-100-1	100	100	61	0.010	<i>onm</i>	4.662	0.0057	0.0094	<i>onm</i>	1.355	227.281	0	0
Рандом-100-2	100	109	59	0.010	<i>onm</i>	2.744	0.0059	0.0097	<i>onm</i>	2.587	187.730	0	0
Рандом-100-3	100	181	48	0.050	<i>onm</i>	3.767	0.0146	0.0116	<i>onm</i>	3.636	1271.030	0.0125	0.0102
Рандом-100-4	100	206	45	0.110	<i>onm</i>	0.895	0.0189	0.0106	<i>onm</i>	3.823	1177.880	0.0200	0.0067
Рандом-100-5	100	231	39	0.050	<i>onm</i>	3.425	0.0256	0.0269	<i>onm</i>	11.788	828.246	0.0064	0.0197
Рандом-100-6	100	321	34	0.230	<i>onm</i>	3.572	0.0162	0.0146	<i>onm</i>	12.033	991.050	0.0118	0.0144
Рандом-100-7	100	317	32	0.170	<i>onm</i>	3.291	0.0156	0.0156	<i>onm</i>	4.657	1389.730	0.0031	0.0094
Рандом-100-8	100	398	29	0.234	<i>onm</i>	0.669	0.0017	0.0075	<i>onm</i>	0.645	340.124	0	0
Рандом-100-9	100	430	27	0.344	<i>onm</i>	0.389	0	0	<i>onm</i>	1.279	354.274	0	0
Рандом-100-10	100	498	24	0.564	<i>onm</i>	3.95	0.0167	0.0204	<i>onm</i>	0.514	1091.677	0	0
Рандом-100-20	100	981	14	0.417	<i>onm</i>	0.086	0	0	<i>onm</i>	0.041	449.634	0	0
Рандом-100-30	100	1477	11	1.304	<i>onm</i>	0.137	0.0091	0.0273	<i>onm</i>	0.395	260.612	0	0
Рандом-100-40	100	1945	8	0.601	<i>onm</i>	0.052	0	0	<i>onm</i>	1.707	152.198	0	0
Рандом-100-50	100	2483	7	0.540	<i>onm</i>	0.049	0.0214	0.0510	<i>onm</i>	0.353	265.367	0	0
Рандом-100-60	100	2985	6	0.439	<i>onm</i>	<0.01	0	0	<i>onm</i>	0.010	166.618	0	0
Рандом-100-70	100	3435	5	0.372	<i>onm</i>	0.044	0	0	<i>onm</i>	0.069	181.945	0	0
Рандом-100-80	100	3935	4	0.100	<i>onm</i>	<0.01	0	0	<i>onm</i>	0.013	34.696	0	0
Рандом-100-90	100	4446	4	0.060	<i>onm</i>	<0.01	0	0	<i>onm</i>	<0.01	224.661	0	0
Рандом-150-1	150	157	94	0.010	<i>onm</i>	22.389	0.0011	0.0032	<i>onm</i>	16.493	780.895	0	0
Рандом-150-2	150	243	78	0.150	<i>onm</i>	234.872	0.0301	0.0158	<i>onm</i>	41.833	2849.191	0.0135	0.0131
Рандом-150-3	150	322	65	0.170	<i>onm</i>	67.784	0.0177	0.0170	<i>onm</i>	11.969	4802.393	0.0038	0.0067
Рандом-150-4	150	437	53	0.358	<i>onm</i>	30.304	0.0274	0.0163	<i>onm</i>	53.291	3420.011	0.0208	0.0157
Рандом-150-5	150	557	46	0.850	<i>onm</i>	2.293	0.0174	0.0147	<i>onm</i>	12.164	4318.266	0.0033	0.0078
Рандом-150-6	150	705	38	3.190	<i>onm</i>	19.279	0.0171	0.0172	<i>onm</i>	33.036	4316.529	0.0132	0.0132
Рандом-150-7	150	778	34	2.492	<i>onm</i>	0.462	0.0059	0.0118	<i>onm</i>	27.439	6203.474	0.0044	0.0105

Табела 13: Резултати тестирања модификоване VNS методе на оптимално решеним инстанцама произвољно генерисаног и нет типа

Инстанце				Cpl/Gur <i>t</i>	VNS-CAI				модификовани VNS				
Име	$ V $	$ E $	<i>opt</i>		<i>sol</i>	<i>t</i>	<i>err</i>	$\sigma$	<i>sol</i>	<i>t</i>	<i>t<sub>tot</sub></i>	<i>err</i>	$\sigma$
Рандом-150-8	150	906	31	6.240	<i>onm</i>	0.865	0	0	<i>onm</i>	4.124	1055.847	0	0
Рандом-150-9	150	965	30	8.940	<i>onm</i>	3.727	0.0067	0.0170	<i>onm</i>	4.158	5875.070	0	0
Рандом-150-10	150	1152	27	17.230	<i>onm</i>	3.128	0.0056	0.0132	<i>onm</i>	8.814	1344.600	0	0
Рандом-150-20	150	2228	16	11.120	<i>onm</i>	1.561	0	0	<i>onm</i>	0.615	1007.232	0	0
Рандом-150-30	150	3318	12	20.210	<i>onm</i>	0.383	0	0	<i>onm</i>	1.356	908.424	0	0
Рандом-150-40	150	4476	9	6.499	<i>onm</i>	0.409	0.0778	0.0509	<i>onm</i>	0.158	676.154	0	0
Рандом-150-50	150	5550	8	7.541	<i>onm</i>	0.014	0	0	<i>onm</i>	0.062	644.087	0	0
Рандом-150-60	150	6734	6	0.775	<i>onm</i>	0.012	0	0	<i>onm</i>	0.290	337.630	0	0
Рандом-150-70	150	7807	6	3.894	<i>onm</i>	0.015	0	0	<i>onm</i>	0.021	838.383	0	0
Рандом-150-80	150	8924	4	0.192	<i>onm</i>	0.019	0	0	<i>onm</i>	0.032	669.221	0	0
Рандом-150-90	150	10043	4	0.150	<i>onm</i>	0.017	0	0	<i>onm</i>	0.025	725.683	0	0
Рандом-200-1	200	229	116	0.010	<u>117</u>	173.552	0.0171	0.0124	<i>onm</i>	390.690	7084.473	0.0138	0.0084
Рандом-200-2	200	390	92	0.355	<u>93</u>	647.247	0.0306	0.0199	<i>onm</i>	1793.969	6824.363	0.0228	0.0197
Рандом-200-3	200	581	69	0.885	<i>onm</i>	507.393	0.0428	0.0281	<u>70</u>	120.810	7200.035	0.0164	0.0182
Рандом-200-4	200	737	60	5.220	<i>onm</i>	568.080	0.0458	0.0235	<i>onm</i>	947.988	7200.053	0.0292	0.0203
Рандом-200-5	200	1010	47	20.580	<i>onm</i>	41.339	0.0372	0.0232	<i>onm</i>	498.988	7200.036	0.0340	0.0217
Рандом-200-6	200	1180	42	88.870	<i>onm</i>	84.363	0.0560	0.0363	<i>onm</i>	2415.873	7200.054	0.0619	0.0218
Рандом-200-7	200	1453	36	75.960	<i>onm</i>	11.272	0.0097	0.0182	<i>onm</i>	91.686	7200.064	0.0056	0.0188
Рандом-200-30	200	5876	12	60.210	<i>onm</i>	9.478	0.0125	0.0397	<i>onm</i>	19.121	7200.009	0	0
Рандом-200-40	200	7907	10	60.885	<i>onm</i>	0.302	0	0	<i>onm</i>	0.574	6751.898	0	0
Рандом-200-50	200	9895	8	14.347	<i>onm</i>	0.248	0	0	<i>onm</i>	0.197	6262.112	0	0
Рандом-200-60	200	11971	6	2.098	<i>onm</i>	0.496	0	0	<i>onm</i>	1.312	6292.355	0	0
Рандом-200-70	200	14059	6	8.075	<i>onm</i>	0.025	0	0	<i>onm</i>	0.057	5300.497	0	0
Рандом-200-80	200	15918	4	0.367	<i>onm</i>	0.038	0	0	<i>onm</i>	0.089	3229.427	0	0
Рандом-200-90	200	17821	4	0.310	<i>onm</i>	0.030	0	0	<i>onm</i>	0.046	1200.033	0	0
Рандом-250-1	250	345	136	0.030	<u>137</u>	1111.594	0.0226	0.0136	<u>137</u>	2051.000	7200.130	0.0091	0.0069
Рандом-250-2	250	633	97	2.318	<u>99</u>	380.006	0.0318	0.0229	<u>98</u>	3852.956	7200.081	0.0173	0.0125
Рандом-250-3	250	956	73	175.521	<i>onm</i>	132.791	0.0322	0.0271	<i>onm</i>	2839.269	7200.158	0.0329	0.0170
Рандом-250-4	250	1194	62	1043.172	<i>onm</i>	148.167	0.0234	0.0231	<i>onm</i>	2215.667	7200.102	0.0218	0.0163
Рандом-250-30	250	9347	13	1018.686	<u>14</u>	1.005	0	0	<i>onm</i>	206.136	7200.117	0.0115	0.0275
Рандом-250-40	250	12500	10	182.286	<i>onm</i>	0.743	0	0	<i>onm</i>	7.929	7200.700	0	0

Табела 13: Резултати тестирања модификоване VNS методе на оптимално решеним инстанцама произвољно генерисаног и нет типа

Инстанце				Cpl/Gur <i>t</i>	VNS-CAI				модификовани VNS				
Име	$ V $	$ E $	<i>opt</i>		<i>sol</i>	<i>t</i>	<i>err</i>	$\sigma$	<i>sol</i>	<i>t</i>	<i>t<sub>tot</sub></i>	<i>err</i>	$\sigma$
Рандом-250-50	250	15605	8	34.325	<i>onm</i>	0.621	0	0	<i>onm</i>	0.334	7200.098	0	0
Рандом-250-60	250	18660	8	95.914	<i>onm</i>	0.037	0	0	<i>onm</i>	0.097	7200.178	0	0
Рандом-250-70	250	21741	6	12.263	<i>onm</i>	0.037	0	0	<i>onm</i>	0.047	7200.201	0	0
Рандом-250-80	250	24836	4	0.981	<i>onm</i>	0.465	0	0	<i>onm</i>	0.236	2398.816	0	0
Рандом-250-90	250	27974	4	0.486	<i>onm</i>	0.052	0	0	<i>onm</i>	0.049	7200.066	0	0
Рандом-300-1	300	481	145	0.106	<u>149</u>	2797.158	0.0228	0.0141	<u>148</u>	5036.068	7200.219	0.0186	0.0100
Рандом-300-2	300	876	103	80.454	<u>105</u>	1057.238	0.0414	0.0207	<u>105</u>	1087.834	7200.223	0.0262	0.0197
Рандом-300-40	300	17934	10	386.037	<i>onm</i>	3.232	0.0200	0.0510	<i>onm</i>	33.455	7200.207	0	0
Рандом-300-50	300	22520	8	106.509	<i>onm</i>	31.909	0	0	<i>onm</i>	90.232	7200.147	0	0
Рандом-300-60	300	26952	8	193.105	<i>onm</i>	0.069	0	0	<i>onm</i>	0.100	7200.184	0	0
Рандом-300-70	300	31390	6	23.656	<i>onm</i>	0.286	0	0	<i>onm</i>	0.150	7200.192	0	0
Рандом-300-80	300	35871	5	11.201	<i>onm</i>	1.725	0.0800	0.0980	<i>onm</i>	2.967	7200.072	0	0
Рандом-300-90	300	40412	4	0.66	<i>onm</i>	0.092	0	0	<i>onm</i>	0.084	7200.007	0	0
Нет-10-10	100	342	28	0.02	<i>onm</i>	0.129	0	0	<i>onm</i>	1.752	961.002	0	0
Нет-10-20	200	712	56	0.02	<i>onm</i>	18.013	0.0018	0.0054	<i>onm</i>	3.649	7200.043	0	0
Нет-20-20	400	1482	98	0.02	<i>onm</i>	944.944	0.0245	0.0345	<i>onm</i>	792.207	7200.243	0.0122	0.0198
Нет-30-20	600	2252	140	0.02	<u>145</u>	6916.392	0.0624	0.0305	<u>155</u>	6758.821	7200.823	0.0455	0.0207

Сумирајући резултате тестирања (Табеле 12 и 13) лако може да се уочи да је применом модификоване VNS методе решење једнако оптималном пронађено код 219, док је применом VNS-CAI методе решење једнако оптималном пронађено код 218 од 231 тестиране инстанце. Поред тога, квалитет решења добијених применом модификоване VNS методе бољи је код скоро свих тестираних инстанци у односу на квалитет решења добијен применом VNS-CAI методе. Ипак, упркос податку да су пронађена квалитетна решења, у поређењу са резултатима добијеним применом оптимизационих решавача, обе методе нису се показале као превише успешне. С тим у вези, извршена су додатна тестирања на инстанцама које нису коришћене за тестирање у [72], односно извршена су тестирања инстанци произвољно генерисаног типа са 400, 450, 500, 600 и 700 чворова, и инстанци планарног типа са 250, 300, 400, 500 и 600 чворова. Резултати додатних тестирања приказани су у наставку, у Табелама 14 и 15.

Пошто су за потребе додатних тестирања све инстанце прво тестиране применом оптимизационих решавача а потом применом VNS метода, примећено је да одређени број инстанци оптимизациони решавачи не могу да реше, односно да не могу ни да започну њихово решавање, док код неких других инстанци могу да започну процес решавања али га убрзо прекидају услед недостатка меморије. Поред ових, ”проблематичних” инстанци, скуп додатно тестираних инстанци чине и инстанце које оптимизациони решавачи, потенцијално, могу да реше али је процес решавања изузетно дуг (може да потраје и до неколико дана). Из тог разлога, време решавања свих инстанци ограничено је на 7200 секунди, као што је то био случај и у Табелама 12 и 13. Резултати тестирања ”проблематичних” инстанци приказани су у Табели 15, док су резултати тестирања инстанци за које су оптимизациони решавачи вратили макар неко решење пре прекида програма услед временског ограничења приказани у Табели 14. Обе методе тестиране су на истом скупу улазних параметара, односно за  $k_{min} = 1$ ,  $k_{max} = 20$ ,  $k_{step} = 1$ ,  $t_{max} = 7200$  и  $p = 0.5$ . Код модификоване VNS методе додатни параметар  $brIter_{max}$  је постављен на 100.

