

UNIVERZITET U BEOGRADU  
SAOBRAĆAJNI FAKULTET

Nenad P. Bjelić

MODELI ZA OPERATIVNO UPRAVLJANJE  
JEDNOM KLASOM PRETOVARNIH  
SREDSTAVA U USLOVIMA DINAMIČKE  
POJAVE ZADATAKA

Doktorska disertacija

Beograd, 2014

UNIVERSITY OF BELGRADE  
FACULTY OF TRANSPORT AND TRAFFIC ENGINEERING

Nenad P. Bjelić

MODELS FOR OPERATIONAL PLANNING  
OF A MATERIALS HANDLING SYSTEM  
WITH DYNAMIC TASK ARRIVALS

Doctoral Dissertation

Belgrade, 2014

MENTOR: **dr Milorad Vidović**, dipl.inž.saob., redovni profesor, Univerzitet u Beogradu, Saobraćajni fakultet

ČLANOVI KOMISIJE: **dr Katarina Vukadinović**, dipl.inž.saob., redovni profesor, Univerzitet u Beogradu, Saobraćajni fakultet

**dr Nenad Mladenović**, dipl.mat., naučni savetnik, Matematički institut Srpske akademije nauka i umetnosti

DATUM ODBRANE: \_\_\_\_\_

## MODELI ZA OPERATIVNO UPRAVLJANJE JEDNOM KLASOM PRETOVARNIH SREDSTAVA U USLOVIMA DINAMIČKE POJAVE ZADATAKA

Rezime: Uštede ostvarene realizacijom jedne upravljačke odluke na operativnom nivou, svakako nisu dovoljan motiv odgovornima u logističkim, ali i svim ostalim, sistemima da se u dovoljnoj meri posvete racionalizaciji upravljačkih odluka. Međutim, kada se u obzir uzme frekventnost realizacije operativnih upravljačkih odluka jasno je da efikasna realizacija operativnih zadataka može predstavljati razliku između uspešnog i neuspešnog poslovanja kompanije, ili sticanje značajne konkurentske prednosti. Iz tog razloga predmet ovog istraživanja je unapređenje efikasnosti upravljačkih odluka na operativnom nivou posebne klase pretovarnih sistema koja se karakteriše dinamičkom pojavom zadataka u vremenu. Posmatrani problem se može predstaviti kao uopštenje problema putujućeg servisera, u kome se javlja heterogena flota servisera a zadaci se karakterišu postojanjem vremenskih prozora. Obzirom da su se pretovarni sistemi usled povećanja intenziteta robnih tokova razvili u veoma kompleksne upravljačke strukture, donošenje upravljačkih odluka je praktično nemoguće bez upotrebe računara. U tom smislu, u ovom istraživanju je pretpostavka o povećanju efikasnosti realizacije pretovarnih operacija zasnovana na implementaciji upravljačkih odluka koje su rezultat matematičkih modela i algoritma predstavljenih u radu, a koji se mogu implementirati u softver koji se koristi kao pomoć ljudskim donosiocima odluka. U istraživanju su za rešavanje problema predstavljeni matematički model mešovitog celobrojnog linearnog programiranja i dva heuristička algoritma zasnovana na unapređenju odluke primenom metode promenljivih okolina. Efikasnost predloženih upravljačkih modela i generisanih odluka je testirana na benčmark instancama, na osnovu čijih rešenja se može uočiti praktična primenljivost predloženih modela.

Ključne reči: pretovarna sredstva, operativno upravljanje, VNS, linearno programiranje, modeliranje, problem putujućeg servisera

Naučna oblast: Saobraćajno inženjerstvo

Uža naučna oblast: Rukovanje materijalom i eko logistika

UDK broj: 621.86/.87(043.3)

## MODELS FOR OPERATIONAL PLANNING OF A MATERIALS HANDLING SYSTEM WITH DYNAMIC TASKS ARRIVALS

Abstract: Level of savings made from one efficient operational control decision is not sufficiently large to motivate logistics, or any other, managers to be interested in improving efficiency of operational control decisions. Nevertheless, the frequency of making such decisions multiplies saving effects and usually differentiates successful from unsuccessful companies. Accordingly, the focus of this thesis is on improving operational control decisions made in a specific material handling systems which are characterized by dynamic tasks arrivals. Respected problem is considered as a generalization of the travelling repairman problem where fleet of repairmen consists of more than one repairman and where tasks are with time windows. Bearing in mind that, due to the significant increase of freight movements, control of material handling systems became very complex structures their efficient control cannot be realized without some kind of decision support system. Therefore, in this research it is supposed that an improvement of material handling processes is based on an improvement of operational control decisions which are outputs of appropriate mathematical models and algorithms implemented in a decision support system. In that sense, considered problem is solved by mixed integer linear programming mathematical model and two heuristic algorithms based on the variable neighborhood search metaheuristic framework. Efficiency of proposed models is tested on several sets of benchmark instances which proved their practical applicability.

Key words: materials handling, operational control, VNS, linear programming, modeling, traveling repairmen problem

Scientific field: Traffic and Transport

Field of Academic Expertise: Materials handling and eco-logistics

UDC number: 621.86/.87(043.3)

# Sadržaj

Spisak slika.....	I
Spisak tabela .....	III
Spisak skraćenica.....	IV
<b>1. Uvod .....</b>	<b>1</b>
<b>2. Hijerarhijski nivoi planiranja i upravljanja logističkim sistemima .....</b>	<b>6</b>
2.1. Strateško, taktičko i operativno planiranje .....	6
2.2. Nivoi planiranja u logističkim sistemima.....	9
2.2.1. Nivoi odlučivanja u kontejnerskom terminalu.....	11
2.2.2. Nivoi odlučivanja u distributivnoj mreži.....	14
2.3. Odnos planiranja i upravljanja.....	15
2.4. Karakteristike zadataka operativnog upravljanja pretovarnim sistemima.....	16
<b>3. Definicija i primenljivost problema .....</b>	<b>20</b>
3.1. Specifičnosti pretovarnog sistema sa kumulativnom ciljnom funkcijom .....	20
3.2. Primeri pretovarnih sistema sa kumulativnom ciljnom funkcijom.....	24
3.2.1. Specijalizovani pretovarni uređaji za istovar šljunka iz potisnica.....	25
3.2.2. Sistemi unutrašnjeg transporta sa automatski vođenim vozilima .....	26
3.2.3. Isporuka pošiljaka u distributivnim sistemima.....	27
3.2.4. Vremensko raspoređivanje brodova na pristajalištu .....	28
<b>4. Upravljanje pretovarnim sistemima sa kumulativnom ciljnom funkcijom</b>	<b>31</b>
4.1. Problemi kombinatorne optimizacije.....	31
4.2. Osnovni pojmovi u analizi algoritama za upravljanje pretovarnim sistemima sa kumulativnom ciljnom funkcijom.....	32
4.2.1. Asimptotska notacija i kompleksnost algoritama .....	32
4.2.2. P, NP i NP-complete klase problema .....	36
4.3. Načini upravljanja pretovarnim sistemima sa kumulativnom ciljnom funkcijom .....	39
4.3.1. Upravljanje koje se odnosi na ceo planski period.....	40
4.3.1.2. Egzaktne metode .....	41
4.3.1.3. Aproksimativne i heurističke metode.....	44
4.3.2. Upravljanje sistemom sa sukcesivnim ažuriranjem informacija .....	48
4.3.3. Upravljanje sistemom sa nultim planskim periodom .....	50
<b>5. Modeliranje pretovarnih sistema sa kumulativnom ciljnom funkcijom i dinamičkom pojavom zadataka.....</b>	<b>52</b>
5.1. Problem upravljanja pretovarnim sistemima sa kumulativnom ciljnom funkcijom kao generalizacija problema putujućeg servisera.....	53
5.2. Pregled relevantne literature i dostignuća u rešavanju hetTRPTW-a .....	55

5.2.1. Sličnost sa drugim problemima.....	55
5.2.2. Dostignuća u radovima koji tretiraju rešavanje TRP klase problema.....	58
5.3. Modeliranje hetTRPTW-a kao problema mešovitog celobrojnog linearnog programiranja .....	62
5.4. Najmanja vrednost rešenja hetTRPTW-a.....	67
5.5. Heuristički algoritam za rešavanje hetTRPTW-a .....	69
5.5.1. Pojam okoline trenutnog rešenja problema i njegove pretrage.....	69
5.5.2. Osnovni koncepti realizacije VNS-a.....	71
5.5.2.1. Metoda promenljivog spusta - VND .....	72
5.5.2.2. Redukovana metoda promenljivih okolina – RVNS.....	74
5.5.2.3. Bazični metod promenljivih okolina - BVNS.....	75
5.5.3. VNS algoritam za rešavanje hetTRPTW-a .....	76
5.5.3.1. Kodiranje rešenja .....	76
5.5.3.2. Proračun vrednosti ciljne funkcije .....	77
5.5.3.3. Početno rešenje problema .....	78
5.5.3.4. Algoritam promenljivog spusta – VND .....	78
5.5.3.5. Redukcija okolina u VND-u .....	82
5.5.3.6. Razmrdavanje rešenja.....	88
5.5.3.7. Uslovi za završetak algoritma.....	90
5.6. Numerički eksperimenti efikasnosti modela .....	91
<b>6. Primena predloženih modela u upravljanju pretovarnim sredstvima za istovar šljunka iz potisnica.....</b>	<b>107</b>
<b>7. Zaključak .....</b>	<b>122</b>
<b>Literatura.....</b>	<b>125</b>
<b>Biografija autora .....</b>	<b>130</b>
<b>Izjava o autorstvu.....</b>	<b>Error! Bookmark not defined.</b>
<b>Izjava o istovetnosti štampane i elektronske verzije doktorskog rada.....</b>	<b>Error! Bookmark not defined.</b>
<b>Izjava o korišćenju.....</b>	<b>Error! Bookmark not defined.</b>

# Spisak slika

<b>Slika 1.1.</b>	Hronološki prikaz nivoa trgovine na svetskom nivou .....	2
<b>Slika 1.2.</b>	Hronološki prikaz nivoa trgovine Republike Srbije u periodu 2004-2012 ..	2
<b>Slika 1.3.</b>	Struktura i odnos između poglavlja istraživanja .....	4
<b>Slika 2.1.</b>	Osnovne funkcije različitih nivoa planiranja .....	8
<b>Slika 2.2.</b>	Nivoi planiranja u kontejnerskom terminalu .....	13
<b>Slika 2.3.</b>	Odnos hijerarhijskih nivoa planiranja i upravljanja .....	15
<b>Slika 3.1.</b>	Primer efekta odstupanja od FIFO principa opsluge .....	22
<b>Slika 3.2.</b>	Primer uticaja dinamičke pojave zadataka na upravljanje pretovarnim sistemom .....	23
<b>Slika 3.3.</b>	Primeri bagera za eksploataciju šljunka sa dna rečnog korita .....	25
<b>Slika 3.4.</b>	Primer pretovarnih sredstava za istovar šljunka iz potisnica .....	25
<b>Slika 3.5.</b>	Primer izgleda AGV-a i primene u skladištima i lukama .....	27
<b>Slika 3.6.</b>	Specifičnosti problema vremenskog raspoređivanja brodova i predmeta istraživanja .....	30
<b>Slika 4.1.</b>	Prikaz asimptotske analize kompleksnosti algoritama .....	35
<b>Slika 4.2.</b>	Primer redukcije problema u okviru NP-a .....	38
<b>Slika 4.3.</b>	Klasifikacija optimizacionih metoda .....	41
<b>Slika 4.4.</b>	Primer klasifikacije metaheuristika .....	47
<b>Slika 4.5.</b>	Modifikovani prikaz vremenskog okvira pojave pojedinih metaheuristika prema Talbi (2009) .....	47
<b>Slika 4.6.</b>	Prikaz upravljanja sistemom sa sukcesivnim ažuriranjem informacija ...	49
<b>Slika 4.7.</b>	Rolling horizon upravljanje sistemima .....	49
<b>Slika 5.1.</b>	Optimalno rešenje hetTRPTW za rc_202.2 instancu .....	67
<b>Slika 5.2.</b>	Idealna raspodela vremena pojave zadataka .....	68
<b>Slika 5.3.</b>	Primer kompletnih Or-opt 4 i Or-opt 6 okolina .....	70
<b>Slika 5.4.</b>	Primer primene VND algoritma .....	73
<b>Slika 5.5.</b>	Primer ugnježdenog načina pretrage okolina .....	74
<b>Slika 5.6.</b>	Or-opt 2 korak .....	80
<b>Slika 5.7.</b>	Swap korak .....	80
<b>Slika 5.8.</b>	Relocation korak .....	81
<b>Slika 5.9.</b>	Vremenski prikaz rešenja instance rc 202.2 .....	84
<b>Slika 5.10.</b>	Ugnježdavanje $\mathcal{N}_{relocate k'}$ okolina .....	88
<b>Slika 5.11.</b>	Primer razmrđavanja u VNS algoritmu .....	89
<b>Slika 5.12.</b>	Kretanje centara lokalne pretrage po prostoru rešenja .....	90



<b>Slika 5.13.</b> Relativna razlika parametara VNS i RdVNS algoritma za slučaj Ashayer-ovog skupa benčmark instanci.....	105
<b>Slika 5.14.</b> Relativna razlika parametara VNS i RdVNS algoritma za slučaj Dumas-ovog skupa benčmark instanci .....	105
<b>Slika 5.15.</b> Relativna razlika parametara VNS i RdVNS algoritma za slučaj Solomon-ovog skupa benčmark instanci.....	105
<b>Slika 5.16.</b> Relativna razlika parametara VNS i RdVNS algoritma za slučaj Langevin-ovog skupa benčmark instanci .....	106
<b>Slika 6.1.</b> Layout posmatranog distributivnog područja.....	108
<b>Slika 6.2.</b> Izgled raspodele i funkcije verovatnoća korišćene beta raspodela.....	111
<b>Slika 6.3.</b> Primer ravnomerno raspoređene i grupisane pojave zadataka .....	111
<b>Slika 6.4.</b> Relativno odstupanje vrednosti ciljnih funkcija rešenja RdVNS algoritma od rešenja VNS algoritma.....	120
<b>Slika 6.5.</b> Efekti primene redukovanih okolina za rešavanje HDPA-a za slučaj ravnomerno raspodeljenih vremena između pojave zadataka .....	121
<b>Slika 6.6.</b> Efekti primene redukovanih okolina za rešavanje HDPA-a kada su vremena između pojave zadataka raspodeljena prema beta raspodeli .....	121

## Spisak tabela

<b>Tabela 4.1.</b> Klase efikasnosti algoritama.....	35
<b>Tabela 4.2.</b> Zavisnost vremena rešavanja problema od njegove veličine .....	42
<b>Tabela 4.3.</b> Maksimalni red veličina optimizacionih problema rešenih nekom egzaktnom metodom .....	44
<b>Tabela 5.1.</b> Relevantne karakteristike posmatranih problema.....	57
<b>Tabela 5.2.</b> Postavka rc_202.2 benčmark instance .....	66
<b>Tabela 5.3.</b> Prosečne vrednosti ciljne funkcije za slučaj Ashayer-ovog skupa benčmark instanci .....	94
<b>Tabela 5.4.</b> Prosečne vrednosti ciljne funkcije za slučaj Dumas-ovog skupa benčmark instanci .....	95
<b>Tabela 5.5.</b> Prosečne vrednosti ciljne funkcije za slučaj Solomon-ovog skupa benčmark instanci .....	96
<b>Tabela 5.6.</b> Prosečne vrednosti ciljne funkcije za slučaj Langevin-ovog skupa benčmark instanci .....	97
<b>Tabela 5.7.</b> Prosečne vreme realizacije algoritama za slučaj Ashayer-ovog skupa benčmark instanci .....	99
<b>Tabela 5.8.</b> Prosečne vreme realizacije algoritama za slučaj Dumas-ovog skupa benčmark instanci .....	100
<b>Tabela 5.9.</b> Prosečne vreme realizacije algoritama za slučaj Solomon-ovog skupa benčmark instanci .....	101
<b>Tabela 5.10.</b> Prosečne vreme realizacije algoritama za slučaj Langevin-ovog skupa benčmark instanci .....	102
<b>Tabela 6.1.</b> Najkraća rastojanja između deponija šljunka.....	109
<b>Tabela 6.2.</b> Karakteristike pretovarnih sredstva relevantne za njihovo upravljanje	110
<b>Tabela 6.3.</b> Podaci o zadacima u toku planskog perioda za slučaj 10 zadataka .....	113
<b>Tabela 6.4.</b> Referentne veličine HDAP ciljne funkcije za slučaj vremena između pojava pretovarnih zadataka prema ravnomernoj raspodeli .....	115
<b>Tabela 6.5.</b> Referentne veličine HDAP ciljne funkcije za slučaj vremena između pojava pretovarnih zadataka prema beta raspodeli.....	116
<b>Tabela 6.6.</b> Referentne veličine vremena rešavanja HDAP instanci za slučaj vremena između pojava pretovarnih zadataka prema ravnomernoj raspodeli.....	117
<b>Tabela 6.7.</b> Referentne veličine vremena rešavanja HDAP instanci za slučaj vremena između pojava pretovarnih zadataka prema beta raspodeli.....	118

## Spisak skraćenica

- 3PL** kompanija koja pruža logističke usluge drugim kompanijama (eng. 3rd Party Logistics service provider)
- AGV** automatski vođena vozila (eng. Automated Guided Vehicles)
- B&B** metoda grananja i ograničavanja (eng. Branch and Bound)
- B&DSP** problem vremenskog raspoređivanja vozača i autobusa (eng. Bus and Driver Scheduling Problem), itd.
- BI** algoritam najvećeg poboljšanja (eng. Best Improvement)
- BQCSP** problem simultanog vremenskog raspoređivanja pristajališta i lučkih dizalica (eng. Berth and Quay Crane Scheduling Problem),
- BSP** problem vremenskog raspoređivanja pristajališta (eng. Berth Scheduling Problem)
- BVNS** bazični metod promenljivih okolina (eng. Basic Variable Neighborhood VNS)
- CCVRP** kumulativni kapacitivni problem rutiranja (eng. Cumulative Capacitative Vehicle Routing Problem)
- CP** programiranje ograničavanjem (eng. Constraint Programming)
- CPU** centralna procesorska jedinica (eng. Central Processor Unit)
- CSP** problem vremenskog raspoređivanja radne snage (eng. Crew Scheduling Problem),
- CumVRP** kumulativni problem rutiranja (eng. Cumulative Vehicle Routing Problem)
- CVRP** kapacitivni problema rutiranja (eng. Capacitated Vehicle Routing Problem)
- DMP** problem dostavljača (eng. Deliveryman Problem)
- DMPTW** problem dostavljača sa vremenskim prozorima (eng. Deliveryman Problem with Time Windows)
- DSS** sistemi za podršku u odlučivanju (eng. Decision Support Systems)
- EOS** veća količina manji jedinični troškovi (eng. Economy of Scale)
- FI** algoritam prvog poboljšanja (eng. First Improvement)
- GRASP** prilagodljiva, pohlepna i slučajna procedura (eng. Greedy Randomized Adaptive Search Procedure)

<b>HDAP</b>	problem alokacije pretovarnih sredstava(eng. Handling Dvices Allocation Problem)
<b>hetTRPTW</b>	problem heterogene flote putujućih servisera sa vremenskim prozorima (eng. heterogenous Traveling Repairmen Problem with Time Windows)
<b>homTRPTW</b>	problem homogene flote putujućih servisera sa vremenskim prozorima (eng. homogenous Traveling Repairmen Problem with Time Windows)
<b>HOSP</b>	problem vremenskog raspoređivanja operacija u mrežnom čvoru (eng. Hub Operation Scheduling Problem), itd.
<b>LB</b>	donja granica vrednosti rešenja problema (eng. lower bound)
<b>LSP</b>	problem vremenskog raspoređivanja lokomotiva (eng. Locomotive Scheduling Problem),
<b>MA</b>	memetski algoritam (eng. Memetic Algorithm)
<b>MILP</b>	mešovito celobrojno linearno programiranje(eng. Mixed Integer Linear Programming)
<b>MLP</b>	problem minimalnog kašnjenja (eng. Minimum Latency Problem)
<b>NP</b>	nepolinomijalno vreme rešavanja (eng. non Polinomial in Time)
<b>NP</b>	problem rešiv algoritmom u polinomijalnom vremenu nedeterminističkom Turingovom mašinom (eng. Non-deterministic Polinomial-time algorithm)
<b>P</b>	problem rešiv algoritmom u polinomijalnom vremenu determinističkom Turingovom mašinom (eng. Polinomial-time algorithm)
<b>QCSP</b>	problem vremenskog raspoređivanja lučkih dizalica (eng. Quey Crane Scheduling Problem),
<b>RVND</b>	metoda promenljivog spusta sa slučajnim rasporedom okolina (eng. Variable Neighborhood Descent with Random neighborhood ordering)
<b>RVNS</b>	redukovana metoda promenljivih okolina (eng. Reduced Variable Neighborhood Search)
<b>TEU</b>	količina robe koja odgovara količini smeštenoj u ISO kontejner dužine 20 stopa (eng. Twenty feet Equivalent Unit)
<b>TRP</b>	problem putujućeg servisera (eng. Traveling Repairman Problem)
<b>TRPTW</b>	problem putujućeg servisera sa vremenskim prozorima (eng. Traveling Repareman Problem with Time Windows)

- TSP** problem trgovačkog putnika (eng. Traveling Salesman Problem)
- TSPTW** problem trgovačkog putnika sa vremenskim prozorima (eng. Traveling Salesman Problem with Time Windows)
- UB** gornja granica vrednosti rešenja problema (eng. upper bound)
- VND** metoda promenljivog spusta (eng. Variable Neighborhood Descent)
- VNS** metoda promenljivih okolina - VNS (eng. Variable Neighborhood Search)
- VRP** problem rutiranja sredstava (eng. Vehicle Routing Problem)
- VRPTW** problem rutiranja sredstava sa vremenskim prozorima (eng. Vehicle Routing Problem)
- VSP** problem vremenskog raspoređivanja vozila (eng. Vehicle Scheduling Problem),
- WRRP** problem rutiranja servisne opreme (eng. Workover Rig Routing Problem)
- WTO** Svetska trgovinska organizacija (eng. World Trade Organization)

## **1. Uvod**

---

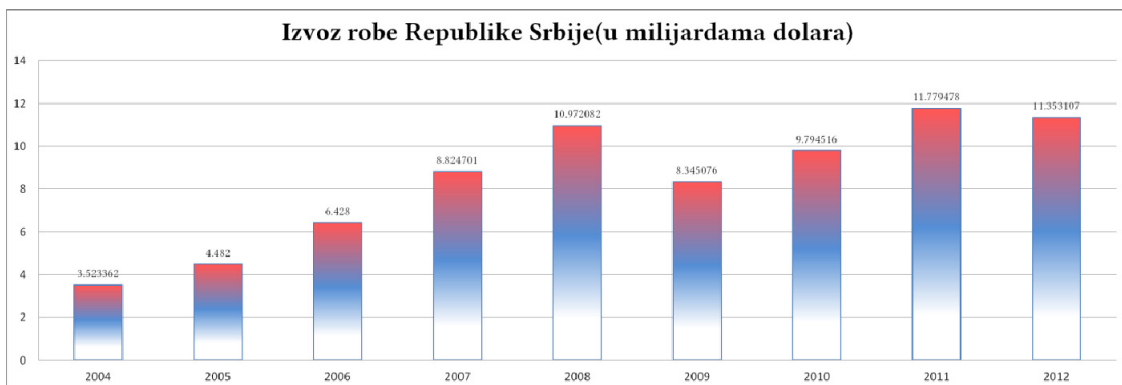
Uštede u troškovima radne snage i efikasni transportni lanci koji omogućavaju brzo i troškovno prihvatljivo dopremanje robe na ciljna tržišta samo su neki od razloga koji su doveli do izmeštanja velikog broja proizvodnih delova kompanija iz matičnih regija, u kojima imaju velike tržišne udele i prepoznatljive brendove, u zemlje udaljene nekoliko hiljada, pa i desetina hiljada kilometara. Vremenom je, u početku veoma siromašna, radna snaga na udaljenim destinacijama postajala platežno sposobnija pa im je roba kompanija koje nisu svoju proizvodnju izmestile u udaljene zemlje postala interesantna i platežno prijemčiva. Kao rezultat, nivo robne razmene rastao je iz godine u godinu u oba smera i osim u periodu izbijanja svetske krize, tj. na kraju 2008. i tokom 2009. ovakav trend je i dalje prisutan. Na slici 1.1. je prikazan grafik koji odlikava odnose u vrednostima izvezene robe na svetskom nivou i nivoima trgovinski najrazvijenijih regiona sveta. Kao što se može videti sa grafika nivo svetske trgovinske razmene je sa početnih cca. 2000 milijardi dolara u 1980. porastao na više od 18000 milijardi dolara u 2012. godini. Pored toga može se uočiti i izrazit porast nivoa izvoza iz Kine, kao tipičnog predstavnika zemalja u koju je većina kompanija izmestila

proizvodne linije, koja je u periodu od 2006. do 2012. nivo izvoza povećala za više nego duplo.



**Slika 1.1.** Hronološki prikaz nivoa trgovine na svetskom nivou (WTO\*)

Porast izvoza, tj. međunarodne robne razmene nije karakteristika samo internacionalnog tržišta. Naime, iako se to na prvi pogled ne bi reklo na osnovu trenutnog stanja privrede, u periodu za koji su dostupni relevantni podaci, tj. u periodu od 2004.-2012. Republika Srbija je svoj izvoz povećala sa početnih 3.52 milijarde na 11.35 milijardi dolara, što je uočljivo na slici 1.2.



**Slika 1.2.** Hronološki prikaz nivoa trgovine Republike Srbije u periodu 2004-2012 (WTO\*)

Ovakve promene na globalnom i lokalnom nivou uslovile su intenziviranje logističkih procesa, a samim tim i pretovarnih operacija, kao neizostavnog elementa u savladavanju prostorne razdvojenosti mesta proizvodnje i mesta potrošnje. Sa povećanjem intenziteta tokova u logističkim terminalima i sve strožijim zahtevima

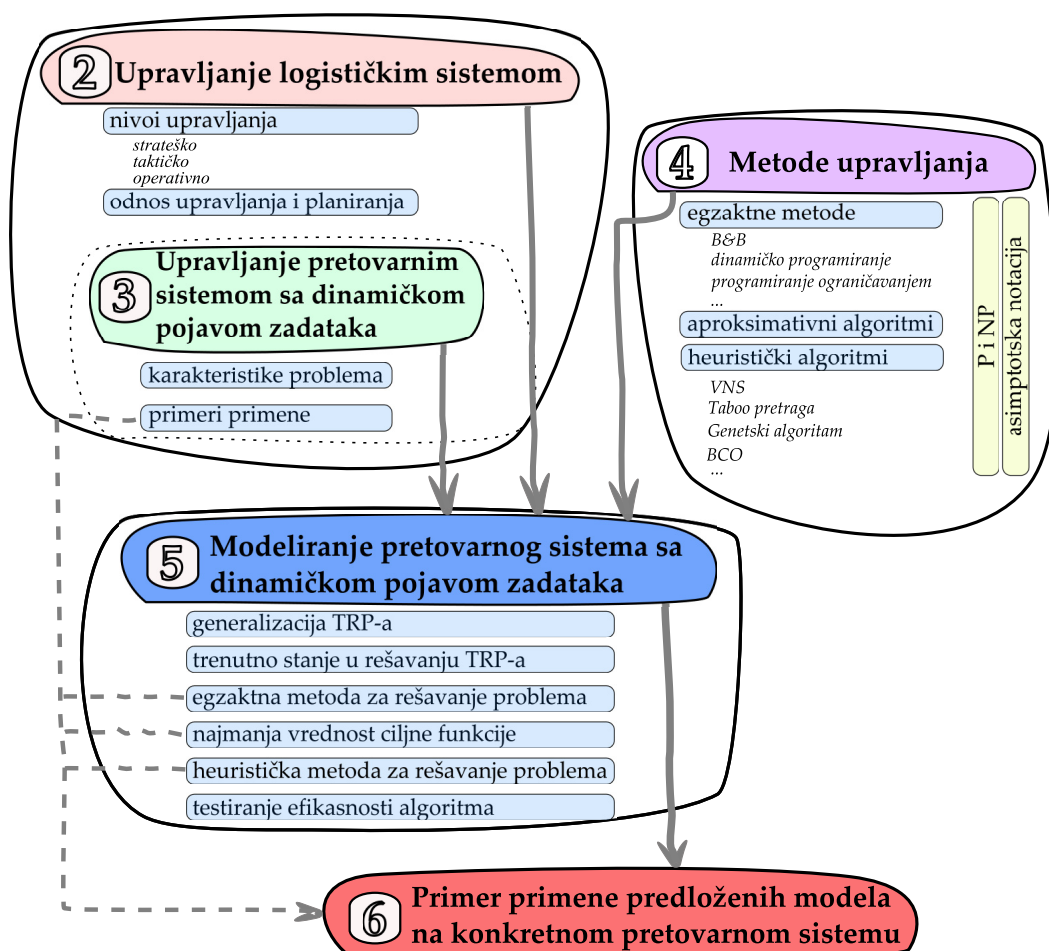
\*Podaci su preuzeti 06.03.2014. sa web adrese [http://www.wto.org/english/res\\_e/statis\\_e/statis\\_bis\\_e.htm](http://www.wto.org/english/res_e/statis_e/statis_bis_e.htm)

klijenata u pogledu kvaliteta očekivane usluge jasno je da procesi u terminalima postaju sve kompleksniji i zahtevniji za efikasnu realizaciju. Intenzitet logističkih, a sa aspekta ovog rada naročito pretovarnih procesa, uslovio je da je efikasno upravljanje ovakvim sistemima jedino moguće realizovati kombinacijom iskusnog i obučenog upravljačkog kadra i odgovarajućih softverskih alata u vidu sistema za podršku odlučivanju – DSS (eng. Decision Support Systems). Pri realizaciji pretovarnih operacija u logističkim procesima potrebno je doneti veliki broj različitih i često, u odnosu na krajnji cilj, međusobno suprotstavljenih odluka. Ukoliko bi se odluke donosile nezavisno jedna od druge, velika većina bi mogla biti dobijena rešavanjem odgovarajućih modela koji reprezentuju realne sisteme u celini, ili delove sistema na koje se problem odnosi. Međutim, kako je nemoguće napraviti model koji bi obuhvatio sve relevantne aspekte u donošenju efikasne upravljačke odluke, to je iskustvo donosioca odluke neprocenjivo za efikasno upravljanje pretovarnim procesima. Sa druge strane, dimenzije pojedinačnih odluka koje je potrebno doneti najčešće nadmašuju ljudske mogućnosti u pogledu sagledavanja svih, a najčešće čak i malog dela svih mogućih rešenja problema. U cilju prevazilaženja ovog problema razvijaju se sistemi za podršku odlučivanju sa ciljem da se ljudskim donosiocima odluke pomogne uzimanjem u razmatranje celog, ili što je moguće većeg, skupa rešenja konkretnog problema. U potrazi za što kvalitetnijim rešenjem sistemi za podršku odlučivanju zasnovani su na različitim metodama rešavanja problema među kojima se naročito izdvajaju metode operacionih istraživanja.

Uzimajući u obzir da pretovarni procesi predstavljaju integralni deo svakog logističkog procesa i da bi neefikasno upravljanje ovim sistemom uticalo na formiranje uskog grla u robnom toku, jasno je da se pretovarnim sistemima mora upravljati na neefikasniji mogući način. Iz tog razloga predmet ove disertacije je upravljanje pretovarnim sistemima na operativnom nivou, ili preciznije jednom specifičnom klasom pretovarnih sredstava u uslovima u kojima se zadaci sukcesivno pojavljuju u toku planskog perioda. Treba naglasiti da se pod efikasnim upravljanjem podrazumeva donošenje kvalitetnih upravljačkih odluka uz trošenje, u odnosu na prirodu problema, razumne količine vremena. Kako je predmet disertacije upravljanje na operativnom nivou to je jasno da je za rešavanje problema na raspolaganju vrlo kratak vremenski



interval<sup>1</sup>. U skladu sa relativno kratkim raspoloživim vremenom za donošenje upravljačkih odluka cilj disertacije je formiranje algoritamske osnove za razvoj DSS-a za praktično upravljanje radom specijalne klase pretovarnih sredstava u uslovima dinamičke pojave zadataka. Cilj disertacije je ostvaren kroz sledeća poglavlja, pri čemu je odnos pojedinih poglavlja prikazan na slici 1.3.



**Slika 1.3.** *Struktura i odnos između poglavlja istraživanja*

Poglavlje dva se bavi problemima upravljanja sa ciljem da se razgraniče različiti nivoi upravljanja, ali i da se definišu zadaci koji se postavljaju pred određene upravljačke nivoe. Pored toga u ovom poglavlju se diferenciraju pojmovi upravljanja i planiranja.

<sup>1</sup> Količina vremena koja stoji na raspolaganju za rešavanje operativnog upravljačkog problema se razlikuje od problema do problema i ne postoji konsenzus među stručnjacima i istraživačima koliko tačno ona iznosi. Tako je, na primer, očekivano da se odluka o rutiranju paketa podataka kroz informacionu mrežu donese za vreme koje je reda veličine dela sekunde. Sa druge strane, rutiranje dostavnih vozila pri isporuci pošiljaka se može definisati u rokovima koji se mogu meriti satima, dok se odluke o dodeli zadataka viljuškarima u distributivnim centrima može realizovati u rokovima koji se mere minutima.

U trećem poglavlju su date relevantne karakteristike operativnog problema upravljanja jednom klasom pretovarnih sistema sa dinamičkom pojavom zadataka, pri čemu su predstavljeni i konkretni sistemi u kojima se upravljanje, na način formulisan u nastavku rada, može implementirati. Četvrto poglavlje se uopšteno bavi načinima za upravljanje, tj. donošenje upravljačkih odluka, pri čemu je, usled algoritamske osnove modela na kojima se zasniva upravljanje, dat i pregled osnovnih pojmova vezanih za klasifikaciju kombinatornih problema i notacije za iskazivanje efikasnosti algoritama. U petom poglavlju je posmatrani upravljački problem modeliran kao generalizacija dobro poznatog problema putujućeg servisera. Za ovako modeliran problem je predložena matematička formulacija kojom je problem moguće rešiti do optimalnosti, ali su, pored toga, predstavljena i dva heuristička algoritma, koji u prihvatljivom vremenu daju rešenja zadovoljavajućeg kvaliteta. Primer primene predloženog modeliranja u upravljanju sistemom čije karakteristike odgovaraju karakteristikama sistema iz poglavlja 3 predstavljen je u šestom poglavlju.

## **2. Hijerarhijski nivoi planiranja i upravljanja logističkim sistemima**

---

Sve blagodeti savremenog ljudskog društva bile bi i dalje nedostižne današnjem čoveku bez procesa planiranja. Drugim rečima, može se reći da je savremena civilizacija rezultat kontinualnog procesa planiranja tokom ljudske istorije. Koliko je proces planiranja deo ljudske prirode najbolje se potvrđuje kroz jednu od parafraza mnogobrojnih pokušaja definicije planiranja koja kaže da svaki put kad na bilo koji način pomislimo o budućnosti mi u stvari planiramo.