У Табели 14 прво су приказани резултати тестирања инстанци планарног а затим резултати тестирања инстанци произвољно генерисаног типа. Из представљених резултата лако се може уочити да је само једна инстанца планарног типа оптимално решена, док су за остале инстанце

пронађене неке вредности. Заправо, за један део тестираних инстанци планарног типа процес решавања је прекинут због "out of memory" статуса, док је за други део инстанци планарног типа процес решавања прекинут без поруке којом би се објаснио разлог прекида, тако да се не зна да ли су те инстанце решене оптимално или не. Применом VNS-CAI методе пронађене су исте или боље вредности у односу на оне које су пронашли оптимизациони решавачи за све инстанце планарног типа. Конкретно, од 6 тестираних, применом VNS-CAI методе боље вредности пронађене су за 3 инстанце, док су за преостале три инстанце пронађене исте вредности као оне које су пронашли оптимизациони решавачи. Што се квалитета пронађених вредности тиче, приметимо да су  $err$  и  $\sigma$  из сегмента (2.18%, 5.59%) тако да се предност VNS-CAI методе највише огледа у њеној брзини обзиром да су времена из колоне  $t$  углавном мања од 300 секунди. Са друге стране, применом модификоване VNS методе вредности из колоне  $best$  пронађене су за свих 6 инстанци. Другим речима, модификованом VNS методом пронађене су исте или боље вредности у односу на оне које су пронашли оптимизациони решавачи и VNS-CAI метода. Иако су времена решавања тестираних инстанци углавном лошија у односу на времена решавања VNS-CAI методе, квалитет пронађених вредности је значајно бољи ( $err, \sigma \in (0.29\%, 6.35\%)$ ).

Што се инстанци произвољно генерисаног типа тиче, оптимизациони решавачи су само за *Рандом-400-60* успели да пронађу и да докажу оптималност добијеног решења, док су код осталих тестираних инстанци успели да пронађу неке вредности али не и да докажу оптималност добијених вредности у задатом року. Што се квалитета пронађених вредности тиче, за разлику од резултата приказаних у Табелама 12 и 13, оптимизациони решавачи нису успели да пронађу најбоље вредности за све приказане инстанце. Заправо, вредности из колоне  $best$  пронашли су за 16 инстанци, док су за 9 инстанци пронашли лошије вредности од вредности из колоне  $best$  (вредности које су пронашли VNS-CAI и/или модификована VNS метода). Са друге стране, применом VNS-CAI методе, вредност из колоне  $best$  пронађена је за 19 инстанци, док је применом модификоване VNS методе вредност из колоне  $best$  пронађена код 24 инстанце. Модификована VNS метода није пронашла вредност из колоне  $best$  само за једну инстанцу (*Рандом-700-20*) али је пронашла бољу вредност у односу на ону коју су пронашли оптимизациони решавачи. Дакле, применом модификоване VNS методе пронађене су вредности које су једнаке или

боље од вредности које су пронашли оптимизациони решавачи за све приказане инстанце. Када су времена решавања у питању, VNS-CAI метода углавном брже долази до вредности из колоне *best* у односу на модификовану VNS методу. Ипак, слично резултатима из претходних табела, применом модификоване VNS методе пронађене вредности су код већине инстанци углавном бољег квалитета (мање вредности у колонама *err* и  $\sigma$ ) када поредимо инстанце за које су модификована VNS и VNS-CAI метода пронашле исте вредности.

У Табели 15 приказани су резултати тестирања преосталих инстанци произвољно генерисаног типа са 400 и више чворова. Иако су све инстанце приказане у Табели 15 решаване применом оптимизационих решавача, применом VNS-CAI и применом модификоване VNS методе, једино је модификована VNS метода решила све приказане инстанце. Заправо, оптимизациони решавачи и VNS-CAI метода решили су само једну инстанцу (*Рандом-600-50*) док су при решавању свих осталих инстанци прекидали процес тестирања пре него што су за њих вратили било које допустиво решење. Додатно, применом VNS-CAI методе пронађене су неке вредности за одређене инстанце, али је уз те вредности приказана и порука да пронађене вредности нису допустиве за посматрани проблем, те се за исте ни не може рећи да су решене. Са друге стране, за неке друге инстанце процес тестирања је прекидан без враћања било каквог решења или поруке којом би се објаснио разлог прекида процеса решавања. У сваком случају, због одсуства изворног кода не може да се утврди прави разлог прекида процеса решавања, те се једино може константовати да се одређене инстанце једноставно не могу решити применом VNS-CAI методе. Пошто су оптимизациони решавачи за одређене инстанце приказали поруку "duplicate entry" пре прекида програма, намеће се утисак да дуплиране гране могу да представљају проблем при решавању. Увидом у садржај тих инстанци примећено је да се одређене гране понављају два или више пута, чиме се оправдава приказана грешка. Међутим, с обзиром да су оптимизациони решавачи овај вид грешке пријавили само за пар инстанци, не и за све приказане, могуће је да дуплиране грешке нису једини разлог прекида процеса решавања како код оптимизационих решавача тако и код VNS-CAI методе.



Табела 14: Резултати додатних тестирања VNS-CAI и модификоване VNS методе

Инстанце				Cpl/Gur	VNS-CAI				модификовани VNS				
Име	$ V $	$ E $	<i>best</i>	<i>val(t)</i>	<i>sol</i>	<i>t</i>	<i>err</i>	$\sigma$	<i>sol</i>	<i>t</i>	<i>t<sub>tot</sub></i>	<i>err</i>	$\sigma$
план250	250	6495	17	17 (1841.74)	<i>onm</i>	827.513	0.0559	0.0128	<i>onm</i>	57.828	7200.111	0.0029	0.0128
план300	300	8199	20	<b>20</b> (7200/нм)	<b>20</b>	3.308	0.0450	0.0472	<b>20</b>	57.628	7200.000	0	0
план400	400	13750	23	<u>25</u> (7200/нм)	<b>23</b>	26.051	0.0522	0.0353	<b>23</b>	989.525	7200.000	0.0370	0.0155
план500	500	19352	26	<u>28</u> (7200/нм)	<u>27</u>	147.449	0.0389	0.0218	<b>26</b>	2149.487	7201.075	0.0635	0.0251
план600	600	24353	32	<u>33</u> (7200/нм)	<b>32</b>	136.083	0.0453	0.0270	<b>32</b>	978.278	7201.878	0.0266	0.0227
план700	700	30827	36	<b>36</b> (7200/нм)	<b>36</b>	258.591	0.0306	0.0262	<b>36</b>	1917.960	7201.770	0.0347	0.0276
Рандом-400-10	400	7894	37	<u>38</u> (7200/нм)	<b>37</b>	339.342	0.0432	0.0248	<b>37</b>	323.989	7200.597	0.0257	0.0181
Рандом-400-20	400	16063	20	<u>22</u> (7200/нм)	<u>22</u>	20.838	0	0	<b>20</b>	872.263	7200.521	0.0850	0.0320
Рандом-400-30	400	23827	14	<u>15</u> (7200/нм)	<u>16</u>	1.780	0	0	<b>14</b>	2586.106	7200.387	0.0821	0.0341
Рандом-400-40	400	32070	12	<b>12</b> (7200/нм)	<b>12</b>	0.562	0	0	<b>12</b>	0.760	7200.352	0	0
Рандом-400-50	400	39820	10	<b>10</b> (7200/нм)	<b>10</b>	0.100	0	0	<b>10</b>	0.208	7200.464	0	0
Рандом-400-60	400	47904	8	8 (2249.111)	<i>onm</i>	0.070	0	0	<i>onm</i>	0.305	7200.285	0	0
Рандом-450-10	450	10054	38	<b>38</b> (7200/нм)	<b>38</b>	22.197	0.0434	0.0279	<b>38</b>	935.490	7200.528	0.0289	0.0234
Рандом-450-20	450	20227	22	<u>23</u> (7200/нм)	<b>22</b>	36.350	0.0545	0.0422	<b>22</b>	223.621	7200.396	0.0114	0.0197
Рандом-450-30	450	30275	15	<u>16</u> (7200/нм)	<u>16</u>	3.702	0	0	<b>15</b>	618.726	7200.556	0.0600	0.0200
Рандом-450-40	450	40364	12	<b>12</b> (7200/нм)	<b>12</b>	1.765	0	0	<b>12</b>	26.335	7200.492	0	0
Рандом-450-50	450	50491	10	<b>10</b> (7200/нм)	<b>10</b>	0.328	0	0	<b>10</b>	0.958	7200.335	0	0
Рандом-500-10	500	12493	39	<u>40</u> (7200/нм)	<b>39</b>	351.107	0.0487	0.0242	<b>39</b>	5091.306	7200.688	0.0410	0.0205
Рандом-500-20	500	24823	22	<b>22</b> (7200/нм)	<u>23</u>	99.211	0.0391	0.0130	<b>22</b>	730.545	7200.803	0.0568	0.0283
Рандом-500-30	500	37504	16	<b>16</b> (7200/нм)	<b>16</b>	4.751	0.0094	0.0223	<b>16</b>	30.626	7200.430	0	0
Рандом-500-40	500	49704	12	<b>12</b> (7200/нм)	<b>12</b>	28.674	0.0458	0.0720	<b>12</b>	262.310	7201.035	0	0
Рандом-600-10	600	18040	42	<b>42</b> (7200/нм)	<b>42</b>	244.169	0.0310	0.0201	<b>42</b>	3404.846	7201.353	0.0381	0.0158
Рандом-600-20	600	36175	24	<b>24</b> (7200/нм)	<b>24</b>	24.013	0.0208	0.0361	<b>24</b>	199.931	7201.187	0.0042	0.0125
Рандом-600-30	600	53871	16	<u>18</u> (7200/нм)	<u>18</u>	5.546	0.0028	0.0121	<b>16</b>	862.325	7201.195	0.0688	0.0272
Рандом-600-40	600	71780	12	<u>14</u> (7200/нм)	<b>12</b>	599.690	0.1583	0.0363	<b>12</b>	1124.884	7200.785	0.0500	0.0486
Рандом-650-10	650	21223	42	<b>42</b> (7200.033)	<b>42</b>	547.931	0.0524	0.0297	<b>42</b>	3900.243	7201.794	0.0536	0.0259
Рандом-650-20	650	42056	24	<b>24</b> (7200/нм)	<u>26</u>	14.827	0	0	<b>24</b>	6433.470	7201.470	0.0646	0.0246
Рандом-700-10	700	24575	44	<b>44</b> (7200/нм)	<b>44</b>	609.908	0.0375	0.0262	<b>44</b>	1980.239	7202.796	0.0398	0.0226
Рандом-700-20	700	48998	24	<u>26</u> (7200/нм)	<b>24</b>	2246.989	0.0771	0.0199	<u>25</u>	5280.892	7202.176	0.0460	0.0191
Рандом-700-30	700	73397	18	<b>18</b> (7200/нм)	<b>18</b>	26.513	0	0	<b>18</b>	184.696	7201.933	0.0028	0.0121
Рандом-700-40	700	97832	14	<b>14</b> (7200/нм)	<b>14</b>	2.937	0	0	<b>14</b>	9.699	7203.916	0	0

Табела 15: Резултати додатних тестирања VNS-CAI и модификоване VNS методе на инстанцама за које отимизациони решавачи нису пронашли ни једно решење пре прекидања програма.

Инстанце				Cpl/Gur <i>val(t)</i>	VNS-CAI				модификовани VNS				
Име	$ V $	$ E $	<i>best</i>		<i>sol</i>	<i>t</i>	<i>err</i>	$\sigma$	<i>sol</i>	<i>t</i>	<i>t<sub>tot</sub></i>	<i>err</i>	$\sigma$
Рандом-400-70	400	111986	6	-	-	-	-	-	<b>6</b>	0.098	1000.137	0	0
Рандом-400-80	400	127690	6	-	-	-	-	-	<b>6</b>	0.108	1000.136	0	0
Рандом-400-90	400	143984	4	-	-	-	-	-	<b>4</b>	0.118	1000.081	0	0
Рандом-450-60	450	121046	8	-	-	-	-	-	<b>8</b>	1.378	1000.209	0	0
Рандом-450-70	450	141168	6	-	-	-	-	-	<b>6</b>	4.496	1000.254	0	0
Рандом-450-80	450	161352	6	-	-	-	-	-	<b>6</b>	0.431	1000.455	0	0
Рандом-450-90	450	181844	4	-	-	-	-	-	<b>4</b>	0.654	1000.578	0	0
Рандом-500-50	500	124834	10	-	-	-	-	-	<b>10</b>	0.088	1000.321	0	0
Рандом-500-60	500	149364	8	-	-	-	-	-	<b>8</b>	0.127	1000.137	0	0
Рандом-500-70	500	174750	6	-	-	-	-	-	<b>6</b>	0.152	1000.312	0	0
Рандом-500-80	500	199522	6	-	-	-	-	-	<b>6</b>	0.182	1000.393	0	0
Рандом-500-90	500	224772	4	-	-	-	-	-	<b>4</b>	0.181	880.035	0	0
Рандом-600-50	600	90102	10	- (7200/нм)	<b>10</b>	6.296	0	0	<b>10</b>	8.836	1000.478	0	0
Рандом-600-60	600	108014	26	-	-	-	-	-	<b>26</b>	0.132	1000.097	0	0
Рандом-600-70	600	125777	69	-	-	-	-	-	<b>69</b>	0.124	1000.059	0	0
Рандом-600-80	600	287740	6	-	-	-	-	-	<b>6</b>	0.265	1000.218	0	0
Рандом-600-90	600	323832	4	-	-	-	-	-	<b>4</b>	1.452	1001.413	0	0
Рандом-650-30	650	127064	16	-	-	-	-	-	<b>16</b>	1173.458	1173.458	0.1219	0.0311
Рандом-650-40	650	169370	9	-	-	-	-	-	<b>9</b>	1040.395	1040.395	0.5278	0.1211
Рандом-650-50	650	211570	10	-	-	-	-	-	<b>10</b>	220.520	1000.438	0.0650	0.0477
Рандом-650-60	650	252886	9	-	-	-	-	-	<b>9</b>	61.771	1000.686	0	0
Рандом-650-70	650	295080	7	-	-	-	-	-	<b>7</b>	74.887	1000.941	0.0929	0.0681
Рандом-650-80	650	336780	6	-	-	-	-	-	<b>6</b>	1.317	1001.212	0	0
Рандом-650-90	650	380158	4	-	-	-	-	-	<b>4</b>	2.609	1001.088	0	0
Рандом-700-50	700	122317	71	-	-	-	-	-	<b>71</b>	988.433	1000.410	0.0296	0.0133
Рандом-700-60	700	146219	124	-	-	-	-	-	<b>124</b>	504.475	1000.345	0.0109	0.0069
Рандом-700-70	700	171363	126	-	-	-	-	-	<b>126</b>	0.230	1000.308	0.0317	0.0163
Рандом-700-80	700	195944	128	-	-	-	-	-	<b>128</b>	0.182	1000.324	0.0492	0.0201
Рандом-700-90	700	440280	4	-	-	-	-	-	<b>4</b>	0.411	1000.414	0	0

Дакле, на основу свега наведеног може се закључити да оптимизациони решавачи и VNS-CAI метода не могу да решавају проблем римске доминације за посматране инстанце, односно да је модификовани VNS алгоритам једини алгоритам који може да решава проблем римске доминације за инстанце свих врста и димензија. Пошто је ово једина метода која проналази нека решења за посматране инстанце, максимално време решавања посматраних инстанци је постављено на 1000 секунди ( $t_{max} = 1000$ ) како би се смањило укупно време њиховог решавања иако би се, потенцијално, применом модификоване VNS методе могле добити и боље вредности (мање вредности у колони *sol* и/или мање вредности у колонама *err* и  $\sigma$ ) за дуже време решавања. Као што се из приказаних резултата може видети, применом модификоване VNS методе решено је свих 29 инстанци. Поред тога што ове инстанце први пут решене од стране неке методе, код већине инстанци вредности из колоне *best* први пут су пронађене за мање од 1 секунде. Такође, код већине инстанци одступања од најбољих пронађених вредности једнаке су нули. Пошто се раније (Табеле 12-13) показало да ова метода често проналази оптимална решења, могло би се закључити да су и вредности пронађене у Табели 15 једнаке или јако близу оптималним.

Најзад, од 6 додатно тестираних инстанци планарног и 54 додатно тестираних инстанци произвољно генерисаног типа (Табеле 14 - 15), модификована VNS метода пронашла је вредности из колоне *best* за 59 инстанци, док су оптимизациони решавачи пронашли вредности из колоне *best* за 20 инстанци а VNS-CAI метода за 26 инстанци. Имајући све наведено у виду може се закључити да је развој модификоване VNS методе свакако оправдан и да сама метода квалитетно решава проблем римске доминације без обзира на тип и величину инстанце.

### 4.2.3 Нова метода за решавање проблема слабе римске доминације заснована на методи променљивих околина

#### Простор решења и структура допустивих решења проблема

Нека се Римско царство може представити графом. Простор решења, трансформације, разлике реда  $k$  и околине решења дефинисане су слично као за проблем римске доминације. Међутим, иако је допустиво решење проблема римске доминације уједно и допустиво решење проблема слабе римске доминације, допустиво решење проблема слабе римске доминације није обавезно допустиво решење проблема римске доминације. Допустивост решења проблема слабе римске доминације ( $WRD$  проблема) дефинише се на следећи начин.

$n$ -торка  $X = (x_1, \dots, x_n)$  се назива **допустивим решењем**  $WRD$  проблема ако постоји функција  $f, f: V \rightarrow \{0, 1, 2\}$  дефинисана са

$$f(i) = x_i, \quad i = 1, \dots, n, \quad (4.2)$$

таква да је  $f$  функција слабе римске доминације. Вредност  $x_i$  назива се тежином чвора  $i$  док се са  $F(X)$  обележава *вредност (тежина)* решења  $X$ ,  $F(X) = \sum_{i=1}^n x_i$ . Другим речима, пошто је проблем слабе римске доминације динамички проблем, свака провинција Римског царства треба да је заштићена пре и након померања једне легије. Стога,  $X$  је допустиво решење  $WRD$  проблема уколико је за свако  $x_i$  ( $i = 1, \dots, n$ ) испуњен један од следећа два услова

1)  $x_i > 0$  или

2)  $x_i = 0$  и

а) постоји најмање једно  $x_j = 2, (i, j) \in E, j = 1, \dots, n$ .

- чвор  $i$  је заштићен од напада и након што се са чвора  $j$  једна легија пресели на било који њему суседни чвор  $s$ . Уколико је  $s = i$ , чвор  $i$  је заштићен од стране своје стациониране легије. Уколико је  $s \neq i$  чвор  $i$  је заштићен зато што има суседа са бар једном стационираном легијом (чвор  $j$ ). Слично, сви остали суседи чвора  $j$  су заштићени будући да је и након трансформације  $x_j > 0$ .

и/или

- б) постоји најмање једно  $x_j = 1$ ,  $(i, j) \in E$ ,  $j = 1, \dots, n$ , такво да, након размене вредности  $x_i \leftarrow 1$  и  $x_j \leftarrow 0$ , за свако  $x_s$  ( $s = 1, \dots, n$ ) са особинама  $x_s = 0$  и  $(s, j) \in E$  постоји најмање једно  $x_k$  са особинама  $x_k > 0$  ( $k = 1, \dots, n$ ) и  $(k, s) \in E$ .

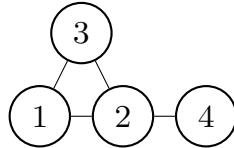
- чворове  $i$  и  $j$  штити легија која је сада стационирана у чвору  $i$ . Сваки други чвор  $s$  који је суседан са  $j$  мора имати бар једног суседа  $k$  таквог да је  $x_k > 0$ .

Условом 1) провинција се сматра сигурном од напада уколико поседује бар једну стационирану легију. Условима 2а) и 2б) провинција се сматра заштићеном од напада уколико има суседа са бар једном стационираном легијом и ако пресељење једне легије не нарушава безбедност суседних провинција. Решење  $X$  сматра се допустивим решењем  $WRD$  проблема уколико све његове координате представљају провинције које су сигурне или заштићене од напада.

Скуп допустивих решења  $WRD$  проблема за граф  $G$  представљају све  $n$ -торке  $X$  ( $X = (x_1, \dots, x_n)$ ,  $x_i \in \{0, 1, 2\}$ ,  $i = 1, \dots, n$ ) дефинисане тако да све њихове координате  $x_i$  задовољавају услове 1) или 2). На пример, скуп допустивих решења  $WRD$  проблема за граф  $\mathcal{G}_1$ , који је дефинисан у претходном поглављу, може се представити скупом

$$X_{WRD(\mathcal{G}_1)} = \left\{ (2, 2, 2), (2, 2, 1), (2, 1, 2), (1, 2, 2), (2, 1, 1), (1, 2, 1), (1, 1, 2), (2, 2, 0), (2, 0, 2), (0, 2, 2), (1, 1, 1), (1, 1, 0), (1, 0, 1), (0, 1, 1), (2, 1, 0), (2, 0, 1), (1, 2, 0), (1, 0, 2), (0, 2, 1), (0, 1, 2), (2, 0, 0), (0, 2, 0), (0, 0, 2), (1, 0, 0), (0, 1, 0), (0, 0, 1) \right\}.$$

Узмимо још један граф за пример. Нека је дат граф  $\mathcal{G}_2 = (V_{\mathcal{G}_2}, E_{\mathcal{G}_2})$ , где је  $V_{\mathcal{G}_2} = \{1, 2, 3, 4\}$  и  $E_{\mathcal{G}_2} = \{(1, 2), (1, 3), (2, 3), (2, 4)\}$ .



Слика 18: Граф  $\mathcal{G}_2$

Скуп допустивих решења  $WRD$  проблема за граф  $\mathcal{G}_2$  може се записати као:  $X_{WRD(\mathcal{G}_2)} = \left\{ (x_1, x_2, x_3, x_4) \mid x_i \in \{0, 1, 2\} \right\} \setminus \left\{ (2, 2, 2, 2), (2, 2, 2, 1), (2, 2, 1, 2), \right.$

$(2, 1, 2, 2), (1, 2, 2, 2), (2, 2, 1, 1), (2, 1, 2, 1), (1, 2, 2, 1), (2, 1, 1, 2), (1, 2, 1, 2), (1, 1, 2, 2),$   
 $(2, 2, 2, 0), (2, 2, 0, 2), (2, 0, 2, 2), (0, 2, 2, 2), (2, 2, 1, 0), (2, 1, 2, 0), (1, 2, 2, 0), (2, 2, 0, 1),$   
 $(2, 1, 0, 2), (1, 2, 0, 2), (2, 0, 2, 1), (0, 2, 2, 1), (2, 0, 1, 2), (0, 2, 1, 2), (1, 0, 2, 2), (0, 1, 2, 2),$   
 $(2, 1, 1, 1), (1, 2, 1, 1), (1, 1, 2, 1), (1, 1, 1, 2), (2, 2, 0, 0), (0, 2, 2, 0), (0, 2, 0, 2), (0, 0, 2, 2),$   
 $(2, 0, 0, 2), (2, 1, 1, 0), (2, 1, 0, 1), (1, 2, 1, 0), (1, 2, 0, 1), (1, 1, 2, 0), (1, 1, 0, 2), (1, 0, 1, 2),$   
 $(1, 0, 2, 1), (0, 1, 1, 2), (0, 1, 2, 1), (0, 2, 1, 1), (2, 0, 1, 1), (0, 2, 1, 1), (1, 1, 1, 1), (2, 1, 0, 0),$   
 $(0, 2, 1, 0), (0, 2, 0, 1), (0, 0, 2, 1), (0, 1, 2, 0), (1, 0, 0, 2), (0, 1, 0, 2), (0, 0, 1, 2), (1, 1, 1, 0),$   
 $(2, 0, 0, 1), (1, 2, 0, 0), (1, 1, 0, 1), (1, 0, 1, 1), (0, 1, 1, 1), (0, 2, 0, 0), (1, 1, 0, 0), (0, 1, 1, 0),$   
 $(1, 0, 0, 1), (0, 1, 0, 1), (0, 0, 1, 1)\}$ .

### Имплементација нове методе променљивих околина за решавање проблема слабе римске доминације

Метода променљивих околина за решавање проблема слабе римске доминације илустрована је Алгоритмом 2. Основне функције *početnoResenje()*, *razmrdavanje()* и *lokalnoPretrazivanje()*, као и помоћна функција *ispitajDopustivost(X, i)* дефинисане су исто као и за  $\mathcal{RD}$  проблем. Међутим, њихова помоћна функција *brNedopustivih(X)* сада проверава допустивост решења  $X$  у односу на  $\mathcal{WRD}$  проблем. Функција *brNedopustivih(X)* проверава допустивост решења  $X$  пре и након померања једне легије користећи помоћну функцију *siguran(k, l)*. Детаљни опис функција *brNedopustivih(X)* и *siguran(k, l)* дат је у наставку. Слично као и за  $\mathcal{RD}$  проблем,  $V_i = \{i_1, \dots, i_{n_i}\}$  представља скуп свих чворова суседних чвору  $i$ .

Функција *siguran(k, l)* имплементирана је на следећи начин. Прво се изврше трансформације  $x_l \leftarrow x_l - 1$  и  $x_k \leftarrow x_k + 1$ . Потом се проверава вредност  $x_j$  за једно по једно  $j \in V_l$ . Уколико је  $x_j \geq 1$  прелази се на проверу вредности  $x_j$  за наредно  $j$ . Иначе се врши провера вредности  $x_s$  за једно по једно  $s \in V_j$ . Ако је за свако  $s \in V_j$   $x_s = 0$ ,  $x_l$  се увећава за један,  $x_k$  умањује за један а функција *siguran(k, l)* враћа 0. Међутим, уколико је  $x_s > 0$  за неко  $s \in V_j$ , испитивање вредности  $x_s$  се прекида а провера вредности  $x_j$  наставља за наредно  $j$ . По завршетку испитивања вредности  $x_j$  за свако  $j$  извршавају се трансформације  $x_l \leftarrow x_l + 1$ ,  $x_k \leftarrow x_k - 1$ , а функција *siguran(k, l)* враћа 1.

Другим речима, у оквиру функције *siguran(k, l)* проверава се да ли би вредности координата текућег решења испуњавале услове допустивости

након што се једна легија пресели из чвора  $l$  у чвор  $k$ . Премештање легије могуће је за било која два чвора  $k$  и  $l$  за која важи  $x_k < 2$  и  $x_l > 0$ , а са обзиром да  $WRD$  проблем испитује заштићеност својих провинција након што се легија пресели из једне у себи суседну провинцију, подразумева се да су  $k$  и  $l$  суседни чворови (иако се у оквиру саме функције не врши провера да ли је  $(k, l) \in E$ ).