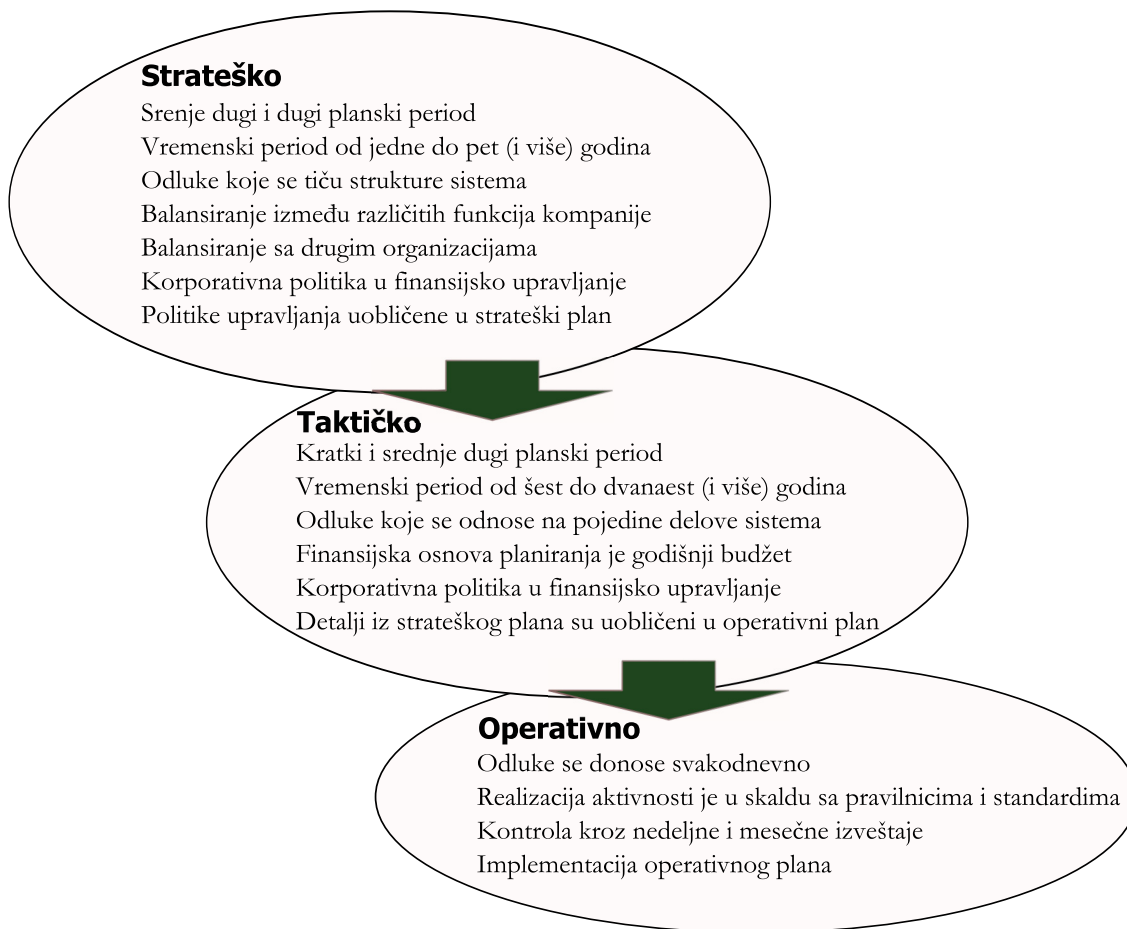
### **2.1. Strateško, taktičko i operativno planiranje**

Planiranje se uopšteno može formulisati kao priprema koju je potrebno sprovesti za donošenje kvalitetne upravljačke odluke. Međutim, kako bi planovi koji bi obuhvatali sve odluke vezane za prevođenje objekta, tj. sistema, nad kojim se plan sprovodi iz postojećeg u željeno stanje bili izuzetno kompleksni, jer se može raditi o veoma kompleksnim objektima i o veoma dugim vremenskim periodima, to se u praksi vremenom iskristalizovala podela planiranja na tri hijerarhijska nivoa. Nivoi se

definišu na osnovu vremenskog perioda na koji se planiranje odnosi, kao i na osnovu značaja upravljačkih odluka. U tom smislu planiranje se može posmatrati kao:

- strateško - podrazumeva donošenje odluka sa dalekosežnim efektima na objekat upravljanja sa ciljem da se obezbedi dalji razvoj sistema. U poslovnom okruženju vremenski okvir strateškog planiranja je najčešće reda veličine godine, a odluke su vezane za opšte ciljeve i svrhu poslovanja kompanije, marketing pristup, strategije zadovoljenja zahteva klijenata, ekspanzija (ili redukcija) mreže objekata i sl. Strateški planovi služe kao ciljevi kojima se teži pri formulisanju planova na nižem hijerarhijskom nivou. Odluke ovog tipa se donose na najvišem upravljačkom nivou kompanije od strane tzv. „top-level“ menadžmenta.
- taktičko - podrazumeva donošenje planova koji su, sa jedne strane usmereni ka zadovoljenju ciljeva postavljenih strateškim planovima, a sa druge definiše okvire za realizaciju operativnih planova. Vremenski okvir na koji se odnose taktički planovi je reda veličine meseca, a odluke donosi tzv. „mid-level“ menadžment kompanije.
- operativno – podrazumeva donošenje planova čije ostvarivanje vodi zadovoljavanju ciljeva postavljenih strateškim i taktičkim planovima. Operativni planovi definišu do krajnjih detalja instrukcije za realizaciju aktivnosti koje ih čine, što, sa druge, strane znači da oni moraju biti realizovani na osnovu preciznih i aktuelnih informacija. Vremenski okviri na koje se odnose ovi planovi su najčešće reda veličine dana, ili smene, a za njihovo donošenje su nadležni rukovodioci, tj. supervizori.

Osnovne funkcionalnosti tri navedena nivoa planiranja su predstavljene na slici 2.1.



**Slika 2.1.** Osnovne funkcije različitih nivoa planiranja (Rushton i ostali, 2010)

Ovde međutim treba naglasiti da jasnu granicu između osnovnih nivoa planiranja nije moguće jasno povući, praktično ni za jedan sistem, jer problemi na koje se odnosi planiranje retko mogu imati karakteristike samo jednog nivoa već poseduju po neku osobinu iz svakog. Iz tog razloga među istraživačima, ali i praktičarima, ne postoji konsenzus o klasifikaciji planiranja na pomenuta tri nivoa. U tom smislu, nije neuobičajeno u literaturi (npr. Fleischmann i ostali, 2007; Meisel, 2009) naći klasifikaciju koja planiranje deli samo na strateški i operativni nivo. U ovom slučaju strateški nivo se podudara sa strateškim nivoom opisanim u prethodnom delu rada, dok se pod operativnim nivoom podrazumevaju prethodno opisani taktički i operativni nivo. Treba napomenuti da je moguće naći i pristup koji planiranje klasifikuje na četiri nivoa. Ovakav pristup podrazumeva postojanje standardna tri nivoa planova, ali pored njih podrazumeva postojanje i rezervnog plana, koji se koristi u situacijama kada se nešto nepredviđeno desi u realizaciji standardnih planova. Ovakvi planovi su u praksi poznati kao „contingency“, „back-up“ ili „worst case

scenario“ planovi, dok su u žargonskoj terminologiji srpskog jezika poznatiji kao „plan B“.

## 2.2. Nivoi planiranja u logističkim sistemima

Usled sve složenijih zahteva koji se postavljaju pred logističke sisteme, kompleksnosti kompanija i/ili sektora specijalizovanih za pružanje logističkih usluga i nezanimljivog učešća logističkih troškova u ukupnim troškovima proizvoda i usluga, tj. mogućnosti ušteda koje bi se ostvarile boljom organizacijom poslova, efikasan proces planiranja je postao preduslov za uspešnu realizaciju logističkih usluga. Kao i u slučaju ostalih privrednih delatnosti i u logističkim sistemima se planiranje, tj. problemi koje je potrebno rešiti, u odnosu na vremenski horizont na koji se odnose, generalno klasifikuju na strateške, taktičke i operativne, mada je u literaturi moguće naići i na malo drugačije pristupe. Na primer, Gunther i Kim (2005) razlikuju probleme projektnog planiranja, probleme operativnog planiranja i probleme kontrole sistema u realnom vremenu. Ovakva podela podrazumeva da projektno planiranje obuhvata probleme opšteprihvaćenih strateškog i taktičkog planiranja, dok se pod operativnim planiranjem i kontrolom sistema u realnom vremenu podrazumeva rešavanje problema kada upravljani sistem već funkcioniše. Razlika između ove dve klase problema je u vremenu koje stoji na raspolaganju za donošenje odluke, jer upravljanje u realnom vremenu podrazumeva donošenje odluke u roku od nekoliko sekundi.

Iako je, kao što je pokazano u prethodnom pasusu, probleme planiranja moguće klasifikovati na različite načine, u odnosu na hijerarhijske nivoe planiranja problemi razmatrani pri planiranju realizacije logističkih aktivnosti se uobičajeno klasifikuju na sledeći način (Rushton i ostali, 2010):

- Strateški nivo
  - Definisavanje kvaliteta opsluge
  - Određivanje kanala distribucije
  - Tačke snabdevanja
  - Lokacije proizvodnje
  - Vrste, broj i tipovi terminala
  - Lokacije i veličine terminala

- Izbor vida transporta
- Odluka o posedovanju ili iznajmljivanju resursa
- Potrebni nivoi zaliha
- Taktički nivo (na primeru terminala)
  - Transport
    - ~ Vrste sredstava
    - ~ Produktivnosti sredstava
    - ~ Broj sredstava
    - ~ Putanje za kretanje
    - ~ Iznajmljivanje sredstava
    - ~ Vozači sredstava
    - ~ Pomoćna sredstva
  - Skladišta
    - ~ Layout
    - ~ Alokacija prostora
    - ~ Tehnologija skladištenja
    - ~ Rukovanje materijalom
    - ~ Broj viljuškara
    - ~ Vrste viljuškara
    - ~ Uključivanje
  - Administracija/informacioni sistemi
    - ~ Informacioni sistem za podršku
    - ~ Kontrolne procedure
    - ~ Kontrola stanja zaliha
    - ~ Sistem pozicioniranja zaliha
    - ~ Obrada narudžbenica
- Operativni nivo
  - Prijem i kontrola robe
  - Komisioniranje
  - Ažuriranje stanja zaliha
  - Regulisanje povrata robe
  - Vremensko određivanje toka procesa
  - Održavanje sredstava
  - Popunjavanje dokumentacije

Kao što se može uočiti u prethodnom delu teksta, problemi koji su klasifikovani kao taktički odnose se samo na slučaj logističkog terminala. Međutim, kako se logističke aktivnosti odvijaju, a samim tim i planiraju, od mikro (sektora u kompaniji) do makro nivoa (internacionalni lanci snabdevanja) to je jasno da je probleme koji se posmatraju kao strateški u jednom logističkom sistemu moguće, u drugom sistemu, posmatrati kao taktičke. Radi boljeg uvida u razlike u nivoima planova i odluka koje se donose u različitim logističkim sistemima u nastavku rada je, u poglavljima 2.2.1 i 2.2.2, dat pregled uobičajenih upravljačkih odluka iz Hendriks (2009) koje se donose u kontejnerskim terminalima i u logističkim kompanijama namenjenim distribuciji robe.

### **2.2.1. Nivoi odlučivanja u kontejnerskom terminalu**

Od 60-ih godina 20. veka transport robe kontejnerima je doživeo rapidnu ekspanziju, tako da se danas u svetu na ovaj način preveze roba kojoj je ekvivalent više od 100 miliona TEU (eng. Twenty feet equivalent Unit). Realizacija tokova ovako velike količine robe omogućena je postojanjem svetske mreže kontejnerskih terminala koji služe, ne samo kao tačke promene transportnog moda u kontejnerskom transportu, već i kao tačke u kojima se vrši preraspodela transporta po brodovima na interkontinentalnim i lokalnim linijama. Za koordinaciju logističkih aktivnosti utovara kontejnera na brodove, istovara sa brodova, transporta i skladištenja do utovara na naredno sredstvo transporta (brodskog kontejnerskog operatera) zadužen je operater terminala. Svaki operater kontejnerskih brodova raspolaže flotom brodova kojom održava skup redovnih kontejnerskih linija koje u redovnim vremenskim intervalima posećuju najveće svetske luke. Učestalost posete lukama je obično na nivou od sedam dana pa se na osnovu ovog parametra definiše potreban broj brodova u floti i na linijama. U skladu sa tim, zadatak operatera kontejnerskog terminala je da brodskim operaterima pruži pomenute logističke usluge koje će im omogućiti održavanje nedeljnog rasporeda kretanja u skladu sa javno objavljenim redom vožnje. Da bi obezbedio zadovoljavajući kvalitet opsluge klijenata operater terminala je suočen sa brojnim odlukama, tj. planovima.

Operateri kontejnerskih terminala obično svoje usluge nude u većem broju terminala širom sveta. Međutim, usled ekspanzije kontejnerskog transporta, potreba i broj kontejnerskih terminala se neprekidno povećavaju, te da bi zadržali svoje učešće na ovom sve zahtevnijem tržištu operateri su prinuđeni da učestvuju u tenderima kojima



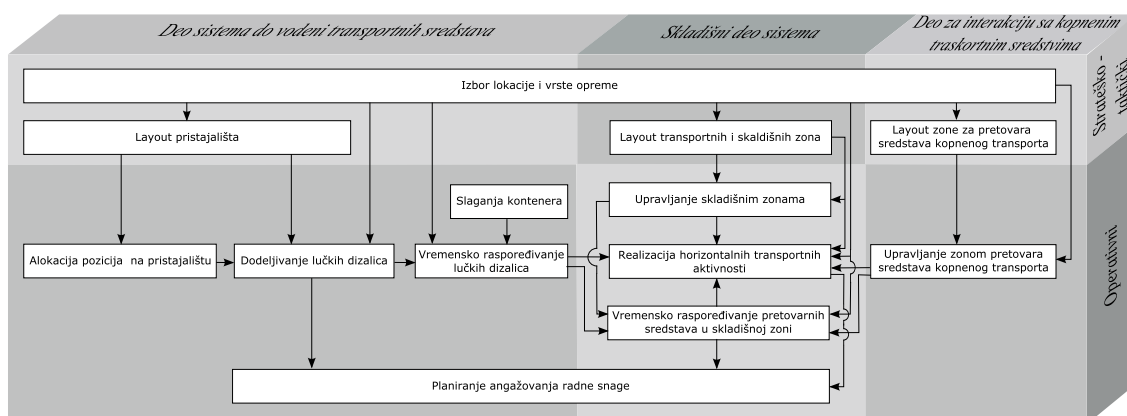
dobijaju pravo na eksploataciju novoizgrađenih, ili već postojećih kontejnerskih terminala. Obzirom da je dobijanje prava na upravljanje kontejnerskim terminalima vrlo skupo i da se potpisuju dugogodišnji ugovori o zakupu, operateri moraju na najvišem upravljačkom nivou da definišu strategije za praćenje tržišta i njegovim eventualnim osvajanjem povećanjem broja terminala. Pored pomenutih, pod strateško planiranje spadaju i odluke vezane za način funkcionisanja terminala, npr. definisanje pretovarnih i transportnih tehnologija koje će biti korišćene za transport kontejnera od operativne obale do skladišne zone, kao i za manipulaciju kontejnera na operativnoj obali i skladišnoj zoni (da li koristiti samo kontejnerske jahače, kombinaciju vučnih sredstava sa prikolicama i ramnih dizalica, kombinaciju oba rešenja ili neku potpuno atipičnu tehnologiju) i definisanje broja lučkih dizalica za manipulaciju kontejnerima pri utovaru i istovaru brodova.

U velikim kontejnerskim lukama upravljanje logističkim procesima za manipulaciju kontejnerima uglavnom realizuje više operatera terminala tako što je svaki operater nadležan za njemu dodeljen terminal. Međutim u sve većem broju luka (npr. Singapur, Rotterdam i Antwerpen), jedan operater na raspolaganju ima više terminala na kojima pruža usluge klijentima. U takvim situacijama prvi korak u taktičkom planiranju operatera je da na bazi podataka o dužini operativne obale terminala, površine dela terminala namenjenog skladištenju kontejnera i strukture robe koja se nalazi na kontejnerskim linijama alokira kontejnerske linije na raspoložive terminale i definiše vremenske periode posvećene pojedinim brodovima na terminalima. Nakon toga, potrebno je definisati tačne pozicije brodova na operativnoj obali i alocirati prostor u skladišnom delu terminala svakom brodu kako bi se minimizovalo vreme potrebno transportnim sredstvima za obavljanje operacija. Ova dva koraka rezultiraju taktičkim planovima opsluge koji se sa vremena na vreme revidiraju kako bi odgovorili na nastale promene.

Alokacija brodova na terminale i raspored zadržavanja na operativnoj obali se realizuju prema definisanim taktičkim planovima samo ukoliko brodovi strogo poštuju planirana vremena dolaska u luku. Kako je u praksi moguće da se brod pojavi ranije, ili, još verovatnije, kasnije (usled lošeg vremena, ili nekog kvara, ili zbog kašnjenja na prethodnim terminalima), ili da struktura robe na brodu značajno odstupa od one na osnovu koje su brodu dodeljeni terminal i pozicija na operativnoj obali, ili da je neka

od lučkih dizalica pokvarene na neodređen vremenski period<sup>2</sup>, to je svaki operater kontejnerskih terminala prinuđen da na nivou radne smene (8h), a u slučajevima velikih poremećaja i češće (obično na 4h), pravi operative planove opsluge. Operativnim planovima operateri terminala upravljaju sistemom na takav način da se klijenti opsluže što je moguće bolje uz minimalne troškove. Prva operativna odluka je dodela brodu konkretnog mesta na operativnoj obali i to obično tako da bude što bliže skladišnoj lokaciji namenjenoj kontejnerima koji se istovaruju sa broda, kako bi se minimizovala rastojanja koja prelaze transportna sredstva. Nakon toga, definiše se vremenski raspored angažovanosti lučke dizalice, transportnih i manipulativnih sredstava, kao i njihovih operatera na opsluzi svakog broda, kako bi bili opsluženi u ugovorom predviđenim intervalima. Detaljni redosledi operacija istovara i utovara kontejnera sa i na brod, transporta do skladišnih lokacija i plana odlaganja i izuzimanja kontejnera sa skladišnih lokacija se ažuriraju sa završetkom svake pretovarne aktivnosti kako bi se minimizovalo vreme opsluge broda.

Kao što je bilo reči u poglavlju 2.1 i u slučaju planiranja u kontejnerskim terminalima postoje autori koji ove aktivnosti posmatraju samo kroz dva nivoa, strateški i operativni. U tom kontekstu, na slici 2.2 je dat primer iz Meisel (2009) u kome autor probleme planiranja u kontejnerskim terminalima klasifikuje u odnosu na dva nivoa planiranja, ali ih pored toga razvrstava i na deo terminala na koji se odnose i daje njihovu međusobnu zavisnost.



**Slika 2.2.** Nivoi planiranja u kontejnerskom terminalu (Meisel, 2009)

<sup>2</sup> Ili zbog bilo kog drugog razloga koji uzrokuje neprimenljivost taktičkih planova opsluge, a kojih u kompleksnim sistemima, kakvi su kontejnerski terminali, ima mnogo.

### **2.2.2. Nivoi odlučivanja u distributivnoj mreži**

Tipična distributivna mreža se sastoji od proizvođača koji proizvodi više vrsta proizvoda na nekoliko prostorno dislociranih proizvodnih lokacija i nekoliko prodavaca na malo, koji zadovoljavaju potrebe krajnjih korisnika. Zavisno od veličine mreže proizvodni resursi proizvođača i lokacije prodavaca mogu biti locirani u različitim regijama, državama, ili čak i kontinentima. Priroda serijske proizvodnje i sezonske potrošnje dovodi do toga da se u pojedinim trenucima javljaju, ili potpuno odsustvo, ili postojanje prekomernih zaliha pojedinih proizvoda. Kako ni proizvođač ni prodavci ne žele da se bave amortizacijom ove pojave najčešće strategija koja se primenjuje u distributivnim sistemima je da se taj posao prepusti 3PL (eng. 3rd Party Logistics service provider) logističkim kompanijama specijalizovanim za usluge ovakvog tipa. 3PL kompanije iz tog razloga imaju za zadatak da potrebnu količinu robe u pravom trenutku prenesu od proizvođača do potrošača. Kako bi minimizovali sezonske efekte varijabilnosti tražnje, ali i minimizovali troškove distribucije primenom EOS (eng. Economy of Scale) principa, ove kompanije na putu od proizvođača do korisnika robu drže u skladišnim objektima. Planiranja u ovakvoj distributivnoj mreži podrazumeva donošenje različitih upravljačkih odluka.

Po pitanju planova sa dugim uticajem na funkcionisanje sistema, tj. strateških planova, 3PL kompanije, pre svega planiraju broj, lokacije i skladišne kapacitete objekata. Ovo je, naravno, pod razumnom pretpostavkom da su lokacije i skladišni kapaciteti proizvođača i prodavaca unapred definisani i poznati. 3PL kompanije odluke ovog tipa ne donose samo na osnovu trenutnih potreba klijenata, već i na osnovu pretpostavki o njihovim budućim potrebama, kao i na osnovu potreba koje će se javiti tržišnom ekspanzijom kompanije.

Kada se planovi prave u već postojećoj distributivnoj mreži, tj. kada su lokacije i veličine svih skladišta već fiksirane, potrebno je definisati njenu topologiju, odnosno međusobnu povezanost objekata u mreži. U distributivnim mrežama to se prvenstveno odnosi na definisanje skupa transportnih linija (drumskih, železničkih, vazdušnih) kojima se roba može transportovati između skladišnih lokacija. U skladu sa EOS principom jasno je da se efikasnost linija povećava sa povećanjem količine robe koja se tim linijama transportuje. Ovo dalje implicira da je najbolji način da se poveća prosečno iskorišćenje transportnih sredstava na linijama transport samo jednom

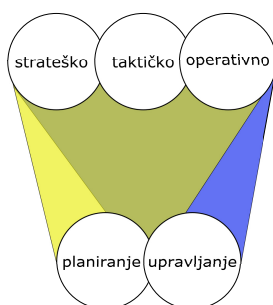
linijom. Iz tog razloga, 3PL kompanije teže da im topologija mreže bude takva da minimizuju broj linija koje realizuju transport, ali uz uslov da nivo usluge na operativnom nivou zadovoljava potrebe sistema. Otvaranje i ukidanje linija za transport robe između čvorova mreže ne može se realizovati u kratkom vremenskom periodu zbog uticaja na veliki broj operativnih aktivnosti u oba čvora uključena u liniju. Na primer, nakon uvođenja nove linije potrebno je izmeniti postojeće redove vožnji, ali je potrebno i obučiti radnike za rad pod novim uslovima kako bi se sve uklopilo u modifikovani plan distribucije. Kako je slična je situacija sa ukidanjem linija to je jasno da je topologiju mreže najbolje menjati sa vremena na vreme, kako bi odgovorila na veće promene u sistemu.

Operativni planovi uključuju dnevne odluke po pitanju definisanja ruta i redosleda opsluge klijenata kroz postojeću mrežu linija. Preciznije, za postojeću topologiju i poznate količine proizvodnje i tražnje u kraćem vremenskom periodu potrebno je definisati količinu robe koja će se transportovati odgovarajućim linkom kako bi se minimizovali troškovi transporta, skladištenja i penala usled prerane ili prekasne isporuke.

### 2.3. Odnos planiranja i upravljanja

U stručnoj literaturi, ali i u praksi, je vrlo često da se pojmovi planiranja i upravljanja koriste kao alternative jedan drugom. Međutim, između njih postoji bitna razlika koja će u nastavku biti detaljnije objašnjena.

Sa slike 2.3 se može uočiti odnos planiranja i upravljanja prema strateškim i hijerarhijskim nivoima odlučivanja. Ono što je primetno je da se i upravljanje i planiranje vezuju za sve hijerarhijske nivoe odlučivanja i da su od elementarnog značaja za efektivno i efikasno odvijanje logističkih procesa.



**Slika 2.3.** Odnos hijerarhijskih nivoa planiranja i upravljanja (Rushton i ostali, 2010)

Razlika između ovih pojmova se nalazi u njihovoj svrsi, a dobar način za njeno objašnjenje je dat u radu Rushton i ostali (2010). Naime, najjednostavnije rečeno autori podrazumevaju da je svrha planiranja „činjenje pravih stvari“, dok je svrha upravljanja „činjenje stvari na pravi način“. Drugačije rečeno planiranje obezbeđuje da će se realizovati prave aktivnosti, tj. da će se planirani procesi realizovati efektivno, dok upravljanje obezbeđuje da će biti realizovane prave aktivnosti, tj. da će se upravljani procesi realizovati efikasno.

Kako je predmet ovog rada operativno planiranje, razlika između planiranja i upravljanja na ovom hijerarhijskom nivou je preciznije definisana u Vidović (2007) gde je operativno upravljanje definisano kao alat za ostvarenje operativnih planova. Preciznije autor definiše da je realizacija operativnih planova zasnovana na upravljačkim odlukama koje se donose na bazi trenutnog stanja sistema i stanja zacrtanog operativnim planom. Naime, na bazi razlika između željenog i trenutnog stanja sistema donose se upravljačke odluke kojima će se iz skupa mogućih stanja sistema odabrati ono koje najviše odgovara ciljnom stanju sistema.

## **2.4. Karakteristike zadataka operativnog upravljanja pretovarnim sistemima**

Pojednostavljeno govoreći, a na osnovu karakteristika hijerarhijskih nivoa upravljanja, može se reći da se na strateškom nivou definiše kakav sistem treba da bude, na taktičkom kako ga organizovati da bude najbolje iskorišćen, dok je operativni nivo onaj koji omogućava da se nešto u sistemu dešava. Iz tog razloga rešavanju operativnih problema radi donošenja kvalitetnih upravljačkih odluka je posvećen veliki broj knjiga i radova i relevantnoj literaturi. Međutim, treba napomenuti da su oni uglavnom okrenuti rešavanju pojedinačnih operativnih problema. Razlog za to je što efikasno rešavanje operativnog problema podrazumeva poznavanje svih, pa i najsitnijih, detalja i specifičnosti vezanih za posmatrani problem, dok je u slučaju taktičkih, a naročito strategijskih problema, moguće rešavati probleme na osnovu generalnih okvira (Ballou, 2004).

Raznovrsnost pretovarnih procesa, kako po funkciji koju obavljaju, tako i po primenjenoj tehnologiji uslovljava je da se, u skladu sa prethodno rečenim, svaki problem mora rešavati pojedinačno. Ono što se, međutim, može navesti kao zajednička

crta svih problema koji spadaju u ovu kategoriju je postojanje dva elementa bez kojih operativno upravljanje ne bi ni postojalo (Bjelić, 2009):

1. **zadatka** koji upravljani sistem treba da realizuje i
2. **resursa** kojima se upravlja radi realizacije zadatka.

U zavisnosti od vrste zadataka i resursa definišu se i različiti problemi operativnog upravljanja, mada se vrlo često kao dodatni element uključuju ciljna funkcija i vremenski aspekt angažovanja resursa, bilo sredstava ili ljudske radne snage. U nastavku su, posmatrani na ovaj način, predstavljeni neki od primera operativnog planiranja/upravljanja, bez pretenzija da se pokrije čitav opseg problema tog tipa (Bjelić, 2009):

- klasičan **problem dodeljivanja** (eng. assignment problem) predstavlja pronalaženje optimalnog skupa parova (zadatak-resurs), dok višestruki problemi dodeljivanja (eng. multiple assignment) predstavljaju optimalan skup “trojki”, “četvorki” itd., kao što je to predstavljeno u radovima autora Vidović i Vukadinović (2006) ili Vidović (1997).
- **problem dispečiranja** se može klasifikovati kao jedan oblik problema dodeljivanja, sa specifičnošću parcijalnog rešavanja i to u trenucima pojave zadataka.
- Za razliku od dispečiranja **problemi rutiranja** podrazumevaju da im je zadatak definisanje redosleda lokacija kroz koje resursi treba da se kreću.
- **problem pakovanja** se posmatra na ovaj način tako što se kao zadatak postavlja da se resursi, u ovom slučaju jedinice koje se pakuju u određeni prostor, rasporede na takav način da zadovolje sve uslove definisane postavkom zadatka. Kao jedan od kompleksnijih primera ove problematike može poslužiti pakovanje pošiljaka u avione tzv. integratora, tj. kompanija za ekspres prenos pošiljaka, na takav način da se pored vođenja računa o iskorišćenosti prostora mora voditi računa i o raspodeli težine pošiljaka u okviru tovarnog prostora, kako ne bi došlo do ugrožavanja bezbednosti letenja letelice. Drugi primer, često i mnogo kompleksniji je redosled utovara kontejnera u brodove. U ovom slučaju se pored ravnomernog rasporeda težine po osama broda zbog stabilnosti broda, mora

voditi računa i o redosledu obilaska luka na putanji broda kako bi se minimizovalo zadržavanje u lukama usled tzv. mrtvog pretovara. Pod ovim pojmom se podrazumeva pretovar jedinica tereta koje nisu namenjene luci istovara kako bi se došlo do onih jedinica koje jesu namenjene luci u kojoj se vrši pretovar.

- ukoliko zadatak podrazumeva da se pri pronalaženju optimalnog rešenja u obzir uzima i vremenski aspekt, tada se radi o tzv. **scheduling<sup>♦</sup> problemima**, ili ukoliko bi se to prevelo u duhu srpskog jezika, problemima **vremenskog raspoređivanja resursa**. Primeri ovakvih problema su vrlo česti u praksi, a zavređivali su i znatnu pažnju istraživača. Iz tog razloga poznat je veliki broj problema koji u sebi sadrže elemente vremenskog raspoređivanja resursa:
  - problem vremenskog raspoređivanja pristajališta – BSP (eng. Berth Scheduling Problem)
  - problem vremenskog raspoređivanja lučkih dizalica – QCSP (eng. Quay Crane Scheduling Problem),
  - problem simultanog vremenskog raspoređivanja pristajališta i lučkih dizalica – BQCSP (eng. Berth and Quay Crane Scheduling Problem),
  - problem vremenskog raspoređivanja lokomotiva – LSP (eng. Locomotive Scheduling Problem),
  - problem vremenskog raspoređivanja vozila – VSP (eng. Vehicle Scheduling Problem),
  - problem vremenskog raspoređivanja operacija u mrežnom čvoru – HOSP (eng. Hub Operation Scheduling Problem), itd.
  - problem vremenskog raspoređivanja radne snage – CSP (eng. Crew Scheduling Problem),
  - problem vremenskog raspoređivanja vozača i autobusa – B&DSP (eng. Bus and Driver Scheduling Problem), itd.

Na osnovu rečenog u ovom poglavlju može se zaključiti da operativno upravljanje pretovarnim procesima omogućuje realizaciju logističkih procesa u skladu sa ciljevima postavljenim na višim hijerarhijskim nivoima. Pored toga značaj operativnog upravljanja ovakvim sistemima se ogleda i kroz ostvarivanje moguće konkurentske prednosti koja se može ostvariti efikasnom realizacijom pretovarnih procesa. Ova prednost se ispoljava ili kroz bržu realizaciju procesa, ili kroz snižavanje troškova

---

♦ U relevantnoj literaturi se neki od ovih problema mogu naći i pod nazivom sequencing, i tada se kao zadatak najčešće postavlja definisanje redosleda kojim će se zadaci izvršavati, bez respektovanja vremenskih ograničenja problema.

realizacije procesa. U svakom slučaju, pruža se mogućnost da se klijentima ponudi kvalitetnija usluga za istu cenu, kako bi se privukli novi klijenti, ili da se ostvarena ušteda iskoristi u neke druge svrhe.

Iako efekti samo jedne efikasne upravljačke odluke u pretovarnim sistemu ne bi mogli da omoguće gore pomenute uštede u vremenu i/ili novcu, kao ni da opravdaju troškove, vreme i energiju menadžmenta uloženu u razvoj sistema koji omogućava donošenje takvih odluka, činjenica da se ovakve odluke donose na nivou dana, sata, a nekada čak i češće, uzrokuje da se razvoj ovakvog sistema višestruko isplati. Kompleksnost pretovarnih sistema u pogledu broja operacija koje treba sinhronizovano realizovati, broja tehnologija kojima je moguće jedan zadatak realizovati, kao i broja jediničnih sredstava koja realizuju zadatke uzrokuje da je njima moguće efikasno upravljati smo uz pomoć DSS-a, kako je to u uvodu i objašnjeno. Kako je reč o vrlo specifičnim sistemima jasno je da se jedan DSS, razvijen za jedan pretovarni sistem ne može primeniti za upravljanje njemu sličnim sistemom, već je potrebno u njegov razvoj implementirati sve specifičnosti upravljanog sistema.

Iz tog razloga, u narednom poglavlju su predstavljene sve specifičnosti pretovarnog sistema koji je predmet ovog rada. Pored toga, predstavljeni su i konkretni pretovarni i nepretovarni sistemi kojima bi se moglo efikasno upravljati DSS-om formiranom na bazi algoritma predstavljenih u poglavlju 5.



## **3. Definicija i primenljivost problema**

---

### **3.1. Specifičnosti pretovarnog sistema sa kumulativnom ciljnom funkcijom**

Nakon uvida u značaj problema operativnog planiranja, u ovom poglavlju će detaljnije biti opisane karakteristike pretovarnih sistema koji su predmet istraživanja, kao i pretpostavke koje važe pri donošenju upravljačkih odluka. U tom smislu, pretovarni sistemi čije upravljanje je predmet ovog rada imaju sledeće karakteristike:

- pretovarne operacije se realizuju flotom pretovarnih sredstava koju čini dva, ili više, pretovarnih sredstava;
- lokacije na kojima se realizuju pretovarne operacije su prostorno dislocirane po oblasti koju pokriva pretovarni sistem;
- broj lokacija na kojima se može javiti potreba za pretovarom je veći od broja pretovarnih sredstava u floti, tako da ih je potrebno premeštati sa jedne na drugu lokaciju kako bi se zadaci opslužili,

- relevantne performanse pretovarnih sredstva bitne za efikasnu realizaciju procesa (brzina kretanja između lokacija i proizvodnost sredstva) se razlikuju od sredstva do sredstva.

Pored ovih karakteristika pretpostavka je da u upravljanoj pretovarnom sistemu važi sledeće:

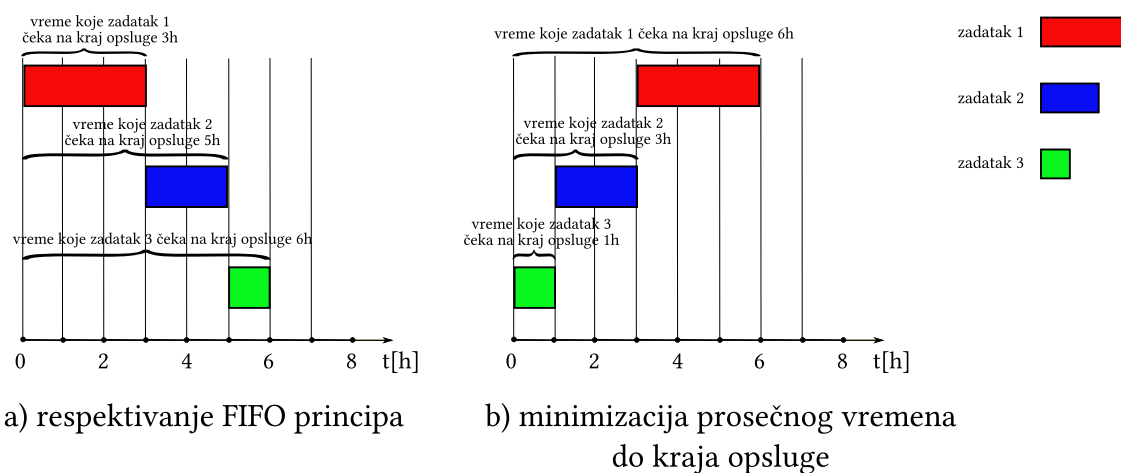
1. cilj pretovarnog sistema je da se pretovarne operacije realizuju na takav način da je prosečno vreme realizacije pretovarnog naloga što je moguće manje;
2. pretovarni zadaci se pojavljuju na različitim lokacijama tokom čitavog planskog perioda, pri čemu je posao moguće realizovati bilo kada nakon pojave zadatka u sistemu;
3. sve relevantne informacije o stanju sistema u trenutku donošenja upravljačke odluke su blagovremeno raspoložive.

Pretpostavka broj jedan, iako intuitivna, ima izuzetno veliki uticaj na prirodu posmatranog problema. Naime, da bi pretovarni sistem mogao da funkcioniše sa ciljem definisanim u pretpostavci potrebno je da su klijenti spremni da odstupe od FIFO principa opsluge ukoliko to vodi postizanju globalno manjeg prosečnog vremena opsluge. Spremnost na odustajanje od prvi ušao prvi opslužen - FIFO (eng. First In First Out) principa opsluge se u pretovarnim sistemima najčešće obezbeđuje garantovanjem krajnjeg trenutka završetka opsluge sredstva spoljnog transporta, koje je predmet ugovora između onoga ko pretovarnu uslugu pruža i onoga ko je zahteva. Na taj način je korisnik pretovarne usluge zaštićen u smislu definisanja vremenskih okvira u kojima svojim korisnicima garantuje usluge, jer se sa svakim prekoračenjem garantovanog vremena opsluge korisniku plaćaju proporcionalni novčani penali čiji nivo je definisan tako da obuhvata nadoknadu štete svim korisnicima klijenta pretovarnog sistema, ali i kaznu pružaocu pretovarnih usluga zbog neispunjenja ugovornih obaveza. Pod takvim uslovima klijenti pretovarnih sistema su raspoloživi da upravljanje pretovarnim procesom u potpunosti prepuste pretovarnim sistemima, čiji su ciljevi funkcionisanja pretovarnog sistema različiti u odnosu na ciljeve klijenata.

Koristi od funkcionisanja sistema u uslovima u kojima se ne poštuje FIFO redosled opsluge je najlakše objasniti na primeru čekanja tri sredstva spoljnog transporta koja

su se jedno za drugim, sa zanemarljivim, ali poznatim, vremenskim razlikama, pojavila u sistemu, pri čemu su očekivana vremena istovara 3h, 2h i 1h, respektivno. Ukoliko prvo transportno sredstvo ne bi bilo spremno da redosled opsluge bude kao u prvoj pretpostavci, već da se primenjuje FIFO princip opsluge, ukupno vreme čekanja sredstava do završetka pretovara bi bilo 14h. U slučaju da su sva sredstva spremna da se opsluga vrši u skladu sa pretpostavkom broj jedan ovo vreme bi iznosilo 10h, uz redosled opsluge treće, drugo pa prvo sredstvo, kako je to predstavljeno na slici 3.1.

Ukupno vreme čekanja sredstava na završetak procesa opsluge se drugačije može posmatrati i kao prosečno vreme koje sredstvo spoljnog transporta, tj. klijent, čeka na kraj opsluge (Lysgaard i Wøhlk, 2013). Na taj način je praćenje kumulante vremena koje sredstva spoljnog transporta čekaju do završetka procesa pretovara ekvivalentno praćenju performanse definisane prvom pretpostavkom. Zbog značaja koji kumulativna priroda performanse pretovarnih sistema ima na njihovo upravljanje pretovarni sistemi koji su predmet istraživanja su u daljem delu teksta referencirani kao *pretovarni sistemi sa kumulativnom ciljnom funkcijom*.



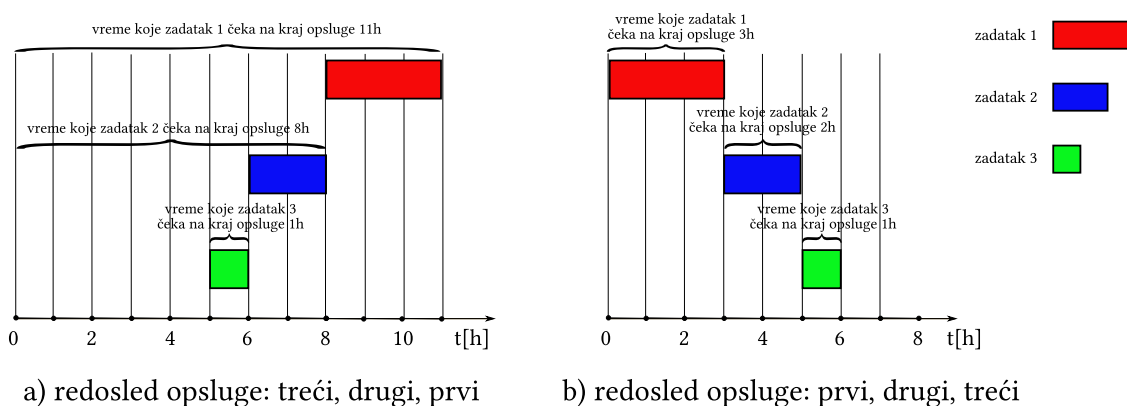
**Slika 3.1.** Primer efekta odstupanja od FIFO principa opsluge

Usled efekta prve pretpostavke na zahtevane osobine klijenata pretovarnog sistema, jasno je da se principi upravljanja ovakvim sistemima ne mogu primeniti na sve pretovarne sisteme. Međutim, svi pretovarni sistemi koji opslužuju druge podsisteme (transportne, proizvodne, skladišne) u okviru istog logističkog sistema funkcionišu u skladu sa pomenutom pretpostavkom jer im je cilj da u proseku što pre oslobode resurse tih podsistema kako bi oni bili što više iskorišćeni, tj. kako bi im se dala mogućnost da maksimiziraju svoju proizvodnost. Naravno, ovakvi uslovi

funkcionisanja važe sve dok je drugim podsistemima jedino bitno da im se oslobodi neki od resursa kako bi ga iskoristili za realizaciju primarne funkcije konkretnog podsistema. Detaljniji opis nekih od pretovarnih sistema sa ovakvim karakteristikama, ali i nekih sistema koji nisu pretovarni, a čije funkcionisanje je u skladu sa prvom pretpostavkom je dat u poglavlju 3.2, u nastavku.

Pretpostavka da se zadaci pojavljuju tokom celog planskog perioda daje pretovarnim sistemima dinamičku karakteristiku, koja je veoma česta u realnim procesima. Iz tog razloga ovakvi sistemi se nazivaju **pretovarni sistemi sa dinamičkom pojavom zadataka**. Ova pretpostavka, zajedno sa trećom, značajno utiče na efikasno upravljanje pretovarnim sistemima. Naime, kako svi pretovarni zadaci nisu prisutni na početku planskog perioda to je potrebno voditi računa i o vremenskom aspektu angažovanja pretovarnih sredstava na pojedinim zadacima kako bi se realizovao krajnji cilj funkcionisanja sistema.

Primer uticaja vremena pojave pretovarnih zadataka tokom planskog perioda na upravljanje sistemom se može videti na primeru pretovara tri sredstva spoljnog transporta, ali sada u situaciji kada prvo sredstvo pristiže na početku planskog intervala, drugo nakon 3h od početka intervala, a treće nakon 5h od početka intervala. Kao što se vidi sa slike 3.2 na ovaj način redosled opsluge prvo, pa drugo, pa treće sredstvo daje znatno bolji rezultat od redosleda opsluge koji se pokazao bolje da su sva sredstva raspoloživa na početku planskog intervala (treće, drugo pa prvo sredstvo).



**Slika 3.2.** Primer uticaja dinamičke pojave zadataka na upravljanje pretovarnim sistemom

Na bazi ovog krajnje jednostavnog primera može se videti da se principi upravljanja sistemom u kome su svi zadaci raspoloživi na početku planskog perioda ne mogu

primeniti na sisteme sa dinamičkom pojavom zadataka. U primeru je pretpostavljeno da se zadaci javljaju na istoj prostornoj lokaciji tako da je potrebno vreme za promenu pretovarne lokacije jednako nuli. Međutim, ukoliko se u upravljanje uključi i ovaj aspekt, kao i aspekt nehomogenosti pretovarnih sredstava, jasno je da se upravljanje takvim sistemom bazira na rešavanju ekstremno kompleksnih problema.

Efekti primene algoritama razvijenih u ovom istraživanju zavise od tačnosti informacija o stanju sistema u trenutku donošenja odluke. Obzirom da metode prikupljanja potrebnih informacija nisu predmet ovog istraživanja to je pretpostavljeno da su informacije o:

- trenutima pojave zadataka u pretovarnim lokacijama;
- očekivanim vremenima pretovara za svako sredstvo;
- vremenu potrebnom sredstvima da od jedne dođu do druge lokacije;
- raspoloživosti pretovarnih sredstava za realizaciju pretovara;

ažurne i poznate pre početka procesa donošenja upravljačke odluke. Naravno, što relevantne informacije više odstupaju od faktičkog stanja sistema to će efekti upravljačkih odluka biti manji.

### **3.2. Primeri pretovarnih sistema sa kumulativnom ciljnom funkcijom**

Pre nego što budu dati konkretni primeri sistema čije funkcionisanje je u skladu sa prethodno definisanim karakteristikama i pretpostavkama, treba napomenuti da se ovde navode isključivo sistemi sa dinamičkom pojavom zadataka u planskom periodu. Međutim, treba naglasiti je da se rezultati ovog istraživanja lako mogu primeniti i na sisteme u kojima su zadaci prisutni u sistemu na početku planskog perioda (statički), jer su oni samo specijalan slučaj. Preciznije, statički sistemi predstavljaju samo specijalni slučaj sistema u kome se vreme pojave svih zadataka poklapa sa početkom planskog perioda. Pri tome, usled velikog uticaja koji dinamička priroda pojave zadataka ima na vrednosti ciljne funkcije, kvalitet upravljačkih odluka dinamičkog sistema primenjenih u uslovima statičke pojave zadataka je upitan budući da je pristup rešavanju dinamičkih problema u svojoj suštini oslonjen na vremenski aspekt problema. Iz tog razloga verovatnije je da bi modeli namenjeni rešavanju isključivo

statičkih problema dali kvalitetnije upravljačke odluke. Naravno, ovo ne važi u slučajevima optimalnog rešavanja problema, jer je rešenje optimalno i za specijalne uslove funkcionisanja sistema.

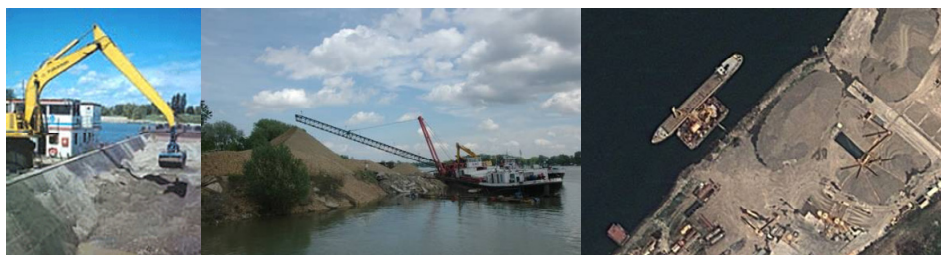
### 3.2.1. Specijalizovani pretovarni uređaji za istovar šljunka iz potisnica

Generalno, realni sistemi koji se bave proizvodnjom i distribucijom šljunka sastoje se od nekoliko bagera (slika 3.3) raspoređenih na mestima eksploatacije šljunka, potiskivane flote kojom se izvađeni šljunak transportuje do deponije i jednog, ili više, pretovarnih sredstava koja se koriste za istovar šljunka iz potisnica na deponije.



**Slika 3.3.** Primeri bagera za eksploataciju šljunka sa dna rečnog korita

Pretovarna sredstva predstavljaju posebnu vrstu plovnih sredstava opremljenih zahvatnim uređajem i trakastim transporterom (slika 3.4). Rad pretovarnog sredstva podrazumeva zahvatanje šljunka iz potisnice pomoću zahvatnog uređaja, istovar u utovarni bunker trakastog transportera i transport trakastim transporterom do deponije na obali reke.



**Slika 3.4.** Primer pretovarnih sredstava za istovar šljunka iz potisnica

Prednost koju vodni transport pruža po pitanju jedinične cene transporta u odnosu na druge vidove transporta za posledicu ima da se kompanije koje se bave ovim poslom odlučuju da šljunak transportuju potiskivanom flotom ili samohodnim teretnjacima što je moguće bliže krajnjim kupcima. Da bi ostvarili ovaj cilj kompanije duž rečnih

tokova otvaraju veliki broj lokacija na kojima skladište šljunak. Da bi se šljunak istovario u ovim lokacijama potrebno je ili da obala bude uređena tako da plovila mogu da joj pristupe da bi se šljunak istovario uz pomoć tipične pretovarne tehnologije, ili da se koriste pomenuti pretovarni uređaji. Izgradnja uređenih obala za pristup plovila je izuzetno skupa investicija, te se kompanije rađe opredeljuju za varijantu istovara specijalizovanim pretovarnim uređajima. Međutim, investicije u ovakva sredstva, iako znatno manja nego u slučaju izgradnje uređenih obala, takođe nisu beznačajna tako da je broj sredstava u sistemu znatno manji od broja lokacija na kojima se vrši skladištenje šljunka. Iz tog razloga upravljanje ovim sredstvima ima veliki uticaj na funkcionisanje i produktivnost čitavog sistema transporta i distribucije šljunka. U slučaju neefikasnog rada, koji podrazumeva veliko čekanje potisnica na istovar, dolazi do smanjenja kapaciteta celokupnog sistema jer tada potisnice prave manji broj obrta u toku planskog vremenskog perioda što rezultuje manjom količinom prevezenog šljunka.

Iz tog razloga od upravljanja pretovarnim sredstvima se zahteva da obezbedi postizanje takvog rasporeda opsluge potisnica koji obezbeđuje minimalna vremena čekanja i realizacije procesa istovara na lokacijama istovara šljunka. Cilj i način funkcionisanja ovog pretovarnog sistema su u potpunosti saglasni sa svim karakteristikama i pretpostavkama pretovarnih sistema sa kumulativnom ciljnom funkcijom.

### **3.2.2. Sistemi unutrašnjeg transporta sa automatski vođenim vozilima**

Automatski vođena vozila – AGV (eng. Automated Guided Vehicles) su nastala sa idejom da se iz unutrašnjih transportnih tokova jednog sistema isključi ljudski faktor, kako bi se ostvarili svi pozitivni efekti automatizacije. Obzirom na pozitivne efekte koji su usledili sa implementacijom ovakvih rešenja tehnologija je unapređivana tako da se ovi sistemi danas osim za transport, koriste i za realizaciju pretovarnih operacija, za obavljanje određenih proizvodnih aktivnosti u toku transporta i sl. Iz tog razloga, danas ih je moguće sresti u različitim pretovarnim, transportnim i proizvodnim sistemima. Na slici 3.5 je prikazan izgled jednog automatski vođenog vozila (AGV-a) za manipulaciju paletama, koncept skladišta gde se ova sredstva koriste, kao i primena u kontejnerskom terminalu u luci Rotterdam.

Zadatak ovih vozila u logističkim sistemima je da realizuju unutrašnje robne tokove između različitih funkcionalnih celina. U zavisnosti od procesa koji se realizuju po celinama moguće je predvideti trenutke pojave zahteva za transportom u planskom periodu koje sistem AGV-a treba da realizuje. Kako bi se omogućilo celom sistemu da postigne maksimalnu iskorišćenost potrebno je da prosečno vreme realizacije transportnog zahteva za prevozom tereta između pojedinih celina bude minimizovano. Iz tog razloga svaki pojedinačni sistem mora biti spreman da se „žrtvuje“ za dobrobit celog sistema, odustajanjem od FIFO redosleda opsluge, usled čeka se može reći da posmatrani sistem funkcioniše u skladu sa prethodno navedenim pretpostavkama.. Jedina specifičnost ovakvih sistema, u odnosu na navedene pretpostavke pretovarnih sistema sa kumulativnom ciljnom funkcijom, jeste da su automatski vođena vozila obično homogena po pitanju karakteristika relevantnih za realizaciju pretovarnog procesa. Međutim, ovo samo upućuje na zaključak da je reč o specijalnom slučaju sistema sa heterogenom flotom AGV-a.



**Slika 3.5.** *Primer izgleda AGV-a i primene u skladištima i lukama<sup>3</sup>*

### 3.2.3. Isporka pošiljaka u distributivnim sistemima

Pojednostavljeno posmatrajući, uobičajeni način funkcionisanja sistema za distribuciju pošiljaka podrazumeva da se sva roba pristigla u prethodnom danu, nakon potrebne

---

<sup>3</sup> Slike su preuzete sa web adresa <http://www.jatorman.com/almacenes-automaticos.html> i [www.portofroterdam.com](http://www.portofroterdam.com)



pripreme u formi sortiranja, na početku radnog dana isporučuje klijentima. U toku faze isporuke se prikupljaju zahtevi klijenata za preuzimanjem pošiljaka, tako da se nakon završetka isporuke kreće sa prikupljanjem pošiljaka. Na kraju radnog dana ove pošiljke se istovaruju u depou i dalje obrađuju kako bi sutra bile spremne za isporuku.

U funkcionisanju ovih sistema često se definišu vremenski okviri (u literaturi najčešće referencirani kao vremenski prozori) u kojima se usluga mora ostvariti. U slučaju isporuke pošiljaka, naročito na početku radnog dana, često se dešava da se pošiljka ne može isporučiti dok klijent nije raspoloživ, tj. postoji samo trenutak od kada se zadatak može izvršiti, ili drugačije posmatrano, trenutak pojave zadataka. U takvim situacijama pre definisanju redosleda isporuke pošiljaka, tj. rutiranja vozila, u obzir se mora uzeti dinamika pojave zadataka i njihov prostorni raspored kako se vreme ne bi trošilo na čekanje klijenta, ali i kako klijent ne bi dugo čekao na pošiljku nakon definisanog vremena isporuke. Period za realizaciju isporuke pošiljaka (planski period) je kraći od radnog vremena te je u ovakvim sistemima opravdano smatrati da se isporuka pošiljke može realizovati u bilo kom trenutku pre kraja planskog perioda.

U procesima prikupljanja pošiljaka je situacija drugačija jer se obično definišu i vremena od kada se neka pošiljka može preuzeti (što je obično uslovljeno nekim drugim procesom, npr. trenutkom završetka proizvodnje robe koja će se distribuirati), ali i vremena do kada se roba mora preuzeti (obično definisanih radnim vremenom klijenta koje ne mora biti jednako vremenu završetka planskog perioda). Na osnovu opšteg cilja sistema za distribuciju pošiljaka, da se minimizuje prosečno vreme opsluge klijenata, jasno je da se deo procesa distribucije vezan za isporuku pošiljaka odvija u skladu sa uslovima definisanim za pretovarne sisteme sa kumulativnom ciljnom funkcijom.

#### **3.2.4. Vremensko raspoređivanje brodova na pristajalištu**

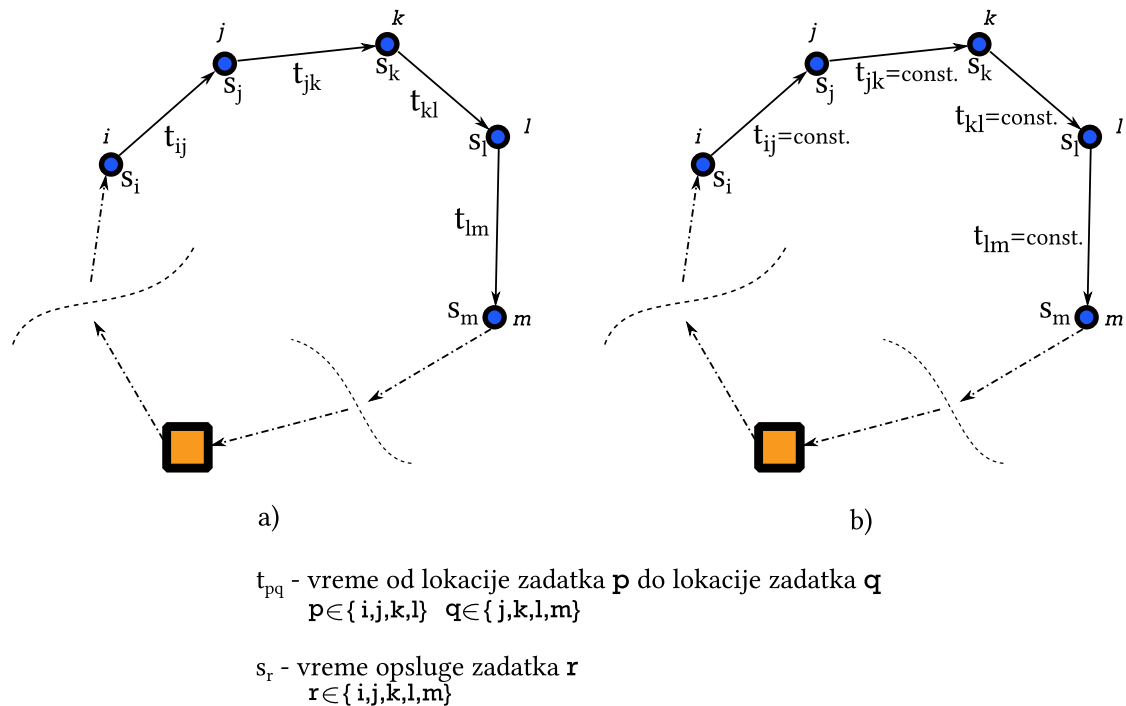
Pristajališta predstavljaju ograničene resurse u lučkim terminalima na koje pristaju brodovi kako bi se sa njih istovarilo teret i/ili na njih utovarilo teret za druge luke. Na primeru kontejnerskog terminala u poglavlju 2.2.1. pokazano je da je upravljanje pristajalištima u okviru terminala uglavnom povereno jednom operateru terminala koji na njemu realizuje sve aktivnosti kako bi zadovoljio ciljeve viših hijerarhijskih nivoa. Usled sve većeg intenziteta robnih tokova u lučkim terminalima broj brodova

koji pristiže na opslugu je sve veći što za posledicu ima malu marginu greške koja se operaterima terminala dozvoljava.

Jedan od osnovnih ciljeva koji se postavlja pred operatere lučkih terminala je da minimizuju zadržavanje brodova u lukama, jer brodovi donose novac vlasnicima samo kada robu transportuju sa jednog mesta na drugo. Ovaj cilj je podudaran sa ciljem postavljenim pred pretovarne sisteme sa kumulativnom ciljnom funkcijom jer se ostvaruje onda kada se minimizuje prosečno vreme zadržavanja broda u luci. Pored toga, odstupanje od FIFO principa opsluge ovde je očekivano, imajući u vidu da su u velikom broju slučajeva iste kompanije ujedno i operateri terminala i brodski operateri.

Vreme pojave brodova u luci je, usled rasprostranjenog prisustva savremenih informacionih i komunikacionih tehnologija – ICT (eng. Information and Communications Technologies), moguće predvideti sa priličnom tačnošću. Zato je u proces donošenja upravljačkih odluka moguće uvrstiti i one brodove koji nisu prisutni u luci na početku planskog perioda, već se njihovo pristizanje očekuje u toku planskog perioda.

Pored svih sličnosti sa pretovarnim sistemom sa kumulativnom ciljnom funkcijom i ovaj problem predstavlja njegov specijalni slučaj. Naime, u problemu koji je predmet ovog istraživanja vremena kretanja pretovarnih sredstava između lokacija u kojima se pojavljuju zahtevi za pretovarom se razlikuju između svih parova lokacija, kao što je to predstavljeno na slici 3.6a. Drugim rečima vreme kada je sredstvo raspoloživo za opslugu sledećeg zadatka se razlikuje za svaki par zadataka. Međutim, u slučaju vremenskog raspoređivanja brodova na pristajalištu to nije slučaj jer svi brodovi na opslugu čekaju na lokacijama koje su približno isto udaljene od pristajališta, te im je svima potrebno približno isto vremena da se pojave na pristajalištu. Odnosno, ukoliko mesta na pristajalištu posmatramo kao flotu resursa, a brodove kao pretovarne zadatke, onda se može reći da je vreme potrebno da resurs dođe u poziciju da opsluži klijenta konstantno, bez obzira na redosled opsluge (slika 3.6b).



**Slika 3.6.** *Specifičnosti problema vremenskog raspoređivanja brodova i predmeta istraživanja*

Pregled sistema na koje se mogu odnositi rezultati ovog istraživanja se opisanim sistemima ne završava, već su ovde predstavljene samo tipični logistički i transportni sistemi kojima se može upravljati modelima koji će biti predstavljeni u nastavku. Neki od primera koji ovde nisu detaljnije prikazani jesu: upravljanje radom građevinske mehanizacije (dizalica velike nosivosti, pumpi za nalivanje betona, mobilnih mešalica za beton i sl), opreme za servisiranje električne mreže, specijalnih mašina za deparafinizaciju cevi na naftnim bušotinama itd.

## **4. Upravljanje pretovarnim sistemima sa kumulativnom ciljnom funkcijom**

---

Upravljanje pretovarnim sistemima sa kumulativnom ciljnom funkcijom se realizuje donošenjem upravljačkih odluka na bazi informacija o stanju sistema na početku planskog perioda i očekivanih trenutaka o pojavama zadataka u toku planskog perioda. Upravljačka odluka podrazumeva alokaciju pretovarnih zadataka na raspoloživa sredstva i definisanje redosleda opsluge alociranih zadataka na svakom od sredstava. Za svaku upravljačku odluku postoji konačan broj mogućih rešenja i potrebno je naći ono rešenje koje će u najvećoj meri odgovarati definisanom cilju upravljanja uz zadovoljenje svih ograničenja sistema.

### **4.1. Problemi kombinatorne optimizacije**

Činjenica da se traži jedno (ili više) rešenje koje sa najvećim stepenom zadovoljava cilj upravljanja svrstava ove probleme u klasu optimizacionih problema. Konkretno ako je sa  $x$  označeno moguće rešenje problema, tada je iz skupa rešenja koja zadovoljavaju

sva ograničenja problema,  $X$ , potrebno naći ono rešenje problema,  $x^*, x^* \in X$ , za koje važi:

$f(x^*) \leq f(x) \forall x \in X$ , u slučaju minimizacionih problema, odnosno

$f(x^*) \geq f(x) \forall x \in X$ , u slučaju maksimizacionih problema

gde je  $f: X \rightarrow \mathbb{R}$ , funkcija koja svakom rešenju  $x$  dodeljuje realnu vrednost i predstavlja funkciju cilja posmatranog problema.

Treba napomenuti da se u optimizacionim problemima bira najbolje,  $x^*$ , iz skupa rešenja koja zadovoljavaju sve uslove posmatranog problema,  $X$ . Međutim, skup svih mogućih rešenja upravljačkog problema,  $S$ , može biti značajno veći, tj. važi  $X \subseteq S$ . Ovo je samo jedna od mogućih formulacija optimizacionih problema. Značaj optimizacionih problema za rešavanje praktičnih problema je razlog što je u velikom broju istraživačkih radova (Papadimitroui i Steiglitz, 1998; Hansen i Mladenović, 2003; Vidović, 2007...) moguće naći različite formulacije ovih problema.

Druga bitna karakteristika posmatranog problema je ta da je broj mogućih rešenja u skupu  $S$  konačan. U zavisnosti od veličine konkretne instance problema broj rešenja u  $S$  može biti manji ili veći, ali u svakom slučaju je konačan. Zbog ove karakteristike upravljanje posmatranim sistemima spada u klasu problema kombinatorne optimizacije.

## **4.2. Osnovni pojmovi u analizi algoritama za upravljanje pretovarnim sistemima sa kumulativnom ciljnom funkcijom**

### **4.2.1. Asimptotska notacija i kompleksnost algoritama**

Usled potencijalno velikog broja rešenja, pretragu skupa  $S$  za pronalaženjem  $x^*$  je praktično nemoguće realizovati bez odgovarajućih računarskih algoritma. Kako je problem upravljanja posmatranom klasom pretovarnih sredstava operativni problem, to je jasno da je upravljačke odluke potrebno donositi u relativno kratkom vremenskom periodu. Iz tog razloga pitanje koliko će vremena trebati algoritmu da reši problem je veoma bitno za upravljanje, a konkretan odgovor je veoma teško dati jer zavisi od uticaja velikog broja faktora, među kojima su verovatno najbitniji

efikasnost hardvera na kome se algoritam realizuje, efikasnost samog algoritma i veličina problema koji se rešava. Uticaj hardvera, tj. konkretno centralne procesorske jedinice – CPU (eng. Central Processor Unit), na vreme realizacije algoritma je očigledan jer njegovo unapređenje dovodi do redukcije vremena potrebnog za dobijanje rešenja problema. Međutim, iako je razvoj CPU-a, koji je impozantan u poslednjih 30-ak godina, znatno uticao na brzinu rešavanja problema, on i dalje nije u mogućnosti da nadomesti uticaj koji na vreme rešavanja problema ima razvoj efikasnih algoritma.

Pod algoritmom se u opštem smislu podrazumeva procedura sastavljena od elementarnih operacija sa ciljem rešavanja konkretnog problema. Na računarima to podrazumeva realizaciju elementarnih računarskih operacija (sabiranja, oduzimanja, dodeljivanja, poređenja, itd.) kojima se zadate ulazne veličine transformišu u izlazne veličine. Iako ove operacije traju izuzetno kratko, one se realizuju veoma veliki broj puta u toku realizacije algoritma, što na kraju definiše vreme trajanja izvršavanja algoritma. Kako je krajnji cilj formulisati algoritam koji će u što kraćem roku ulazne informacije transformisati u izlazne, za sve veličine ulaznih podataka, tj. algoritam čije će vreme realizacije u najmanjem stepenu rasti sa povećanjem veličine problema, to je jasno da efikasni algoritam mora realizovati što je moguće manje elementarnih računarskih operacija. Broj ovakvih operacija je u direktnoj vezi sa veličinom problema koji se rešava, odnosno količinom ulaznih informacija.

Odnos između vremena realizacije algoritama, tj. broja elementarnih operacija u realizaciji algoritma i veličine problema koji se rešava je predmet teorije kompleksnosti algoritama<sup>4</sup>(eng. Computation Complexity Theory). Mera kompleksnosti algoritma se izražava kao funkcionalna zavisnost broja elementarnih operacija algoritma u zavisnosti od veličine ulaznog problema,  $n$ , gde je  $n$  obično broj zadataka koji obuhvata problem. U realizaciji algoritma vrlo veliku ulogu ima i stohastičnost ulaznih podataka. Na primer, isti algoritam za sortiranje niza brojeva u rastući niz će znatno brže dati sortirani niz na izlazu ukoliko mu je ulazni niz [2,1,3,4,5,6,7,8,9], nego da se na ulazu našao niz [9,8,7,6,5,4,3,2,1]. Iz tog razloga, kada

---

<sup>4</sup> Ova oblast pored vremena tretira i memorijski aspekt realizacije algoritama. Međutim, kako je vremenski aspekt značajniji za rešavanje problema operativne prirode to memorijskom aspektu u ovom radu nije posvećena posebna pažnja.

se govori o vremenu realizacije algoritama moguće je razlikovati tri slučaja (Hetland, 2010):

- Najkraće vreme realizacije, koje predstavlja vreme realizacije algoritma u slučaju kada su ulazni podaci potpuno prilagođeni funkcionisanju algoritma. Na primer, kada se na ulazu algoritma za sortiranje niza brojeva u rastući niz pojavi niz brojeva [1,2,3,4,5,6,7,8,9].
- Najduže vreme realizacije, kojim se definiše vreme realizacije algoritma kada su ulazni podaci potpuno neprilagođeni algoritmu. Na primer, kada je ulazni niz brojeva u algoritam za sortiranje niza brojeva u rastući niz dat kao [9,8,7,6,5,4,3,2,1].
- Prosečno vreme realizacije, koje je, najjednostavnije rečeno, definisano za slučajne ulazne podatke, pod uslovom da je definisana raspodela verovatnoća ulaznih podataka. Primer ulaznih podataka za algoritam za sortiranje u rastući niz bi bio [5,6,3,8,9,2,1,7,4].

Tačno definisanje vremena realizacije algoritma, kao što je ranije rečeno, zavisi od velikog broja faktora te ga je nemoguće precizno odrediti. Iz tog razloga ono se samo procenjuje u postupku asimptotske analize. Cilj asimptotske analize je pronalaženje jednostavne funkcije  $g(n)$ , kojom je moguće što verodostojnije opisati rast stvarnog vremena izvršavanja algoritma  $t(n)$ , sa povećanjem veličine problema. Uzimajući u obzir tri slučaja vremena realizacije algoritama rezultat asimptotske analize su tri mere kompleksnosti algoritama:

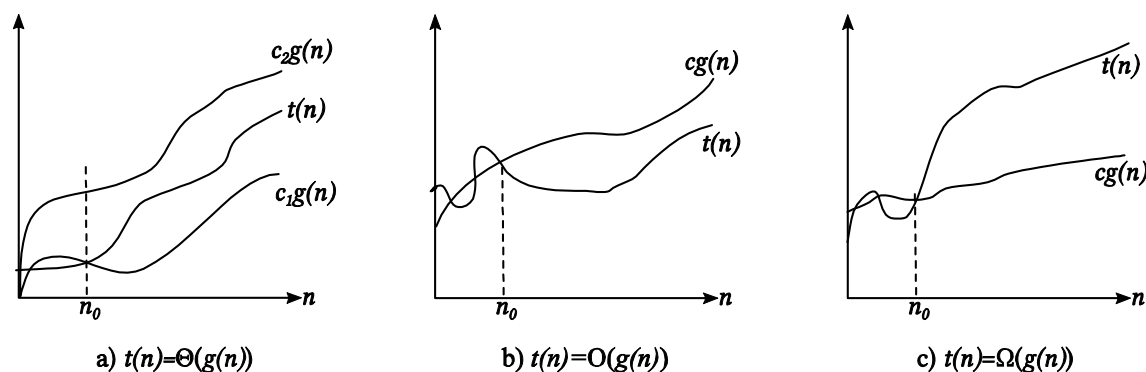
- „big omega“ –  $\Omega(g(n))$ , kojom se definiše zakonitost rasta minimalnog vremena realizacije algoritma
- „big oh“ –  $O(g(n))$ , kojom se definiše zakonitost maksimalnog vremena rada algoritma, i
- „big theta“ –  $\Theta(g(n))$ , kojom se definiše zakonitost očekivanog vremena rada algoritma.

Kao najznačajnija među ovim merama se izdvojila „big oh“ mera jer definiše zakon porasta maksimalnog vremena realizacije algoritma sa porastom veličine problema, tj. ograničava vreme rada algoritma sa gornje strane. Generalno gledano,  $O(g(n))$  je

skup svih funkcija čiji je stepen rasta manji od  $g(n)$  za dovoljno veliko  $n$ . Preciznije, kao što je to opisano u Levitin (2012), za funkciju  $t(n)$  se kaže da pripada  $O(g(n))$ , tj. da važi  $t(n) \in O(g(n))$  ukoliko je  $t(n)$  za velike vrednosti  $n$  ograničeno sa gornje strane proizvodom  $g(n)$  i neke konstante. Drugim rečima,  $t(n) \in O(g(n))$  ukoliko postoji pozitivna konstanta  $c$  i nenegativni ceo broj  $n_0$  tako da je

$$t(n) \leq c \cdot g(n) \text{ za svako } n \geq n_0$$

Formulacija funkcija koje pripadaju skupovima „big omega“ i „big theta“ je analogna i može se naći u bilo kojoj knjizi koja za temu ima algoritme. Vizuelni prikaz mera kompleksnosti algoritama je prikazan na slici 4.1.



**Slika 4.1.** Prikaz asimptotske analize kompleksnosti algoritama (Cormen i ostali, 2001)

Na osnovu asimptotske analize, algoritmi se mogu podeliti na nekoliko klasa efikasnosti. U tabeli 4.1. je predstavljena jedna takva tipična podela. Vremenska efikasnost algoritma je obrnuto proporcionalna njegovoj kompleksnosti tako da funkcije manjeg rednog broja imaju veću efikasnost.

**Tabela 4.1.** Klase efikasnosti algoritama (Hetland, 2010)

R.B.	Asimptotska funkcija	Ime
1	1	konstantna
2	$\log n$	logaritamska
3	$n$	linearna
4	$n \log n$	loglinearna
5	$n^2$	kvadratna
6	$n^3$	kubna
7	$n^k$	polinomijalna
8	$k^n$	eksponencijalna
9	$n!$	faktorielna



#### 4.2.2. P, NP i NP-complete klase problema

U prethodnom poglavlju u tabeli 4.1. su predstavljene različite kompleksnosti algoritama, ali se, zbog svog značaja za mogućnost praktične primene u rešavanju realnih problema, dve od njih izdvajaju. Naime, svi problemi koji se mogu rešiti algoritmicima polinomijalne, ili manje, kompleksnosti se u praksi smatraju rešivim problemima, dok se ostali problemi smatraju nerešivim problemima. To u praktičnom smislu znači da je samo polinomijalne algoritme moguće koristiti za rešavanje praktičnih problema, dok su svi algoritmi eksponencijalne, ili veće kompleksnosti, praktično neupotrebljivi. Ovo naravno ne važi u opštem slučaju, jer je sigurno da algoritmi kompleksnosti  $O(n^{100})$  nisu pogodniji za praktičnu primenu od algoritama kompleksnosti  $O(2^n)$ , ali se iskustveno pokazalo kao validan kriterijum za razlikovanje upotrebljivih od neupotrebljivih algoritama. Iz tog razloga za svaki algoritam kompleksnosti manje ili jednake  $O(n^k)$ , za svako  $k > 0$ , se kaže da je polinomijalan u vremenu, dok se za sve algoritme kompleksnosti  $O(k^n)$  i  $O(n!)$  kaže da su eksponencijalni u vremenu.

U teoriji računarstva - CS (eng. Computer Science) problemi odlučivanja rešivi u (determinističkom) polinomijalnom vremenu se svrstavaju u skup problema koji se označava sa  $P$ . Ovde treba napomenuti da se pod problemima odlučivanja podrazumevaju svi problemi na koje se može odgovoriti sa „da“ ili „ne“.

Druga bitna klasa problema u CS-u su problemi odlučivanja za koje do sada nisu razvijeni algoritmi polinomijalni u vremenu. Iz tog razloga za ove probleme se kaže da su rešivi u nedeterminističkom polinomijalnom vremenu -  $NP$  (eng. Non-deterministic Polynomial-time). Pojam determinističkog i nedeterminističkog vremena rešavanja algoritma je usko vezan sa osnovama CS teorije i izvršavanjem algoritma na determinističkoj i nedeterminističkoj Turingovoj mašini čija problematika u znatnoj meri prevazilazi obuhvat ovog istraživanja. Ipak, za okvirno razumevanje ovih pojmova može da posluži definicija nedeterminističkog algoritma iz Levitin (2012) po kojoj se pod nedeterminističkim algoritmom podrazumeva procedura koja na ulazu dobija instancu problema  $I$  a zatim kroz dve faze daje odgovor na problem odlučivanja. Iterativne i repetitivne faze algoritma su:

- Nedeterministička faza (faza „pogađanja“ rešenja): u kojoj se na proizvoljan način generiše niz  $B$ , koji predstavlja moguće rešenje instance  $I$  (ali može biti i potpuni promašaj), i
- Deterministička faza (faza „verifikacije“ rešenja): u kojoj deterministički algoritam na ulazu dobija  $I$  i  $R$ , a na izlazu daje odgovor „da“ ukoliko je  $R$  stvarno rešenje  $I$ . Ukoliko  $R$  nije rešenje  $I$ , algoritam vraća „ne“ kao odgovor, ali mu je dozvoljeno i da nikada ne završi sa radom.

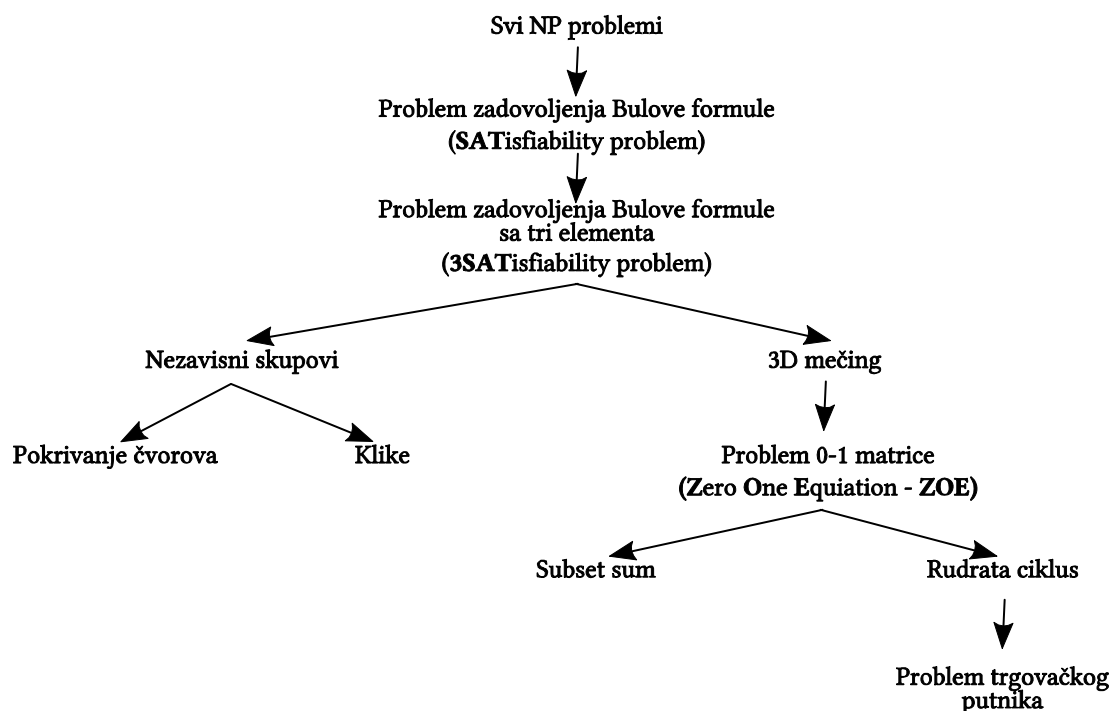
Smatra se da nedeterministički algoritam rezultuje rešenjem problema odlučivanja ako, i samo ako, za svaku instancu problema, za koju je stvarno rešenje „da“, u nekoj iteraciji rešavanja on da odgovor „da“. Drugim rečima, od nedeterminističkog algoritma se očekuje da bar u nekoj iteraciji „pogodi“ rešenje problema, a zatim i da verifikuje validnost rešenja. Pored toga podrazumeva se da algoritam nikada neće vratiti odgovor „da“ za problem za koji je stvarno rešenje „ne“. Nedeterministički algoritam je polinomijalan u vremenu onda kada je kompleksnost verifikacione faze polinomijalna. Iz tog razloga se problemi odlučivanja koji pripadaju  $NP$  klasi često definišu kao problemi čija se rešenja (bez obzira kako se do njih dođe) mogu verifikovati u polinomijalnom vremenu.