Слично као и за  $RD$  проблем, функција  $brNedopustivih(X)$  враћа број координата решења  $X$ , означен са  $nd$ , који не задовољавају нити један од услова допустивости  $WRD$  проблема, тј. функција враћа нулу уколико је решење  $X$  допустиво решење  $WRD$  проблема. Функција је имплементирана на следећи начин. На почетку је  $nd = 0$ . Затим се за једно по једно  $i = 1, \dots, n$  проверава вредност  $x_i$ . Уколико је  $x_i > 0$  провера се наставља за наредно  $i$ ,  $nd$  остаје непромењено. Иначе се врше додатна испитивања а чвор  $i$  добија статус "незаштићеног чвора". У оквиру додатног испитивања, за једно по једно  $j \in V_i$  проверава се вредност  $x_j$  ( $V_j = \{i_1, \dots, i_{n_i}\}$ ):

1. за  $x_j = 2$  чвор  $i$  добија статус "заштићеног чвора" чиме се додатно испитивање завршава.
2. за  $x_j = 1$  проверава се вредност функције  $siguran(i, j)$ . Уколико функција  $siguran(i, j)$  врати 1, чвор  $i$  добија статус "заштићеног чвора" а додатно испитивање се прекида. У случају да функција  $siguran(i, j)$  врати 0, проверава се вредност елемента  $x_j$  за наредну вредност бројача  $j$ .

Случај  $x_j = 0$  се не разматра. Након што су испитане вредности  $x_j$  за свако  $j \in V_i$ , или су додатна испитивања прекинута, вредност  $nd$  се увећава за 1 уколико чвор  $i$  и даље има статус "незаштићен". У супротном, вредност  $nd$  остаје непромењена. У оба случаја, испитивање вредности  $x_i$  се наставља за наредну вредност бројача  $i$ , све док нису испитане све вредности  $n$ -торке  $X$ . Функција  $brNedopustivih(X)$  враћа вредност  $nd$ .

Примера ради, за граф  $\mathcal{G}_1$  функција  $brNedopustivih(X)$  враћа 0 за свако  $X \in X_{WRD(\mathcal{G}_1)}$ . За решење  $X = (0, 0, 0)$ , једино који није из скупа допустивих решења за тај граф, функција  $brNedopustivih(X)$  враћа 3: узимајући

да је  $i = 1, 2, 3$ , функција  $brNedopustivih(X)$  редом проверава да ли координате решења  $X$  задовољавају услове допустивости за  $WRD$  проблем; како је  $x_1 = 0$  чвор 1 ће добити статус "незаштићеног чвора"; испитују се вредности елемената  $x_2$  и  $x_3$  ( $V_1 = \{2, 3\}$ ); како је за свако  $j = \{2, 3\}$   $x_j = 0$ , у оквиру додатних испитивања статус чвора 1 остаје непромењен, што значи да се вредност  $nd$  увећава за један; описани поступак се понавља за  $i = 2$  и  $i = 3$ ; с обзиром да су вредности  $x_2$  и  $x_3$  такође једнаке нули, вредност  $nd$  ће се увећати још два пута. Слично, за граф  $\mathcal{G}_2$  функција  $brNedopustivih(X)$  враћа 0 за све четворке  $X \in X_{WRD(\mathcal{G}_2)}$ . Међутим, за  $X = (1, 0, 2, 0)$  функција  $brNedopustivih(X)$  враћа 1: узимајући да је  $nd = 0$  и са обзиром да је  $x_1 = 1$  прелази се на проверу вредности  $x_2$ ; како је  $x_2 = 0$  чвор 2 добија статус "незаштићеног чвора"; врши се додатно испитивање вредности координата које одговарају чворовима који су суседни са чвором 2; у оквиру додатног испитивања проверавају се вредности  $x_j$ , где је  $j = \{1, 3, 4\}$  ( $V_2 = \{1, 3, 4\}$ ); са обзиром да је  $x_1 \geq 1$  даље се проверава вредност функције  $siguran(2, 1)$ ; у оквиру функције  $siguran(2, 1)$  прво се изврше трансформације  $x_2 \leftarrow x_2 + 1$ ,  $x_1 \leftarrow x_1 - 1$ ; потом се за свако  $j = \{2, 3\}$  ( $V_1 = \{2, 3\}$ ) проверава вредност  $x_j$ ; имајући у виду да је  $x_2 = 1$ , односно да је  $x_3 = 2$ , у оквиру функције  $siguran(2, 1)$  врше се трансформације  $x_1 \leftarrow x_1 + 1$ ,  $x_2 \leftarrow x_2 - 1$  и одређује број координата текућег решења које не задовољавају услове допустивости; функција  $siguran(2, 1)$  враћа 1 као резултат; чвор 2 добија статус заштићеног чвора чиме су додатна испитивања завршена; провера вредности координата решења  $X$  се наставља за  $i = 3$ ; како је  $x_3 = 2$  проверава се вредност наредне координате решења  $X$ ; будући да је  $x_4 = 0$ , чвор 4 добија статус "незаштићеног чвора"; додатна испитивања се врше над елементима  $x_j$ ,  $j = \{2\}$  ( $V_4 = \{2\}$ ); са обзиром да је  $x_2 \notin \{1, 2\}$  чвор 4 остаје "незаштићен", вредност бројача  $nd$  се увећава за 1. Дакле, испитавши вредности свих координата решења  $X$ , функција  $brNedopustivih(X)$  враћа 1 за улазни аргумент  $X = (1, 0, 2, 0)$ .

Улазни параметри модификоване VNS методе су најмањи ( $k_{min}$ ) и највећи ( $k_{max}$ ) ред околине, корак  $k$  ( $k_{step}$ ), максимално време извршавања кода ( $t_{max}$ ) изражено у секундама, максимални број итерација ( $brIter_{max}$ ) и вероватноћа преласка у решења истог квалитета ( $p$ ) изражена у процентима.



## Резултати тестирања модификоване VNS методе за решавање WRD проблема

VNS алгоритам за решавање WRD проблема тестиран је под истим условима као и за RD проблем, тј. на истом скупу инстанци као и VNS-CAI метода у [72] и, додатно, на још неким инстанцама произвољно генерисаног типа са 100 и на свим инстанцама произвољно генерисаног типа са 150, 200 и 250 чворова. Дакле, тестиране су инстанце за које су оптимизациони решавачи успели да докажу оптималност пронађеног решења и на инстанцама за које постоји бар неко решење у литератури. Резултати тестирања приказани су у Табелама 16-19 на следећи начин: резултати тестирања модификоване VNS методе на истом скупу инстанци као у [72] приказани су у Табелама 16-17, док су резултати додатних тестирања од стране обе VNS методе и оптимизационих решавача приказани у Табелама 18 и 19. Табеле су формиране на сличан начин као и за RD проблем (Табеле 16 и 18 садрже резултате тестирања оптимално решених инстанци, док Табеле 17 и 19 садрже резултате тестирања инстанци код којих оптималне вредности нису познате). Слично као и код резултата који се односе на RD проблем, табеле су због већег броја колона приказане у *landscape* формату. С тим у вези, пре приказа сваке табеле дат је њен детаљни опис.

Модификовани VNS алгоритам тестиран је за  $k_{min} = 1$ ,  $k_{max} = 20$ ,  $k_{step} = 1$ ,  $brIter_{max} = 500$ ,  $t_{max} = 7200$  и  $p = 5$ , док су за VNS-CAI методу коришћени параметри  $k_{min} = 1$ ,  $k_{step} = 1$ ,  $k_{max} = 30$ ,  $t_{max} = 7200$  и  $p = 0.5$ . Слично као и за RD проблем, свака инстанца тестирана је по 20 пута (са 20 различитих семена)<sup>16</sup>.

Резултати тестирања 42 инстанце грид типа (34 оптимално решених и 8 инстанци за које су позната нека али није доказано да су пронађена решења оптимална) изложени су на почетку Табела 16 и 17. Применом VNS-CAI методе решење једнако оптималном пронађено је за 33 инстанце, док је применом модификованог VNS-а решење једнако оптималном пронађено за 32 грид инстанце. Када су времена решавања у питању, инстанце са највише 63 чворова изузетно се брзо решавају оптимизационим решавачима (време решавања сваке од ових инстанци мање је од 100

---

<sup>16</sup>Слично као и за RD проблем, разлика у вредностима коришћених параметара је потпуно ненамерна и примећена тек након што су све инстанце тестиране.

секунди), док се инстанце са 64 и више чворова решавају нешто спорије (време решавања ових инстанци углавном је веће од 100 секунди). Пошто максимално време потребно да се реши инстанца није познато за VNS-CAI методу, поређење максималних времена решавања инстанци вршиће се само између колоне  $t$  која садржи време решавања оптимizacionих решавача и колоне  $t_{tot}$  која садржи податке које се односе на максимално време решавања модификоване VNS методе. Лако се може закључити да су све инстанце са највише 63 чворова брже решене оптимizacionим решавачима (потколоне  $t$  колоне *Cpl/Gur*), док је већина инстанци са 64 и више чворова Табеле 16 брже решена применом модификованог VNS-а (потколоне  $t_{tot}$  колоне *модификовани VNS*). Када су резултати приказани у Табели 17 у питању, свих 8 инстанци-grid типа брже је решено применом модификоване VNS методе. Додатно, применом модификоване VNS методе је за једну инстанцу (*grid09×11*) пронађена мања вредност од оне коју су пронашли оптимizacionи решавачи и VNS-CAI метода.

Ако поредимо резултате рада VNS-CAI и модификованог VNS-а, приметимо да су за све-grid инстанце (Табеле 16 и 17) вредности колоне  $t$  углавном мање код VNS-CAI методе, док су вредности у потколонема  $err$  и  $\sigma$  које одговарају модификованом VNS-у углавном мање од вредности потколоне  $err$  и  $\sigma$  које одговарају VNS-CAI методи. Заиста, код 27 инстанци вредности потколоне  $t$  која одговара VNS-CAI методи мање су од или једнаке вредностима колоне  $t$  која одговара модификованој VNS методи. Са друге стране, код свих, осим код једне-grid инстанце (*grid09×11*), вредности колоне  $err$  модификоване VNS методе мање су од или једнаке вредностима колоне  $err$  VNS-CAI методе. Конкретно, код 35 од 42-grid инстанце вредности колоне  $err$  модификованог VNS-а строго су мање од вредности колоне  $err$  VNS-CAI методе, док су код 6 инстанци те вредности међусобно једнаке. Додатно, при решавању инстанце *grid09×11*, модификованом VNS методом добијене су веће вредности у колонема  $err$  и  $\sigma$  у односу на вредности из колоне  $err$  и  $\sigma$  VNS-CAI методе, али је зато модификованом VNS методом за ову инстанцу пронађено боље решење.

Резултати тестирања инстанци-планарног типа приказани су у наставку Табела 16 и 17. Упркос чињеници да оптимizacionи решавачи брже решавају инстанце са највише 50 чворова од модификоване VNS методе (Табела 16), инстанце са 100 и 150 чворова не само што су брже

решене модификованом VNS методом, већ су за њих пронађене и боље вредности (Табела 17). Пошто се инстанце планарног типа брзо решавају и применом VNS-CAI методе, једино што се може закључити јесте да обе методе брзо проналазе вредности из колоне *opt* за инстанце планарног типа, односно брзо проналазе најмање познате вредности за одређене инстанце и да је, специјално, за инстанцу *план100* модификована VNS метода успешнија у смислу да има мање вредности у потколонема *err* и  $\sigma$  у односу на вредности које се налазе у потколонема *err* и  $\sigma$  VNS-CAI методе.

Од 90 произвољно генерисаних инстанци, односно од 30 оптимално решених произвољно генерисаних инстанци (Табеле 16 и 18), VNS-CAI методом вредности једнаке оптималним нису пронађене за 4 инстанце. Од 60 инстанци код којих оптимална решења нису позната, VNS-CAI методом лошије вредности у односу на оне који су пронашли оптимизациони решавачи пронађене су код 5 инстанци, вредности једнаке оним које су пронашли оптимизациони решавачи такође код 5 инстанци, односно боље вредности у односу на оне које су пронашли оптимизациони решавачи код 48 инстанци. Са друге стране, модификованом VNS методом решење једнако оптималном није пронађено само за две инстанце (*Рандом-100-6* и *Рандом-150-2*), а од 60 инстанци за које су позната нека али није потврђено да су пронађена решења оптимална (Табеле 17 и 19), модификованом VNS методом су за две инстанце пронађене веће вредности од оних које су пронашли оптимизациони решавачи, исте вредности као оне које су пронашли оптимизациони решавачи код 6 инстанци, односно боље вредности од оних које су пронашли оптимизациони решавачи код 52 инстанце. Поред тога, модификованом VNS методом су код свих 30 оптимално решених инстанци пронађене исте или боље вредности у односу на оне које су пронађене VNS-CAI методом, односно код 58 од 60 инстанци код којих нису позната оптимална али су позната нека решења. Поред тога што су применом модификоване VNS методе пронађене боље вредности код већине инстанци, пронађене вредности су уједно и бољег квалитета имајући у виду да су вредности у колонема  $\sigma$  и *err* модификоване VNS методе мање од или једнаке вредностима колоне  $\sigma$  и *err* VNS-CAI методе код свих инстанци код којих су вредности у колонема *sol* међусобно једнаке.

Резултати тестирања инстанци нет типа приказани су на крају Табе-

ла 16 и 17. Тестирања су извршена за 4 инстанце, са 100, 200, 400 и 600 чворова. Као што се из резултата може видети, оптимизациони решавачи нису успели да пронађу и докажу оптималност добијеног решења за инстанце нет типа са 200 и више чворова. Међутим, они су пре прекидања програма дали неке резултате за ове инстанце. С тим у вези, резултати тестирања обе методе, VNS-CAI и модификовани VNS, поредиће се са свим познатим резултатима које су дали оптимизациони решавачи. За почетак, вредност из колоне *opt* за инстанцу *Het-10-10* пронађена је од стране обе методе (Табела 16). За друге две инстанце по величини (*Het-10-20* и *Het-20-20*), модификованом VNS методом пронађене су нове најмање вредности (Табела 17). За инстанцу са највише чворова (*Het-30-20*) применом VNS-CAI методе пронађена је вредност која је за мање од 1% лошија од вредности коју су пронашли оптимизациони решавачи и модификована VNS метода (Табела 17). Када су времена решавања у питању, инстанца *Het-10-10* најбрже је решена оптимизационим решавачима, док су преостале три инстанце оптимизациони решавачи и обе методе решавали по 7200 секунди, односно до прекидања програма услед временског ограничења.