Uzimajući u obzir realizaciju nedeterminističkih algoritama, tj. osobinu da u nekoj iteraciji moraju da daju rešenje problema, jasno je i da svi problemi koji pripadaju skupu  $P$ , pripadaju i skupu  $NP$ , odnosno da važi  $P \subset NP$ . U odnosu skupova stoji znak  $\subset$  iz prostog razloga što je mišljenje većine autora (Sipser, 2013; Dasgupta i ostali, 2008; Levitin, 2012; Cormen i ostali, 2001) iz oblasti koje tretiraju ovo pitanje, tj. iz matematike, teorije kompleksnosti i  $CS$ -a, da skup  $P$  nikada ne može biti jednak skupu  $NP$ , tj. pretpostavljeno je da važi  $P \neq NP$ . Ova tvrdnja nikada nije dokazana i predstavlja jedno od najznačajnijih nerešenih matematičkih pitanja<sup>5</sup>.

---

<sup>5</sup> Za rešavanje ovog, kao i još šest najbitnijih problema savremene matematičke teorije, Klej-ev matematički institut (Clay Mathematical Institute) je 2000. god. ponudio nagradu od milion američkih dolara. Na žalost, do današnjeg dana, sem jednog koji je rešen 2003. godine, ostali, među kojima i problem  $P = NP?$ , ostaju bez odgovora. Značaj ovog pitanja je vrlo velik jer pozitivan odgovor znači da se svaki problem iz  $NP$ , a većina problema odlučivanja spada u ovaj skup, može rešiti algoritmom polinomijalne kompleksnosti. U suprotnom, ukoliko se dokaže da je  $P \neq NP$  tada je jasno da energiju treba usmeriti na rešavanje  $NP$ -complete problema.

Posebno značajna klasa  $NP$  problema jesu  $NP$ -complete problemi koji se karakterišu time da se svi problemi iz  $NP$  mogu redukcijom transformisati u njih. Uz to redukcijom je moguće jedan  $NP$ -complete problem transformisati u drugi. Ovo znači da pronalaženje polinomijalnog algoritma za rešavanje nekog od  $NP$ -complete problema ujedno znači i mogućnost rešavanja svih problema iz  $NP$  u polinomijalnom vremenu. Primer redukcije problema u okviru  $NP$  skupa je prikazan na slici 4.2.



**Slika 4.2.** *Primer redukcije problema u okviru  $NP$ -a (Dasgupta i ostali, 2008)*

Naravno, proces redukcije takođe zahteva odgovarajuće angažovanje procesorskih resursa, odnosno vreme za realizaciju. Ukoliko je ovo vreme polinomijalno u odnosu na veličinu problema, onda će ukupno vreme rešavanja početnog problema biti jednako zbiru dva polinomijalna algoritma, što je opet polinomijalan algoritam, ali sa većim stepenom polinoma, ili većim koeficijentom elementa polinoma najvećeg stepena. Ukoliko je pak za redukciju potreban algoritam eksponencijalne zavisnosti u odnosu na veličinu problema tada se i za rešavanje početnog problema koristi algoritam eksponencijalne kompleksnosti. Ovo znači da se problem svrstava u  $NP$ -complete skup problema kada pripada skupu  $NP$  i kada se svaki problem iz skupa  $NP$  može u polinomijalnom vremenu redukovati u posmatrani problem.

Kada se radi o rešavanju optimizacionih problema, jasno je da oni ne spadaju u klasu problema odlučivanja i da je njihovo rešavanje znatno kompleksnije. Drugačije rečeno, rešavanje optimizacionog problema je u najboljem slučaju kompleksno kao rešavanje najkompleksnijeg  $NP$  problema. Ukoliko ovi problemi imaju odgovarajuću verziju koja pripada  $NP$  skupu, tj. ukoliko se dato rešenje može verifikovati u polinomijalnom vremenu, tada se ovi problemi svrstavaju u  $NP$ -hard klasu problema. Treba napomenuti da se u  $NP$ -hard klasu problema svrstavaju i oni problemi čije se rešenje ne može verifikovati u polinomijalnom vremenu, tj. koji ne pripadaju  $NP$  skupu.

Činjenica da pozitivan odgovor na pitanje da li je  $P=NP$  znači da je ista kompleksnost algoritama za rešavanje bilo kog problema iz  $NP$  i algoritma za verifikaciju rešenja je ujedno i osnovni razlog zašto istraživači sumnjaju u njenu istinitost. Iz tog razloga, velika većina istraživača je svoju pažnju usmerila ka prevazilaženju kompleksnosti u rešavanju problema. Načini za to su predstavljeni u nastavku rada.

### **4.3. Načini upravljanja pretovarnim sistemima sa kumulativnom ciljnom funkcijom**

Način na koji će se upravljati pretovarnim sistemima sa kumulativnom ciljnom funkcijom, ili generalno posmatrano način rešavanja problema, zavisi pre svega od veličine problema koji se rešava. Naime, ukoliko su problemi malih dimenzija moguće ih je rešiti do optimalnosti i krajnje neefikasnim algoritmima. Preciznije, ukoliko su za donošenje odluke raspoloživi neograničeno vreme i memorijski resursi svaki problem je moguće rešiti do optimalnosti nekim od enumerativnih algoritama. Međutim, kako je operativno upravljanje vezano za kratke vremenske rokove to sledi da je upravljačke odluke potrebno doneti u kratkim vremenskim rokovima, što znači da je odluka o načinu upravljanja posmatranim pretovarnim sistemima definisana i vremenskim aspektom problema.

Kada se govori o vremenskom aspektu problema, treba naglasiti da se u ovom istraživanju pod njim podrazumeva aspekt isključivo vezan za vreme realizacije algoritama kojima se problem rešava, tj. vreme za koje se dobija rešenje problema. Sa druge strane, vremenski aspekt u upravljanju se često povezuje sa vremenskim periodom na koje se upravljanje odnosi. Razlog za to je mogućnost upravljanja sistemima u periodima različite dužine (dan, smena, radni sat, itd.). Međutim, dužina

vremenskog perioda na koje se upravljanje odnosi je u direktnoj vezi sa veličinom problema koji se rešava i zato je treba posmatrati samo kao okvirnu meru za broj zadataka koje je potrebno opslužiti. Posmatranje dužeg planskog perioda podrazumeva i rešavanje problema većih dimenzija jer se u njemu javlja i veći broj pretovarnih zadataka. U tom smislu u nastavku rada su prvo opisani načini upravljanja pretovarnim sistemima kada se u obzir uzimaju svi zadaci iz planskog perioda, a nakon toga i načini kada se, zbog velikog broja zadatka, umesto celog planskog perioda posmatraju njegovi delovi, tj. kada se broj tretiranih zadataka umanjuje smanjivanjem vremena na koji se upravljanje odnosi.

#### **4.3.1. Upravljanje koje se odnosi na ceo planski period**

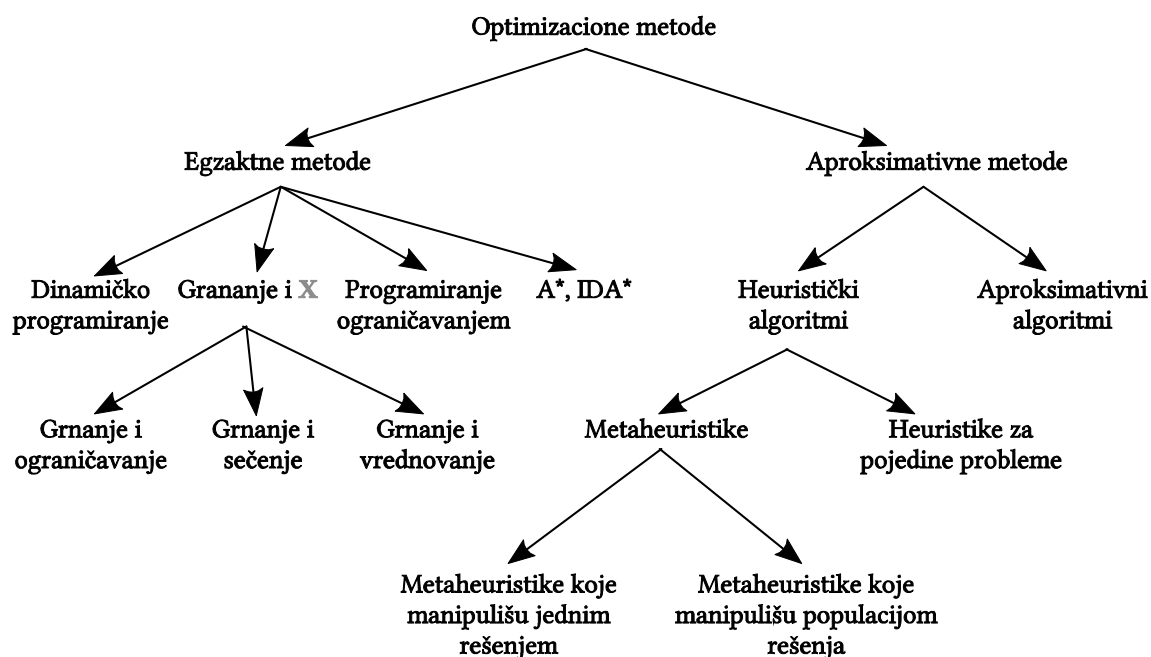
Upravljanje sistemima kada se u obzir uzimaju podaci o celom planskom periodu karakteriše se simultanim tretiranjem svih pretovarnih zadataka koji se javljaju u planskom periodu, što za posledicu ima mogućnost dobijanja optimalnih upravljačkih odluka za posmatrani period. Preduslov za ovakav način upravljanja je blagovremena raspoloživost informacija o stanju sistema u toku planskog perioda, što se pre svega odnosi na informacije koje se tiču zadataka koji se ne nalaze u sistemu na početku planskog perioda, već se dinamički pojavljuju tokom planskog perioda.

Najveći nedostatak ovakvog načina upravljanja je stohastička priroda logističkih, a samim tim i pretovarnih procesa, jer se može desiti da se informacije o stanju sistema u nekom trenutku u toku planskog perioda pokažu kao pogrešne, usled nekog od mnogobrojnih razloga (kvarovi na pretovarnim uređajima, pojava loših vremenskih uslova, itd.). Operativni plan napravljen na bazi pogrešnih informacija rezultira i slabijim performansama sistema, a u ekstremnim slučajevima se i odbacuje kao neupotrebljiv.

Međutim, razvoj informacionih, ali i svih drugih tehnologija sve više doprinosi redukciji neizvesnosti u pretovarnim procesima. Na primer: skoro sva sredstva spoljnog transporta su opremljena uređajima za praćenje prostorne pozicije sredstva u realnom vremenu; moguće je sa sve većom preciznošću predvideti meteorološke uslove koji mogu uzrokovati kašnjenje transportnih sredstava koja učestvuju u kašnjenju; dijagnostifikovanje kvarova na pretovarnim sredstvima je takođe ubrzano permanentnim nadzorom te su sredstva manje vremena u stanju kvara... Sve ovo

doprinosi značaju i povećava mogućnost upravljanja pretovarnim sistemima kada se u obzir uzima ceo planski period.

Mogućnost pronalaženja optimalnih ili, kada njih nije moguće naći, njima bliskih upravljačkih odluka je cilj koji je prisutan u svim istraživanjima koja tretiraju ovu oblast. Zato ne čudi da je do danas razvijen veliki broj optimizacionih metoda koje tretiraju ceo planski period. Jedan način njihove klasifikacije je predstavljen u Talbi (2009) i prikazan je na slici 4.3., a opis onih koje su na neki način bitne za ovo istraživanje je dat u nastavku.



**Slika 4.3.** Klasifikacija optimizacionih metoda (Talbi, 2009)

#### 4.3.1.2. Egzaktne metode

Osnovna karakteristika egzaktnih metoda je da se njihovom primenom garantuje dobijanje optimalnog rešenja problema. Za razliku od aproksimativnih metoda, koje mogu rezultirati optimalnim rešenjem, ali ne mogu garantovati da je rezultujuće rešenje optimalno, egzaktne metode to mogu.

Najprostija egzaktna metoda se dobija jednostavnom enumeracijom svih rešenja problema, koja podrazumeva listanje svih mogućih rešenja problema i odabir onog rešenja koje najviše odgovara zadatoj ciljnoj funkciji. Na taj način je dokazano da ne postoji rešenje koje više zadovoljava postavljeni cilj. Međutim, reč je o veoma neefikasnoj metodi, što se može pokazati na primeru dobro poznatog problema

trgovačkog putnika – TSP (eng. Traveling Salesman Problem). Naime, kako je rešenje TSP-a sekvenca lokacija koje trgovački putnik obilazi po redosledu, počevši od krajnjeg levog, to je jasno da je broj rešenja problema reda  $n!$ , gde je  $n$  broj zadataka, odnosno da je minimalna kompleksnost ovakvog algoritma  $O(n!)$ . Slikovita prezentacija neefikasnosti algoritma je predstavljena u Yalaoui i ostali (2012), a prikazana je u tabeli 4.2., iz koje se vidi da je i za relativno male probleme potrebno neprihvatljivo mnogo vremena za njihovu realizaciju.

**Tabela 4.2.** Zavisnost vremena rešavanja problema od njegove veličine (Yalaoui i ostali, 2012)

Veličina problema ( $n$ )	Broj mogućih rešenja ( $n!$ )	Vreme rešavanja*			
		[s]	[dan]	[mesec]	[godina]
1	1	0.000001	-	-	-
2	2	0.000002	-	-	-
3	6	0.000006	-	-	-
4	24	0.000024	-	-	-
5	120	0.00012	-	-	-
6	720	0.00072	-	-	-
7	5 040	0.00504	-	-	-
8	40 320	0.040320	-	-	-
9	362 880	0.36288	-	-	-
10	3 628 800	3.6288	-	-	-
11	39 916 800	39.9168	-	-	-
12	479 001 600	479.0016	-	-	-
13	6 227 020 800	6 227,0208	-	-	-
14	87 178 291 200	87 178.2912	1.009008	-	-
15	1.30767E+12	1 307 674.368	15.13512	-	-
16	2.09228E+13	20 922 789.89	242.16192	8.072064	-
17	3.55687E+14	355 687 428.1	4 116.75264	137.225088	11.435424
18	6.40237E+15	6 402 373 706	74 101.54752	2 470.051584	205.837632

\* pretpostavljeno je da je računaru za jednu elementarnu operaciju potrebna  $1\mu\text{s}$

Zbog praktične neprimenljivosti enumeracije, kao metode za rešavanje problema istraživači su bili primorani da pristupe „sophisticiranijim“ metodama koje direktno tretiraju mnogo manji broj rešenja, ali koje, na osnovu iskorišćenih karakteristika problema, mogu garantovati optimalnost dobijenog rešenja. Osnovna osobina problema koja je iskorišćena je mogućnost predstavljanja prostora rešenja u formi drveta, što je omogućilo različite načine pretrage radi pronalaženja optimalnog rešenja.

Metoda grananja i ograničavanja – B&B (eng. Branch and Bound) je često korišćena metoda za rešavanje celobrojnih problema linearnog programiranja – ILP (eng. Integer Linear Programming), mada se može koristiti i za rešavanje mešovitih celobrojnih problema linearnog programiranja – MILP (eng. Mixed Integer Linear Programming). Najveći broj softverskih rešenja za rešavanje ILP i MILP problema je zasnovan na ovoj metodi, koja je bazirana na pretrazi prostora rešenja dinamičkim građenjem drveta, počevši od korenog čvora. Rešenja problema su predstavljena kao listovi drveta, dok čvorovi između korenog čvora i listova, tj. grane drveta predstavljaju podskupove prostora rešenja sa zajedničkim karakteristikama. Metoda redukuje broj posmatranih rešenja problema tako što ne pretražuje one grane čija je minimalna vrednost najboljeg rešenja u grani lošija od trenutno najboljeg rešenja.

Dinamičko programiranje je još jedna često korišćena egzaktna metoda koja je bazirana na zapažanju da delovi optimalnog rešenja moraju takođe biti optimalni u svom domenu. Iz tog razloga, rešavanje problema ovom metodom se svodi na rekurzivnu redukciju problema i rešavanje do optimalnosti ovako redukovanih problema. Odbacivanjem onih podproblema, tj. delova rešenja, koja ne vode ka optimalnom redukuje se broj tretiranih rešenja i ubrzava se proces rešavanja problema.

Programiranje ograničavanjem – CP (eng. Constraint Programming) je metoda koja, kao što joj i ime sugeriše, do rešenja dolazi dovođenjem u relaciju skupa vrednosti koje promenljive mogu uzeti i skupa ograničenja koja moraju zadovoljiti, pri čemu ograničenja mogu imati matematičku ili simboličku formu. Baza metode je zasnovana na logičkim konceptima i implikacijama pretrage podataka predstavljenih u formi drveta.

Kako većina upravljačkih problema spada u klasu optimizacionih (pripada *NP*-hard skupu) problema, to je jasno da se egzaktni algoritmi mogu koristiti samo za rešavanje realnih problema malih dimenzija. Primer maksimalne veličine *NP*-hard problema koji su rešeni nekom od savremenih egzaktnih metoda je predstavljen u tabeli 4.3.



**Tabela 4.3.** *Maksimalni red veličina optimizacionih problema rešenih nekom egzaktnom metodom (Talbi, 2009)*

Problem	Kvadratni problem dodeljivanja	Flow-shop vremensko raspoređivanje	Bojenje grafa	Kapacitivni problem rutiranja vozila
Veličina instance	30 objekata	100 poslova i 20 mašina	100 čvorova	60 klijenata

#### 4.3.1.3. *Aproksimativne i heurističke metode*

Uprkos tome što ih neki autori (Dasgupta, 2008) nazivaju „inteligentnim“, metode za pretragu drвета rešenja su i dalje eksponencijalni problem, što znači da mogu trajati neprihvatljiv vremenski period. Kako bi se izborili sa ovim problemom istraživači su bili prinuđeni na kompromis u vidu odustajanja od optimalnosti rešenja u korist kraćeg vremena rešavanja algoritma. Naime, iako je optimalno rešenje ono koje u najvećoj mogućoj meri zadovoljava postavljeni cilj, vrlo često je u znatno kraćem vremenu moguće dobiti rešenje koje se, u relativnom odnosu veličina, minimalno razlikuje od optimalnog.

U zavisnosti od spremnosti donosioca odluke da se odrekne kvaliteta upravljačke odluke zavisi i vreme rešavanja upravljačkog problema. Naime, vrlo je verovatno da je svaki donosilac odluke u operativnom nivou upravljanja spreman da se odrekne par procenata uštede koja bi se postigla optimalnim rešenjem problema, zarad smanjenja vremena dobijanja rešenja na prihvatljiv nivo. Ovo tvrdnja dobija na težini ukoliko se zna da savremene alternative „inteligentnim“ metodama za pretragu drвета rešenja u velikom procentu generišu rešenja čije relativno odstupanje od optimalnih je unutar jednog procenta, ili je pak veoma blisko optimalnom, za veličine problema prisutne u praksi. Naravno, tačno odstupanje je u većini slučajeva nemoguće odrediti jer se problemi i ne mogu rešiti do optimalnosti.

Metode koje su alternativa „inteligentnim“ metodama za pretragu drвета rešenja smeštaju se u dve osnovne klase:

- aproksimativne metode
- heurističke metode

Osnovna karakteristika metoda iz obe klase je da na osnovu saznanja o karakteristikama problema koji se rešava do kvalitetnih rešenja problema dolaze

znatno brže od egzaktnih metoda. Razlika između metoda leži u garanciji kvaliteta rešenja. Naime, u klasu aproksimativnih metoda se svrstavaju one za koje je teorijski dokazan kvalitet rešenja i vreme izvršavanja. U Talbi (2009) se pod  $\beta$ -aproksimativnim algoritmom podrazumeva polinomijalni algoritam za rešavanje posmatranog problema koji za svaku instancu generiše rešenje  $R$  tako da je zadovoljeno<sup>6</sup>

$$f(R) \leq \beta \cdot f(G) \text{ ukoliko je } \beta > 1$$

$$\beta \cdot f(G) \leq f(R) \text{ ukoliko je } \beta < 1$$

gde je  $G$  optimalno rešenje problema. Faktor  $\beta$  definiše relativnu garanciju efikasnosti algoritma i može biti konstantan, ili u funkciji od veličine problema. Na osnovu rečenog jasno je da se primenom aproksimativnih algoritma dobijaju rešenja koja će od vrednosti ciljne funkcije optimalnog rešenja odstupati maksimalno za faktor  $\beta$ .

Za razliku od aproksimativnih, heurističke metode ne nude nikakve teorijske garancije za efikasnost rešavanja problema, već se ona dokazuje brojnim empirijskim eksperimentima. Heurističke metode su uglavnom rezultat dobrog razumevanja problema koji se rešava, intuicije, dosetke i dovtljivosti istraživača da će određeni načini pristupa i rešavanja problema dati očekivane rezultate. Uprkos odsustvu teorijske podloge za garanciju efikasnosti i efektivnosti metoda, zbog izuzetno povoljnog odnosa kvaliteta rešenja i vremena potrebnog za njihovo dobijanje heuristički algoritmi su naišli na veliko prihvatanje od strane istraživača, a sa tim i na brojne praktične primene kako u fundamentalnim naukama (fizika, hemije, ...), tako i u praktičnim procesima (logistike, lanaca snabdevanja, pretovarnih procesa, ...).

Iako realizacija heurističkih algoritama podrazumeva individualni pristup svakom problemu kako bi se iskoristile sve specifičnosti problema za njegovo efikasno rešavanje, mogu se razlikovati dva bazična pristupa uobličavanju ovih algoritama. Na jednoj strani su algoritmi koji su razvijeni isključivo na osnovu uvida u karakteristike, strukturu i specifičnosti posmatranog problema, gde se sve osobenosti problema koriste za razvijanje efikasnog načina rešavanja problema. Ovi algoritmi su isključivo namenjeni rešavanju konkretnog problema, jer upotreba bilo kog iskorišćenog principa pri rešavanju drugih problema najverovatnije neće imati nikakav pozitivan

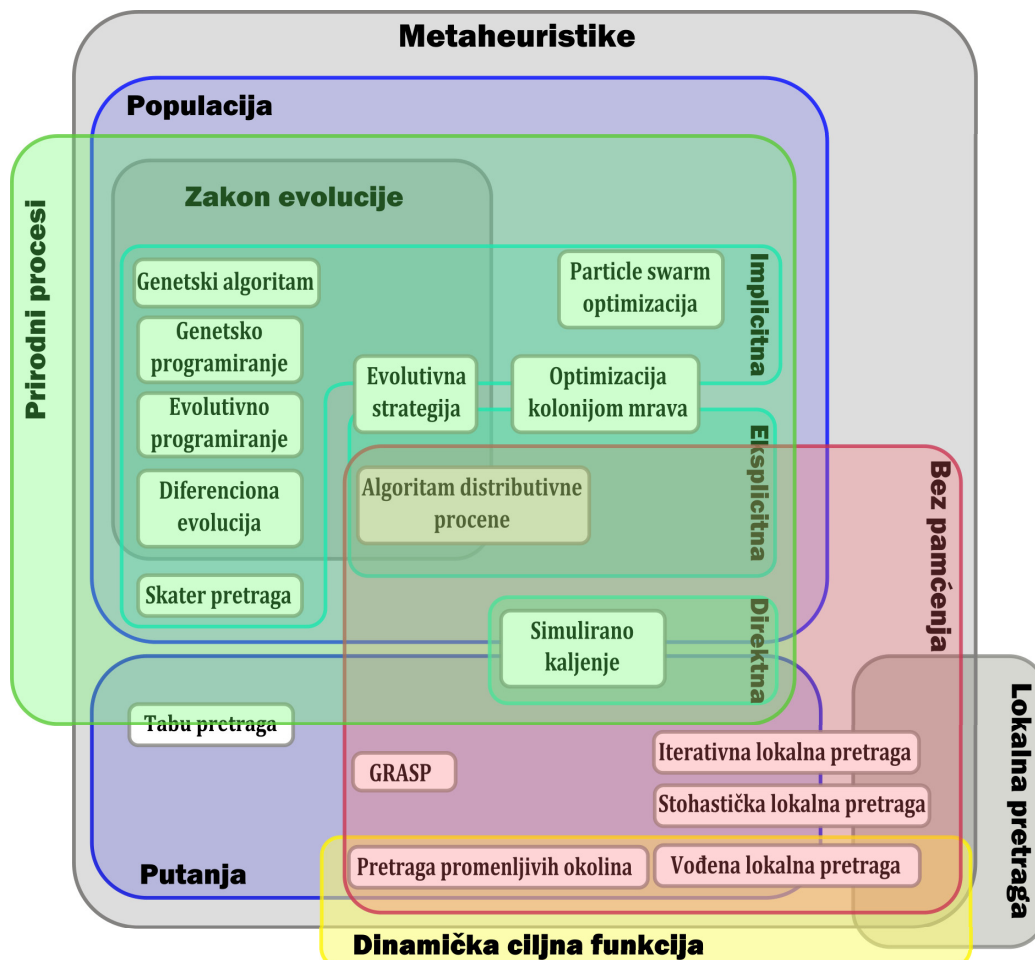
---

<sup>6</sup> Uz pretpostavku o minimizacionoj prirodi ciljne funkcije

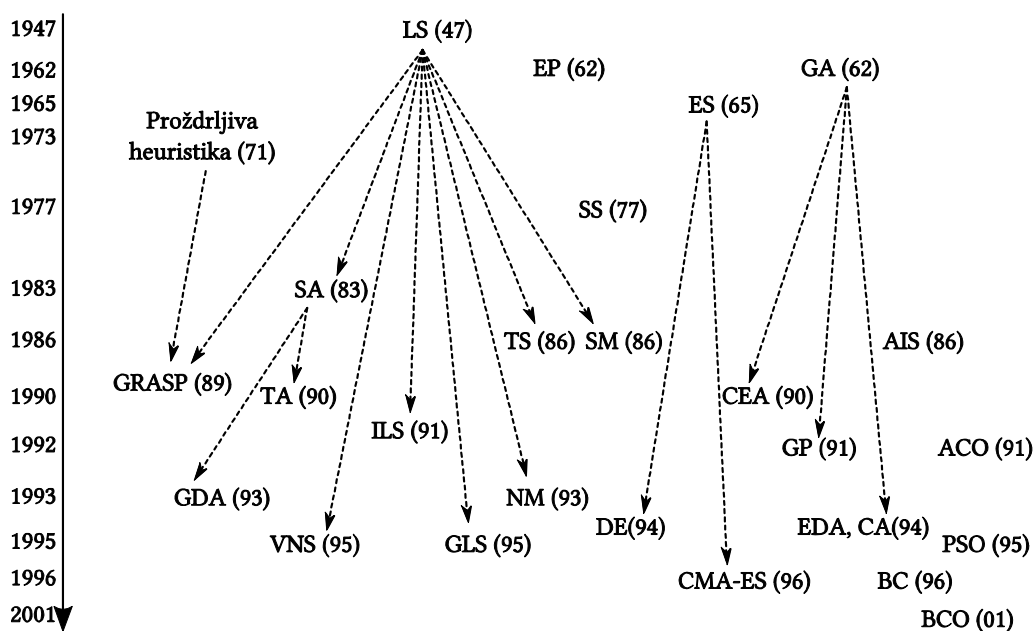
efekat na efikasnost drugog algoritma. Usled iskorišćenja specifičnosti posmatranog problema u njegovom rešavanju, ovi algoritmi su po pravilu izuzetno efikasni u pogledu vremena rešavanja.

Sa druge strane su heuristički algoritmi koji do rešenja dolaze okvirno definisanom strategijom izgradnje rešenja, ili načinom pretrage skupa mogućih rešenja problema. Strategije na osnovu kojih se generiše konačno rešenje algoritma se nazivaju metaheuristike, zbog čega se i heuristički algoritmi razvijeni na bazi strateški definisane ideje o rešavanju problema nazivaju metaheuristički algoritmi. Generalno gledano, metaheuristike svoju efikasnost i strukturu crpe ili iz analogije sa efikasnošću i načinom realizacije procesa koji se odvijaju u prirodi (npr. sposobnost mrava da nađu najkraći put do izvora hrane, kaljenje gvožđa u proizvodnji čelika, principi na osnovu kojih funkcioniše evolucija, ...), ili sa strategijama koje dokazano efikasno pretražuju prostor rešenja problema.

Usled praktične efikasnosti metaheurističkih algoritama i njihove popularnosti među istraživačima, odnosno velikog broja do sada definisanih algoritama, praktično ih je nemoguće klasifikovati na univerzalno prihvatljiv način. Ovome doprinosi i to što se „neprestano“ pojavljuju nove metaheuristike, kao i to što se u rešavanju problema vrlo često one kombinuju kako bi se na efikasniji način došlo do kvalitetnih rešenja. Kako bi se stekla bar nekakva slika o raznolikosti kriterijuma na osnovu koji je moguće klasifikovati metaheuristike, kao i na dinamiku pojave najvažnijih od njih na slici 4.4. je prikazan jedan primer klasifikacije metaheuristika na osnovu funkcionalnih karakteristika, dok je na slici 4.5. prikazan hronološki redosled pojave i odnos među pojedinim metaheuristikama.



Slika 4.4. Primer klasifikacije metaheuristika (preuzeto sa <http://metah.nojhan.net>)



Slika 4.5. Modifikovani prikaz vremenskog okvira pojave pojedinih metaheuristika prema Talbi (2009)

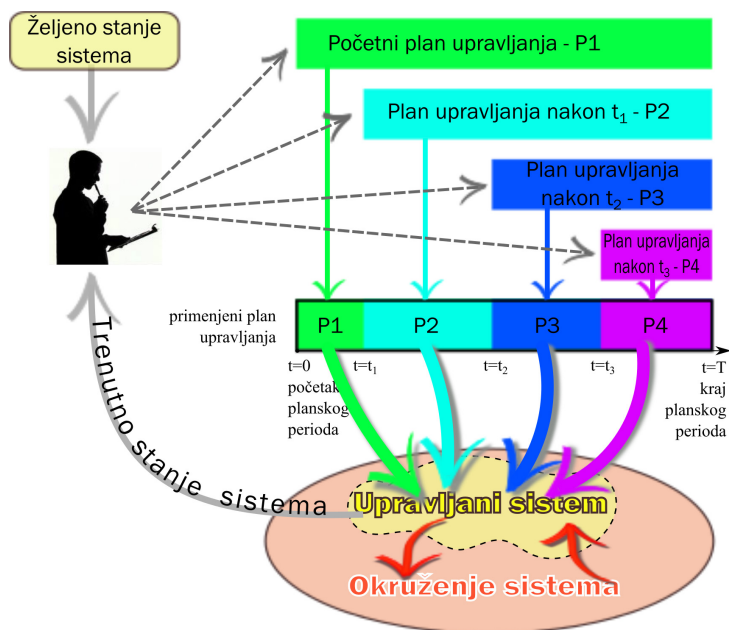
Obzirom na postojanje različitih alternativa za pomoć pri donošenju upravljačkih odluka koje se odnose na ceo planski period logično se postavlja pitanje koju alternativu koristiti u kojim uslovima. Kao što je ranije rečeno vremenski aspekt ima verovatno ključnu ulogu u odgovoru na ovo pitanje, ali ne treba zanemariti ni uticaj kvaliteta rešenja. U tom smislu, ukoliko je to vremenski prihvatljivo na prvom mestu je rešavanje problema nekom od egzaktnih metoda, pri čemu treba napomenuti da iako one podrazumevaju algoritme eksponencijalne kompleksnosti određene strukture i/ili instance problema se mogu rešiti u prihvatljivom vremenu čak i za probleme praktičnih dimenzija.

Ukoliko se ipak pokaže da se struktura problema i/ili instanca ne može rešiti u prihvatljivom vremenu kao sledeći izbor se nameće neki od aproksimativnih algoritama zbog garancija za kvalitet rešenja koje se dobija na izlazu. Iako su ovi algoritmi praktično primenljiviji u odnosu na egzaktne, njihov nedostatak je što su kao i heuristički algoritmi vezani za rešavanje posmatranog problema tako da je velika šansa da za konkretan problem ne postoji aproksimativni algoritam.

Na kraju preostaje rešavanje problema nekim od heurističkih algoritama. Izbor konkretnog metaheurističkog okvira za rešavanje problema je stvar iskustva, intuicije, primenljivosti algoritma i ličnih preferencija istraživača. Naravno, u konstrukciji heurističkih algoritma metaheuristika pruža samo okvir za rad algoritma, tako da se istraživači često odlučuju za implementaciju segmenata drugih (meta)heurističkih algoritma, aproksimativnih algoritama, ili najčešće, specifičnih karakteristika konkretnog problema.

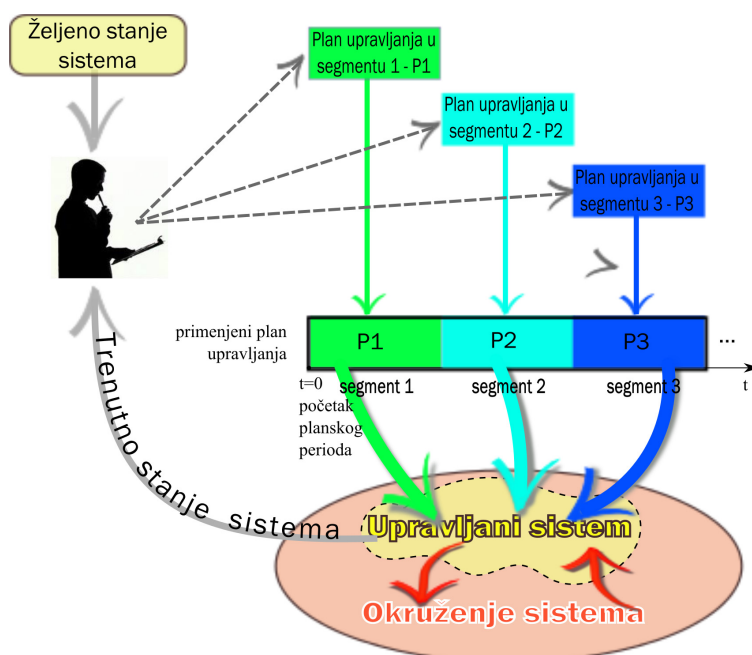
#### **4.3.2. Upravljanje sistemom sa sukcesivnim ažuriranjem informacija**

U slučaju sistema sa izrazitom stohastičnošću procesa u kojima kvalitet upravljačkih odluka donetih na početku planskog perioda degradira sa protokom vremena usled razlikovanja planiranog od faktičkog stanja sistema, upravljanje se obično realizuje podelom planskog perioda na segmente manjih dužina (slika 4.6). Na početku svakog segmenta se na bazi trenutnih informacija o stanju sistema pravi novi plan upravljanja koji se odnosi na sve segmente do kraja planskog perioda. Na ovaj način se upravljačke odluke sukcesivno usaglašavaju sa faktičkim stanjem sistema i sistemom se upravlja na osnovu poslednje donetog, tj. najžaurnijeg plana.



**Slika 4.6.** Prikaz upravljanja sistemom sa sukcesivnim ažuriranjem informacija

Ukoliko se dužina perioda na koji se doneti plan odnosi, tj. dužina svih segmenata obuhvaćenih planom, uzme kao konstantna veličina (npr. 8h, 24h, i sl.) tada se sistemom može upravljati do beskonačnosti jer se na poslednje posmatranu dužinu planskog perioda sa svakim donošenjem novog plana planski period produžava za dužinu segmenta. Usled načina na koji se planovi ažuriraju ovakav način upravljanja je u literaturi poznat kao rolling horizon upravljanje i prikazan je na slici 4.7.



**Slika 4.7.** Rolling horizon upravljanje sistemima

U zavisnosti od trenutaka u kojima se vrši generisanje novog plana razlikuje se vremenski inicirano planiranje (eng. time driven planning) u kome se novi planovi prave u tačno definisanim trenucima vremena i planiranje inicirano događajima (eng. event driven panning) u kome se planovi ažuriraju u trenucima realizacije, za sistem značajnih događaja (Fleischmann i ostali, 2005). Kako upravljanje sistemom u nezavisnim segmentima ne može garantovati optimalno upravljanje u celom planskom periodu, čak i kada su upravljačke odluke u svakom segmentu optimalne, u vremenski iniciranom upravljanju simultano sagledavanje više od jednog segmenta značajno doprinosi kvalitetu dobijenih rešenja, iako za posledicu ima rešavanje problema većih dimenzija. Iz tog razloga se prvi plan uvek donosi na bazi sagledavanja celokupnog planskog perioda.

#### **4.3.3. Upravljanje sistemom sa nultim planskim periodom**

Kao i kod upravljanja sistemom koje se bazira na ažuriranju operativnog plana na osnovu informacija do kraja planskog perioda, a koje se realizuje u trenucima koji su značajni za funkcionisanje sistema, i upravljanje sistemom sa nultim planskim periodom se takođe realizuje u „značajnim“ trenucima funkcionisanja sistema. Međutim, u ovom slučaju upravljačka odluka se donosi smo na bazi trenutnog stanja sistema, bez ikakvog uvida u stanje sistema u budućnosti. U praksi je ovaj način upravljanja poznat pod nazivom dispečiranje.

U odnosu na odstupanje trenutnog od planiranog stanja sistema dispečiranje se smatra najotpornijim načinom upravljanja, jer se nikakve informacije o stanju sistema u budućnosti i ne uključuju u proces donošenja odluke. To za posledicu ima odluke do kojih se dolazi skoro trenutno, ali sa druge strane, njihov kvalitet je daleko lošiji od odluka baziranih na celom planskom periodu, ili odluka donetih na osnovu preostalih segmenata do kraja planskog perioda.

Upravljanje dispečiranjem podrazumeva korišćenje određenog skupa upravljačkih pravila, tzv. pravila dispečiranja, koja se primenjuju po realizaciji predefinisano dogadaja. U odnosu na događaj na koji reaguju, pravila se dele na ona inicirana pojavom zahteva za opslugom i na pravila inicirana pojavom slobodnog resursa, pri čemu je za realizaciju upravljanja potrebno implementirati barem po jedno pravilo svake vrste.