Сумирајући резултате тестирања модификованог VNS алгоритма за решавање *WRD* проблема на свим инстанцама за које су оптимизациони решавачи дали неко решење (Табеле 16 - 19), лако се може закључити да су, поред тога што описани алгоритам може да решава *WRD* проблем на инстанцама свих типова и димензија, добијена иста решења, а неретко и боља у односу на она која су добијена коришћењем оптимизационих решавача. Конкретно, од 69 оптимално решених инстанци, модификованом VNS методом решења једнака оптималним пронађена су за 65 инстанци. Код преостале четири инстанце пронађене су вредности код којих су релативна одступања од оптималних вредности приближно једнака 4%, 3.7%, 3.85% и 1.64%. Од 73 инстанце за које су позната нека решења али није потврђено да су пронађене вредности оптималне, модификованом VNS методом најбоље познате вредности пронађене су код 71 инстанце. Код инстанци код којих модификованом VNS методом нису пронађене вредности из колоне *best*, релативна одступања од вредности из колоне *best* износе 1.45% и 1.3%. Дакле, модификованом методом су најбоље познате вредности пронађене код 136 од 142 тестиране инстанце (VNS-CAI методом најбоље познате вредности пронађене су код 116 од 142 инстанце). Имајући у виду да оптимизациони решавачи нису успели

да прикажу ни једну вредност пре прекида програма услед недостатка меморије или неког другог ограничења за 19 инстанци, а да су код чак 38 инстанци пронашли лошије вредности у односу на оне које су пронађене применом модификоване VNS методе, може се закључити да је математички модел за решавање проблема слабе римске доминације изузетно захтеван и да је развој описане метахеуристике потпуно оправдан.

Табела 16: Резултати тестирања модификоване VNS методе на оптимално решеним инстанцама.

Име	Инстанце			Cpl/Gur <i>val(t)</i>	VNS-CAI				модификовани VNS				
	$ V $	$ E $	<i>best</i>		<i>sol</i>	<i>t</i>	<i>err</i>	$\sigma$	<i>sol</i>	<i>t</i>	<i>t<sub>tot</sub></i>	<i>err</i>	$\sigma$
грид04×10	40	66	15	4.109	<i>onm</i>	0.015	0	0	<i>onm</i>	0.028	19.200	0	0
грид05×08	40	67	14	4.64	<i>onm</i>	0.047	0.0333	0.0333	<i>onm</i>	0.250	95.282	0	0
грид03×14	42	67	16	4.829	<i>onm</i>	<0.01	0	0	<i>onm</i>	0.018	22.564	0	0
грид06×07	42	71	15	5.801	<i>onm</i>	0.080	0.0063	0.0188	<i>onm</i>	0.034	28.095	0	0
грид04×11	44	73	16	5.5	<i>onm</i>	0.031	0.0088	0.0210	<i>onm</i>	0.067	27.881	0	0
грид03×15	45	72	17	7.789	<i>onm</i>	0.012	0	0	<i>onm</i>	0.063	28.970	0	0
грид05×09	45	76	16	7.908	<i>onm</i>	0.139	0.0235	0.0288	<i>onm</i>	0.069	31.660	0	0
грид04×12	48	80	17	12.84	<i>onm</i>	0.069	0.0361	0.0265	<i>onm</i>	0.687	190.513	0	0
грид06×08	48	82	18	25.499	<i>onm</i>	<0.01	0	0	<i>onm</i>	0.016	43.262	0	0
грид07×07	49	84	18	9.845	<i>onm</i>	0.021	0	0	<i>onm</i>	0.045	47.705	0	0
грид05×10	50	85	18	10.61	<i>onm</i>	0.055	0.0053	0.0158	<i>onm</i>	0.201	47.455	0	0
грид04×13	52	87	19	11.813	<i>onm</i>	0.035	0.0050	0.0150	<i>onm</i>	0.154	53.591	0	0
грид06×09	54	93	19	25.539	<i>onm</i>	0.331	0.0450	0.0150	<i>onm</i>	1.061	294.252	0.0026	0.0115
грид05×11	55	94	19	11.424	<i>onm</i>	0.388	0.0300	0.0245	<i>onm</i>	7.175	324.409	0.0211	0.0258
грид04×14	56	94	20	35.326	<i>onm</i>	0.082	0.0214	0.0237	<i>onm</i>	0.905	334.031	0.0025	0.0109
грид07×08	56	97	20	21.882	<i>onm</i>	0.076	0.0286	0.0233	<i>onm</i>	0.625	332.826	0	0
грид04×15	60	101	22	40.256	<i>onm</i>	0.163	0	0	<i>onm</i>	0.469	88.252	0	0
грид05×12	60	103	21	14.88	<i>onm</i>	4.036	0.0271	0.0260	<i>onm</i>	0.218	408.966	0.0048	0.0143
грид06×10	60	104	21	35.713	<i>onm</i>	0.746	0.0273	0.0223	<i>onm</i>	11.011	460.456	0.0071	0.0170
грид07×09	63	110	22	70.259	<i>onm</i>	0.318	0.0370	0.0155	<i>onm</i>	4.742	560.501	0.0114	0.0197
грид08×08	64	112	23	171.925	<i>onm</i>	0.037	0.0063	0.0149	<i>onm</i>	0.644	136.133	0	0
грид05×13	65	112	23	67.007	<i>onm</i>	0.928	0.0208	0.0208	<i>onm</i>	0.212	545.956	0	0
грид06×11	66	115	24	381.771	<i>onm</i>	0.757	0.0040	0.0120	<i>onm</i>	0.067	156.791	0	0
грид05×14	70	121	24	73.489	<i>onm</i>	27.030	0.0491	0.0202	<i>onm</i>	9.094	803.952	0.0292	0.0191
грид07×10	70	123	25	618.089	<i>onm</i>	0.670	0.0077	0.0154	<i>onm</i>	0.624	782.534	0	0
грид06×12	72	126	26	1166.405	<i>onm</i>	0.544	0.0074	0.0148	<i>onm</i>	0.193	192.534	0	0
грид08×09	72	127	25	435.146	<i>onm</i>	15.935	0.0383	0.0117	<u>26</u>	0.141	243.24	0	0
грид05×15	75	130	26	288.06	<i>onm</i>	8.133	0.0313	0.0174	<i>onm</i>	16.067	1030.223	0.0154	0.0188
грид07×11	77	136	27	988.596	<i>onm</i>	0.582	0.0268	0.0155	<i>onm</i>	14.594	1073.358	0.0111	0.0170
грид06×13	78	137	27	1005.126	<u>28</u>	0.407	0.0086	0.0149	<u>28</u>	0.399	240.895	0	0
грид08×10	80	142	28	2162.812	<i>onm</i>	10.011	0.0375	0.0178	<i>onm</i>	0.471	1349.913	0.0161	0.0178
грид09×09	81	144	28	737.579	<i>onm</i>	12.521	0.0437	0.0251	<i>onm</i>	33.286	1321.442	0.0214	0.0175

Табела 16: Резултати тестирања модификоване VNS методе на оптимално решеним инстанцама.

Име	Инстанце			Cpl/Gur <i>val(t)</i>	VNS-CAI				модификовани VNS				
	$ V $	$ E $	<i>best</i>		<i>sol</i>	<i>t</i>	<i>err</i>	$\sigma$	<i>sol</i>	<i>t</i>	<i>t<sub>tot</sub></i>	<i>err</i>	$\sigma$
грид07×12	84	149	29	4637.38	<i>onm</i>	47.642	0.0441	0.0181	<i>onm</i>	193.300	1632.750	0.0293	0.0123
грид07×14	98	175	34	1720.86	<i>onm</i>	143.804	0.0433	0.0181	<i>onm</i>	25.353	2626.519	0.0162	0.0146
план10	10	27	3	0.156	<i>onm</i>	<0.01	0	0	<i>onm</i>	<0.01	0.36	0	0
план20	20	105	3	1.36	<i>onm</i>	<0.01	0	0	<i>onm</i>	<0.01	236.135	0	0
план30	30	182	5	7.49	<i>onm</i>	<0.01	0	0	<i>onm</i>	<0.01	35.369	0	0
план50	50	465	6	98.49	<i>onm</i>	0.01	0	0	<i>onm</i>	0.045	328.218	0	0
Рандом-50-1	50	49	24	0.281	<i>onm</i>	<0.01	0	0	<i>onm</i>	<0.01	48.632	0	0
Рандом-50-2	50	49	23	0.343	<i>onm</i>	0.034	0	0	<i>onm</i>	0.016	37.090	0	0
Рандом-50-3	50	58	24	0.39	<i>onm</i>	0.062	0	0	<i>onm</i>	0.088	63.400	0	0
Рандом-50-4	50	54	24	0.484	<i>onm</i>	0.225	0	0	<i>onm</i>	0.037	53.512	0	0
Рандом-50-5	50	67	22	0.968	<i>onm</i>	0.377	0.0196	0.0216	<i>onm</i>	5.639	274.454	0.0182	0.0223
Рандом-50-6	50	86	19	2.053	<i>onm</i>	0.030	0	0	<i>onm</i>	0.118	60.128	0	0
Рандом-50-7	50	84	19	3.171	<i>onm</i>	0.889	0.0175	0.0238	<i>onm</i>	1.811	202.007	0	0
Рандом-50-8	50	95	17	3.093	<i>onm</i>	0.131	0.0333	0.0272	<i>onm</i>	0.307	239.582	0	0
Рандом-50-9	50	108	17	26.373	<i>onm</i>	0.129	0.0028	0.0121	<i>onm</i>	0.161	62.742	0	0
Рандом-50-10	50	112	16	6.781	<i>onm</i>	0.047	0	0	<i>onm</i>	0.109	65.603	0	0
Рандом-50-20	50	248	9	346.264	<i>onm</i>	<0.01	0	0	<i>onm</i>	0.070	170.726	0	0
Рандом-50-30	50	373	7	476.278	<i>onm</i>	0.038	0	0	<i>onm</i>	0.059	283.017	0	0
Рандом-50-40	50	475	6	1447.318	<i>onm</i>	0.092	0	0	<i>onm</i>	0.019	234.611	0	0
Рандом-50-50	50	597	5	1545.06	<i>onm</i>	0.013	0	0	<i>onm</i>	0.015	1106.759	0	0
Рандом-50-60	50	739	4	210.71	<i>onm</i>	0.014	0	0	<i>onm</i>	<0.01	590.625	0	0
Рандом-50-70	50	860	3	156.14	<i>onm</i>	0.059	0	0	<i>onm</i>	0.076	1839.431	0	0
Рандом-50-80	50	980	3	90.813	<i>onm</i>	<0.01	0	0	<i>onm</i>	0.011	3215.716	0	0
Рандом-50-90	50	1103	2	36.53	<i>onm</i>	0.030	0	0	<i>onm</i>	0.046	20.843	0	0
Рандом-100-1	100	100	46	0.64	<i>onm</i>	157.329	0.0354	0.0145	<i>onm</i>	152.173	3529.966	0.0141	0.0158
Рандом-100-2	100	109	46	0.843	<i>onm</i>	36.052	0.0148	0.0117	<i>onm</i>	28.777	2811.837	0.0011	0.0047
Рандом-100-3	100	181	37	7.421	<i>onm</i>	23.640	0.0445	0.0261	<i>onm</i>	34.160	3532.942	0.0162	0.0232
Рандом-100-4	100	206	34	61.702	<i>onm</i>	12.367	0.0213	0.0175	<i>onm</i>	7.070	3440.746	0.0088	0.0324
Рандом-100-5	100	231	32	164.502	<i>onm</i>	60.361	0.0299	0.0186	<i>onm</i>	2.737	4913.243	0.0063	0.0159
Рандом-100-6	100	321	26	5806.74	<u>27</u>	12.441	0.0265	0.0217	<u>27</u>	12.376	5075.340	0	0
Рандом-100-7	100	317	25	4009.377	<i>onm</i>	204.939	0.0434	0.0234	<i>onm</i>	29.346	6240.692	0.0040	0.0120

Табела 16: Резултати тестирања модификоване VNS методе на оптимално решеним инстанцама.

Име	Инстанце			Cpl/Gur	VNS-CAI				модификовани VNS				
	$ V $	$ E $	<i>best</i>	<i>val(t)</i>	<i>sol</i>	<i>t</i>	<i>err</i>	$\sigma$	<i>sol</i>	<i>t</i>	$t_{tot}$	<i>err</i>	$\sigma$
Нет-10-10	100	342	20	148.213	<i>onm</i>	4.290	0.0095	0.0190	<i>onm</i>	9.485	7200.080	0.0025	0.0109

Табела 17: Резултати тестирања модификоване VNS методе на инстанцама код којих нису познате оптималне али су познате неке вредности.