Iako je efikasnost ovog načina upravljanja znatno lošija od upravljanja egzaktnim i heurističkim algoritmima, što se može videti i u Bjelić, (2009), ono je široko rasprostranjeno u praktičnim sistemima. Razlog za to je jednostavnost pravila na kojima je bazirano, kao i različiti načini realizacije upravljanja, tretiranje više od jednog kriterijuma pri donošenju odluke, široka primenljivost itd. Kao potvrda popularnosti ovakvog načina upravljanja u praksi može da posluži podatak da je u literaturi moguće naći preko stotinu, što jednostavnih, što kompleksnih, pravila dispečiranja.

Uprkos vremenskoj efikasnosti i praktičnoj primenljivosti, upravljanje dispečiranjem nije predmet ovog istraživanja zbog lošeg kvaliteta upravljačkih odluka. Pored toga, zbog pojava koje su u poglavlju 4.3.1. navedene kao razlozi za smanjenje stohastičnosti procesa u pretovarnim sistemima iz razloga pouzdanosti informacija o pojavama zahteva u planskom periodu, kao i mogućnosti da se heurističkim algoritmima za relativno kratko vreme dođe do kvalitetnih upravljačkih odluka koje obuhvataju ceo planski period, kao predmet ovog istraživanja se nametnulo upravljanje pretovarnim sistemima na bazi informacija iz celog planskog perioda. Naravno, obzirom da se radi o kompleksnim kombinatornim optimizacionim problemima, za očekivati je da se radi o *NP*-hard problemima, za koje ne postoje polinomijalni algoritmi i koji ne mogu biti rešeni do optimalnosti za probleme većih dimenzija. U tom smislu, sa ciljem dokazivanja kompleksnosti posmatranog problema, u narednom poglavlju će problem biti formulisan kao generalizacija, u literaturi poznatog, problema putujućeg servisera – TRP (eng. Traveling Repairman Problem). Nakon modeliranja problema kao TRP, predstavljeni su načini za njegovo rešavanje, kako za probleme malih dimenzija, koje je moguće rešiti do optimalnosti, tako i za probleme dimenzija praktičnih problema. Na osnovu rezultata istraživanja objavljenih u literaturi u poslednjih par godina za rešavanje TRP-a i njemu srodnih problema praktičnih veličina, po efikasnosti, kako sa aspekta kvaliteta rešenja, tako i sa aspekta vremena rešavanja, se izdvojila metoda pretrage promenljivih okolina - VNS (eng. Variable Neighborhood Search) zbog čega je problem upravljanja pretovarnim sistemima sa kumulativnom ciljnom funkcijom u nastavku rešavan upravo ovom metaheuristikom.



## **5. Modeliranje pretovarnih sistema sa kumulativnom ciljnom funkcijom i dinamičkom pojavom zadataka**

---

Generalno gledano pod modelom se podrazumeva pojednostavljena reprezentacija posmatranog sistema formirana sa svrhom njegovog izučavanja (Banks i ostali, 2009). U skladu sa tim, modeliranje se može posmatrati kao proces definisanja granica posmatranog sistema, cilja funkcionisanja, uočavanja osnovnih elemenata, njihovih međusobnih odnosa, kao i odnosa sa okruženjem sistema, simplifikacije i apstrakcije elemenata sa zanemarljivim ili nebitnim uticajem na funkcionisanje sistema koje na kraju rezultira modelom. Iako su po definiciji pojednostavljene reprezentacije realnih sistema, modeli moraju obuhvatiti sve njihove relevantne karakteristike tako da se saznanja o modeliranom sistemu, dobijena primenom modela sa različitim ulaznim podacima ili različitim scenarijima funkcionisanja, mogu smatrati verodostojnim.

U odnosu na način predstavljanja strukture elemenata realnog sistema, kao i njihovog međusobnog odnosa, za ovo istraživanje su od značaja modeli kod kojih je odnos među elementima moguće izraziti raspoloživim matematičkim aparatom, tj. matematički modeli. Predstavljanje međusobnog odnosa elemenata pretovarnog sistema u formi matematičkog modela omogućuje da se za rešavanje posmatranog problema koristi neka od raspoloživih metoda (i/ili alata) koje su do sada razvijene i korišćene u rešavanju matematičkih modela. Pored njih, za rešavanje problema heurističkim metodama, od značaja su računarski modeli, tj. modeli koji obuhvataju sve osobine heurističkih metoda za rešavanje posmatranog problema, tako da na izlazu, za dati skup ulaznih veličina, daju izlazne veličine kojima se definiše rešenje heurističkog modela.

Uzimajući u obzir dostignuća do kojih su istraživači došli u efikasnom rešavanju problema, kao logičan prvi korak pri rešavanju bilo kog problema operativnog upravljanja nameće se provera mogućnosti predstavljanja posmatranog problema kao jednog od problema kojim su se istraživači već bavili. Ukoliko je to slučaj sva akumulirana znanja je moguće iskoristiti u rešavanju posmatranog problema. Čak i u situaciji kada posmatrani upravljački problem ne odgovara u potpunosti nekom od poznatih i istraživanih problema, moguće je na osnovu saznanja drugih istraživača o sličnim problemima definisati smer istraživanja u pravcu koji je dao dobre rezultate pri rešavanju drugih problema. U skladu sa tim, u narednom poglavlju je problem operativnog upravljanja jednom klasom pretovarnih sredstava u uslovima dinamičke pojave zadataka predstavljen kao generalizacija poznatog problema putujućeg servisera.

### **5.1. Problem upravljanja pretovarnim sistemima sa kumulativnom ciljnom funkcijom kao generalizacija problema putujućeg servisera**

Kao što je navedeno u poglavlju 3.1 osnovna karakteristika upravljanja pretovarnim sistemima sa kumulativnom ciljnom funkcijom je kumulativna priroda ciljne funkcije koju treba minimizovati u rešavanju problema. Obzirom da ta osobina upravljanja u znatnoj meri utiče na specifičnost rešavanja problema, u dostupnoj literaturi je moguće uočiti nekoliko problema koji se takođe odlikuju ovakvom prirodom ciljne funkcije, među kojima je najpoznatiji problem putujućeg servisera – TRP (eng.

Traveling Repairman Problem). Sličnosti i razlike TRP-a i ostalih problemima sa kumulativnom ciljnom funkcijom detaljnije su obrađeni u sledećem poglavlju.

Obzirom da TRP podrazumeva postojanje jednog servisera koji opslužuje kvarove koji postoje u sistemu od početka planskog perioda, jasno je da se on ne podudara sa posmatranim problemom po pitanju svih karakteristika i pretpostavki iz poglavlja 3.1. U tom smislu posmatrani problem predstavlja generalizaciju TRP-a tako što:

- posmatra zadatke koji na početku planskog perioda nisu prisutni u sistemu, odnosno čija opsluga, iz bilo kog razloga ne može početi do definisanog trenutka u vremenu, koji se naziva trenutak pojave zadatka. Ovako generalizovani problem se naziva problem putujućeg servisera sa vremenskim prozorima – TRPTW (eng. Traveling Repairman Problem with Time Windows). Treba naglasiti da i u slučaju TRPTW-a i u slučaju posmatranog problema zadatke je moguće opsluživati bilo kada nakon trenutka pojave, ali odlaganje početka opsluge doprinosi povećanju vrednosti ciljne funkcije te od kvalitetnih rešenja treba očekivati da počeci opsluge zadataka budu što bliže trenutcima pojava.
- omogućava da servisiranje kvarova (pretovar) realizuje više od jednog servisera, odnosno, dozvoljava postojanje skupa (flote) servisera. Pored toga, kako je za posmatrani problem karakteristično da su pretovarna sredstva različitih performansi po pitanju brzine kretanja između lokacija zadataka i proizvodnosti, to i generalizacija problema mora tretirati flotu heterogenih servisera.

Ovako generalizovan problem se naziva problemom heterogene flote putujućih servisera sa vremenskim prozorima – hetTRPTW (eng. heterogeneous Traveling Repairmen Problem with Time Windows). Modeliranje posmatranog problema kao hetTRPTW-a rezultira modelima koji obuhvataju sve relevantne karakteristike posmatranog problema.

Kao što je već rečeno modeliranje problema upravljanja pretovarnim sistemima sa kumulativnom ciljnom funkcijom i dinamičkom pojavom zadatka kao hetTRPTW-om omogućava korišćenje svih dostignuća koja si omogućila efikasno rešavanje TRP-a, ili sličnih problema. Međutim, specifične karakteristike posmatranog problema, odnosno neophodnost modeliranja kao hetTRPTW-a, je razlog da ona nisu brojna, što je i predstavljeno u poglavlju 5.2.2.

Ipak, jedno od dostignuća u rešavanju TRPTW- a je vrlo značajno za pokazivanje NP-hard prirode posmatranog optimizacionog problema. Naime, kako je hetTRPTW generalizacija TRPTW-a i kako je Tsistsiklis (1992) dokazao da TRPTW problem odlučivanja, u kome su dati samo vremenski trenuci pojave zahteva za servisiranjem, pripada NP-complete skupu problema, to je jasno da i hetTRPTW problem odlučivanja pripada NP-complete skupu problema, odnosno da njegova optimizaciona verzija pripada NP-hard skupu problema.

## 5.2. Pregled relevantne literature i dostignuća u rešavanju hetTRPTW-a

### 5.2.1. Sličnost sa drugim problemima

Kao što je ranije rečeno, osnovna karakteristika TRP-a je kumulativna priroda ciljne funkcije čijim minimizovanjem se ustvari minimizuje prosečno vreme čekanja na opslugu (Lysgaard i Wøhlk, 2013). Imajući u vidu široke mogućnosti za praktičnu primenu ovakve ciljne funkcije, u literaturi je prisutan veliki broj problema koji koriste neki njen oblik. Problemi koji se sreću u literaturi i koji dele prirodu ciljne funkcije sa TRP-om su:

- *problem minimalnog kašnjenja – MLP* (eng. Minimum Latency Problem) je problem koji je dobro poznat i proučavan u teoriji vremenskog raspoređivanja (eng. Scheduling theory). Kao i kod TRP-a problem se sastoji u pronalaženju takvog rasporeda poslova na mašini koji će rezultirati najmanjim ukupnim vremenom čekanja poslova na završetak svakog od njih.
- *problem dostavljača – DMP* (eng. Deliveryman problem) je problem koji za cilj ima minimizovanje vremena koje svi klijenti dostavljača čekaju na dostavu pošiljke. Uslovi u kojima radi dostavljač su isti kao i u slučaju putujućeg serviseru. Ono što je potrebno naglasiti, je da je osnovna razlika između TRP-a i DMP-a u tome da je vreme realizacije dostave na lokaciji klijenta kraće u odnosu na vreme potrebno serviseru za otklanjanje kvara (Angel-Bello i ostali, 2013).
- *kumulativni kapacitivni problem rutiranja – CCVRP* (eng. Cumulative Capacitative Vehicle Routing Problem) je varijanta poznatog kapacitivnog

problema rutiranja - CVRP (eng. Capacitated Vehicle Routing Problem). Kao kod CVRP-a, CCVRP podrazumeva postojanje homogene flote vozila u lokaciji polaska i skupa zahteva za prevozom. Vozila su ograničenog kapaciteta, pri čemu se pod kapacitetom podrazumeva težinska nosivost vozila, a svaki zahtev je okarakterisan težinom tereta koju je potrebno prevesti. Cilj je da se, respektujući ograničenja nosivosti, zadovolje svi zahtevi za opslugom, ali da se, za razliku od CVRP-a, minimizuje suma trenutaka dolazaka u lokacije zahteva.

- *kumulativni problem rutiranja - CumVRP* (eng. Cumulative Vehicle Routing Problem) je generalizacija CCVPR-a. Naime, ciljna funkcija u CumVRP-u minimizuje sumu proizvoda trenutaka dolaska u lokacije zahteva i količine robe koja se u njima nalazi (Kara i ostali, 2008). Sve ostale postavke problema su iste kao i kod CCVRP-a.
- *problem rutiranja servisne opreme – WRRP* (eng. Workover Rig Routing Problem) je praktični problem nastao iz potrebe da se upravlja flotom specijalizovane opreme za servisiranje uređaja koji omogućavaju ispumpavanje nafte iz kontinentalnih bušotina u Brazilu (Aloise i ostali, 2006). Ciljna funkcija koja se koristi u WRRP-u je veoma bliska ciljnoj funkciji CumVRP-a jer minimizuje ukupne gubitke ispumpavanja nafte, tj. sumu proizvoda trenutka završetka servisiranja i gubitaka usled smanjenih performansi uređaja.

Pored osnovnih klasa nabrojanih problema, u literaturi je moguće naći i njihove uopštene forme u smislu postojanja vremenskih prozora i/ili postojanja homogene ili heterogene flote vozila. Tako, na primer, Heilporn i ostali (2010) razmatraju problem dostavljača sa vremenskim prozorima - DMPTW (eng. Deliveryman Problem with Time Windows), van der Meer (2000) razmatra problem homogene flote putujućih servisera sa vremenskim prozorima – homTRPTW (eng. homogenous Traveling Repairmen Problem with Time Windows), dok Bjelić i ostali (2013a, 2013b) razmatraju hetTRPTW problem.

Iako se pomenuti problemi u literaturi često posmatraju kao ekvivalenti jedni drugima sa aspekta primenljivosti predloženih algoritma za njihovo rešavanje, pojedine njihove karakteristike u znatnoj meri to ograničavaju. Kako bi se jasno uočile razlike između pojedinih klasa problema, u tabeli 5.1 je dat uporedni pregled karakteristika

pomenutih problema na osnovu kojih je moguće uvideti njihove osnovne razlike. Prikazane karakteristike su identifikovane na osnovu modela kojima su problemi modelovani u dostupnoj literaturi.

**Tabela 5.1.** Relevantne karakteristike posmatranih problema

Karakteristike	Problemi								
	TRP	MLP	DMP	CCVRP	CumVRP	WRRP	DMPTW	homTRPTW	hetTRPTW
a) više od jednog sredstva u floti				✓	✓	✓		✓	✓
b) homogena flota				✓	✓			✓	✓
c) tretira vremenske prozore							✓	✓	✓
d) zatvoreni vremenski prozori							✓		
e) doprinos ciljnoj funkciji od početka planskog perioda	✓	✓	✓	✓	✓	✓	✓		
f) doprinos ciljnoj funkciji od početka vremenskog prozora								✓	✓
g) otvorene rute sredstava						✓			
h) sva sredstva su na početku u depou	✓	✓	✓	✓	✓		✓	✓	✓
i) sredstva imaju ograničenu nosivost				✓	✓				

Kao što se vidi iz tabele CCVRP i CumVRP imaju identične osobine po pitanju respektovanih karakteristika. Međutim, treba naglasiti da je njihova osnovna razlika u samoj ciljnoj funkciji, kako je to ranije objašnjeno.

Iz tabele se vidi da se tri karakteristike (c, d, f) tiču prisustva vremenskih prozora u posmatranom problemu. Karakteristika c) se odnosi na to da li model podrazumeva postojanje vremenskih prozora, a to da li vremenski prozor podrazumeva postojanje i leve i desne granice prozora je informacija koja je predmet karakteristike d). Ovdje treba naglasiti da je za obe varijante (i homogenu u heterogenu) TRPTW-a karakteristično to da postoji samo leva granica prozora. Razlog za to je priroda posmatranog problema koja podrazumeva da se kvar može otkloniti u bilo kom vremenskom trenutku nakon njegovog detektovanja, tj. generalno gledano ne postoji trenutak do koga se kvar mora ukloniti. U slučaju postojanja takvog trenutka problem poseduje zatvorene vremenske prozore. Na osnovu uvida u dostupnu literaturu, Heilporn i ostali (2010) su jedini koji su tretirali problem sa kumulativnom ciljnom funkcijom i zatvorenim vremenskim prozorima.

Treba napomenuti da prisustvo vremenskih prozora u modelu uopšte ne mora imati direktan uticaj na vrednost ciljne funkcije, već samo na definisanje intervala u kome je moguće posetiti klijente. Preciznije, u pomenutom radu Heilporna i ostalih (2010) iako

autori tretiraju postojanje intervala u kome je moguće posetiti klijenta, pri računanju vrednosti ciljne funkcije u obzir se uzimaju trenuci dolaska na lokaciju klijenta mereno od početka planskog perioda. Ostali modeli koji tretiraju postojanje vremenskih prozora mere čekanje klijenta od leve granice prozora, tj. trenutka od kada ga je moguće opslužiti. Razlika između modela u odnosu na ovu karakteristiku je definisana karakteristikom f).

Naravno, obzirom da prisustvo vremenskih prozora generalizuje posmatrani model, to se svi problemi koji ne tretiraju vremenske prozore mogu se posmatrati kao problemi sa otvorenim vremenskim prozorima, tj. vremenskim prozorima čija leva granica odgovara početku planskog perioda, a desna granica ne postoji.

Na osnovu podataka iz tabele 5.1 svi modeli, osim modela za rešavanje WRRP-a podrazumevaju da sredstva kreću iz depoa i završavaju u njemu, tj. da imaju zatvorene rute. Međutim, iako modeli za rešavanje svih problema iz tabele 5.1., osim WRRP-a, sadrže ograničenja koja respektuju ovu karakteristiku, to nema uticaj na ciljnu funkciju jer dužina segmenta rute koja završava u depou ne doprinosi ciljnoj funkciji. Iz tog razloga karakteristiku g) treba shvatiti uslovno, u smislu postojanja ograničenja koja nameću formiranje zatvorenih ruta kako bi sredstva sledeći planski period krenula iz depoa. Sa druge strane, svi modeli vrednost ciljne funkcije računaju na način kao da su formirane rute otvorene, tj. bez vrednosti segmenta rute koji vodi u depo. Izuzetak predstavljaju modeli za rešavanje WRRP-a. Naime, u WRRP-u ne postoji depo u koji se servisna oprema vraća već se neprekidno nalazi u radnom režimu krećući se od jedne do druge bušotine. Iz tog razloga, na početku planskog perioda WRRP-a svaki kamion sa servisnom opremom se nalazi na lokaciji bušotine na kojoj je bio na kraju prethodnog planskog perioda. Lokacija na kojoj će kamion završiti planski period je definisana otvorenom rutom koja je rezultat primenjenog WRRP-a modela.

### 5.2.2. Dostignuća u radovima koji tretiraju rešavanje TRP klase problema

Kao što je rečeno, u relevantnoj literaturi problemi opisani u prethodnom poglavlju često se tretiraju kao ekvivalentni. Kako bi se stekao uvid u trenutno stanje njihovog

rešavanja u ovom delu rada je dat pregled najznačajnijih radova iz raspoložive literature.

Iako je TRP bio predmet velikog broja istraživanja tokom prethodnih par decenija, isto se ne može reći za njegove generalizacije u pogledu prisustva većeg broja servisera u floti i/ili posmatranja vremenskih prozora. Preciznije, ove klase problema su tek u poslednjih nekoliko godina počele intenzivnije da privlače pažnju istraživača.

U pogledu najopštijeg slučaja TRP-a, tj. slučaja postojanja više od jednog servisera u floti, ali i postojanja vremenskih prozora, u raspoloživoj literaturi je moguće pronaći pet radova. U radu Bjelić i ostali (2010) autori su posmatrajući problem upravljanja flotom pretovarnih sredstava za istovar šljunka iz barži predstavili dva matematička modela za optimalno rešavanje problema. Memetski heuristički algoritam – MA (eng. Memetic Algorithm) za rešavanje hetTRPTW-a zasnovan na kombinaciji genetskog algoritma – GA (eng. Genetic Algorithm), kao nosioca pretrage po prostoru rešenja i metoda promenljivog spusta – VND (eng. Variable Neighborhood Descent), kao tehnike lokalnog pretraživanja je predložen u radu Bjelić i Vidović (2011). U radu Bjelić i ostali (2013a) predložen je GA za rešavanje hetTRPTW-a. Rad sa najboljim performansama predloženih heurističkih algoritama za rešavanje TRPTW-a je rad Bjelića i ostalih (2013b). Za optimalno rešavanje problema autori su hetTRPTW matematički formulisali kao MILP model. Međutim, kako je hetTRPTW *NP*-hard problem to je jasno da se predložena matematička formulacija može koristiti samo za probleme malih dimenzija. Kao potvrda ove tvrdnje mogu poslužiti i rezultati numeričkih eksperimenata iz rada u kome su optimalna rešenja dobijena za probleme sa maksimalno 20 zadataka i flotu od dva servisera, dok su u slučajevima flote od 5 i 10 servisera optimalno rešene instance sa maksimalno 5 zadataka.

Za rešavanje problema većih dimenzija, bilo u smislu broja zadataka, bilo broja raspoloživih servisera, autori su predložili heuristički algoritam zasnovan na VNS metaheuristici. Predloženi VNS heuristički algoritam je testiran na problemima srednjih i velikih dimenzija. Za rešavanje problema velikih dimenzija predložena je strategija za redukovanje vremena rešavanja kojom se vreme formiranja rešenja smanjuje za više od 50% u proseku, dok se rešenja degradiraju za manje od 1% u proseku.



Van der Meer (2000) u svojoj doktorskoj disertaciji posmatra problem operativnog upravljanja flotom AGV-a u različitim stanjima internog transportnog sistema. Autor je problem posmatrao kao homTRPTW. Za optimalno rešavanje problem je formulisan MILP modelom, dok je za rešavanje instanci velikih dimenzija predložen heuristički algoritam zasnovan na heuristici ubacivanja (eng. insertion heuristics).

U pogledu radova koji tretiraju samo postojanje vremenskih prozora, tj. koji rešavaju TRPTW u literaturi su dostupna dva rada. Tsitsiklis (1992) se u svom radu bavi specijalnim slučajevima TSPTW-a i TRPTW-a. Za specijalne slučajeve problema za koje je to moguće, autor predstavlja polinomijalne algoritme za njihovo rešavanje, dok u suprotnom predstavlja njihovu *NP*-complete prirodu. U ovom radu je autor pokazao da je za slučaj TRPTW-a u kome su dati samo vremenski trenuci pojave zahteva za opslugom, tj. servisiranjem, problem jako *NP*-complete.

U radu Heilporn i ostali (2010) autori razmatraju DMPTW, pri čemu su vremenski prozori ograničeni sa obe strane. DMPTW je prvo formulisan koristeći MILP model zasnovan na mrežnom protoku (AF-DMP), a zatim i pomoću MILP modela zasnovanog na sekvencionalnom dodeljivanju zadataka (S-DMP). Radi analize efikasnosti predloženih modela, za slučaj u kome zadaci imaju otvoren vremenski prozor sa desne strane dok su vremenski prozori u depou zatvoreni, sprovedena je analiza poliedra formiranog primenom S-DMP-a. Pored toga, autori su predložili egzaktni i heuristički algoritam za rešavanje DMPTW-a.

Problemi u kojima ne postoji ograničenje po pitanju vremenskih prozora, ali se za realizaciju zadataka koristi flota resursa, se mogu podeliti na klasu u kojoj postoji ograničenje nosivosti resursa i klasu u kojoj ovo ograničenje ne postoji. Prva klasa je u literaturi poznata kao CCVRP i za njeno rešavanje su Lysgaard i Wøhlk (2013) predstavili, do sada jedini, egzaktni algoritam. Heuristički pristupi rešavanju CCVRP-a su predstavljeni u radovima Nguveu i ostali (2010), Ribeiro i Laporte (2012) i Ke i Feng (2013) u kojima su autori koristili MA, ALNS i dvofaznu pretragu, respektivno, kao okvir za realizaciju heurističkih algoritama. Na osnovu rezultata iz pomenutih radova može se zaključiti da su najbolja rešenja testnih problema dobijena pri korišćenju ALNS i dvofaznog algoritma. Poseban slučaj CCVRP-a, u kome je ograničeno rastojanje (vreme, troškovi i sl.) koje stoji na raspolaganju resursu u

planskom periodu je razmatrano u radu Zhixing i ostali (2014) u kome su autori predstavili egzaktni algoritam za rešavanje problema.

Klasa problema u kojoj ne postoji ograničenje nosivosti resursa, niti njegove bilo koje druge karakteristike relevantne za opslugu, je poznata u literaturi kao  $k$ TRP, gde je  $k$  broj jedinica resursa u floti za realizaciju zadataka.  $k$ TRP u kome je vreme servisiranja na lokaciji zadatka isto za sve klijente je razmatran u radu Fakcharoenphol i ostali (2007). Autori su predstavili  $8.497\alpha$ -aproksimativni algoritam za rešavanje razmatranog problema, gde je  $\alpha$  najmanji aproksimativni faktor rešavanja problema drveta najmanjeg razapinjanja sa jednim korenim i ukupno  $i$  čvorova u drvetu. Slučaj kada trajanje servisiranja nije jednako za sve zadatke je razmatran u radu Jothi i Raghavachari (2007) u kome autori predstavljaju  $\frac{3}{2}\beta + \frac{1}{2}$  aproksimativni algoritam, gde je  $\beta$  aproksimativni faktor rešavanja  $k$ TRP-a. Na osnovu dostupne literature, heuristički pristupi rešavanju  $k$ TRP do sada nisu bili predmet istraživanja.

$k$ TRP kod kojeg je  $k=1$ , tj. TRP, DMP i MLP, je predmet velikog broja istraživanja. Dok su egzaktni i aproksimativni pristupi rešavanju TRP-a privlačili pažnju istraživača u prošlosti, heuristički pristup rešavanju problema im je zaokupirao pažnji tek u poslednjih desetak godina. Egzaktni algoritmi za rešavanje TRP-a su zasnovani ili na dinamičkom programiranju (Wu, 2000), ili na B&B algoritmu (Wu i ostali, 2004), ili na kombinaciji ova dva algoritma. U pogledu rešavanja problema aproksimativnim algoritmima treba napomenuti da najbolji dostignuti aproksimativni faktor iznosi 3.59 za algoritam predstavljen u radu Chaudhuri i ostali (2003).

U pogledu heurističkog pristupa rešavanju TRP-a u literature je moguće pronaći tri rada. U prvom objavljenom radu (Salehipour i ostali, 2011) heuristički algoritam za rešavanje TRP-a je zasnovan na kombinaciji prilagodljive pohlepne i slučajne procedure – GRASP (eng. Greedy Randomized Adaptive Search Procedure) i VNS algoritma. Rešenja testnih primera, kao i vreme generisanja rešenja su unapređeni VNS algoritmom predloženim u radu Mladenović i ostali (2013) i kombinacijom GRASP-a, iterativnog lokalnog pretraživanja – ILS (eng. Iterative Local Search) i metode promenljivog spusta sa slučajnim rasporedom okolina - RVND (eng. Variable Neighborhood Descent with Random neighborhood ordering), predstavljenog u Silva i ostali (2012).

Na osnovu svega rečenog u ovom poglavlju, može se zaključiti da TRP i svi, po pitanju prirode ciljne funkcije, njemu srodni problemi predstavljaju praktično primenljive probleme koji su intenzivniju pažnju istraživačke zajednice privukli tek u poslednjih desetak godina. Koji od pomenutih TRP modela se može koristiti za rešavanje konkretnog praktičnog problema zavisi od karakteristika posmatranog sistema. U tom smislu, upravljanje flotom pretovarnih sredstava sa dinamičkom pojavom zadataka u planskom periodu se može realizovati primenom heTRPTW modela predstavljenih u Bjelić i ostali (2013b). Iz tog razloga u sledećim poglavljljima su detaljno predstavljeni MILP matematička formulacija heTRPTW-a i heuristički algoritmi za njegovo rešavanje.

### 5.3. Modeliranje heTRPTW-a kao problema mešovitog celobrojnog linearnog programiranja

Formulacija heTRPTW-a kao MILP modela je zasnovana na mrežnom protoku kroz kompletan, usmeren, asimetričan i otežan<sup>7</sup> graf  $G=(N,E,t)$ . Ukoliko je sa  $V$  označen skup raspoloživih putujućih serviseru, pri čemu je  $m = |V|$  njihov broj, a sa  $P$  je označen skup zadataka, gde  $n$  ( $n = |P|$ ) označava njihov ukupan broj, zadatak rešavanja heTRPTW-a se svodi na pronalaženje  $m$  Hamiltonovih putanja kroz  $G$ , na takav način da se minimizuje ukupno vreme koje svi zadaci čekaju na završetak opsluge.

Graf  $G$  je definisan skupom čvorova,  $N$ , i skupom grana,  $E$ , pri čemu je svakoj grani iz  $E$  pridružena težina,  $t$ . Skup čvorova,  $N$ , je definisan kao  $N = P \cup \{0, n + 1\}$ , gde su čvorovi  $0$  i  $n+1$  „virtuelni“ čvorovi koji referenciraju na depo, tj. ne predstavljaju zadatke koje je potrebno servisirati. Čvor  $0$  referencira na depo kao polaznu tačku rute serviseru, dok čvor  $n+1$  referencira na depo kao na poslednji čvor rute. Skup grana,  $E$ , se definiše kao  $E = \{(v, i, j) : \forall v \in V, \forall i, j \in N\}$ . Na ovaj način je svaki par čvorova  $(i, j) \forall i, j \in N$  međusobno povezan sa  $m$  paralelnih grana, gde je po jedna grana namenjena svakom serviseru iz  $V$ . Svakoj grani iz  $E$  je dodeljena težina  $t_{ij}^v$  koja se definiše kao vreme potrebno serviseru  $v$  da nakon čvora  $i$  ode do čvora  $j$  i da opsluži

---

<sup>7</sup> Svaka grana grafa je okarakterisana težinom, koja u slučaju heTRPTW-a predstavlja vreme potrebno da se dođe do krajnjeg čvora grane i da se zadatak u krajnjem čvoru opsluži.

zadatak koji predstavlja taj čvor. Na osnovu ovakve definicije jasno je da se  $t_{ij}^v$  računa prema (1).

$$t_{ij}^v = \frac{d_{ij}}{v_v} + s_j^v \quad (1)$$

Gde je  $d_{ij}$  rastojanje između čvorova  $i$  i  $j$ , a  $s_j^v$  vreme potrebno serviseru  $v$  da opsluži zadatak  $j$ .

Ukoliko se kao promenljive odlučivanja u modelovanju problema koriste:

- $D_i$  – trenutak završetka opsluge zadatka  $i$ ,  $\forall i \in P$
- $D_0^v$  - trenutak u kome serviser  $v$  napušta depo (čvor 0 grafa  $G$ ),  $\forall v \in V$
- $D_{n+1}^v$  - trenutak u kome se serviser  $v$  vraća u depo (čvor  $n+1$  grafa  $G$ ),  $\forall v \in V$
- $x_{ij}^v = \begin{cases} 1 & \text{– ukoliko serviser } v \text{ nakon čvora } i \text{ ide u čvor } j \\ 0 & \text{– u suprotnom} \end{cases}$ ,  $\forall v \in V, \forall i, j \in N$ .

a koeficijenti imaju sledeće značenje:

- $e_i$  - predstavlja trenutak pojave kvara, tj. zadataka
- $A^v$  - predstavlja trenutak dostupnosti servisera  $v$ , tj. trenutak od koga je serviser  $v$  na raspolaganju za opslugu zadataka

tada se matematički MILP model za rešavanje hetTRPTW-a može formulirati kao:

$$\min \sum_{i \in P} (D_i - e_i) \quad (2)$$

uz ograničenja

$$\sum_{v \in V} \sum_{j \in N} x_{ij}^v = 1, \quad \forall i \in P \quad (3)$$

$$\sum_{j \in N} x_{ij}^v - \sum_{j \in N} x_{ji}^v = 0, \quad \forall i \in P, v \in V \quad (4)$$

$$\sum_{j \in P} x_{0j}^v \leq 1, \quad v \in V \quad (5)$$

$$\sum_{i \in P} x_{i_{n+1}}^v \leq 1, \quad v \in V \quad (6)$$

$$x_{ij}^v = 1 \Rightarrow D_j \geq D_i + t_{ij}^v, \quad \forall i, j \in P, v \in V \quad (7a)$$

$$x_{0j}^v = 1 \Rightarrow D_j \geq D_0^v + t_{0j}^v, \quad \forall j \in P, v \in V \quad (8a)$$

$$x_{i_{n+1}}^v = 1 \Rightarrow D_{n+1}^v \geq D_i + t_{i_{n+1}}^v, \quad \forall i \in P, v \in V \quad (9a)$$

$$\sum_{j \in P \cup \{0\}} x_{ji}^v = 1 \Rightarrow D_i \geq e_i + s_i^v, \quad \forall i \in P, v \in V \quad (10a)$$

$$D_0^v \geq A^v, \quad v \in V \quad (11)$$

$$D_{n+1}^v \geq 0, \quad v \in V \quad (12)$$

$$\sum_{v \in V} \sum_{j \in N} x_{n+1j}^v = 0 \quad (13)$$

$$\sum_{v \in V} \sum_{i \in N} x_{i0}^v = 0 \quad (14)$$

$$x_{ij}^v \in \{0,1\}, \quad \forall i, j \in N, v \in V \quad (15)$$

Ciljna funkcija (2) minimizuje sumu razlika između trenutka popravke i pojave kvara, za svaki od zadataka koji se posmatra u planskom periodu. Ograničenja (3)-(6), (13) i (14) obezbeđuju funkcionalnu realizaciju tokova kroz graf. Ograničenja (3) obezbeđuju da je svaki od zadataka opslužen tačno jednom, dok ograničenja (4) služe da omoguće kretanje servisera između čvorova tako što uslovljavaju da broj ulazaka servisera u čvor bude jednak broju izlazaka iz čvora. Izrazi (5) daju mogućnost serviserima da napuste čvor 0 (depo), dok im izrazi (14) onemogućuju da se vrate u čvor, jer je to prvi čvor u ruti. Slično prethodnom paru ograničenja, ograničenja (13) i (6) onemogućuju servisere da napuste čvor  $n+1$ , ali im i omogućuju da mogu da se vrate u čvor.

Nepostojanje cikličnih ruta servisera, kao i vremenska usaglašenost kretanja servisera kroz graf je obezbeđena izrazima (7a), (8a), (9a) i (10a). Drugačije posmatrano, skup ovih ograničenja obezbeđuje da je zadovoljeno

$$D_j = \max \left( D_i + \frac{d_{ij}}{v_v} + s_j^v, e_j + s_j^v \right) \quad (16)$$

Kao što se vidi iz formulacije problema, ograničenja (7a) - (10a) nisu u linearnom obliku. Kako je za formulisanje problema kao MILP modela potrebno da sva ograničenja budu linearnog tipa, to je pomenuta ograničenja potrebno linearizovati. Konkretno, u ovom slučaju je to moguće učiniti primenom transformacije sa velikim brojem  $M$ , tj. takozvane big- $M$  transformacije, gde je  $M$  dovoljno veliki broj. Konkretno, na primeru skupa ograničenja (7a), uslov da će ograničenje važiti samo u slučaju da je promenljiva  $x_{ij}^v = 1$  se može napisati kao

$$D_i + t_{ij}^v - D_j \leq M(1 - x_{ij}^v), \quad \forall i, j \in P, v \in V \quad (7)$$

Na ovaj način, ukoliko je promenljiva  $x_{ij}^v$  jednaka 1 ograničenje postaje  $D_i + t_{ij}^v \leq D_j$ , dok u suprotnom postaje  $D_i + t_{ij}^v - D_j \leq M$ . Kako je rečeno da je  $M$  dovoljno veliki broj, to je jasno da je  $D_i + t_{ij}^v - D_j \leq M$  uvek zadovoljeno i da ograničenje nema nikakav uticaj na rešenje problema.

Na osnovu primera transformacije skupa ograničenja (7a) u skup (7), moguće je transformisati i ograničenja (8a), (9a) i (10a) u ograničenja (8), (9) i (10).

$$D_0^v + t_{0j}^v - D_j \leq M(1 - x_{0j}^v), \quad \forall j \in P, v \in V \quad (8)$$

$$D_i + t_{i, n+1}^v - D_{n+1}^v \leq M(1 - x_{i, n+1}^v), \quad \forall i \in P, v \in V \quad (9)$$

$$e_i + s_i^v - D_i \leq M \left( 1 - \sum_{j \in P \cup \{0\}} x_{ji}^v \right), \quad \forall i \in P, v \in V \quad (10)$$

Na ovaj način linearni oblik MILP modela za rešavanje hetTRPTW-a je definisan izrazima (2) – (15).

Primer optimalnog rešenja hetTRPTW-a predloženim MILP modelom predstavljen je na slici 5.1. U pitanju je jedna od 31 benčmark instance<sup>8</sup> koje su generisane u radu Potvin i Bengio (1996). Skup pomenutih benčmark instanci je zasnovan na Solomonovom RC2 skupu benčmark instanci za problem rutiranja sredstava sa

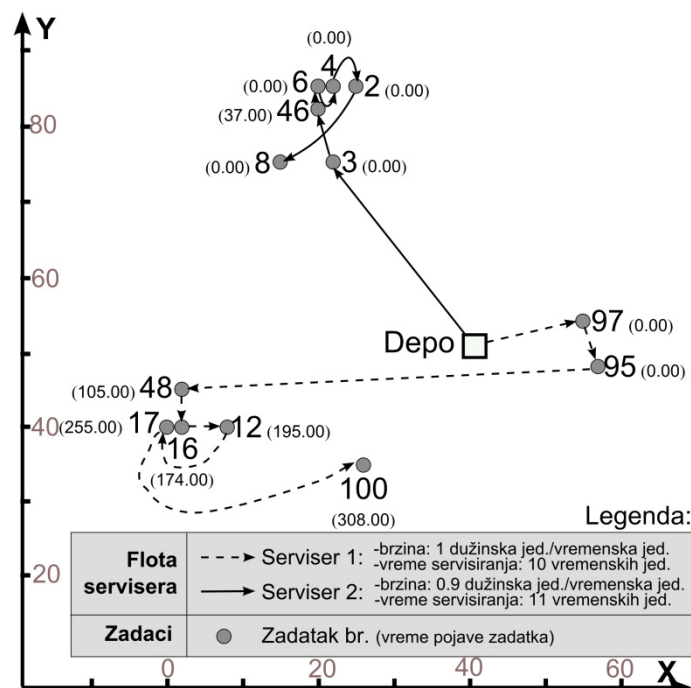
<sup>8</sup> Instance su dostupne za preuzimanje na web adresi <http://myweb.uiowa.edu/bthoa/TSPTWBenchmarkDataSets.htm>.

vremenskim prozorima – VRPTW (eng. Vehicle Routing Problem with Time Windows), a korišćen je i u radu Ohlmann i Thomas (2007) za formiranje skupa TSPTW benčmark instanci. Obzirom da pomenuti skup benčmark problema originalno nije namenjen hetTRPTW-u, prilagođavanje instanci posmatranom problemu je podrazumevalo ignorisanje zahteva klijenata u pogledu količine tereta koja se u čvorovima grafa utovaruje na (istovaruje sa) sredstava. Pored toga, ignorisana je karakteristika zahteva u pogledu vremenskog roka do kada je potrebno realizovati zahtev, odnosno desna granica vremenskog prozora je zanemarena. Treba naglasiti da je heterogenost flote servisera respektovana kroz različite brzine kretanja servisera po granama grafa, ali i kroz različita vremena servisiranja zadataka. Detalji o prilagođavanju benčmark instanci hetTRPTW-u su preciznije objašnjene u poglavlju 5.6. Respektovani i zanemareni podaci iz originalnog fajla rc\_202.2 problema su prikazani u tabeli 5.