Име	Инстанце			Cpl/Gur	VNS-CAI				модификовани VNS				
	$ V $	$ E $	<i>best</i>	<i>val(t)</i>	<i>sol</i>	<i>t</i>	<i>err</i>	$\sigma$	<i>sol</i>	<i>t</i>	$t_{tot}$	<i>err</i>	$\sigma$
грид06×14	84	148	30	<b>30</b> (7200/нм)	<b>30</b>	2.319	0.0097	0.0148	<b>30</b>	1.706	1541.939	0.0017	0.0073
грид08×11	88	157	31	<b>31</b> (7200/нм)	<b>31</b>	3.412	0.0278	0.0190	<b>31</b>	23.951	2230.121	0.0113	0.0154
грид06×15	90	159	32	<b>32</b> (7200/нм)	<b>32</b>	1.196	0.0179	0.0218	<b>32</b>	3.103	1778.921	0	0
грид09×10	90	161	31	<b>31</b> (7200/нм)	<b>31</b>	40.651	0.0443	0.0197	<b>31</b>	5.791	2500.137	0.0065	0.0129
грид07×13	91	162	32	<b>32</b> (7200/нм)	<b>32</b>	16.778	0.0272	0.0130	<b>32</b>	19.704	1918.521	0.0078	0.0135
грид08×12	96	172	33	<b>33</b> (7200/нм)	<b>33</b>	107.765	0.0403	0.0184	<b>33</b>	145.485	2533.369	0.0273	0.0132
грид09×11	99	178	34	<u>35</u> (7200/нм)	<u>35</u>	2.261	0.0181	0.0132	<b>34</b>	531.761	2628.962	0.0279	0.0064
грид10×10	100	180	35	<b>35</b> (7200/нм)	<b>35</b>	6.63	0.0302	0.0168	<b>35</b>	1.469	2633.736	0.0057	0.0114
план100	100	1540	8	<u>9</u> (7200/нм)	<b>8</b>	4.916	0	0	<b>8</b>	2.699	5981.246	0	0
план150	150	2867	10	<u>13</u> (7200/нм)	<b>10</b>	88.248	0.0273	0.0417	<b>10</b>	24.547	7221.162	0.0250	0.0433
Рандом-100-8	100	317	23	<b>23</b> (7200/нм)	<b>23</b>	313.924	0.0448	0.0279	<b>23</b>	33.771	6867.432	0.0174	0.0213
Рандом-100-9	100	430	21	<b>21</b> (7200/нм)	<b>21</b>	4.980	0.0269	0.0293	<b>21</b>	44.186	7020.981	0.0095	0.0190
Рандом-100-10	100	498	19	<b>19</b> (7200/нм)	<b>19</b>	460.905	0.0445	0.0260	<b>19</b>	196.600	7200.052	0.0316	0.0307
Рандом-100-20	100	981	11	<u>12</u> (7200/нм)	<b>11</b>	8.951	0.0250	0.0382	<b>11</b>	2.308	7202.277	0.0091	0.0273
Рандом-100-30	100	1477	8	<u>11</u> (7200/нм)	<b>8</b>	1462.462	0.1056	0.0242	<b>8</b>	31.278	7200.936	0.0563	0.0622
Рандом-100-40	100	1945	7	<u>9</u> (7200/нм)	<b>7</b>	1.501	0	0	<b>7</b>	0.444	7200.698	0	0
Рандом-100-50	100	2483	5	<u>7</u> (7200/нм)	<b>5</b>	37.134	0	0	<b>5</b>	79.823	7204.286	0	0
Нет-10-20	200	712	39	<u>40</u> (7200/нм)	<u>40</u>	67.323	0.0146	0.0119	<b>39</b>	589.779	7202.664	0.0295	0.0147
Нет-20-20	400	1482	79	<u>81</u> (7200/нм)	<u>81</u>	2066.577	0.0180	0.0132	<b>79</b>	3584.317	7210.371	0.0386	0.0157
Нет-30-20	600	2252	122	<b>122</b> (7200/нм)	<u>123</u>	6034.018	0.0474	0.0352	<b>122</b>	3022.467	7253.234	0.0242	0.0126



Табела 18: Резултати додатних тестирања VNS-CAI и модификоване VNS методе на оптимално решеним инстанцама

Инстанце				Cpl/Gur	VNS-CAI				модификовани VNS				
Име	$ V $	$ E $	<i>best</i>	<i>val(t)</i>	<i>sol</i>	<i>t</i>	<i>err</i>	$\sigma$	<i>sol</i>	<i>t</i>	$t_{tot}$	<i>err</i>	$\sigma$
Рандом-150-1	150	157	70	0.719	<i>onm</i>	5.784	0.0136	0.0106	<i>onm</i>	31.432	3194.479	0.0014	0.0043
Рандом-150-2	150	243	61	23.018	<u>62</u>	13.182	0.0298	0.0163	<u>62</u>	97.083	3053.275	0.0137	0.0105
Рандом-150-3	150	322	50	1607.303	<u>52</u>	19.882	0.0240	0.0250	<i>onm</i>	1670.271	4213.027	0.0350	0.0209
Рандом-200-1	200	229	92	0.625	<i>onm</i>	27.482	0.0179	0.0155	<i>onm</i>	243.649	7201.07	0.0016	0.0039
Рандом-250-1	250	345	107	10.815	<u>109</u>	215.935	0.0257	0.0138	<i>onm</i>	797.02	7201.043	0.0178	0.0106

Табела 19: Резултати додатних тестирања VNS-CAI и модификоване VNS методе на инстанцама код којих нису познате оптималне али су познате неке вредности

Инстанце				Cpl/Gur	VNS-CAI				модификовани VNS				
Име	$ V $	$ E $	<i>best</i>	<i>val(t)</i>	<i>sol</i>	<i>t</i>	<i>err</i>	$\sigma$	<i>sol</i>	<i>t</i>	$t_{tot}$	<i>err</i>	$\sigma$
Рандом-100-60	100	2985	4	<u>100</u> (7200/нм)	<b>4</b>	0.437	0.0625	0.1083	<b>4</b>	72.434	7206.583	0	0
Рандом-100-70	100	3435	4	<u>100</u> (7200/нм)	<b>4</b>	0.045	0	0	<b>4</b>	0.013	7208.769	0	0
Рандом-100-80	100	3935	3	<u>100</u> (7200/нм)	<b>3</b>	0.025	0	0	<b>3</b>	1.131	7206.989	0	0
Рандом-100-90	100	4446	3	<u>100</u> (7200/нм)	<b>3</b>	0	0	0	<b>3</b>	0.019	7214.638	0	0
Рандом-150-4	150	437	41	<u>43</u> (7200/нм)	<u>44</u>	6.036	0.0330	0.0283	<b>41</b>	1257.691	3592.144	0.0500	0.0211
Рандом-150-5	150	557	37	<u>38</u> (7200/нм)	<u>38</u>	27.003	0.0342	0.0251	<b>37</b>	2044.549	7200.099	0.0338	0.0168
Рандом-150-6	150	705	30	<u>31</u> (7200/нм)	<b>30</b>	131.499	0.0800	0.0371	<b>30</b>	419.636	7200.276	0.0400	0.0226
Рандом-150-7	150	778	27	<u>28</u> (7200/нм)	<u>28</u>	3.561	0.0482	0.0379	<b>27</b>	830.549	7201.580	0.0259	0.0206
Рандом-150-8	150	906	24	<u>26</u> (7200/нм)	<b>24</b>	74.872	0.0813	0.0360	<b>24</b>	1684.669	7200.308	0.0417	0.0264
Рандом-150-9	150	965	25	<u>29</u> (7200/нм)	<b>25</b>	40.569	0.0520	0.0286	<b>25</b>	11.648	7201.968	0.0160	0.0196
Рандом-150-10	150	1152	21	<u>23</u> (7200/нм)	<u>22</u>	5.779	0.0295	0.0260	<b>21</b>	216.471	6099.083	0.0262	0.0237
Рандом-150-20	150	2228	12	<u>17</u> (7200/нм)	<u>13</u>	1.433	0.0308	0.0449	<b>12</b>	3630.186	7207.652	0.0833	0.0264
Рандом-150-30	150	3318	9	<u>150</u> (7200/нм)	<b>9</b>	111.754	0.1056	0.0242	<b>9</b>	5183.858	7216.550	0.1056	0.0242
Рандом-150-40	150	4476	7	<u>150</u> (7200/нм)	<b>7</b>	1.01	0	0	<b>7</b>	6.377	7212.160	0	0
Рандом-150-50	150	5550	6	- (7200/нм)	<b>6</b>	0.301	0	0	<b>6</b>	1.343	7219.616	0	0
Рандом-150-60	150	6734	5	- (7200/нм)	<b>5</b>	0.081	0	0	<b>5</b>	0.972	7216.176	0	0
Рандом-150-70	150	7807	4	- (7200/нм)	<b>4</b>	0.265	0	0	<b>4</b>	7.08	7267.152	0	0
Рандом-150-80	150	8924	3	- (7200/нм)	<b>3</b>	1.803	0	0	<b>3</b>	98.015	7238.961	0	0
Рандом-150-90	150	10043	3	- (7200/нм)	<b>3</b>	0.031	0	0	<b>3</b>	1.374	7341.438	0	0

Табела 19: Резултати додатних тестирања VNS-CAI и модификоване VNS методе на инстанцама код којих нису познате оптималне али су познате неке вредности

Име	Инстанце			Cpl/Gur	VNS-CAI				модификовани VNS				
	$ V $	$ E $	$best$	$val(t)$	$sol$	$t$	$err$	$\sigma$	$sol$	$t$	$t_{tot}$	$err$	$\sigma$
Рандом-200-2	200	390	69	<b>69</b> (7200/нм)	<u>71</u>	62.339	0.0380	0.0189	<u>70</u>	648.308	7200.176	0.0236	0.0122
Рандом-200-3	200	581	55	<b>55</b> (7200/нм)	<u>58</u>	151.38	0.0276	0.0176	<b>55</b>	4829.422	7201.879	0.0355	0.0135
Рандом-200-4	200	737	46	<u>49</u> (7200/нм)	<u>49</u>	134.943	0.0622	0.0285	<b>46</b>	2654.908	7201.196	0.0772	0.0326
Рандом-200-5	200	1010	39	<u>40</u> (7200/нм)	<b>39</b>	218.109	0.0603	0.0365	<b>39</b>	792.238	7204.249	0.0346	0.0260
Рандом-200-6	200	1180	36	<u>37</u> (7200/нм)	<b>36</b>	115.925	0.0583	0.0247	<b>36</b>	533.056	7204.490	0.0292	0.0186
Рандом-200-7	200	1453	30	<b>30</b> (7200/нм)	<b>30</b>	95.571	0.0617	0.0264	<b>30</b>	286.001	7204.908	0.0450	0.0264
Рандом-200-8	200	1580	28	<u>31</u> (7200/нм)	<u>29</u>	23.275	0.0241	0.0221	<b>28</b>	74.533	7210.582	0.0411	0.0283
Рандом-200-9	200	1781	26	<b>26</b> (7200/нм)	<b>26</b>	34.457	0.0500	0.0246	<b>26</b>	621.742	7204.031	0.0212	0.0257
Рандом-200-10	200	1984	24	<u>29</u> (7200/нм)	<b>24</b>	290.275	0.0688	0.0272	<b>24</b>	4369.946	7201.297	0.0500	0.0250
Рандом-200-20	200	4116	13	<u>200</u> (7200/нм)	<u>14</u>	5.264	0.0464	0.0341	<b>13</b>	127.704	7222.167	0.1115	0.0515
Рандом-200-30	200	5876	10	<u>200</u> (7200/нм)	<b>10</b>	22.338	0.0550	0.0497	<b>10</b>	470.2	7258.864	0.0250	0.0433
Рандом-200-40	200	7907	8	- (7200/нм)	<b>8</b>	1.655	0	0	<b>8</b>	19.552	7272.771	0	0
Рандом-200-50	200	9895	6	- (7200/нм)	<b>6</b>	14.753	0.0250	0.0595	<b>6</b>	175.337	7280.975	0	0
Рандом-200-60	200	11971	5	- (7200/нм)	<b>5</b>	0.951	0	0	<b>5</b>	29.721	7325.941	0	0
Рандом-200-70	200	14059	4	- (7200/нм)	<b>4</b>	0.718	0	0	<b>4</b>	22.998	7394.337	0	0
Рандом-200-80	200	15918	4	- (7200/нм)	<b>4</b>	0.000	0	0	<b>4</b>	0.129	7602.310	0	0
Рандом-200-90	200	17821	3	- (7200/нм)	<b>3</b>	0.014	0	0	<b>3</b>	0.153	7632.036	0	0
Рандом-250-2	250	633	77	77 (7200/нм)	<u>78</u>	327.501	0.0609	0.0210	<u>78</u>	4705.878	7201.934	0.0340	0.0173
Рандом-250-3	250	956	60	<u>61</u> (7200/нм)	<u>61</u>	591.823	0.0377	0.0244	<b>60</b>	7122.483	7205.279	0.0300	0.0145
Рандом-250-4	250	1194	50	<u>54</u> (7200/нм)	<u>52</u>	301.061	0.0308	0.0214	<b>50</b>	1297.013	7206.840	0.0300	0.0232
Рандом-250-5	250	1540	43	<u>44</u> (7200/нм)	<u>44</u>	516.513	0	0	<b>43</b>	684.635	7208.166	0.0291	0.0231
Рандом-250-6	250	1869	37	<u>44</u> (7200/нм)	<b>37</b>	987.268	0	0	<b>37</b>	3363.253	7210.818	0.0500	0.0299
Рандом-250-7	250	2159	34	<u>36</u> (7200/нм)	<u>35</u>	82.719	0.0371	0.0273	<b>34</b>	3657.473	7208.054	0.0529	0.0256
Рандом-250-8	250	2420	29	<u>34</u> (7200/нм)	<u>32</u>	70.707	0.0328	0.0251	<b>29</b>	3829.234	7211.048	0.1034	0.0408
Рандом-250-9	250	2880	27	<u>33</u> (7200/нм)	<u>29</u>	21.613	0.0345	0.0289	<b>27</b>	3622.312	7224.904	0.0815	0.0277
Рандом-250-10	250	3167	25	<u>29</u> (7200/нм)	<u>26</u>	399.371	0.0404	0.0257	<b>25</b>	2874.293	7202.959	0.0600	0.0297
Рандом-250-20	250	6280	15	- (7200/нм)	<b>15</b>	28.354	0.0500	0.0357	<b>15</b>	678.819	7211.186	0.0300	0.0332
Рандом-250-30	250	9347	10	- (7200/нм)	<b>10</b>	1109.467	0.0950	0.0218	<b>10</b>	2483.681	7270.351	0.0900	0.0300
Рандом-250-40	250	12500	8	- (7200/нм)	<b>8</b>	26.584	0.0063	0.0272	<b>8</b>	44.401	7319.904	0	0
Рандом-250-50	250	15605	6	- (7200/нм)	<b>6</b>	36.364	0.1167	0.0764	<b>6</b>	1049.087	7551.006	0.0917	0.0829
Рандом-250-60	250	18660	5	- (7200/нм)	<b>5</b>	39.514	0.0100	0.0436	<b>5</b>	0.04	7279.655	0	0

Табела 19: Резултати додатних тестирања VNS-CAI и модификоване VNS методе на инстанцама код којих нису познате оптималне али су познате неке вредности

Инстанце				Cpl/Gur	VNS-CAI				модификовани VNS				
Име	$ V $	$ E $	<i>best</i>	<i>val(t)</i>	<i>sol</i>	<i>t</i>	<i>err</i>	$\sigma$	<i>sol</i>	<i>t</i>	<i>t<sub>tot</sub></i>	<i>err</i>	$\sigma$
Рандом-250-70	250	21741	4	- (7200/нм)	<b>4</b>	122.784	0.1500	0.1225	<b>4</b>	233.755	7680.901	0.0625	0.1083
Рандом-250-80	250	24836	4	- (7200/нм)	<b>4</b>	0.031	0	0	<b>4</b>	5.915	1855.069	0	0
Рандом-250-90	250	27974	3	- (7200/нм)	<b>3</b>	0.035	0	0	<b>3</b>	7.524	1454.522	0	0

## 5 Закључак

Иако је први утисак да је проблем римске доминације само историјски, овај проблем се може применити и на велики број проблема данас. Наиме, оптимизациони проблеми римске доминације могу да се примене на велики број проблема који се тичу распоређивања различитих врста ресурса. При том, могу да се примене како на статичке, тако и на динамичке проблеме. Примера ради, ако се уместо легија римског царства могу посматрати возила хитне помоћи, тада се проблем распоређивања возила хитне помоћи може еквивалентно моделирати као проблем организовања римских легија. Уместо возила хитне помоћи могу се распоређивати радио пријемници, болнице, ватрогасне станице, ресторани и многи други ресурси. Поред распоређивања статичких објеката, проблем римске доминације може се искористити приликом одбранбених стратегија или пак, приликом постављања надзорног система. Проблем ”трпи” промене којима се додају/одузимају ресурси или се увећава/умањује број пријемника (учесника).

У овом раду разматрани су проблем римске доминације и његове две модификације, проблем ограничене и проблем слабе римске доминације. У релевантној литератури постојале су две математичке формулације проблема римске доминације. Примећено је да се обе формулације могу побољшати увођењем одређених релаксација и смањивањем одређеног броја ограничења. Побољшањем је омогућено да се за већи број инстанци добије оптимална вредност. Ипак, како време решавања проблема римске доминације за одређене класе графова може бити изузетно дуго, или се, пак, неке инстанце не могу решити због ограничења у меморији, развијен је нови алгоритам, заснован на методи променљивих околина, којим се овај проблем може решити. Поред тога што нови алгоритам достиже оптимално решење за велики број инстанци изузетно брзо и квалитетно, као што се из приказаних резултата може видети, пронађене вредности су углавном боље у односу на вредности добијене метахеуристичком методом из литературе. За проблем ограничене римске доминације дефинисана је прва математичка формулација. Делови резултата овог дела истраживања објављени су [84] и [85]. Што се проблема слабе римске доминације тиче, уочено је да иако у литератури постоје две различите дефиниције овог проблема, постојећа математичке формулације

проблема не одговара ни једној. Грешка је отклоњена, доказано је да исправљена формулација одговара основној дефиницији проблема, а затим су предложени побољшана математичка формулација за коју је такође показано да одговара основној дефиницији проблема слабе римске доминације и два алгорита за решавање проблема слабе римске доминације. Први алгоритам оптимално решава проблем слабе римске доминације на блок графовима, док је други алгоритам нови алгоритам заснован на методи променљивих околина. Иако нови алгоритам представља метахеуристичку методу, на основу приказаних резултата се може се видети да представљен алгоритам достиже оптимално решење за велики број инстанци а неретко и боље вредности у односу на вредности добијене метахеуристичком методом из литературе. Делови поменутих резултата објављени су у [72], [81] и [93].

Будући да су посматрани проблеми из класе NP проблема, један од доприноса ове тезе се огледа управо у алгоритмима за решавање тих проблема, на страну што су за одређене инстанце достигнута нова најбоља решења. Допринос ове тезе такође се огледа у техникама за унапређење одговарајућих проблема, моделирању проблема чија математичка формулација није позната, као и у техникама решавања тих проблема алгоритамски. Описане технике могу се применити и на друге верзије проблема римске доминације, па тако ова теза представља увод у једно много обимније истраживање.

## Литература

- [1] Edward Luttwak. *The grand strategy of the Roman Empire: From the first century AD to the third*. JHU Press, 1979.
- [2] Michael R. Garey and David S. Johnson. A guide to the theory of np-completeness. *WH Freeman, New York*, 70, 1979.
- [3] Miodrag Živković. *Algoritmi*. Matematički fakultet, 2010.
- [4] Stephen A. Cook. The complexity of theorem-proving procedures. In *Proceedings of the third annual ACM symposium on Theory of computing*, pages 151–158. ACM, 1971.
- [5] Richard M. Karp. Reducibility among combinatorial problems. In *Complexity of computer computations*, pages 85–103. Springer, 1972.
- [6] R. Gary Parker and Ronald L. Rardin. *Discrete optimization*. Elsevier, 2014.
- [7] Teresa W. Haynes, Stephen Hedetniemi, and Peter Slater. *Fundamentals of domination in graphs*. CRC Press, New York, 1998.
- [8] Ernie J. Cockayne, Paul A. Dreyer, Sandra M. Hedetniemi, and Stephen T. Hedetniemi. Roman domination in graphs. *Discrete Mathematics*, 278(1):11–22, 2004.
- [9] Roushini Leely Pushpam and Padmapriya Sampath. Restrained roman domination in graphs. *Transactions on Combinatorics*, 4(1):1–17, 2015.
- [10] Michael A. Henning and Stephen T. Hedetniemi. Defending the roman empire — a new strategy. *Discrete Mathematics*, 266(1):239–251, 2003.
- [11] Ian Stewart. Defend the roman empire! *Scientific American*, 281:136–138, 1999.
- [12] Charles S. ReVelle and Kenneth E. Rosing. Defendens imperium romanum: a classical problem in military strategy. *American Mathematical Monthly*, pages 585–594, 2000.
- [13] Paul Andrew Dreyer Jr. *Applications and variations of domination in graphs*. PhD thesis, Citeseer, 2000.
- [14] Vincenzo Currò. *The Roman Domination Problem on Grid Graphs*. PhD thesis, Università di Catania, 2014.

- [15] Weiping Shang and Xiaodong Hu. The roman domination problem in unit disk graphs. In *Computational Science–ICCS 2007*, pages 305–312. Springer, 2007.
- [16] Chun-Hung Liu and Gerard J. Chang. Roman domination on strongly chordal graphs. *Journal of Combinatorial Optimization*, 26(3):608–619, 2013.
- [17] Aris Pagourtzis, Paolo Penna, Konrad Schlude, Kathleen Steinhöfel, David Scot Taylor, and Peter Widmayer. Server placements, roman domination and other dominating set variants. In *Foundations of Information Technology in the Era of Network and Mobile Computing*, pages 280–291. Springer, 2002.
- [18] Nader Jafari Rad and Lutz Volkmann. Changing and unchanging the roman domination number of a graph. *Utilitas Mathematica*, 89:79–95, 2012.
- [19] Mae Targhi, N. Jafari Rad, and M. Saied Moradi. Properties of independent roman domination in graphs. *Australasian Journal of Combinatorics*, 52:11–18, 2012.
- [20] Nader Jafari Rad and Lutz Volkmann. Roman domination perfect graphs. *An. Stiint. Univ. Ovidius Constanta Ser. Mat*, 19:167–174, 2011.
- [21] Robert R. Rubalcaba and Peter J. Slater. Roman dominating influence parameters. *Discrete Mathematics*, 307(24):3194–3200, 2007.
- [22] P. Roushini Leely Pushpam and T.N.M. Malini Mai. Edge roman domination in graphs. *J. Combin Math. Combin. Comput*, 69:175–182, 2009.
- [23] Robert A. Beeler, Teresa W. Haynes, and Stephen T. Hedetniemi. Double roman domination. *Discrete Applied Mathematics*, 211:23–29, 2016.
- [24] M. Chellali, T.W. Haynes, S.T. Hedetniemi, and A.A. McRae. Roman 2-domination. *Discrete Applied Mathematics*, 204:22–28, 2016.
- [25] Hossein Abdollahzadeh Ahangar, Michael A. Henning, Vladimir Samodivkin, and Ismael G. Yero. Total roman domination in graphs. *Applicable Analysis and Discrete Mathematics*, 10(2):501–517, 2016.
- [26] Lutz Volkmann. Signed total roman domination in graphs. *Journal of Combinatorial Optimization*, 32(3):855–871, 2016.
- [27] Guoliang Hao, Lutz Volkmann, and Doost Ali Mojdeh. Total double roman domination in graphs. *Communications in Combinatorics and Optimization*, 5(1):27–39, 2020.

- [28] Suitberto Cabrera García, Abel Cabrera Martínez, and Ismael G. Yero. Quasi-total roman domination in graphs. *Results in Mathematics*, 74(4):1–18, 2019.
- [29] Nasrin Dehgardi and Lutz Volkmann. Nonnegative signed total roman domination in graphs. *Communications in Combinatorics and Optimization*, 5(2):139–155, 2020.
- [30] Zehui Shao, Doost Ali Mojdeh, and Lutz Volkmann. Total roman  $\{3\}$ -domination in graphs. *Symmetry*, 12(2):268, 2020.
- [31] Jafar Amjadi, Sakineh Nazari-Moghaddam, and Seyed Mahmoud Sheikholeslami. Global total roman domination in graphs. *Discrete Mathematics, Algorithms and Applications*, 9(04):1750050, 2017.
- [32] M. Chellali, N. Jafari Rad, S.M. Sheikholeslami, and L. Volkmann. Varieties of roman domination. *Structures of Domination in Graphs*, pages 273–307, 2021.
- [33] M. Chellali, N. Jafari Rad, S.M. Sheikholeslami, and L. Volkmann. Varieties of roman domination ii. *AKCE International Journal of Graphs and Combinatorics*, 17(3):966–984, 2020.
- [34] H. Abdollahzadeh Ahangar, M.P. Álvarez, M. Chellali, Seyed Mahmoud Sheikholeslami, and Juan Carlos Valenzuela-Tripodoro. Triple roman domination in graphs. *Applied Mathematics and Computation*, 391:125444, 2021.
- [35] H. Abdollahzadeh Ahangar, Teresa W. Haynes, and J.C. Valenzuela-Tripodoro. Mixed roman domination in graphs. *Bulletin of the Malaysian Mathematical Sciences Society*, 40(4):1443–1454, 2017.
- [36] P. Roushini Leely Pushpam and S. Padmapriya. Global roman domination in graphs. *Discrete applied mathematics*, 200:176–185, 2016.
- [37] Ayotunde T. Egunjobi and Teresa W. Haynes. Perfect double roman domination of trees. *Discrete Applied Mathematics*, 284:71–85, 2020.
- [38] Doost Ali Mojdeh and Lutz Volkmann. Roman  $\{3\}$ -domination (double italian domination). *Discrete Applied Mathematics*, 283:555–564, 2020.
- [39] V. Zverovich and Anush Poghosyan. On roman, global and restrained domination in graphs. *Graphs and Combinatorics*, 27(5):755–768, 2011.
- [40] E.J. Cockayne, P.J.P. Grobler, W.R. Griundlingh, J. Mungangao, and J.H. Van Vuuren. Protection of a graph. *Utilitas Mathematica*, 67:19–32, 2005.



- [41] Erin W. Chambers, Bill Kinnersley, Noah Prince, and Douglas B. West. Extremal problems for roman domination. *SIAM Journal on Discrete Mathematics*, 23(3):1575–1586, 2009.
- [42] Odile Favaron, Hosein Karami, R. Khoeilar, and Seyed Mahmoud Sheikholeslami. On the roman domination number of a graph. *Discrete Mathematics*, 309(10):3447–3451, 2009.
- [43] Chun-Hung Liu and Gerard Jennhwa Chang. Upper bounds on roman domination numbers of graphs. *Discrete Mathematics*, 312(7):1386–1391, 2012.
- [44] B.P. Mobaraky and S.M. Sheikholeslami. Bounds on roman domination numbers of graphs. *Matematički vesnik*, 60(4):247–253, 2008.
- [45] Antoaneta Klobučar and Ivona Puljić. Some results for roman domination number on cardinal product of paths and cycles. *Kragujevac Journal of Mathematics*, 38(1):83–94, 2014.
- [46] Antoaneta Klobučar and Ivona Puljić. Roman domination number on cardinal product of paths and cycles. *Croatian Operational Research Review*, 6(1):71–78, 2015.
- [47] Derya Doğan Durgun and Emre Niyazi Toprakkaya. Roman domination of the comet, double comet, and comb graphs. *arXiv preprint arXiv:2102.07902*, 2021.
- [48] Fu Xueliang, Yang Yuansheng, and Jiang Baoqi. On the domination number of generalized Petersen graphs  $P(n, 3)$ . *Ars Combinatoria*, 84:373–383, 2007.
- [49] Xueliang Fu, Yuansheng Yang, and Baoqi Jiang. On the domination number of generalized Petersen graphs  $P(n, 2)$ . *Discrete Mathematics*, 309(8):2445–2451, 2009.
- [50] Haoli Wang, Xirong Xu, Yuansheng Yang, and Chunnian Ji. Roman domination number of generalized Petersen graphs  $P(n, 2)$ . *arXiv preprint arXiv:1103.2419*, 2011.
- [51] P. Jeyanthi and T. Saratha Devi. Edge pair sum labeling of spider graph. *Journal of Algorithms and Computation*, 45(1):25–34, 2014.
- [52] Hua-Ming Xing, Xin Chen, and Xue-Gang Chen. A note on roman domination in graphs. *Discrete mathematics*, 306(24):3338–3340, 2006.