**Tabela 5.2.** Postavka rc\_202.2 benčmark instance

R.B. Zadatka	Koordinate		Količina za manipulaciju (zanemareno)	Leva granica vremenskog prozora	Desna granica vremenskog prozora (zanemareno)	Trajanje zadržavanje u čvoru
	X	Y				
1	40	50	0	0	960	0
2	25	85	20	0	911	10
3	22	75	30	0	919	10
4	22	85	10	0	910	10
6	20	85	20	0	909	10
8	15	75	20	0	914	10
12	8	40	40	195	315	10
16	2	40	20	174	294	10
17	0	40	20	255	375	10
46	20	82	10	37	157	10
48	2	45	10	105	225	10
95	57	48	23	0	932	10
97	55	54	26	0	934	10
100	26	35	15	308	428	10

Uz pretpostavku da su dva raspoloživa servisera raspoloživi za opslugu od početka planskog perioda, tj. da su  $A^1 = 0$  i  $A^2 = 0$ , MILP matematički model primenjen na rc\_202.2 instancu daje optimalno rešenje koje podrazumeva da je redosled opsluge zadataka u slučaju servisera 1  $97 \rightarrow 95 \rightarrow 48 \rightarrow 16 \rightarrow 12 \rightarrow 17 \rightarrow 100$ , a u slučaju servisera 2  $3 \rightarrow 46 \rightarrow 6 \rightarrow 4 \rightarrow 2 \rightarrow 8$ . Ovakva alokacija zadataka i redosled opsluge rezultiraju ukupnim čekanjem klijenata na završetak opsluge od 574.171 vremensku jedinicu.



Slika 5.1. Optimalno rešenje hetTRPTW za  $rc_{202.2}$  instancu iz Potvin i Bengio (1996)

#### 5.4. Najmanja vrednost rešenja hetTRPTW-a

Najmanja vrednost, tj. donja granica – LB (eng. lower bound) je jedna od karakteristika problema koja svoju primenu nalazi u raznim metodama za rešavanje problema<sup>9</sup>. Najpoznatija je njena primena kao sastavnog dela B&B metode koja se koristi za rešavanje MILP modela, ali se može koristiti i u svim drugim metodama u kojim je potrebno ograničiti vrednost ciljne funkcije sa donje strane.

LB, kao što joj ime kaže, predstavlja procenu vrednosti ciljne funkcije ispod koje stvarna vrednost ciljne funkcije sigurno neće ići, tj. vrednost ciljne funkcije će sigurno biti veća, ili jednaka vrednosti LB.. Preciznije, pošto se LB dobija relaksacijom (zenemarivanjem) nekog od postojećih ograničenja modela<sup>10</sup> onda su rešenja u takvim uslovima bolja od rešenja problema u kome su sva ograničenja respektovana. Ukoliko

<sup>9</sup> U ovom slučaju se, obzirom na prirodu problema koji se razmatra podrazumeva troškovna priroda ciljne funkcije. U slučaju problema sa ciljnom funkcijom koja nije troškovne prirode ekvivalentna karakteristika je najveća moguća vrednost ciljne funkcije, odnosno gornja granica - UB (eng. upper bound). Sve rečeno u ovom poglavlju za LB se odnosi i na UB u slučajevima profitnih ciljnih funkcija, ali je u praksi češći slučaj da se profitna ciljna funkcija transformiše u troškovnu množenjem koeficijenata sa -1.

<sup>10</sup> U B&B se relaksira celobrojna, ili binarna, priroda promenljivih koje se nalaze u modelu, što rezultira rešenjima u kojima promenljive ne zadovoljavaju postavljeni uslov celobrojnosti, ili binarnosti.

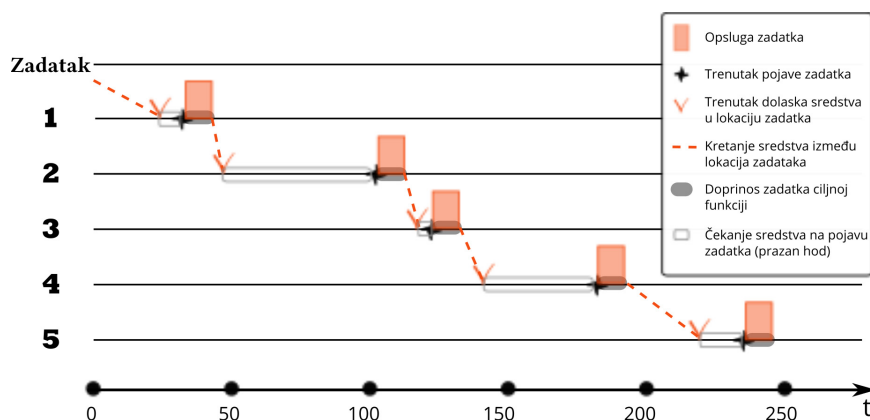


se nakon rešavanja relaksiranog problema, kroz proveru rešenja utvrdi da su zadovoljena sva ograničenja problema onda je LB ujedno i optimalno rešenje problema. Iz tog razloga se u istraživanjima teži onoj metodi za dobijanje LB koji rezultira vrednostima koje su bliske, ili u najboljem slučaju jednake, optimalnim vrednostima. Drugačije rečeno, što su veće vrednosti LB koje primenjena metoda daje, to je metoda bolja. Naravno, uz obavezno respektovanje uslova da je to garantovano najmanja vrednost ciljne funkcije.

Predlog metode za dobijanje LB u slučaju hetTRPTW-a je zasnovan na sledećim pretpostavkama:

1. svaki zadatak u planskom periodu se opslužuje sredstvom, tj. serviserom sa najvećom proizvodnošću
2. opsluga uvek započinje u trenutku pojave zadatka

Na ovaj način sredstva ne čekaju na početak opsluge i uvek su opslužena najbrže moguće. Vrednost LB je ujedno i vrednost optimalnog rešenja kada su vremenski periodi između pojava zadataka dovoljno veliki da serviser sa najvećom proizvodnošću može da opsluži trenutni zadatak i pređe rastojanje do lokacije sledećeg zadatka pre trenutka pojave sledećeg zadatka. Primer ovakve opsluge je predstavljen na slici 5.2. Druga situacija kada je vrednost LB ujedno i vrednost optimalnog rešenja hetTRPTW-a je kada se u sistemu nalazi onoliko serviseri sa maksimalnim intenzitetom opsluge koliko je zadataka koji se očekuju u sistemu, pri čemu su serviseri locirani na lokacijama pojave zadataka. Na taj način bi sa svakom pojavom zadataka odmah započinjala i njihova opsluga.



Slika 5.2. Idealna raspodela vremena pojave zadataka

Međutim, kako je malo verovatno da će se zadaci pojavljivati na način da je moguće da ih opsluži sredstvo sa najvećom proizvodnošću tako da ni jedno od njih ne čeka na pojavu sredstva, ili pak da je broj sredstva jednak broju zadataka<sup>11</sup>, to je jasno da će vrednost ciljne funkcije praktično uvek imati vrednost koja je veća od vrednosti LB.

Uzimajući u obzir sve prethodno rečeno jasno je da se vrednost LB za hetTRPTW dobija na način definisan jednačinom (17).

$$LB = \sum_{i \in P} \min_{v \in V} (s_i^v) \quad (17)$$

## 5.5. Heuristički algoritam za rešavanje hetTRPTW-a

Na osnovu uvida u raspoloživu literaturu jasno je uočljivo da hetTRPTW do sada nije bio u fokusu istraživača, te se ne može naći veliki broj heurističkih algoritama za njegovo rešavanje. Ipak, u radu Bjelić i ostali (2013b) je predstavljen heuristički algoritam zasnovan na VNS metaheuristici, koji će u ovom poglavlju biti detaljno predstavljen.

### 5.5.1. Pojam okoline trenutnog rešenja problema i njegove pretrage

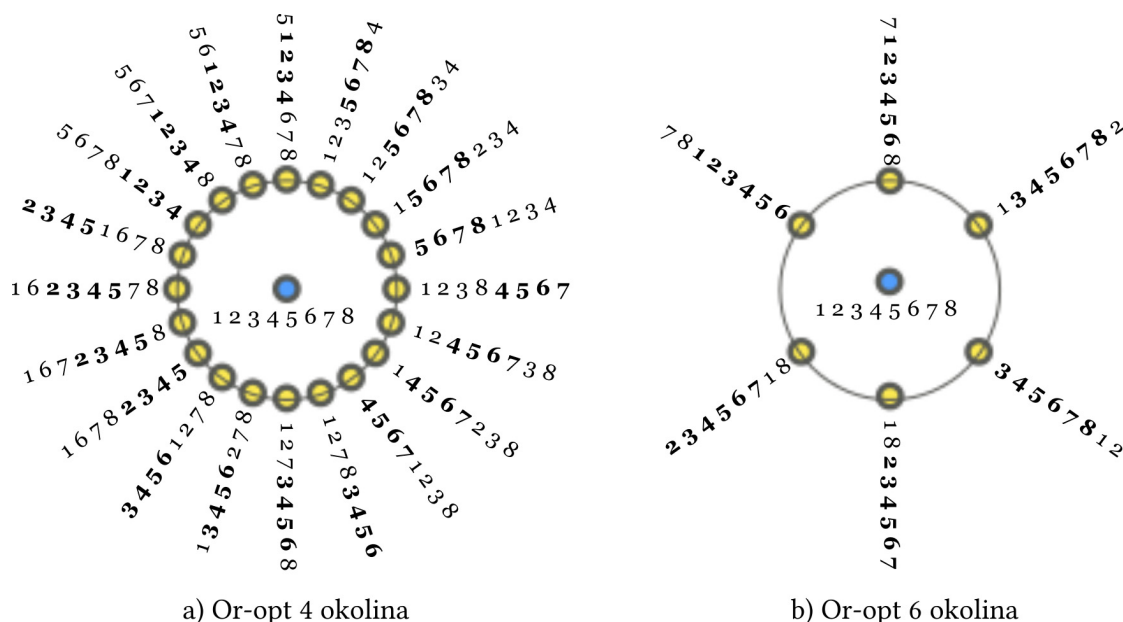
VNS je, kao što mu to i ime sugerise, algoritam zasnovan na rešavanju problema pretragom okolina postojećeg rešenja. Iz tog razloga, pre nego što budu predstavljeni osnovni principi funkcionisanja VNS metaheuristike, kao i detalji heurističkog algoritma za rešavanje hetTRPTW-a zasnovanog na tim principima, potrebno je objasniti šta je okolina rešenja i kako funkcioniše pretraga okoline.

U radu Pisinger i Ropke (2010) okoline rešenja  $x$ ,  $\mathcal{N}(x)$ , se definiše kao  $\mathcal{N}(x) \subseteq X$  gde je  $X$  skup mogućih rešenja problema, tako da važi i  $x \in X$ . Drugim rečima,  $\mathcal{N}$  je funkcija koja dovodi u vezu  $x$  sa skupom drugih rešenja. Ukoliko se sa  $f$  označi funkcija koja u vezu dovodi svako rešenje iz  $X$  sa vrednošću njegove ciljne funkcije, tj. ukoliko je  $f: X \rightarrow \mathbb{R}$ , tada se rešenje  $x$  smatra lokalnim optimumom u odnosu na okolinu  $\mathcal{N}$  ukoliko važi da je  $f(x) \leq f(x'), \forall x' \in \mathcal{N}(x)$ .

---

<sup>11</sup> U kom slučaju problem operativnog upravljanja sistemom ni ne postoji

Na koji način će se  $x$  dovesti u vezu sa drugim rešenjima, tj. koja okolina će biti upotrebljena zavisi od konkretnog problema koji se rešava i cilja koji se želi postići. Međutim, postoje i okoline koje se nalaze u velikom broju implementiranih algoritama baziranih na pretrazi okolina. Jedna od njih je i Or-opt okolina koju je moguće primeniti u potrazi za najboljim redosledom opsluge zadataka opsluženih resursom. Ideja Or-opt okoline je da se sve moguće grupe od  $k$  sukcesivnih zadataka u trenutnom rešenju  $x$  relociraju na sve moguće pozicije kako bi se proverilo da li neki od promenjenih redosleda opsluge postiže bolje rešenje problema. U zavisnosti od broja,  $k$ , sukcesivnih zadataka koji se premeštaju Or-opt okoline se označavaju sa Or-opt  $k$ . U tom smislu na slici 5.2 su prikazane celokupne Or-opt 4 i Or-opt 6 okoline za slučaj TSP problema od 8 zadataka sa trenutnim rešenjem  $x=1,2,3,4,5,6,7,8$ .



**Slika 5.3.** Primer kompletne Or-opt 4 i Or-opt 6 okoline

Na osnovu pojmova okolina i lokalni optimum u odnosu na okolinu moguće je definisati i pojam pretrage okolina. Svaki algoritam pretrage okolina na ulazu dobija  $x$  kao početno rešenje problema. Dalje se algoritmom računa najbolje rešenje,  $x'$ , u okolini,  $\mathcal{N}$ , rešenja  $x$  kao  $x' = \operatorname{argmin}_{x'' \in \mathcal{N}(x)} \{f(x'')\}$ . Ukoliko je  $f(x') \leq f(x)$  tada se rešenje problema  $x$  ažurira, tj.  $x = x'$  i pretraga se nastavlja pretragom okoline  $\mathcal{N}$  u odnosu na novo rešenje problema. Pretraga se završava kada  $x$  predstavlja lokalni optimum u odnosu na okolinu  $\mathcal{N}$ .

Prethodno opisani algoritam pretrage okolina se naziva algoritam najvećeg poboljšanja – BI (eng. Best Improvement), jer ispituje kvalitet  $x$  u odnosu na sva rešenja u okolini  $\mathcal{N}$ . Međutim, u slučaju izuzetno velikih okolina istraživači se često odlučuju za primenu strategije koja vrši ažuriranje trenutnog rešenja  $x$  kada god se u pretrazi nađe na bolje rešenje u okolini  $\mathcal{N}$ . Ovakva strategija pretrage okolina se realizuje algoritmom prvog poboljšanja – FI (eng. First Improvement) i za razliku od BI u njoj se  $x'$  ne računa kao  $\operatorname{argmin}_{x'' \in \mathcal{N}(x)} \{f(x'')\}$ , već se sukcesivno za svako rešenje iz  $\mathcal{N}(x)$  proverava uslov  $f(x') \leq f(x)$ . Naravno, ukoliko je uslov zadovoljen vrši se ažuriranje  $x = x'$  i restartuje pretraga okoline  $\mathcal{N}$  sa novim  $x$  kao centrom pretrage, a u suprotnom se prelazi na novo rešenje iz  $\mathcal{N}(x)$ . Uslov završetka FI algoritma je isti kao i u slučaju BI algoritma, tj. algoritam staje kada u  $\mathcal{N}(x)$  nema rešenja koja su bolja od  $x$ .

### 5.5.2. Osnovni koncepti realizacije VNS-a

VNS je metaheuristički algoritam koji je prvi put predstavljen u radu Mladenović i Hansen (1997) a baziran je na sistematskom korišćenju više okolina u potrazi za rešenjem problema. Ova metaheuristika svoju efikasnost bazira na tri elementarne činjenice (Hansen i Mladenović, 2003):

1. Rešenje koje se pokaže kao lokalni minimum<sup>12</sup> u odnosu na jednu okolinu ne mora biti lokalni minimum i u odnosu na neku drugu okolinu
2. Globalni minimum je lokalni minimum u odnosu na sve moguće okoline
3. Za mnoge probleme se pokazalo da su lokalni minimumi u odnosu na bilo koju okolinu vrlo bliski

Sve tri navedene činjenice opravdavaju uvođenje više okolina u pretragu za rešenjem problema. Dok je prva činjenica poprilično intuitivna po pitanju uvođenja više okolina trenutnog rešenja, drugu činjenicu, kako je to navedeno u Davidović (2006), ne treba shvatiti kao garanciju da je optimalno rešenje u odnosu na sve pretraživane okoline ujedno i optimalno rešenje problema, već da ukoliko rešenje nije optimalno u odnosu na jedno okruženje onda sigurno nije globalni optimum. Treća činjenica inicira

---

<sup>12</sup> Uz pretpostavlu da je ciljna funkcija troškovne prirode

potrebu za intenziviranjem pretrage rešenja u blizini postojećeg rešenja jer se očekuje da lokalni optimum u odnosu na neku okolinu sadrži informacije o globalnom optimumu. Međutim, kako se nikada ne zna koje informacije mogu biti zajedničke globalnom i lokalnom optimumu to je potrebno intenzivirati pretragu okruženja u odnosu na lokalni optimum. Treba napomenuti da treća činjenica nije teoretski dokazana već je rezultat brojnih empirijskih eksperimanata.

Tri činjenice na kojima je zasnovana reliazacija VNS metaheuristike mogu se primeniti na tri načina u odnosu na prisustvo stohastike u realizaciji algoritma: deterministički, stohastički i kombinovani. U tom smislu razlikuju se tri načina realizacije algoritma pretrage promeljivih okolina koji su predstavljeni u sledećim delovima ovog poglavlja.

#### 5.5.2.1. Metoda promenljivog spusta - VND

Metoda promenljivog spusta, tj. VND metoda podrazumeva determinističku promenu okolina, odnosno, u ovoj metodi se bira lokani optimum u odnosu na  $k_{max}$  odabranih okolina, pri čemu je redosled promene okolina determinističan. Konkretno, realizacija VND metode podrazumeva realizaciju koraka predstavljenu algoritmom 1.

##### **Algoritam 1** – Pseudokod za realizaciju metode promenljivog spusta

Inicijalizacija: Definirati skup okolina  $\mathcal{N}_k$ ,  $k = 1, \dots, k_{max}$ , koje će se koristiti u spustu; odrediti početno rešenje problema  $x \in X$

Ponavljaj sledeće korake dok sva rešenja u svim okolinama nisu ispitana:

(1)  $k \leftarrow 1$

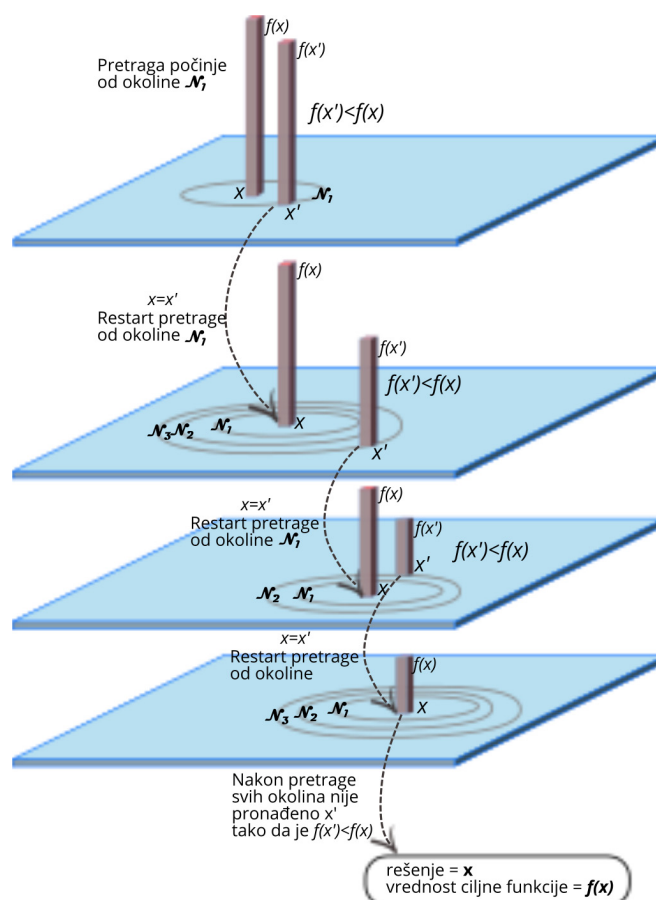
(2) Ponavljaj sledeće korake dok je  $k \leq k_{max}$

(a) **Pretraga okoline  $\mathcal{N}_k$** : iz okoline  $\mathcal{N}_k$  odaberi sledeće rešenje  $x'$ , ukoliko nema neispitanih rešenja  $k \leftarrow k + 1$  i vrati se na (2)

(b) **Pomeranje pretrage**: ukoliko je  $f(x') \geq f(x)$  preći na (a), u suprotnom  $x \leftarrow x'$ ,  $k \leftarrow 1$

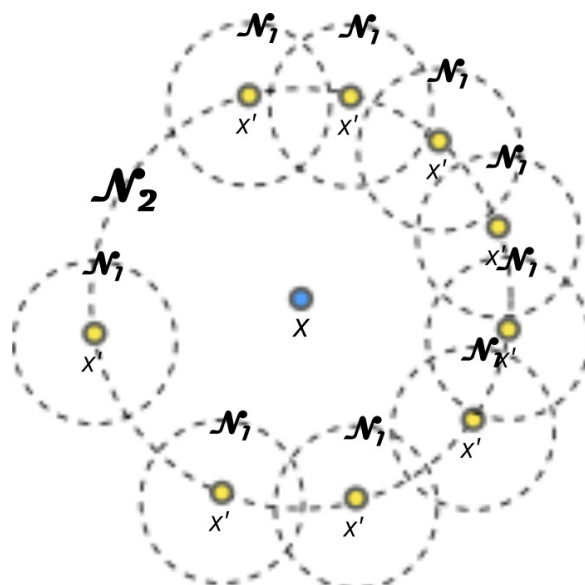
Na osnovu koraka algoritma 1, jasno je da algoritam startuje sa početnim rešenjem kao najboljim,  $x$ , i da pretraga okolina započinje od okoline  $\mathcal{N}_1$ . Rešenja iz okoline  $\mathcal{N}_1$ ,  $x'$ , se pretražuju kako bi se našlo ono koje zadovoljava uslov  $f(x') \leq f(x)$ . Ukoliko se pronade takvo rešenje ono postaje centar pretrage,  $x = x'$ , i pretraga se opet vraća na okolinu  $\mathcal{N}_1$ . Ukoliko se u okolini  $\mathcal{N}_1$  ne nađe rešenje koje zadovoljava uslov, pretraga se prebacue u okolinu  $\mathcal{N}_2$ . Ukoliko se u njoj nađe rešenje koje zadovoljava uslov, to

rešenje postaje centar pretrage i pretraga opet počinje od  $\mathcal{N}_1(x)$ , a ukoliko se ne nađe rešenje koje zadovoljava uslov, pretraga se seli u sledeću okolinu, tj.  $\mathcal{N}_3$ . Na ovaj način dobija se rešenje koje je lokalni optimum u odnosu na  $k_{max}$  primenjenih okolina. Slikoviti primer primene VND algoritma je prikazan na slici 5.3. Naravno, što je više okolina u odnosu na koje je rešenje lokalni optimum veća je šansa da se radi o globalnom optimumu. Mađutim, veliki broj primenjenih okolina, tj. veliko  $k_{max}$  ujedno znači i više vremena potrebnog za njihovu pretragu.



Slika 5.4. Primer primene VND algoritma

Treba napomenuti da je pored sekvencijalnog redosleda pretrage okolina, kako je to prethodno objašnjeno, okoline moguće pretraživati i po tzv. ugnježđenom redosledu. Ukoliko je  $k_{max} = 2$ , ugnježđenje okolina  $\mathcal{N}_1$  u okolinu  $\mathcal{N}_2$  se vrši kada se za svako rešenje  $x'$  iz  $\mathcal{N}_2$ , vrši pretraga okolina  $\mathcal{N}_1$  sa  $x'$  kao centrom pretrage, tj. kada se za svako  $x'$  iz  $\mathcal{N}_2(x)$  pretražuje  $\mathcal{N}_1(x')$ , kako je to predstavljeno na slici 5.5. Na taj način se intenzivira pretraga prostora rešenje po okolini  $\mathcal{N}_2(x)$ .



Slika 5.5. Primer ugnježenog načina pretrage okolina

#### 5.5.2.2. Redukovana metoda promenljivih okolina – RVNS

Redukovana metoda promenljivih okolina – RVND (eng. Reduced Variable Neighborhood Search) podrazumeva stohastički pristup realizaciji algoritma u smislu da se u pretrazi ne proveravaju sva rešenja iz okolina koje se pretražuju, već se iz svake okoline na slučajan način bira po jedno rešenje. Konkretno RVNS algoritam podrazumeva realizaciju koraka predstavljenih pseudokodom u algoritmu 2.

#### Algoritam 2 – Pseudokod za realizaciju redukovane metode promenljivih okolina

Inicijalizacija: Definisati skup okolina  $\mathcal{N}_k$ ,  $k = 1, \dots, k_{max}$ , koje će se koristiti u spustu; odrediti početno rešenje problema  $x \in X$ ; odrediti uslove zaustavljanja.

Ponavljaj sledeće korake dok se ne zadovolji neki od uslova zaustavljanja:

(1)  $k \leftarrow 1$

(2) Ponavljaj sledeće korake dok je  $k \leq k_{max}$

(a) **Razmrdavanje**: na slučajan način odabрати rešenje  $x'$  iz okoline  $\mathcal{N}_k(x)$

(b) **Pomeranje pretrage**: ukoliko je  $f(x') < f(x)$ ,  $x \leftarrow x'$ ,  $k \leftarrow 1$ ; u suprotnom  $k \leftarrow k + 1$

Na osnovu algoritma 2 jasno je da RVNS i dalje zadržava principe sistematske promene okolina, kao kod VND algoritma, ali sa izborom samo po jednog rešenja iz ispitivane okoline. Ovakav način pretrage okolina je pogodan u slučaju okolina čija

kompletna pretraga zahteva veliku količinu vremena, ili problema velikih dimenzija. Međutim, odsustvo intenzivne pretrage u okolini rešenja  $x$  uzrokuje da ova metoda bez neke metode lokalne pretrage ne daje kvalitetne rezultate. Sa druge strane, brzina realizacije RVNS-a je uzrokovala da se ona često koristi kao metoda za generisanje početnog rešenja perturbacionih metoda. Kao kriterijum za zaustavljanje algoritma kod RVNS-a se najčešće koristi broj iteracija bez unapređenja rešenja, mada je, u zavisnosti od postavljenog cilja, moguće koristiti i bilo koji drugi kriterijum.

### 5.5.2.3. Bazični metod promenljivih okolina - BVNS

Bazični metod promenljivih okolina – BVNS (eng. Basic Variable Neighborhood VNS) je metod koji se obično poistovećuje sa VNS metodom. U tom smislu, u nastavku rada će VNS označavati BVNS metodu. Ovaj metod predstavlja kombinaciju determinističkog i stohastičkog metoda, a redosled koraka je dat pseudokodom u algoritmu 3.

#### **Algoritam 3** – Pseudokod za realizaciju bazičnog metoda promenljivih okolina

*Inicijalizacija:* Definisati skup okolina  $\mathcal{N}_k$ ,  $k = 1, \dots, k_{max}$ , koje će se koristiti u spustu; odrediti početno rešenje problema  $x \in X$ ; odrediti uslove zaustavljanja.

*Ponavljaj* sledeće korake dok se ne zadovolji neki od uslova zaustavljanja:

(1)  $k \leftarrow 1$

(2) *Ponavljaj* sledeće korake dok je  $k \leq k_{max}$

(a) **Razmrđavanje:** na slučajan način odaberi rešenje  $x'$  iz okoline  $\mathcal{N}_k(x)$

(b) **Lokalna pretraga:** Primeniti neki algoritam za lokalnu pretragu sa  $x'$  kao početnim rešenjem; dobijeni lokalni optimum označiti kao  $x''$ .

(b) **Pomeranje pretrage:** ukoliko je  $f(x'') < f(x)$ ,  
 $x \leftarrow x''$ ,  $k \leftarrow 1$ ; u suprotnom  $k \leftarrow k + 1$

Kao i kod RVNS metode, ni u VNS metodi se ne pretražuju cele okoline rešenja  $x$ , već se intenzivira pretraga, nekom metodom lokalne pretrage, oko slučajno odabranih rešenja  $x'$ . Najčešća metoda lokalne pretrage koja se koristi za intenziviranje pretrage oko  $x'$  je VND metoda, pri čemu okruženja koja se koriste u razmrđavanju i lokalnom pretraživanju ne moraju biti ista, kako po prirodi tako ni po broju.



Na kraju treba napomenuti da se VNS metoda pokazala kao vrlo kompetitivna u oblasti problema kombinatorne optimizacije. U tom smislu našla je primenu u rešavanju velikog broja praktičnih problema, među kojima su i problemi iz oblasti logistike. Za rad sa problemima velikih dimenzija VNS je dobila i odgovarajuće modifikacije, kako bi se nosila sa okolinama velikih dimenzija. Međutim, kako zalaženje u detalje ovih proširenja prevazilazi potrebe ovog istraživanja, zainteresovani čitalac se o njima može informisati u Hansen i ostali (2010) i Hansen i Mladenović (2001,2003).

Detalji primene VNS metode za rešavanje hetTRPTW-a su predstavljeni u nastavku ovog poglavlja.

### 5.5.3. VNS algoritam za rešavanje hetTRPTW-a

VNS algoritam za rešavanje hetTRPTW-a, predstavljen u Bjelić i ostali (2013b) je zasnovan na VNS algoritmu predstavljenom algoritmom 3, pri čemu je za intenziviranje pretrage u okolini slučajno odabranog rešenja  $x'$  korišćen VND pristup predstavljen algoritmom 1. Ostale karakteristike predloženog heurističkog algoritma su date u nastavku.

#### 5.5.3.1. Kodiranje rešenja

Efikasnost načina kodiranja rešenja može značajno doprineti efikasnosti realizacije algoritama, ali je, sa druge strane u znatnoj meri određena načinom rešavanja problema. Kako je VNS metoda bazirana na funkcionalnom povezivanju rešenja preko promena u strukturi rešenja, koja rezultiraju formiranjem okolina, to je radi efikasnije realizacije promena na jednom rešenju iskorišćena karakteristika hetTRPTW-a da je to problem koji podrazumeva postojanje skupa serviseri,  $V$ , koji opslužuju skup kvarova,  $P$ . Kako je broj serviseri jednak  $m = |V|$ , a broj kvarova  $n = |P|$  i kako rešenje treba da definiše alokaciju kvarova po serviserima, ali i redosled opsluge zadataka dodeljnih svakom serviseru, to je rešenje hetTRPTW-a predstavljeno kao  $m$  uređenih skupova rednih brojeva zadataka,  $H^v$ . Indeks  $v$  u oznaci  $H^v$  označava redni broj serviseri na koji se uređeni skup rednih brojeva zadataka odnosi, pri čemu važi da je  $\sum_{v \in V} |H^v| = n$ , gde je  $|H^v|$  kardinalnost skupa  $H^v$ . Ovakav način kodiranja rešenja omogućava primenu koraka koji su do sada korišćeni za rešavanje srodnih problema i koji

omogućavaju formiranje skupa okruženja koja će biti korišćena u postupcima razmrdavanja i lokalne pretrage.

Kao i u slučaju optimalnog rešavanja hetTRPTW-a, predstavljanje načina kodiranja rešenja će biti predstavljeno pomoću instance rc\_202.2 iz Potvin i Bengio (1996), koja je data u poglavlju 5.3. Konkretno, kodiranje optimalnog rešenja problema sa slike 5.1 bi bilo  $H^1 = \{97, 95, 48, 16, 12, 17, 100\}$  i  $H^2 = \{3, 46, 6, 4, 2, 8\}$  jer su svih 13 kvarova opsluženi sa dva servisera.

### 5.5.3.2. Proračun vrednosti ciljne funkcije

U skladu sa karakteristikom hetTRPTW-a da se zadaci pojavljuju tokom planskog perioda, ali i predstavljenim načinom kodiranja rešenja, vrednost ciljne funkcije rešenja  $x$ ,  $f(x)$ , se računa prema izrazu (18)

$$f(x) = \sum_{v=1}^m \sum_{i=1}^{|H^v|} D_{h_i^v} - e_{h_i^v} \quad (18)$$

pri čemu je značenje promenljivih  $D$  i  $e$  isto kao u poglavlju 5.3, a  $h_i^v$  označava redni broj zadatka koji se u skupu zadataka  $H^v$  nalazi na  $i$ -toj poziciji. Na primer, za skup  $H^v$  optimalnog rešenja instance rc\_202.2 iz Potvin i Bengio (1996) vrednosti indeksa  $h_i^v$  su:  $h_1^2 = 3$ ,  $h_2^2 = 46$ ,  $h_3^2 = 6$ ,  $h_4^2 = 4$ ,  $h_5^2 = 2$  i  $h_6^2 = 8$ .

Vremena završetka opsluge, odnosno vremena popravke kvarova,  $D_{h_i^v}$ , se računa prema izrazu (19):

$$D_{h_i^v} = \begin{cases} \max\left(A^v + \frac{d_{0h_i^v}}{v_v}, e_{h_i^v}\right) + s_{h_i^v}^v, & i = 1 \\ \max\left(D_{h_{i-1}^v} + \frac{d_{h_{i-1}^v h_i^v}}{v_v}, e_{h_i^v}\right) + s_{h_i^v}^v, & i > 1 \end{cases} \quad (19)$$

Ovo znači da otpočinjanje popravke kvara kreće po realizaciji poslednjeg, u smislu vremenskih koordinata, od dva kritična događaja: pojave kvara, ili pristizanja servisera u lokaciju. Uzimanje realizacije poslednjeg od ova dva događaja je rezultat činjenice da je potrebno da su se oba događaja realizovala kako bi krenula popravka. Name, ukoliko se pojavi kvar, popravka će početi tek kada se pojavi serviser, a ukoliko je

serviser u lokaciji u kojoj se očekuje potreba za uslugom, ona neće otpočeti dok se ne pojavi kvar.

### 5.5.3.3. Početno rešenje problema

Kao početno rešenje VNS algoritma se koristi rešenje dobro poznate konstruktivne heuristike ubacivanja. Ovu heuristiku je van der Meer (2000) predložio kao metodu za rešavanje homTRPTW problema velikih dimenzija. Osnovna ideja ove heuristike je da se u svakoj iteraciji, sukcesivno za po jedan zadatak iz  $P$  traži pozicija najboljeg ubacivanja, u smislu dobijanja ciljne funkcije najmanje vrednosti.

Pseudokod algoritma ubacivanja korišćen za generisanje početnog rešenja je prikazan algoritmom 4.

---

**Algoritam 4** – Pseudokod za realizaciju algoritma ubacivanja

---

Inicijalizacija: Inicirati konačno rešenje iniciranjem  $H^v$  za svako  $v \in V$ ; elemente skupa  $P$  iskopirati u skup  $W$ .

Ponavljaj sledeće korake dok je  $W \neq \{\emptyset\}$ :

(1)selektuj zadatak  $w, w \in W$

Ponavljaj sledeće korake za svaku moguću poziciju ubacivanja zadatka  $w$  u konačno rešenje

(a)Ubaci na poziciju ubacivanja zadatak  $w$  i izračunaj vrednost ciljne funkcije privremenog rešenja

(b)Zapamti vrednost ciljne funkcije i pozicije ubacivanja za zadata  $w$

(c) Izbaci zadatak  $w$  sa trenutne pozicije ubacivanja

(2)Zadatak  $w$  ubaci na onu poziciju u konačno rešenje, za koju privremeno rešenje ima najmanju vrednost ciljne funkcije; izbriši vrednosti ciljnih funkcija i pozicija ubacivanja za zadatak  $w$ .

(3) $W \leftarrow W \setminus \{w\}$ .

---

### 5.5.3.4. Algoritam promenljivog spusta – VND

VND algoritam za pronalaženje lokalnog optimuma u okolini trenutno posmatranog rešenja je baziran na FI strategiji pretrage okolina, kako je to predstavljeno algoritmom 1. U cilju dostizanja što većeg skupa rešenja kroz primenjene okoline  $\mathcal{N}$ , čime se povećava kvalitet dostignutog optimuma VND algoritma, potrebno je inicirati promene koje će rezultirati zadržavanjem postojeće strukture rešenja, u smislu broja zadataka dodeljenih serviserima, ali i one koje će dovesti do promene polazne strukture rešenja. U tom cilju okoline koje su korišćene u VND-u omogućavaju:

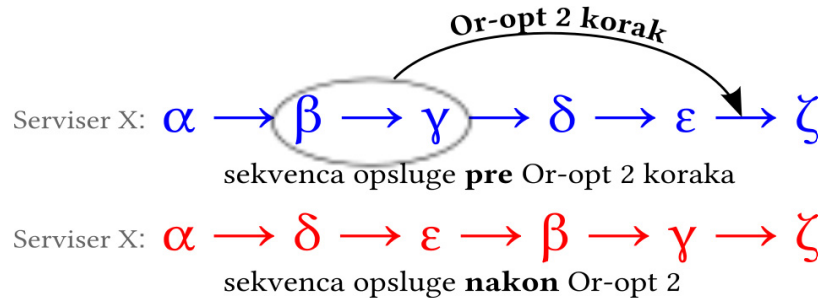
1. promenu redosleda opsluge zadataka dodeljenih serviserima, pri čemu se teži poboljšanju rešenja posmatranjem svakog servisera ponaosob;
2. razmenu zadataka dodeljenih različitim serviserima, čime se proverava kvalitet alokacije zadataka serviserima, uz zadržavanje postojeće strukture rešenja u smislu broja zadataka dodeljenih serviserima;
3. relokaciju zadataka između servisera, čime se menja struktura prvobitnog rešenja, jer se zadatak sa prvobitnog servisera dodeljuje nekom od ostalih servisera, tj. broj zadataka dodeljenih serviserima se menja. Naravno, ovakva promena podrazumeva da se pored promene strukture rešenja menja i redosled opsluge zadataka na serviseru kome je zadatak dodeljen, ali i na prvobitnom serviseru .

Implementacijom ovakvih okolina u odnosu na trenutno rešenje dobijaju se rešenja koja nisu preterano udaljena jedna od drugih u prostoru rešenja, što, u skladu sa trećom činjenicom na kojoj je baziran VNS, daje dobre šanse za nalaženje unapređenog rešenja problema.

U skladu sa prethodno rečenim, VND se sastoji od tri okoline:

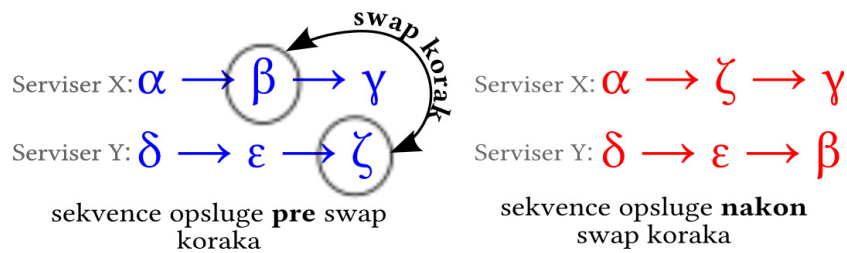
- $\mathcal{N}_{Or-opt}$  okolina se sastoji od rešenja koja se dobijaju primenom Or-opt promene na početnom rešenju. Or-opt promenom, ili korakom, kako se drugačije naziva promena na rešenju koje rezultira novim rešenjem, se menja redosled opsluge dodeljenih zadataka na svakom serviseru nezavisno. U cilju posmatranja većeg broja rešenja, posmatrana su Or-opt okruženja u kojima je redosled opsluge menjan za sukcesivnih  $1, 2, \dots, \min(3, |H^v| - 1)$  zadataka iz početnog redosleda. Drugačije rečeno, posmatrano  $\mathcal{N}_{Or-opt}$  okruženje se sastoji od rešenja dobijenih Or-opt  $k$  koracima, gde je  $k = 1, 2, \dots, \min(3, |H^v| - 1)$ .

Primer Or-opt 2 koraka je predstavljen na slici 5.6. Na slici se vidi da su dva sukcesivna zadatka,  $\beta$  i  $\gamma$ , koja se, pre primene Or-opt 2 koraka, opslužuju nakon zadataka  $\alpha$ , prebačena između zadataka  $\varepsilon$  i  $\zeta$ , nakon primene Or-opt 2 koraka.



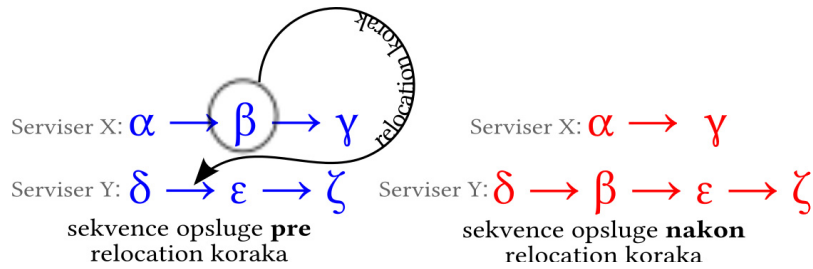
Slika 5.6. Or-opt 2 korak

- Okolina  $\mathcal{N}_{\text{swap}}$  se sastoji od rešenja dobijenih primenom swap koraka. Osnovna ideja ovog koraka je da se, zadržavajući postojeću strukturu rešenja, novo rešenje generiše zamenom dva zadatka sa različitih serviserera. Okolina  $\mathcal{N}_{\text{swap}}$  se sastoji od svih rešenja dobijenih zamenama svih parova rešenja sa različitih zadataka. Primer swap koraka u kome zadaci  $\beta$  i  $\zeta$  menjaju servisere i pozicije opsluge je prikazan na slici 5.7.



Slika 5.7. Swap korak

- Okolina  $\mathcal{N}_{\text{relocate}}$  je sačinjena od rešenja koja se dobijaju relocation korakom. Ideja ovog koraka je da se promenom strukture rešenja, u smislu broja zadataka dodeljenih serviserima, pronade najbolje moguće rešenje. Pored promene broja zadataka alociranih serviserima menja se i redosled opsluge na serviserima uključenim u korak, jer je kod jednog serviserera zadatak uklonjen, a kod drugog je dodat. Okolinu  $\mathcal{N}_{\text{relocate}}$  čine sva rešenja dobijena premeštanjem svakog od zadataka u rešenju na svaku poziciju ubacivanja na ostalim serviserima. Primer ovog koraka je dat na slici 5.8 gde se zadatak  $\beta$ , opslužen serviserom X pre premeštanja, prebacuje serviseru Y, na opslugu između zadataka  $\delta$  i  $\epsilon$ .



Slika 5.8. Relocation korak

Redosled pretrage okolina rešenja u VND-u je veoma bitna karakteristika predloženih algoritama. Opšte pravilo koje se primenjuje u metodama koje su zasnovane na pretragama okolina rešenja je da se pretraga uvek započinje sa okolinama manje kardinalnosti. Uprkos tome što je veća šansa da se poboljšano rešenje nađe u skupu rešenja veće kardinalnosti, njihova pretraga je vremenski zahtevnija, tako da je praksa pokazala da se bolji odnos kvalitet rešenja/vreme rešavanja dobija promenom okruženje u skladu sa porastom njihove kardinalnosti.

Na osnovu karakteristika okolina koja se primenjuju u VND algoritmu mogu se izračunati njihove kardinalnosti. Kardinalnost skupova rešenja koji čine okoline  $\mathcal{N}_{Or-opt}$ ,  $\mathcal{N}_{swap}$  i  $\mathcal{N}_{relocate}$  je data jednačinama (20), (21) i (22), respektivno.

$$|\mathcal{N}_{Or-opt}| = 3 \sum_{v=1}^m |H^v|^2 - 9n + 8m \quad (20)$$

$$|\mathcal{N}_{swap}| = \frac{n^2 - \sum_{v=1}^m |H^v|}{2} \quad (21)$$

$$|\mathcal{N}_{relocate}| = n^2 + n(m-1) - \sum_{v=1}^m |H^v|^2 \quad (22)$$

Na osnovu izraza (20)-(21) jasno je da svi skupovi imaju kardinalnost koja je kvadratna funkcija broja zadataka, jer je  $n = |P| = \sum_{v=1}^m |H^v|$ . Da bi se definisao redosled pretrage po okolinama potrebno je utvrditi odnos koeficijenata uz kvadratne elemente polinoma, pri čemu je najbitnije razmotriti slučaj Or-opt okoline jer njegova kardinalnost najviše zavisi od zbira kvadratnih elemenata.

Kako je  $\sum_{v=1}^m |H^v|^2 = n^2$  u slučaju kada u sistemu postoji samo jedan serviser, tj. kada je  $m = 1$ , u kom slučaju se ne radi o hetTRPTW-u, ili kada su svi zadaci dodeljeni samo jednom sredstvu, što je malo verovatno da se desi u hetTRPTW-u, to je jasno da će u najvećem broju slučajeva važiti da je  $n^2 > \sum_{v=1}^m |H^v|^2$ . Ovo je posledica činjenice

da je zbir kvadrata elemenata koji u sumi daju neki broj,  $\varrho$ , uvek manji od kvadrata tog broja,  $\varrho$ . Na ovaj način je sigurno, sem u pomenutom slučaju kada su svi zadaci dodeljeni samo jednom sredstvu, da je veličina  $\mathcal{N}_{Or-opt}$  okoline manja od okolina  $\mathcal{N}_{swap}$  i  $\mathcal{N}_{relocate}$ , dok je odnos ove dve okoline lako utvrditi, obzirom da kvadratni element u polinomu  $\mathcal{N}_{swap}$  okoline ima koeficijent  $\frac{1}{2}$ . Na osnovu svega rečenog jasno je da je u očekivanim rešenjima, u kojima su zadaci raspodeljeni na više servisera, odnos veličina okruženja  $|\mathcal{N}_{relocate}| > |\mathcal{N}_{swap}| > |\mathcal{N}_{Or-opt}|$ , odnosno da je redosled okolina takav da je  $\mathcal{N}_1 = \mathcal{N}_{Or-opt}$ ,  $\mathcal{N}_2 = \mathcal{N}_{swap}$  i  $\mathcal{N}_3 = \mathcal{N}_{relocate}$ .

### 5.5.3.5. Redukcija okolina u VND-u

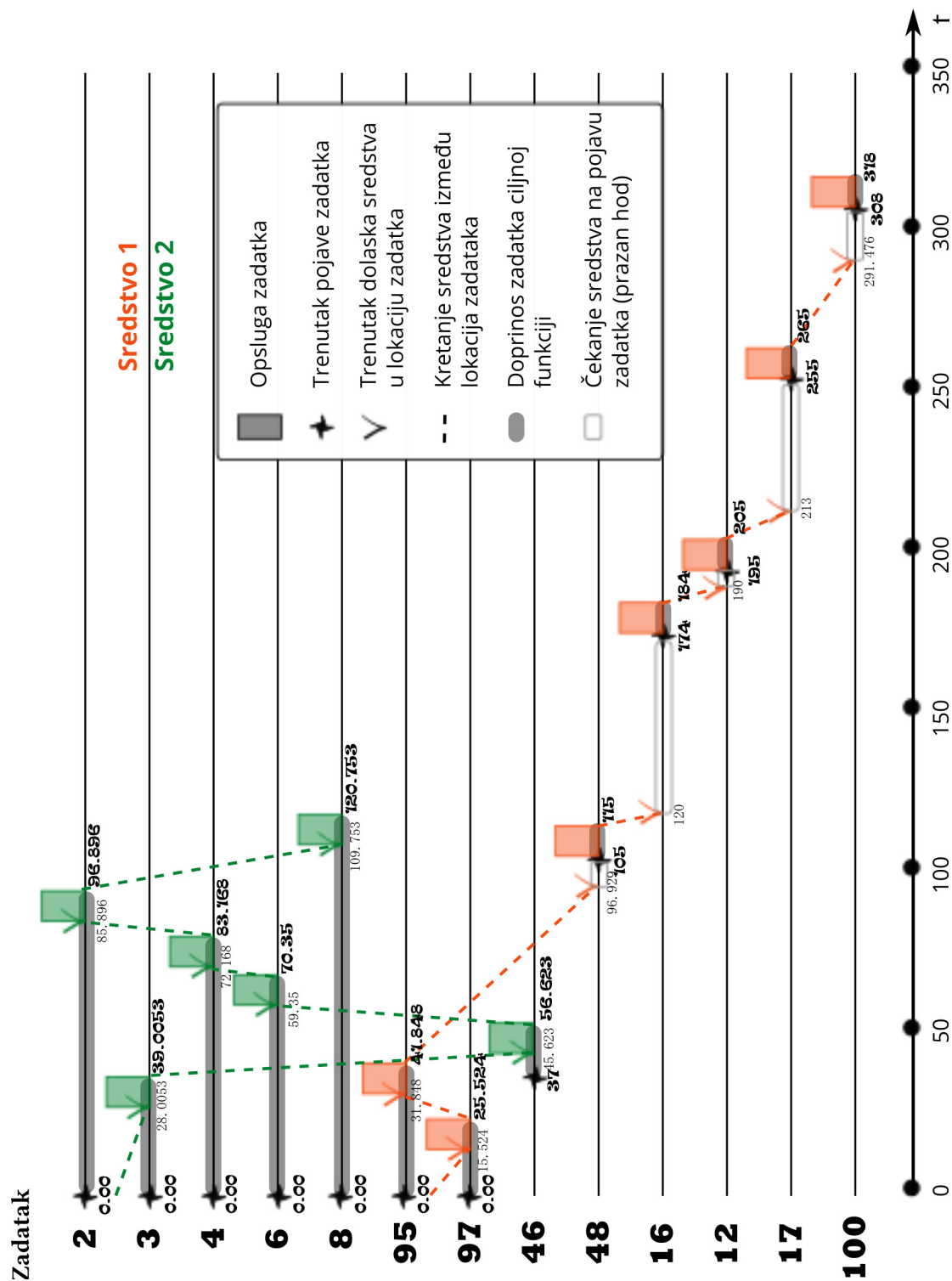
Povećanje veličine problema koji se rešava, bilo sa strane broja zadataka koji se opslužuju ( $n$ ), ili broja sredstava koja su prisutna u sistemu ( $m$ ), u znatnoj meri vodi redukciji vremenske efikasnosti VND algoritma. Kao prikaz ove tvrdnje mogu poslužiti rezultati iz tabela P1-5 do P1-8, iz priloga 1, gde je jasno uočljiv trend rasta vremena rešavanja sa rastom veličine problema. Obzirom, na to da na primenu VND algoritma otpada najveći deo vremena realizacije VNS algoritma, ali i to da VNS ne garantuje dobijanje optimalnog rešenja, jedna od mogućnosti da se smanji vreme rešavanja problema velikih dimenzija je da se ne ispituju sva rešenja u primenjenim okolinama. Naime, na osnovu karakteristika sistema neka rešenja se, nakon brze provere definisanih uslova, mogu ukloniti iz okolina, čime se redukuje veličina okoline te se njegova pretraga ubrzava. Karakteristike funkcionisanja hetTRPTW sistema na osnovu kojih su generisane redukcije okruženja, kao i same redukcije, su date u nastavku ovog poglavlja.

Generalno gledano, kumulativna ciljna funkcija problema znači da se vreme čekanja prvoopsluženog zadataka na kraj opsluge prenosi i na sve preostale zadatke u sekvenci opsluge. U sistemima sa dinamičkom pojavom zadataka, ova pojava se može amortizovati kada se u opsluzi naiđe na zadatak u kome serviser mora da čeka na pojavu zadatka, ali to ne mora uvek da bude slučaj. Na primer, za zadatke 97 i 95 sa slike 5.9, na kojoj je prikazan vremenski aspekt optimalnog rešenja rc 202.2 instance iz Potvin i Bengio (1996), vidi se da je vreme opsluge zadatka 95 zavisno od opsluge zadatka 97, tj. vreme kretanja servisera do lokacije zadatka 97 i vreme njegove opsluge

se dodaju ciljnoj funkciji i u slučaju zadatka 95, jer je trenutak otpočinjanja opsluge zadatka 95 nakon zadatka 97, a zadatak 95 je sve vreme prisutan u sistemu, gde čeka na opslugu. Međutim, vreme opsluge zadatka 97, a u konkretnom slučaju i zadatka 95, nema nikakvog uticaja na doprinos ciljnoj funkciji sledećeg zadatka u sekvenci, tj. zadataka 48, jer njegov doprinos počinje da se računa tek od 105. vremenske jedinice kada počinje i njegova opsluga. Drugim rečima, pošto serviser čeka na pojavu zadatka dolazi do potpune amortizacije vremena čekanja prethodnih zadataka.

Prebacivanje zadataka 48 na početak sekvence opsluge bi ciljnu funkciju povećao za istu vrednost, kao i kada se opslužuje kao treći u nizu. Međutim, doprinos svih ostalih zadataka bi bio veći jer su oni već bili u sistemu dok je serviser čekaao na pojavu zadatka 48. U tom smislu, može se zaključiti da je, za razliku od TRP-a u kome je akcenat pri rešavanju problema na pronalaženju najbolje kombinacije vremena za savladavanje rastojanja između lokacija i vremena opsluge zadataka, jer su svi zadaci uvek na raspolaganju za opslugu, u slučaju hetTRPTW-a rešavanje dodatno otežano i respektovanjem dinamičke pojave zadataka. Tako se u slučaju servisera 2 u instanci rc202.2 opsluga zadataka nakon zadatka 3 (čiji je kraj opsluge u 39.0053 v.j.), kada su svi zadaci prisutni u sistemu, svodi na pronalaženje najkraće rute servisera kroz model. Ovde treba naglasiti da se u ovom slučaju svodi na najkraću rutu kroz lokacije zadataka 46, 2, 4, 6 i 8, jer je pretpostavljeno da su zadaci isti te su i trajanja njihovih opsluga ista (11 vremenskih jedinica). U suprotnom i ovaj aspekt bi morao biti uključen u razmatranje i zadatak bi se sveo na TRP. U slučaju servisera 1 jasno je da, nakon zadataka 97 i 95, dinamički aspekt pojave zadataka ima primarnu ulogu u definisanju rešenja jer nakon zadatka 48 serviser ne traži najkraću rutu kroz lokacije zadataka (koja bi intuitivno sa slike 5.9 bila  $17 \rightarrow 16 \rightarrow 12 \rightarrow 100$ ) već lokacije opslužuje po vremenu pojavljivanja, kako bi isključio čekanje zadataka na pojavu servisera.





Slika 5.9. Vremenski prikaz rešenja instance rc 202.2

Okruženje  $\mathcal{N}_{Or-opt}$  je redukovano na osnovu opažanja da u kvalitetnim rešenjima, kakvo je rešenje sa slike 5.1, svaka zamena dva susedna zadatka na svakom od servisera vodi ka kasnijem završetku prvoopsluženog zadatka, što za posledicu može

imati veću akumulaciju vremena čekanja na kraj opsluge kod svih sledećih zadataka u sekvenci. Na primer, zamena zadataka 97 i 95 u slučaju serviser 1 sa slike 5.9 bi dovela do kasnijeg završetka opsluge zadatka 95 (kao prvoopsluženog u novoj sekvenci), jer je rastojanje od depoa do lokacije zadataka 95 veće nego do lokacije zadataka 97 (lako uočljivo na slici 5.1), a kako su oba prisutna u sistemu taj produžetak opsluge prvoopsluženog zadatka bi se akumulirao i u doprinosu ciljnoj funkciji zadatka 97, jer je kasnije počeo sa opslugom. Slična je situacija i sa zadacima opsluženim serviserom 2, jer zamena redosleda opsluge zadataka 3 i 46 znači da će opsluga prvoopsluženog zadatka umesto u 28.0053 v.j. početi tek sa pojavom zadatka 46, tj. u 37. v.j. te će se sav doprinos ciljnoj funkciji zadataka 46 akumulirati i u doprinosu zadatka 3, koji je sve vreme prisutan u sistemu.

U tom smislu, redukcija  $\mathcal{N}_{Or-opt}$  okoline je bazirana na ideji da ukoliko zamena pozicija opsluge dva zadatka rezultira sekvencom opsluge u kojoj je prvoopsluženi zadatak (u daljem tekstu novi prvoopsluženi zadatak) opslužen kasnije nego prvoopsluženi zadatak u početnoj sekvenci (u daljem tekstu početni prvoopsluženi zadatak), vrlo je verovatno da bi početni prvoopsluženi zadatak na nekoj kasnijoj poziciji u sekvenci opsluge imao mnogo više akumulacije vremena opsluge njemu prethodnih zadataka, imajući u vidu dinamičku pojavu zadataka u heTTRPTW-u. Iz tog razloga iz Or-opt k okoline mogu se isključiti sva rešenja u kojima početni prvoopsluženi zadatak nije na početnoj lokaciji u sekvenci opsluge. Sa druge strane, ukoliko se desi da zamena pozicija opsluge rezultuje smanjenjem vremena završetka novog prvoopsluženog zadatka, onda je jasno da početni prvoopsluženi zadatak nije na dobrom mestu u sekvenci opsluge. Definisane bolje pozicije je zadatak Or-opt k okoline te je u  $\mathcal{N}_{Or-opt}$  potrebno uključiti sve zadatke od početnog prvoopsluženog do kraja sekvence opsluge.

Preciznije, redukcija Or-opt k okoline podrazumeva da se, počevši od krajnje leve strane sekvence opsluge, redosled opsluge dva susedna zadatka menja i da se poredi vreme završetka prvoopsluženih zadataka pre i nakon zamene. Ukoliko je vreme završetka opsluge novog prvoopsluženog zadatka manje od vremena završetka opsluge početnog prvopsluženog zadataka u Or-opt k okolinu se svrstavaju svi zadaci od početnog prvoopsluženog do kraja sekvence opsluge i kreće se sa formiranjem rešenja u okolini. Ukoliko je ipak vreme završetka opsluge novog prvoopsluženog

zadatka veće od vremena završetka početnog prvoopsluženog zadatka, tada se početni prvoopsluženi isključuje iz sekvence nad kojom će se sprovesti Or-opt k koraci u formiranju okoline, a postupak zamene susednih zadataka se pomera za jednu poziciju u desno.

U slučaju pretrage  $\mathcal{N}_{Or-opt}$  okoline rešenja u kome su svi zadaci na serviserima dobro locirani u sekvenci opsluge, tj. kada se zamenom bilo koja dva susedna zadatka ne postiže brže vreme završetka novog prvoopsluženog zadatka, postiže se najveća redukcija okruženja, jer pretraga okoline podrazumeva  $|H^v|-1$  zamena pozicija opsluge, na svakom serviseru  $v$ , i poređenje vremena završetka početnog i novog prvoopsluženog zadatka. Sa druge strane, redukcija  $\mathcal{N}_{Or-opt}$  okoline ne postoji u slučajevima u kojima se na prvim mestima u sekvencama opsluge nalaze „loše pozicionirani“ zadaci. Međutim, u takvim rešenjima je unapređenje rešenja izvesno pretragom  $\mathcal{N}_{Or-opt}$  okoline.

Redukcija  $\mathcal{N}_{swap}$  okoline je bazirana na pretpostavci da zamena zadataka na velikom rastojanju po pitanju vremena pojave ima malu verovatnoću za generisanje kvalitetnih rešenja. Razlog za to je da se zadaci koji se javljaju kasnije u planskom periodu smeštaju na lokaciju u sekvenci opsluge zadatka sa početka opsluge, i obrnuto. Ovo rezultira, sa jedne strane, čekanjem servisera na pojavu zadatka sa kraja opsluge, a samim tim i većim doprinosom ciljnoj funkciji svih zadataka koji se opslužuju nakon zadatka sa kraja opsluge, a sa druge strane velikim doprinosom ciljnoj funkciji zadataka sa početka planskog perioda, jer svo vreme dok se opslužuju zadaci on je prisutan u sistemu i čeka na početak opsluge. Kako bi se odredio stepen vremenske razdvojenosti dva zadatka uvodi se parametar  $\alpha$  koji predstavlja odnos vremenske razlike u vremenima pojave dva zahteva i dužine trajanja planskog perioda. Redukcija  $\mathcal{N}_{swap}$  okoline podrazumeva da u okolinu mogu ući samo rešenja kod kojih je  $\alpha$  manje od unapred zadate vrednosti  $\alpha_{crit}$ .

Veličina  $\mathcal{N}_{swap}$  okoline, a samim tim i intenzitet redukcije, je obrnuto proporcionalna vrednosti parametra  $\alpha_{crit}$ , jer sa smanjenjem vrednosti parametra sve manji broj parova zadataka zadovoljava uslov i pretraga okoline je brža. Sa druge strane, veći broj rešenja u okolini rezultira i većom verovatnoćom pronalaska boljeg rešenja. Kako bi se utvrdila vrednost parametra  $\alpha_{crit}$ , Bjelić i ostali (2013b) su sproveli preliminarnu

analizu rešenja slučajno izabranog skupa instanci dobijenih primenom VNS algoritma sa redukovanom okolinom  $\mathcal{N}_{swap}$  za različite vrednosti parametra  $\alpha_{crit}$ . Rezultati ove analize su pokazali da su najbolja rešenja dobijena za vrednost parametra  $\alpha_{crit}$  od 0.3, tj. kada se okolina formira razmenama zadataka čija razlika u vremenima pojave su manja od 30% širine planskog perioda.

Redukcija  $\mathcal{N}_{relocate}$  okoline je takođe zasnovana na ideji da se iz okoline izuzmu rešenja u kojima se javljaju zadaci sa velikom akumulacijom opsluge prethodnika iz sekvence opsluge. Konkretno, redukcija  $\mathcal{N}_{relocate}$  podrazumeva da se iz okoline eliminišu rešenja koja se dobijaju premeštanjem zadatka na novu poziciju ukoliko je vreme pojave zadatka koji se premešta veće od trenutka pojave zadatka na čiju poziciju u sekvenci opsluge se zadatak ubacuje. Na taj način se izbegavaju rešenja u kojima bi serviser čekao na pojavu premeštenog zadatka, odnosno u kojima bi svi zadaci akumulirali čekanje serviseru i opslugu premeštenog zadatka u svoj doprinos ciljnoj funkciji. Na primeru instance rc 202.2 je jasno da bi premeštanje zadatka 100 iz sekvence opsluge serviseru 1 u sekvencu opsluge serviseru 2 na bilo koju drugu poziciju, sem poslednje u sekvenci, značilo da bi svi zadaci koji su trenutno u sekvenci opsluge serviseru 2 od pozicije ubacivanja pa do kraja sekvence morali da čekaju do 318 v.j. da serviser krene da njih opslužuje. Iz tog razloga u okolinu  $\mathcal{N}_{relocate}$  ne bi ušla rešenja u kojima se zadatak 100 ubacuje na poziciju zadatka čiji trenutak opsluge je manji od 308. Sa druge strane, zadatke 95 i 97 je moguće premestiti na bilo koju poziciju u sekvencu opsluge serviseru 1 jer im je trenutak pojave u 0.00 v.j. tj. serviser nikada neće čekati na njihovu pojavu i time uzrokovati bespotrebnu akumulaciju vremena zadatka<sup>13</sup> koji slede premešteni zadatak.

Intenzitet redukcije  $\mathcal{N}_{relocate}$  okoline u ovom slučaju zavisi od strukture trenutaka pojave zadataka. Ovo znači da bi u slučaju hetTRPTW problema u kome svi zadaci stižu u sistem istovremeno sva premeštanja bila dozvoljena i redukcije okoline ne bi ni bilo, dok je u dinamičkom hetTRPTW-u broj mogućih rešenja određen strukturom polaznog rešenja i vremenskim rasporedom pojave zadataka u sistemu.

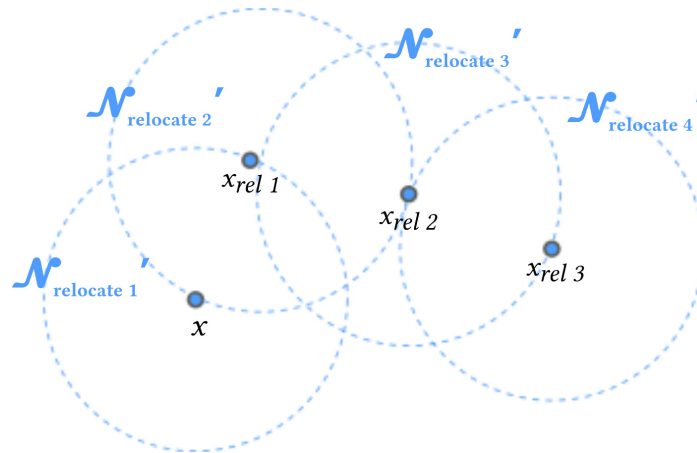
---

<sup>13</sup> Do povećanja doprinosa ciljnoj funkciji zadataka posluženih nakon premeštenog zadatka može doći, ali ne zbog toga što serviser čeka na pojavu zadatka umesto da opslužuje zadatke koji su već prisutni u sistemu, već zbog lošijeg mesta zadatka u sekvenci opsluge serviseru.

### 5.5.3.6. Razmrdavanje rešenja

Korak 2(a) iz algoritma 3 u kome se na slučajan način definiše novi centar lokalne pretrage  $x'$  je, zbog očekivanja da su lokalni optimumi blizu jedni drugima u prostoru rešenja, realizovan primenom koncepta ugnježenih okolina. Koraci koji su korišćeni u formiranju okolina su relocate i ExtSwap.

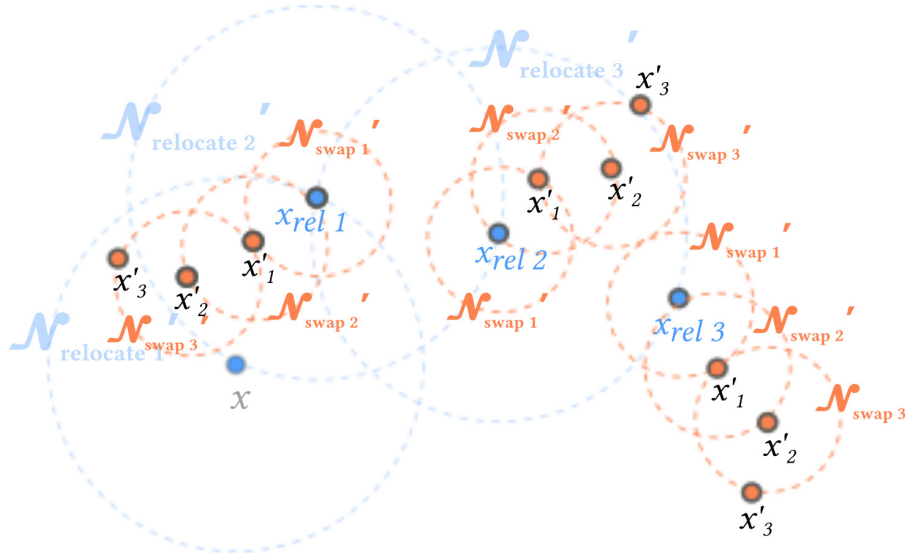
Prvi skup ugnježenih okolina  $\mathcal{N}_{relocate\ k}'$  ( $k = 1, 2, \dots, k_{relocate}^{max}$ ) se dobija sukcesivnim primenama relocate koraka, korišćenog u formiranju  $\mathcal{N}_{relocate}$  okoline u VND algoritmu, na trenutno najboljem rešenju VNS algoritma. Treba napomenuti da sukcesivna primena relocate koraka znači da je početno rešenje za  $k+1$  okolinu slučajno odabrano rešenje,  $x_{rel\ k}$ , iz  $k$ -te okoline, dok je za dobijanje rešenja iz prve okoline,  $x_{rel\ 1}$ , primenjen relocate korak na rešenju  $x$ . Grafički prikaz ugnježdavanja rešenja iz  $\mathcal{N}_{relocate\ k}'$  okolina, kada je  $k_{relocate}^{max} = 4$  je dat na slici 5.10.



**Slika 5.10.** Ugnježdavanje  $\mathcal{N}_{relocate\ k}'$  okolina

ExtSwap korak je prošireni swap korak, koji je, u suštini identičan swap koraku korišćenom za formiranje  $\mathcal{N}_{swap}$  u VND algoritmu, ali sa tom razlikom što je omogućena i razmena zadataka dodeljenih istom sredstvu. Ugnježdene okoline dobijene primenom ovog koraka,  $\mathcal{N}_{swap\ k}'$  ( $k = 1, 2, \dots, k_{swap}^{max}$ ) se dobijaju sukcesivnim primenama ExtSwap koraka na početnom rešenju. Slično kao i kod  $\mathcal{N}_{relocate\ k}'$  okolina i u ovom slučaju se kao početno rešenje za  $\mathcal{N}_{swap\ k+1}'$  okolinu uzima slučajno generisano rešenje iz  $\mathcal{N}_{swap\ k}'$  okoline. Međutim, kao početno rešenje za okolinu  $\mathcal{N}_{swap\ 1}'$  se uzima rešenje  $x_{rel\ k}$  iz odgovarajuće  $\mathcal{N}_{relocate\ k}'$  okoline. Na taj način se ugnježdavanje  $k_{swap}^{max}$   $\mathcal{N}_{swap\ k}'$  okolina ponavlja za svako rešenje iz

odgovarajuće  $\mathcal{N}_{relocate\ k'}$  okolina. Slikoviti primer ugnježdavanja okolina u procesu razmrđavanja rešenja za slučajeve kada je  $k_{relocate}^{max} = 3$ , a  $k_{swap}^{max} = 3$  je prikazan na slici 5.11.



Slika 5.11. Primer razmrđavanja u VNS algoritmu

Pseudo kod koji u korak 2(a) algoritma 3, vraća novi centar lokalne pretrage ( $x'$ ) iz  $k$ -te okoline trenutno najboljeg rešenja  $x$  je prikazan algoritmom 5.

**Algoritam 5** – Pseudokod za realizaciju procesa razmrđavanja

*Inicijalizacija:* Definisati vrednosti  $k_{relocate}^{max}$  i  $k_{swap}^{max}$

Ako je  $k = 1$ :

$z \leftarrow 1$ ;  $Rel \leftarrow \{ \}$ ;  $x_c \leftarrow x$ ;

dok je  $z < k_{relocate}^{max}$ :

generisati  $x_{rel\ z}$  realizacijom relocate koraka za slučajno odabrane zadatak i poziciju ubacivanja u  $x_c$

$Rel \leftarrow Rel \cup x_{rel\ z}$ ;  $x_c \leftarrow x_{rel\ z}$ ;  $z \leftarrow z + 1$

Ako je  $(k - 1) \% k_{swap}^{max} = 0$ :

$w \leftarrow (k - 1) \% k_{swap}^{max} + 1$ ;  $z \leftarrow (k - 1) \setminus k_{swap}^{max} + 1$ ;  $Swa \leftarrow \{ \}$ ;

$x_c \leftarrow x_{rel\ z}$  (rešenje koje se u skupu  $Rel$  nalazi na poziciji  $z$ )

dok je  $w < k_{swap}^{max}$ :

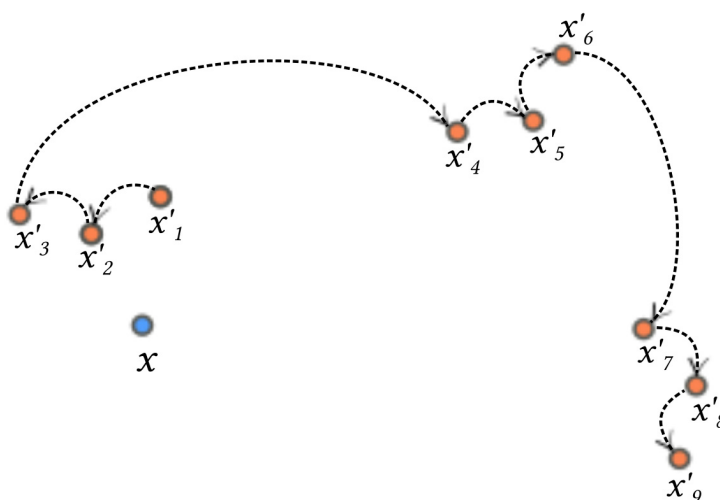
generisati  $x_w'$  realizacijom ExtSwap koraka za slučajno odabran par zadataka u rešenju  $x_c$

$Swa \leftarrow Swa \cup x_w'$ ;  $x_c \leftarrow x_w'$ ;  $w \leftarrow w + 1$

$q \leftarrow (k - 1) \% k_{swap}^{max} + 1$

Kao novo  $x'$  vrati rešenje  $x_q'$  (rešenje koje se u skupu  $Swa$  nalazi na poziciji  $q$ )

Operator  $\%$  i operator  $\setminus$  u algoritmu 5 se respektivno odnose na modularno i celobrojno deljenje brojeva. Na osnovu algoritma 5, kao i opisa ugnježenih okolina, novi centri lokalne pretrage,  $x'$ , se menjaju u okolini trenutnog centra VNS algoritma. Primer promene centara lokalne pretrage, za situaciju sa slike 5.11, je prikazan na slici 5.12. Treba napomenuti da je ukupan broj centara lokalne pretrage jednak proizvodu  $k_{relocate}^{max}$  i  $k_{swap}^{max}$ , te su oni, radi razlikovanja, označeni sa  $x'_k$ ,  $k = 1, 2, \dots, k_{relocate}^{max} \cdot k_{swap}^{max}$ .



**Slika 5.12.** Kretanje centara lokalne pretrage po prostoru rešenja

#### 5.5.3.7. Uslovi za završetak algoritma

Kao uslov za završetak VNS algoritma obično se koriste uslovi vezani za broj iteracija koje se realizuju algoritmom (bilo kao ograničenje ukupnog broja iteracija, ili kao ograničenje broja iteracija bez unapređenja najboljeg rešenja), ili se ograničava vreme rada algoritma. U predloženom VNS algoritmu je korišćen kriterijum koji se zasniva na ograničenju broja iteracija bez unapređenja rešenja. Konkretno, svaki put kada se VND algoritmom rešenje  $x$  ne unapredi centar lokane pretrage ( $x'$ ) se, realizacijom procesa razmrdavanja, tj. algoritmom 5, premešta u novi centar pretrage. Ukoliko nakon  $k_{max}$  ovakvih koraka VND ne uspe da poboljša trenutno  $x$ , ono se smatra rešenjem algoritma. U tom smislu uslov za završetak rada VNS algoritma je  $k_{max}$  realizacija VND algoritma bez unapređenja rešenja.

Obzirom na ugnježdenu strukturu okolina iz kojih se generišu centri lokalne pretrage, u ovom slučaju je  $k_{max} = k_{max}^1 \cdot k_{max}^2$  tj. jednoznačno je definisano tek kada su

definisani  $k_{max}^1$  i  $k_{max}^2$ . Što je veći broj okolina veća je i šansa za pronalaženjem optimalnog rešenja problema, ali to sa sobom nosi i veće vreme realizacije algoritma usled većeg broja realizacija VND algoritma. Definisanje odnosa vremena realizacije algoritma i kvaliteta dobijenih rešenja je bilo predmet testiranja u istraživanju Bjelić i ostali (2013b). U njemu su autori, za slučajno odabrane instance problema različitih veličina i različite kombinacije vrednosti parametara  $k_{max}^1$  i  $k_{max}^2$ , prateći vrednosti vremena realizacije algoritma i vrednost ciljne funkcije rešenja zaključili da se njihov najbolji odnos dobija kada su vrednosti parametara jednaki pet.