- [53] Fu Xueliang, Yang Yuansheng, and Jiang Baoqi. Roman domination in regular graphs. *Discrete Mathematics*, 309(6):1528–1537, 2009.
- [54] Michael A. Henning. A characterization of roman trees. *Discussiones Mathematicae Graph Theory*, 22(2):325–334, 2002.
- [55] X-X Song and X-F Wang. Roman domination number and domination number of a tree. *Chinese Quarterly Journal of Mathematics*, 21(3):358, 2006.
- [56] Tadeja Kraner Šumenjak, Polona Pavlič, and Aleksandra Tepeh. On the roman domination in the lexicographic product of graphs. *Discrete Applied Mathematics*, 160(13):2030–2036, 2012.
- [57] Jan Arne Telle and Andrzej Proskurowski. Algorithms for vertex partitioning problems on partial k-trees. *SIAM Journal on Discrete Mathematics*, 10(4):529–550, 1997.
- [58] Gayla S. Domke, Johannes H. Hattingh, Stephen T. Hedetniemi, Renu C. Laskar, and Lisa R. Markus. Restrained domination in graphs. *Discrete Mathematics*, 203(1):61–69, 1999.
- [59] Gayla S. Domke, Johannes H. Hattingh, Michael A. Henning, and Lisa R. Markus. Restrained domination in trees. *Discrete Mathematics*, 211(1):1–9, 2000.
- [60] Nader Jafari Rad and Marcin Krzywkowski. On the restrained roman domination in graphs.
- [61] Mustapha Chellali and Nader Jafari Rad. Strong equality between the roman domination and independent roman domination numbers in trees. *Discussiones Mathematicae Graph Theory*, 33(2):337–346, 2013.
- [62] P. Roushini Leely Pushpam and T.N.M. Malini Mai. Weak edge roman domination in graphs. *Australasian Journal of Combinatorics*, 51:125–138, 2011.
- [63] P. Roushini Leely Pushpam and M. Kamalam. Efficient weak roman domination in graphs. *International Journal of Pure and Applied Mathematics*, 101(5).
- [64] Abel Cabrera Martínez, Luis P. Montejano, and Juan A. Rodríguez-Velázquez. Total weak roman domination in graphs. *Symmetry*, 11(6):831, 2019.
- [65] Mustapha Chellali, Teresa W. Haynes, and Stephen T. Hedetniemi. Bounds on weak roman and 2-rainbow domination numbers. *Discrete Applied Mathematics*, 178:27–32, 2014.

- [66] Mathieu Liedloff, Ton Kloks, Jiping Liu, and Sheng-Lung Peng. Roman domination over some graph classes. In *Graph-Theoretic Concepts in Computer Science*, pages 103–114. Springer, 2005.
- [67] P. Roushini Leely Pushpam and M. Kamalam. Effect of vertex deletion on the weak roman domination number of a graph. *AKCE International Journal of Graphs and Combinatorics*, 2018.
- [68] P. Roushini Leely Pushpam and T.N.M. Malini Mai. Weak roman domination in graphs. *Discussiones Mathematicae Graph Theory*, 31(1):161–170, 2011.
- [69] Magdalena Valveny, Hebert Pérez-Rosés, and Juan A. Rodríguez-Velázquez. On the weak roman domination number of lexicographic product graphs. *Discrete Applied Mathematics*, 263:257–270, 2019.
- [70] Mathieu Liedloff, Ton Kloks, Jiping Liu, and Sheng-Lung Peng. Efficient algorithms for roman domination on some classes of graphs. *Discrete Applied Mathematics*, 156(18):3400–3415, 2008.
- [71] Michel Jonathan Mizrahi. *Algoritmos y complejidad para algunos problemas de dominación*. PhD thesis, Facultad de Ciencias Exactas y Naturales. Universidad de Buenos Aires, 2014.
- [72] Marija Ivanović and Dragan Urošević. Variable neighborhood search approach for solving roman and weak roman domination problems on graphs. *Computing & Informatics*, 38(1), 2019.
- [73] Chakradhar Padamutham and Venkata Subba Reddy Palagiri. Algorithmic aspects of roman domination in graphs. *Journal of Applied Mathematics and Computing*, 64(1):89–102, 2020.
- [74] E.J. Cockayne, P.A. Dreyer Jr., S.M. Hedetniemi, S.T. Hedetniemi, and A.A. McRae. The algorithmic complexity of roman domination.
- [75] A.P. Burger, E.J. Cockayne, W.R. Grundlingh, C.M. Mynhardt, J.H. Van Vuuren, and W. Winterbach. Finite order domination in graphs. *Journal of Combinatorial Mathematics and Combinatorial Computing*, 49:159–176, 2004.
- [76] EJ Cockayne, O Favaron, and CM Mynhardt. Secure domination, weak roman domination and forbidden subgraphs. *Bull. Inst. Combin. Appl*, 39:87–100, 2003.

- [77] Michael A. Henning. Defending the roman empire from multiple attacks. *Discrete Mathematics*, 271(1):101–115, 2003.
- [78] Polona Pavlič and Janez Žerovnik. Roman domination number of the cartesian products of paths and cycles. *The Electronic Journal of Combinatorics*, 19(3):P19, 2012.
- [79] Limin Wang, Yalin Shi, Zhao Zhang, Zan-Bo Zhang, and Xiaoyan Zhang. Approximation algorithm for a generalized roman domination problem in unit ball graphs. *Journal of Combinatorial Optimization*, 39(1):138–148, 2020.
- [80] Chih-Shan Liu Sheng-Lung Peng and Chuan Yi Tang. Weak roman domination on block graphs. *Proceedings of the 27th Workshop on Combinatorial Mathematics and Computation Theory, Providence University, Taichung, Taiwan, April 30-May 1*, pages 86–89., 2010.
- [81] Marija Ivanović and Dragan Urošević. A new linear-time algorithm for computing the weak roman domination number of a block graph. In *XIII Balkan Conference on Operational Research Proceedings*, page 25. FON, 2018.
- [82] Mathieu Chapelle, Manfred Cochefert, Jean-François Couturier, Dieter Kratsch, Mathieu Liedloff, and Anthony Perez. Exact algorithms for weak roman domination. In *Combinatorial Algorithms*, pages 81–93. Springer, 2013.
- [83] A.P. Burger, A.P. de Villiers, and J.H. van Vuuren. A binary programming approach towards achieving effective graph protection. In *Proceedings of the 2013 ORSSA Annual Conference*, pages 19–30. ORSSA, 2013.
- [84] Marija Ivanović. Improved mixed integer linear programming formulations for roman domination problem. *Publications de l’Institut Mathématique*, 99(113):51–58, 2016.
- [85] Marija Ivanović. A mixed integer linear programming formulation for restrained roman domination problem. *Theory and Applications of Mathematics & Computer Science*, 5(2):110–115, 2015.
- [86] Christian Blum and Andrea Roli. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Computing Surveys (CSUR)*, 35(3):268–308, 2003.
- [87] F. Harary. *Graph theory*. Addison-Wesley MR0256911, 1969.

- [88] F. Harary. A characterization of block-graphs. *Canadian Mathematical Bulletin*, 6(1):1–6, 1963.
- [89] Nenad Mladenović and Pierre Hansen. Variable neighborhood search. *Computers & Operations Research*, 24(11):1097–1100, 1997.
- [90] Nenad Mladenovic. A variable neighborhood algorithm-a new metaheuristic for combinatorial optimization. In *papers presented at Optimization Days*, page 112, 1995.
- [91] Pierre Hansen and Nenad Mladenović. Variable neighborhood search: Principles and applications. *European journal of operational research*, 130(3):449–467, 2001.
- [92] Pierre Hansen, Nenad Mladenović, and Dionisio Perez-Britos. Variable neighborhood decomposition search. *Journal of Heuristics*, 7(4):335–350, 2001.
- [93] Marija Ivanović. Improved integer linear programming formulation for weak roman domination problem. *Soft Computing*, 22(19):6583–6593, 2018.

## Биографија аутора

Марија Ивановић рођена је 6.11.1982. године у Пожаревцу. Основну школу "Јован Цвијић" завршила је у Костолцу док је Гимназију "Јован Шербановић" завршила у Пожаревцу. На смеру Нумеричка математика и оптимизација, на Математичком факултету, дипломирала је 2007. године. На истом смеру наредне године уписала је мастер студије а две године касније и докторске студије. Мастер рад под темом "Теорија игара - игре сусретања и игре налажења" одбранила је септембра 2011. године и тиме стекла звање Мастер-математичар.

Марија Ивановић је била учесница European Study Group with Industry ESGI 99 у организацији Природно-Математичког факултета Универзитета у Новом Саду као члан тима који се бавио развојем математичког модела за решавање проблема вишенивоског транспорта и дистрибуције. Од 2009.-2021. године била је учесник на пројекту под редним бројем TR36006 "Математички модели и методе оптимизације са применама", финансиран од стране Министарства за науку Републике Србије. У оквиру докторских студија учествовала је у Темпус пројекту SEE Doctoral Studies in Mathematical Sciences 44703 - TEMPUS-1 -2008- 1 –BATEMPUS - JPCR. На Математичком факултету Универзитета у Београду била је запослена од 2008-2021. године, где је држала вежбе на Катедри за нумеричку математику и оптимизацију из неколико различитих предмета на основним и мастер студијама. Од децембра 2021. године запослена је на Институту за Физику у Београду.

Објављени радови:

1. М. Ивановић, *Improved integer linear programming formulation for weak roman domination problem*, Soft Computing, 22: 6583–6593, 2018.  
<https://doi.org/10.1007/s00500-017-2706-4>
2. М. Ивановић, Д. Уршевић, *Variable neighborhood search approach for solving roman and weak roman domination problems on graphs*, Computing & Informatics, 38(1): 57-84, 2019.
3. М. Ивановић, *A mixed integer linear programming formulation for restrained*

*roman domination problem* , Theory of Applications of Mathematics & Computer Science, 5(2): 110-115, 2015.

4. М.Ивановић, *A mixed integer linear programming formulation for roman domination problem*, Publications de l' Mathematique, 99(113) : 51-58, 2016.
5. М. Ивановић, А. Савић, Д. Урошевић, Ђ. Дугошија, *A new Variable Neighborhood Search approach for solving Dynamic Memory Allocation Problem*, Yugoslav Journal of Operations Research, 28(3): 291-314, 2018.
6. L. Chen, H. Jiang, Y. Shao, M. Ivanovic1, *Dominator and total dominator coloring in vague graphs*, Engineering and Applied Science Letters, 2(2): 10-17, 2019, doi:10.30538/psrp-easl2019.0017

Саопштења са међународног скупа штампано у целини:

7. М. Ивановић, Ђ. Дугошија, А. Савић, Д. Урошевић, *A New Integer Linear Formulation for a Memory Allocation Problem*, Proceedings of XI Balcan Conference on Operational Research, стране 284-288. Београд и Златибор, 7-11, септембар, 2013.
8. М. Ивановић, Д. Урошевић, *Book of abstracts of the 4th International Conference on Variable Neighborhood Search*, ISBN 978-84-617-5068-9. страна 23. Малага, Шпанија, 3-5. октобар, 2016.

Саопштења са међународног скупа штампано у изводу:

9. М.Ивановић, Д.Урошевић, *Variable Neighborhood Search Solution for some variants of the Roman Domination problem*, 4th International VNS conference, Book of abstracts p.23 Малага, Шпанија, 2016.

10. J.Бримберг, М. Ивановић, Н.Младеновић, Д.Урошевић, Primal-Dual VNS for large p-center problem, ICVNS2018 – 6th International Conference on Variable Neighborhood Search p.47, Ситонија, Грчка, 2018.
11. М. Радмиловић-Рађеновић, М. Ивановић, Бранислав Рађеновић, *The Effect of the Antenna Design on the Characteristics of Microwave Ablation Treatment on Liver Cancer*, MAS 16th International European Conference On Mathematics, Engineering, Natural & Medical Sciences, Mardin, Turkey, February 22-23, 2022, Conference program, pp.6.
12. М. Радмиловић-Рађеновић, Н. Бошковић, М. Ивановић, Б. Радјеновић, *Finite Element Analysis of the Efficiency of Multislot Antenna in Microwave Tissue Ablation*, 6-th Ankara International Congress on Scientific , Ankara, Turkey, April 1-3, 2022, Congress program, pp. 51.

#### Збирка задатака

13. Збирка задатака 1. А. Делић, З. Дражић, С. Живановић, М. Ивановић, *Збирка решених задатака из Увода у нумеричку математику*, издавач: Математички факултет, Универзитет у Београду, Србија.



Прилог 1.

## Изјава о ауторству

Потписани-а Марија Ивановић  
број индекса 2028/2018

### Изјављујем

да је докторска дисертација под насловом

Нови приступи у решавању оптимизационог  
проблема римске доминације на графовима

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

Потпис докторанда

У Београду, 1.06.2022.

Марија Ивановић

Прилог 2.

**Изјава о истоветности штампане и електронске  
верзије докторског рада**

Име и презиме аутора Марија Иванковић  
Број индекса 2028/2018  
Студијски програм МАТЕМАТИКА  
Наслов рада Нови приступи у решавању оптимизационог проблема  
римске доминације на графовима  
Ментор др. Александар Савић  
Потписани/а Марија Иванковић

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, 1.06.2022.

Марија Иванковић

Прилог 3.

## Изјава о коришћењу

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

Нови приступи у решавању оптимизационог проблема римске доминације на графовима

која је моје ауторско дело.

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

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

1. Ауторство
2. Ауторство - некомерцијално
3. Ауторство – некомерцијално – без прераде
4. Ауторство – некомерцијално – делити под истим условима
5. Ауторство – без прераде
6. Ауторство – делити под истим условима

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

Потпис докторанда

У Београду, 1.06.2022.

Milovan Dukić