U cilju minimizacije vremena realizacije algoritma pored VNS algoritma u kome se lokalna pretraga realizuje VND algoritmom sa celim okolinama, posmatran je i VNS algoritam u kome su okoline korišćene u VND algoritmu redukovane. Takav VNS algoritam je referenciran u nastavku rada kao RdVNS.

## 5.6. Numerički eksperimenti efikasnosti modela

Kao što je to već napomenuto, predloženi algoritam ne garantuje kvalitet dobijenog rešenja, pa se njegova efikasnost proverava empirijskim putem, testiranjem na različitim instancama problema. Istraživačka praksa je da se predloženi algoritmi testiraju na skupovima testnih problema, tzv. benčmark instanci, koji su dostupni svim istraživačima kako bi performanse algoritma bile uporedive. U tom smislu, predloženi VNS algoritam je testiran na četiri skupa benčmark instanci: Dumas, Solomon, Langvin (dostupni na web adresi <http://my-web.uiowa.edu/bthoa/TSPTWBenchmarkDataSets.htm>) i Ashayer (dostupan na <http://ftp.zib.de/pub/mp-testdata/tsp/atsptw/index.html>).

Ono što treba napomenuti na početku je da namenske benčmark instance za hetTRPTW ne postoje, jer problem i nije bio predmet interesovanja istraživača. Iz tog razloga, korišćene su benčmark instance problema koji poseduju najbitnije, u smislu testiranja algoritma, karakteristike hetTRPTW-a. Kako je jedna od osnovnih karakteristika hetTRPTW-a dinamička pojava zadataka, ili drugačije posmatrano, postojanje leve granice vremenskih prozora koje označavaju trenutke od kojih je zadatke moguće opslužiti, to je za benčmark instance od najbitnijeg značaja da poseduju informacije o vremenskim prozorima zadataka. Iz tog razloga, kao benčmark instance korišćene su pomenute instance namenjene testiranju TSPTW problema.



Međutim, kako TSPTW benčmark instance imaju i leve i desne granice vremenskih prozora, pri testiranju je desna granica prozora zanemarena.

Dalje prilagođavanje instanci hetTRPTW-u je podrazumevalo respektovanje prisustva heterogene flote servisera pri realizaciji opsluge, što je realizovano uvođenjem korektivnog koeficijenta  $c_v$ . Obzirom da se serviseri u floti mogu razlikovati u brzini (odnosno vremenu) kretanja između lokacija zadataka, ali i u brzini opsluge, a imajući u vidu i da TSPTW podrazumeva prisustvo samo jednog trgovačkog putnika, to je koeficijent  $c_v$  korišćen dvojako, tj. i za korekciju potrebnog vremena kretanja i za korekciju dužine opsluge. Konkretno, vrednost koeficijenta  $c_v$  za servisera  $v$  je računata kao,  $c_v = 1 + 0.1(v - 1)$  gde je  $v = \{1, 2, \dots, m\}$  indeks servisera iz skupa  $V$ . Korigovano vreme potrebno serviseru  $v$  da pređe rastojanje od lokacije zadatka  $i$  do lokacije zadatka  $j$  se računa kao  $t_{ij}^v = \frac{t_{ij}}{c_v}$ , a korigovano vreme opsluge zadatka  $j$  se računa kao  $s_j^v = c_v \cdot s_j$ . Vremena  $t_{ij}$  i  $s_j$  su respektivno, vremena potrebna putujućem trgovcu da savlada rastojanje od lokacije zadatka  $i$  do lokacije zadatka  $j$  i da opsluži klijenta  $j$  u TSPTW benčmark instancama. Flote servisera koje su posmatrane u benčmark instancama se sastoje od 2, 5 i 10 servisera.

Obzirom na povećanje kompleksnosti problema sa povećanjem broja posmatranih zadataka kao benčmark instance su korišćene samo instance sa manje od 100 zadataka. Iz tog razloga je skup Ashayerovih benčmark instanci redukovan na 25 instanci sa manje od 100 zadataka. Pored toga, imajući u vidu da je jedan od ciljeva postavljenih pred TSPTW benčmark instance da provere efikasnost algoritama za različite širine vremenskih prozora, što kod hetTRPTW-a nije slučaj, jer postoji samo leva granica prozora, iz Dumas skupa benčmark instanci su korišćene samo instance sa širinom prozora od  $100v$ .j. pri čemu je ova širina odabrana na slučajan način. Iz Solomonovog i Langevinovog skupa benčmark instanci su korišćene sve instance problema jer je u svim instancama prisutno manje od 100 zadataka i ne tretiraju različite širine vremenskih prozora za opslugu zadataka.

U cilju pokušaja optimalnog rešavanja problema, instance manjih dimenzija (sa manje od 20 zadataka) su rešavane primenom B&B algoritma implementiranog u softverski paket CPLEX 12.2, pri čemu je memorija za organizaciju rešenja u formi drveta ograničena na 1GB. Uključujući i instance malih dimenzija, za sve benčmark instance

je računata najmanja vrednost rešenja (LB), kao i vrednosti ciljne funkcije dobijene primenom predstavljenih VNS i RdVNS algoritama.

Svi algoritmi su realizovani na računaru koji radi pod Windows XP SP3 operativnim sistemom sa AMD Phenom II 2.61GHz CPU-om sa 2GB RAM, pri čemu je kodiranje svih algoritma realizovano u programskom jeziku Python 2.5. Svaka instanca problema je rešavana po jednom za slučaj determinističkog B&B algoritama, dok je u slučajevima stohastičkih algoritama (VNS i RdVNS) rešavana po pet puta.

Efikasnost predloženih algoritama ocenjivana je respektovanjem dva parametra njihovog rada, prosečnom vrednošću ciljne funkcije dobijenih rešenja i prosečnim vremenom potrebnom algoritmu za generisanje rešenja. U tom cilju, prosečne vrednosti ciljnih funkcija rešenja dobijenih rešavanjem Aschayer-ovog, Dumas-ovog, Solomon-ovog i Lengevin-ovog skupa benčmark instanci su date u tabelama 5.3 do 5.6, respektivno, dok su prosečna vremena realizacije algoritama data u tabelama 5.7 do 5.10. Vrednosti ovih parametara su prikazane za svaki respektovani broj servisera u floti. Kolone „Instanca“ u tabelama daju informaciju o benčmark instanci za koju su prikazani rezultati, dok kolona „n“ sadrži informaciju o broju zadataka koji se tretira u instanci. U koloni „LB“ svake tabele se nalazi informacija o najmanjoj vrednosti rešenja instance (eng. lower bound).

Ostatak podataka u tabelama je organizovan na takav način da je vrednost parametra za slučaj VNS algoritma posmatran kao bazna vrednost ( $f_{base}$ ) i njegova nominalna vrednost je data u koloni „VNS“ za svaki parametar i broj servisera u floti. Relativno odstupanje, izraženo u procentima, nominalne vrednosti parametra ( $f_{param}$ ) primenjenog algoritma od nominalne vrednosti parametra za slučaj VNS algoritma ( $f_{base}$ ) je dato u kolonama „B&B“, „Insertion“ i „RdVNS“ za slučajeve B&B, algoritma ubacivanja i RdVNS algoritma, respektivno. Konkretno, vrednost relativnog odstupanja je računata kao  $\frac{f_{param}-f_{base}}{f_{base}} \cdot 100\%$ . Korišćenjem ovakve metrike u merenju performansi lako je uočiti odnos performansi svakog algoritma u odnosu na bazni algoritam, tj. u ovom slučaju VNS. Naime, što je negativnija vrednost relativnog odstupanja to je performansa konkretnog algoritma bolja u odnosu na VNS algoritam, i obrnuto.

**Tabela 5.3.** Prosečne vrednosti ciljne funkcije za slučaj Ashayer-ovog skupa benčmark instanci

Instanca	n	LB	m=2			m=5			m=10		
			$\Delta B\&B$ [%]	VNS	$\Delta RdVNS$ [%]	$\Delta B\&B$ [%]	VNS	$\Delta RdVNS$ [%]	$\Delta B\&B$ [%]	VNS	$\Delta RdVNS$ [%]
rbg010a.tw	10	522	<b>0.0%</b>	522	0.00%	0.0%	522	0.00%	0.0%	522	0.00%
rbg016b.tw	16	1162	<b>-0.1%</b>	1189.34	0.00%	3.2%	1188.5	0.00%	15.9%	1188.5	0.07%
br17a.07.tw	17	976	-0.1%	2719.61	0.17%	3.0%	1612.46	0.04%	10.6%	1320.2	0.08%
br17a.10.tw	17	976	1.1%	2136.57	0.23%	6.0%	1303.62	0.33%	8.4%	1170.42	0.02%
br17b.08.tw	17	473	3.7%	1287.31	1.24%	2.3%	708.28	0.59%	10.2%	591.8	0.41%
br17c.05.tw	17	470	0.2%	1494.59	0.01%	3.4%	812.18	1.25%	17.8%	647.52	2.60%
br17f.03.tw	17	508	0.2%	1944.34	0.22%	3.9%	1037.41	0.02%	4.7%	796.22	1.12%
br17f.10.tw	17	508	0.6%	1164.05	0.04%	5.1%	728.03	0.24%	6.4%	629.9	0.00%
rbg017a.tw	17	4150	<b>0.0%</b>	4486.609	0.00%	1.4%	4226.8	0.00%	1.6%	4226.8	0.02%
rbg019a.tw	19	1045	<b>0.0%</b>	1064.1	0.00%	9.5%	1064.1	0.00%	25.7%	1064.1	0.00%
rbg021.3.tw	21	4346		4513.38	0.00%		4427.3	0.00%		4427.3	0.00%
rbg27.a.01.tw	27	732		5478.35	0.17%		2577.87	0.88%		1724.00	0.69%
rbg27.b.10.tw	27	693		3784.37	1.06%		1766.42	0.34%		1219.80	0.23%
rbg27.b.19.tw	27	693		1457.82	2.86%		912.36	1.54%		790.7	1.18%
rbg27.c.03.tw	27	788		5560.33	0.46%		2530.73	0.09%		1664.98	0.55%
rbg27.c.06.tw	27	788		4847.20	1.70%		2115.26	2.51%		1450.87	2.99%
rbg035a.2.tw	35	1890		5040.58	0.12%		2855.52	-0.07%		2379.70	-0.01%
rbg035a.tw	35	1890		1959.2	0.01%		1954.74	-0.02%		1954.96	-0.03%
rbg040a.tw	40	1992		2078.58	0.00%		2048.8	0.00%		2048.8	0.00%
rbg042a.tw	42	2361		2789.71	0.00%		2507.52	-0.01%		2508.79	-0.06%
rbg050a.tw	50	2539		2813.4	0.00%		2635.1	0.00%		2635.1	0.04%
rbg050b.tw	50	9351		10776.18	-0.03%		9599.15	0.00%		9599.95	-0.01%
rbg067a.tw	67	3577		3846.07	-0.04%		3696.06	-0.01%		3695.6	0.00%
rbg088a.tw	68	4572		4872.18	-0.04%		4723.04	0.00%		4725.02	0.03%
rbg092a.tw	92	6065		6338.44	0.00%		6247.48	0.04%		6248.49	0.00%
<i>Prosečno</i>			0.57%	3366.57	0.33%	3.76%	2552.03	0.31%	10.13%	2369.26	0.40%

**Tabela 5.4.** Prosečne vrednosti ciljne funkcije za slučaj Dumas-ovog skupa benčmark instanci

Instanca	n	LB	m=2			m=5			m=10		
			$\Delta B\&B$ [%]	VNS	$\Delta RdVNS$ [%]	$\Delta B\&B$ [%]	VNS	$\Delta RdVNS$ [%]	$\Delta B\&B$ [%]	VNS	$\Delta RdVNS$ [%]
n20w100.001.txt	20	0	<b>0.0%</b>	153.71	0.00%	2.4%	98.30	0.00%	11.5%	72.06	0.00%
n20w100.002.txt	20	0	<b>0.0%</b>	142.82	0.00%	4.3%	89.60	-0.04%	12.4%	62.45	0.06%
n20w100.003.txt	20	0	5.2%	170.57	3.92%	13.6%	100.31	0.00%	1.1%	68.41	1.13%
n20w100.004.txt	20	0	5.6%	163.17	0.00%	3.5%	72.84	0.00%	13.4%	52.52	-0.13%
n20w100.005.txt	20	0	<b>0.0%</b>	130.6	1.78%	9.0%	67.57	0.13%	16.5%	44.24	0.00%
n40w100.001.txt	40	0		217.46	1.61%		70.42	-0.53%		46.14	0.42%
n40w100.002.txt	40	0		202.23	0.10%		67.61	0.00%		49.36	-0.47%
n40w100.003.txt	40	0		265.92	-0.19%		134.88	-0.19%		91.62	-0.20%
n40w100.004.txt	40	0		315.76	1.62%		130.36	-0.13%		86.94	0.21%
n40w100.005.txt	40	0		183.44	1.30%		66.47	0.00%		45.90	0.12%
n60w100.001.txt	60	0		340.01	-0.02%		112.48	0.71%		76.75	0.01%
n60w100.002.txt	60	0		334.58	-0.22%		86.41	-1.16%		56.23	-0.08%
n60w100.003.txt	60	0		471.12	-0.92%		164.00	0.27%		109.82	0.72%
n60w100.004.txt	60	0		362.71	0.66%		162.15	-0.66%		115.32	-0.53%
n60w100.005.txt	60	0		413.71	0.43%		114.94	3.63%		64.42	-0.20%
n80w100.001.txt	80	0		679.726	0.15%		182.01	-1.22%		121.88	0.29%
n80w100.002.txt	80	0		392.72	0.74%		52.98	1.51%		32.62	-0.13%
n80w100.003.txt	80	0		551.51	1.35%		133.62	-3.16%		88.98	0.28%
n80w100.004.txt	80	0		494.29	2.46%		89.05	-0.05%		57.69	0.28%
n80w100.005.txt	80	0		469.61	-0.20%		126.34	-0.77%		85.49	0.15%
n100w100.001.txt	100	0		507.83	-0.03%		76.54	0.17%		48.92	-0.02%
n100w100.002.txt	100	0		550.89	1.17%		156.04	0.00%		95.31	-0.05%
n100w100.003.txt	100	0		489.03	-0.28%		119.83	0.43%		74.32	0.19%
n100w100.004.txt	100	0		489.905	-2.70%		84.94	0.00%		54.81	0.67%
n100w100.005.txt	100	0		492.48	0.07%		173.47	0.09%		116.45	0.00%
<i>Prosečno</i>			2.16%	359.43	0.51%	6.56%	109.33	-0.04%	10.97%	72.75	0.11%

**Tabela 5.5.** Prosečne vrednosti ciljne funkcije za slučaj Solomon-ovog skupa benčmark instanci

Instanca	n	LB	m=2			m=5			m=10		
			$\Delta B\&B$ [%]	VNS	$\Delta RdVNS$ [%]	$\Delta B\&B$ [%]	VNS	$\Delta RdVNS$ [%]	$\Delta B\&B$ [%]	VNS	$\Delta RdVNS$ [%]
rc_206.1.txt	3	30	<b>0.0%</b>	41.18	0.00%	<b>0.0%</b>	33.01	0.00%	<b>0.0%</b>	33.01	0.00%
rc_207.4.txt	5	50	<b>0.0%</b>	62.58	0.00%	<b>0.0%</b>	53.18	0.00%	<b>0.0%</b>	53.18	0.00%
rc_202.2.txt	13	130	<b>0.0%</b>	547.17	0.40%	0.4%	359.74	-0.17%	1.1%	304.96	0.00%
rc_204.4.txt	13	130	<b>0.0%</b>	181.77	0.00%	1.3%	143.08	0.00%	1.7%	143.08	0.00%
rc_205.1.txt	13	130	<b>0.0%</b>	181.77	0.00%	1.3%	143.08	0.00%	1.7%	143.08	0.00%
rc_203.4.txt	14	140	-0.2%	926.05	-0.32%	0.3%	540.52	0.12%	1.5%	426.34	0.05%
rc_203.1.txt	18	180	5.9%	570.87	0.11%	2.6%	366.42	0.46%	7.5%	311.07	0.26%
rc_201.1.txt	19	190	<b>0.0%</b>	224.82	0.61%	2.3%	200.595	0.24%	11.3%	200.8398	0.15%
rc_204.3.txt	23	230		2213.50	1.03%		1272.18	0.24%		981.57	-0.04%
rc_206.3.txt	24	240		392.97	0.95%		259.45	0.69%		259.69	0.39%
rc_201.2.txt	25	250		330.97	2.57%		271.50	0.11%		271.41	0.10%
rc_201.4.txt	25	250		319.38	0.00%		268.355	0.00%		268.355	0.00%
rc_205.2.txt	26	260		420.47	0.09%		280.20	0.00%		280.40	0.07%
rc_202.4.txt	27	270		796.96	0.12%		530.28	-0.05%		463.64	0.04%
rc_205.4.txt	27	270		382.97	4.37%		289.05	0.33%		289.05	0.43%
rc_202.3.txt	28	280		496.79	1.07%		356.34	0.20%		352.74	0.17%
rc_208.2.txt	28	280		1141.35	-0.17%		446.98	0.05%		362.19	0.25%
rc_207.2.txt	30	300		624.64	5.12%		342.42	1.18%		337.84	1.03%
rc_201.3.txt	31	310		417.34	-0.20%		333.59	-0.09%		333.16	0.09%
rc_202.1.txt	32	320		1021.901	0.11%		686.7072	0.89%		596.546	0.25%
rc_203.2.txt	32	320		2012.38	1.07%		1153.61	-0.14%		842.69	-0.03%
rc_204.2.txt	32	320		2914.90	1.40%		1499.98	1.78%		1111.45	3.01%
rc_207.3.txt	32	320		671.18	3.29%		364.19	0.78%		356.90	1.35%
rc_207.1.txt	33	330		696.09	2.02%		378.31	1.69%		373.96	1.27%
rc_205.3.txt	34	340		903.96	1.13%		434.29	-0.05%		405.44	0.05%
rc_208.3.txt	35	350		1176.071	0.85%		466.3478	1.91%		425.4158	1.59%
rc_203.3.txt	36	360		2570.94	0.57%		1367.04	0.96%		1075.92	1.99%
rc_206.2.txt	36	360		833.80	0.61%		417.61	0.12%		404.80	0.10%
rc_206.4.txt	37	370		674.71	-0.58%		409.20	0.03%		409.82	-0.20%
rc_208.1.txt	37	370		1631.99	1.06%		514.84	2.59%		462.31	1.43%
rc_204.1.txt	45	450		5882.42	1.70%		2851.24	0.70%		1954.17	0.26%
<i>Prosečno</i>			0.7%	1136.093	1.02%	1.0%	596.5139	0.54%	3.1%	490.3908	0.54%

**Tabela 5.6.** Prosečne vrednosti ciljne funkcije za slučaj Langevin-ovog skupa benčmark instanci

Instanca	n	LB	m=2			m=5			m=10		
			$\Delta B\&B$ [%]	VNS	$\Delta RdVNS$ [%]	$\Delta B\&B$ [%]	VNS	$\Delta RdVNS$ [%]	$\Delta B\&B$ [%]	VNS	$\Delta RdVNS$ [%]
N20ft204.dat	19	0	<b>0.0%</b>	11.618	0.00%	0.0%	0	0.00%	0.0%	0	0.00%
N20ft204new.txt	19	0	<b>0.0%</b>	11.62	0.00%	0.0%	0.00	0.00%	0.0%	0.00	0.00%
N20ft205.dat	19	0	<b>0.0%</b>	12.64	0.00%	0.0%	3.29	0.00%	0.0%	0.00	0.00%
N20ft206.dat	19	0	<b>0.0%</b>	8.18	0.00%	0.0%	4.93	0.00%	0.0%	3.37	0.00%
N20ft207.dat	19	0	<b>-0.4%</b>	21.64	-0.36%	0.0%	0.43	0.00%	0.0%	0.00	0.00%
N20ft208.dat	19	0	<b>0.0%</b>	9.00	0.00%	0.0%	7.07	0.00%	0.0%	5.21	0.00%
N20ft209.dat	19	0	<b>0.0%</b>	5.00	0.00%	169.3%	0.71	0.00%	0.0%	0.00	0.00%
N20ft210.dat	19	0	<b>0.0%</b>	14.46	0.00%	0.0%	10.07	0.00%	0.0%	5.84	0.00%
N20ft301.dat	19	0	<b>0.0%</b>	47.75	0.00%	9.5%	24.71	0.00%	0.0%	14.53	0.00%
N20ft302.dat	19	0	<b>0.0%</b>	13.91	0.00%	0.0%	7.39	0.00%	0.0%	3.68	0.00%
N20ft303.dat	19	0	<b>0.0%</b>	13.36	2.10%	8.4%	10.29	0.00%	0.0%	7.32	0.00%
N20ft304.dat	19	0	<b>0.0%</b>	26.26	0.00%	0.0%	2.86	0.00%	0.0%	0.00	0.00%
N20ft305.dat	19	0	<b>-1.16%</b>	19.53	3.37%	6.5%	9.29	0.00%	0.0%	1.47	0.00%
N20ft306.dat	19	0	<b>0.0%</b>	9.18	0.00%	0.0%	5.93	0.00%	5.6%	4.37	0.00%
N20ft307.dat	19	0	<b>0.0%</b>	30.82	0.00%	0.0%	7.96	0.00%	0.0%	0.00	0.00%
N20ft308.dat	19	0	<b>0.0%</b>	9.00	0.00%	0.0%	7.07	0.00%	0.0%	5.21	0.00%
N20ft309.dat	19	0	<b>0.0%</b>	10.90	2.00%	75.7%	2.95	0.00%	0.0%	0.00	0.00%
N20ft310.dat	19	0	<b>0.0%</b>	21.46	0.00%	0.0%	16.07	0.00%	0.0%	11.84	0.00%
N20ft401.dat	19	0	<b>-0.76%</b>	78.48	-0.77%	2.6%	43.07	-0.30%	5.0%	24.53	0.00%
N20ft402.dat	19	0	<b>0.0%</b>	21.36	1.84%	0.0%	11.39	0.00%	4.4%	5.15	0.00%
N20ft403.dat	19	0	<b>0.0%</b>	17.46	2.44%	0.0%	11.29	0.00%	0.0%	8.32	0.00%
N20ft404.dat	19	0	<b>0.0%</b>	41.09	3.75%	28.3%	7.14	0.00%	0.0%	0.63	0.00%
N20ft405.dat	19	0	<b>0.0%</b>	28.00	0.00%	9.4%	17.29	0.00%	0.0%	8.26	0.00%
N20ft406.dat	19	0	<b>0.0%</b>	11.18	4.44%	0.0%	5.93	0.00%	119.5%	4.37	0.00%
N20ft407.dat	19	0	<b>0.0%</b>	47.72	0.00%	65.5%	21.96	0.00%	5.8%	3.23	1.39%
N20ft408.dat	19	0	<b>0.0%</b>	14.00	0.00%	0.0%	10.46	0.30%	0.0%	7.10	0.00%
N20ft409.dat	19	0	<b>0.0%</b>	23.90	0.00%	0.0%	12.95	0.00%	0.0%	5.41	0.00%
N20ft410.dat	19	0	<b>0.0%</b>	26.36	0.00%	6.2%	16.30	0.00%	0.0%	11.84	0.00%
N40ft201.dat	39	0		15.36	0.00%		11.43	0.00%		8.42	0.00%
N40ft202.dat	39	0		23.73	0.00%		16.07	0.00%		9.74	0.00%
N40ft203.dat	39	0		21.03	0.00%		1.79	0.00%		0.00	0.00%
N40ft204.dat	39	0		31.26	6.04%		11.14	0.00%		4.79	0.00%
N40ft205.dat	39	0		30.93	0.00%		14.99	0.00%		9.63	0.00%
N40ft206.dat	39	0		9.64	0.00%		6.07	0.00%		2.63	0.00%
N40ft207.dat	39	0		11.18	1.46%		0.00	0.00%		0.00	0.00%
N40ft208.dat	39	0		24.13	6.38%		4.64	0.00%		1.05	0.00%
N40ft209.dat	39	0		12.82	0.00%		7.36	0.00%		5.42	0.00%
N40ft210.dat	39	0		13.75	-9.13%		1.43	0.00%		0.00	0.00%
N40ft301.dat	39	0		31.13	0.00%		14.81	0.00%		8.42	0.00%
N40ft302.dat	39	0		32.96	6.07%		22.07	0.00%		15.74	0.00%
N40ft303.dat	39	0		45.02	-0.10%		9.71	0.00%		0.95	0.00%
N40ft304.dat	39	0		49.77	2.46%		18.68	0.00%		10.79	0.00%
N40ft305.dat	39	0		56.25	-0.26%		27.56	0.00%		14.52	0.00%
N40ft306.dat	39	0		15.64	0.00%		12.07	0.00%		8.63	0.00%

**Tabela 5.6.** Prosečne vrednosti ciljne funkcije za slučaj Langevin-ovog skupa benčmark instanci (*nastavak*)

Instanca	n	LB	m=2			m=5			m=10		
			$\Delta B\&B$ [%]	VNS	$\Delta RdVNS$ [%]	$\Delta B\&B$ [%]	VNS	$\Delta RdVNS$ [%]	$\Delta B\&B$ [%]	VNS	$\Delta RdVNS$ [%]
N40ft307.dat	39	0		34.30	3.08%		0.45	0.00%		0.00	0.00%
N40ft308.dat	39	0		60.66	0.30%		10.86	0.00%		5.05	0.00%
N40ft309.dat	39	0		30.42	-0.90%		7.36	0.00%		5.42	0.00%
N40ft401.dat	39	0		49.46	0.00%		22.81	0.00%		16.36	0.00%
N40ft402.dat	39	0		46.60	1.96%		24.07	0.00%		17.74	0.00%
N40ft403.dat	39	0		65.47	0.18%		20.93	0.00%		6.95	0.00%
N40ft404.dat	39	0		82.77	5.14%		29.93	2.23%		17.35	0.00%
N40ft405.dat	39	0		84.29	0.00%		41.06	0.00%		24.63	0.00%
N40ft406.dat	39	0		18.62	3.94%		13.07	0.00%		9.63	0.00%
N40ft407.dat	39	0		49.99	2.94%		9.45	0.00%		4.47	0.00%
N40ft408.dat	39	0		106.03	0.61%		30.14	0.00%		9.05	0.00%
N40ft409.dat	39	0		44.61	-1.48%		11.64	0.00%		6.88	0.00%
N40ft410.dat	39	0		61.63	3.39%		9.14	0.00%		2.42	0.00%
N60ft201.dat	59	0		34.89	2.28%		13.78	0.76%		9.26	0.00%
N60ft202.dat	59	0		18.09	0.00%		4.86	0.00%		1.26	0.00%
N60ft203.dat	59	0		28.22	-0.66%		8.36	0.00%		6.16	0.00%
N60ft204.dat	59	0		25.63	-1.59%		8.79	0.00%		3.05	0.00%
N60ft205.dat	59	0		23.21	1.43%		9.92	0.20%		6.70	0.16%
N60ft206.dat	59	0		35.85	4.05%		22.41	0.00%		13.42	0.00%
N60ft207.dat	59	0		11.72	0.09%		6.84	0.00%		3.93	0.00%
N60ft208.dat	59	0		23.99	0.00%		14.98	0.00%		9.28	0.00%
N60ft209.dat	59	0		22.08	0.66%		6.50	0.00%		4.00	0.00%
N60ft210.dat	59	0		31.86	0.68%		11.97	0.00%		8.75	0.00%
N60ft301.dat	59	0		46.06	7.67%		16.78	0.00%		12.26	0.23%
N60ft302.dat	59	0		35.16	2.44%		11.86	0.00%		5.58	0.00%
N60ft303.dat	59	0		44.94	5.40%		9.64	0.00%		6.16	0.00%
N60ft304.dat	59	0		41.71	-4.26%		17.94	0.00%		8.63	0.00%
N60ft305.dat	59	0		33.23	-4.76%		11.92	0.00%		8.70	0.12%
N60ft306.dat	59	0		55.38	1.24%		33.41	0.00%		24.42	0.00%
N60ft307.dat	59	0		24.31	-9.46%		10.84	0.00%		7.93	0.00%
N60ft308.dat	59	0		45.78	-1.56%		19.98	0.00%		12.60	0.00%
N60ft309.dat	59	0		37.71	-2.25%		9.50	0.00%		7.00	0.00%
N60ft310.dat	59	0		52.48	-2.90%		15.31	0.54%		8.75	0.00%
N60ft401.dat	59	0		59.90	1.88%		18.78	0.82%		13.73	0.41%
N60ft402.dat	59	0		60.07	-1.90%		16.86	0.00%		10.58	0.00%
N60ft403.dat	59	0		67.12	3.94%		18.66	9.21%		6.16	0.00%
N60ft404.dat	59	0		79.77	0.59%		27.00	0.00%		17.63	0.00%
N60ft405.dat	59	0		44.48	11.07%		11.92	0.41%		8.70	0.06%
N60ft406.dat	59	0		70.01	1.97%		37.62	0.51%		27.34	0.57%
N60ft407.dat	59	0		44.38	0.60%		10.84	0.00%		7.93	0.00%
N60ft408.dat	59	0		71.43	-0.63%		24.98	0.00%		17.60	0.00%
N60ft409.dat	59	0		69.25	0.00%		11.96	0.00%		7.00	0.00%
N60ft410.dat	59	0		82.602	1.30%		23.419	1.33%		13.09	0.00%
<i>Prosečno</i>			<i>-0.08%</i>	<i>47.65098</i>	<i>0.63%</i>	<i>13.62%</i>	<i>16.87623</i>	<i>0.51%</i>	<i>5.01%</i>	<i>10.84643</i>	<i>0.06%</i>

**Tabela 5.7.** Prosečne vreme realizacije algoritama za slučaj Ashayer-ovog skupa benčmark instanci

Instanca	n	LB	m=2			m=5			m=10		
			$\Delta$ B&B [%]	VNS [s]	$\Delta$ RdVNS [%]	$\Delta$ B&B [%]	VNS [s]	$\Delta$ RdVNS [%]	$\Delta$ B&B [%]	VNS [s]	$\Delta$ RdVNS [%]
rbg010a.tw	10	522	129%	0.29	-90.4%	800586%	0.34	-82.9%	543595%	0.44	-70.6%
rbg016b.tw	16	1162	434%	1.64	-88.5%	142576%	1.70	-84.4%	126782%	2.19	-70.6%
br17a.07.tw	17	976	233390%	2.93	-66.5%	120728%	2.38	-46.1%	191773%	1.86	-48.6%
br17a.10.tw	17	976	278934%	2.36	-66.6%	125608%	2.05	-55.3%	165515%	1.85	-50.8%
br17b.08.tw	17	473	230242%	3.31	-75.6%	119113%	2.29	-57.3%	184740%	1.84	-18.3%
br17c.05.tw	17	470	258799%	2.65	-77.6%	157176%	1.93	-68.4%	12662%	2.41	-57.9%
br17f.03.tw	17	508	284166%	2.95	-49.1%	142687%	2.63	-22.0%	260757%	3.48	-54.4%
br17f.10.tw	17	508	179612%	3.52	-71.7%	113104%	2.87	-74.7%	205715%	1.64	-36.1%
rbg017a.tw	17	4150	1813%	3.95	-81.3%	140081%	1.67	-59.4%	99294%	2.66	-61.6%
rbg019a.tw	19	1045	268%	4.43	-91.5%	79759%	3.18	-87.2%	56422%	4.32	-81.1%
rbg021.3.tw	21	4346		5.94	-90.0%		3.10	-75.6%		3.33	-65.4%
rbg27.a.01.tw	27	732		20.96	-67.6%		12.36	-31.0%		14.29	-39.6%
rbg27.b.10.tw	27	693		18.57	-60.5%		10.80	-25.3%		20.46	-58.2%
rbg27.b.19.tw	27	693		15.29	-72.8%		13.49	-86.2%		7.91	-56.3%
rbg27.c.03.tw	27	788		11.48	-31.4%		13.58	-20.6%		20.21	-13.2%
rbg27.c.06.tw	27	788		10.62	-64.2%		10.12	-75.4%		17.59	-88.3%
rbg035a.2.tw	35	1890		32.70	-53.9%		30.00	-63.2%		24.61	-61.7%
rbg035a.tw	35	1890		34.31	-79.6%		36.12	-85.6%		29.82	-73.6%
rbg040a.tw	40	1992		50.68	-86.3%		35.81	-74.8%		41.44	-71.4%
rbg042a.tw	42	2361		59.56	-81.8%		46.87	-74.1%		61.02	-65.6%
rbg050a.tw	50	2539		149.88	-89.9%		95.35	-84.3%		126.01	-84.7%
rbg050b.tw	50	9351		268.09	-51.3%		74.10	-72.1%		85.66	-69.8%
rbg067a.tw	67	3577		461.86	-91.1%		349.98	-88.5%		341.37	-80.2%
rbg088a.tw	68	4572		1096.20	-86.6%		646.05	-72.7%		779.31	-81.5%
rbg092a.tw	92	6065		1093.93	-80.6%		869.92	-85.4%		776.75	-60.7%
<i>Prosečno</i>			146779%	134.32	-73.9%	194142%	90.75	-66.1%	184726%	94.90	-60.8%



**Tabela 5.8.** Prosečne vreme realizacije algoritama za slučaj Dumas-ovog skupa benčmark instanci

Instanca	n	LB	m=2			m=5			m=10		
			$\Delta$ B&B [%]	VNS [s]	$\Delta$ RdVNS [%]	$\Delta$ B&B [%]	VNS [s]	$\Delta$ RdVNS [%]	$\Delta$ B&B [%]	VNS [s]	$\Delta$ RdVNS [%]
n20w100.001.txt	20	0	630790%	2.85	-74.0%	658176%	1.68	-42.2%	622956%	2.62	-33.4%
n20w100.002.txt	20	0	319280%	3.28	-77.7%	513902%	1.97	-61.3%	428408%	3.24	-49.1%
n20w100.003.txt	20	0	855406%	3.99	-78.4%	478213%	1.75	-79.7%	507697%	3.07	-72.4%
n20w100.004.txt	20	0	984950%	2.93	-73.8%	422838%	2.40	-59.4%	532797%	3.42	-48.3%
n20w100.005.txt	20	0	545149%	3.34	-79.1%	413334%	2.08	-65.4%	544465%	2.86	-53.2%
n40w100.001.txt	40	0		65.00	-81.1%		21.27	-51.7%		23.01	-26.9%
n40w100.002.txt	40	0		49.69	-55.4%		16.12	-53.3%		18.94	-42.4%
n40w100.003.txt	40	0		36.99	-75.4%		19.82	-68.5%		21.72	-54.0%
n40w100.004.txt	40	0		50.93	-53.2%		21.93	-47.4%		21.79	-49.2%
n40w100.005.txt	40	0		49.70	-72.6%		19.17	-56.0%		24.54	-37.1%
n60w100.001.txt	60	0		199.50	-51.6%		120.47	-69.8%		134.46	-52.9%
n60w100.002.txt	60	0		166.80	-68.2%		64.25	-69.0%		85.45	-73.6%
n60w100.003.txt	60	0		386.12	-76.3%		68.59	-54.0%		109.99	-47.8%
n60w100.004.txt	60	0		275.79	-75.2%		83.40	-73.9%		79.19	-64.5%
n60w100.005.txt	60	0		281.67	-80.4%		87.36	-64.6%		79.58	-31.6%
n80w100.001.txt	80	0		657.60	-54.6%		286.89	-58.6%		159.44	-53.4%
n80w100.002.txt	80	0		692.20	-73.1%		220.30	-79.1%		250.55	-75.2%
n80w100.003.txt	80	0		570.70	-62.4%		301.82	-53.9%		222.54	-47.0%
n80w100.004.txt	80	0		779.40	-68.3%		155.86	-58.1%		271.80	-66.4%
n80w100.005.txt	80	0		893.14	-60.0%		217.61	-56.7%		272.23	-49.8%
n100w100.001.txt	100	0		1083.09	-39.9%		391.79	-67.2%		501.76	-46.4%
n100w100.002.txt	100	0		1848.77	-60.3%		294.66	-53.9%		511.10	-47.1%
n100w100.003.txt	100	0		1721.07	-68.9%		340.90	-61.2%		525.41	-54.3%
n100w100.004.txt	100	0		1791.75	-58.5%		324.69	-74.4%		466.13	-61.1%
n100w100.005.txt	100	0		1579.99	-39.6%		346.60	-54.5%		424.32	-44.1%
<i>Prosečno</i>			667115%	527.85	-66.3%	497293%	136.54	-61.4%	527265%	168.77	-51.2%

**Tabela 5.9.** Prosečne vreme realizacije algoritama za slučaj Solomon-ovog skupa benčmark instanci

Instanca	n	LB	m=2			m=5			m=10		
			$\Delta$ B&B [%]	VNS [s]	$\Delta$ RdVNS [%]	$\Delta$ B&B [%]	VNS [s]	$\Delta$ RdVNS [%]	$\Delta$ B&B [%]	VNS [s]	$\Delta$ RdVNS [%]
rc_206.1.txt	3	30	343%	0.01	-22.9%	474%	0.01	-25.9%	3781%	0.03	-15.9%
rc_207.4.txt	5	50	181%	0.03	-54.0%	9346%	0.04	-33.2%	92072%	0.08	-25.5%
rc_202.2.txt	13	130	796964%	1.05	-78.7%	1886604%	1.07	-49.3%	5896983%	0.68	-31.3%
rc_204.4.txt	13	130	1294%	0.72	-77.1%	856443%	0.68	-55.5%	1002887%	0.88	-35.2%
rc_205.1.txt	13	130	1191%	0.78	-80.7%	742314%	0.79	-58.9%	789530%	1.12	-56.0%
rc_203.4.txt	14	140	1163973%	1.31	-68.8%	1124869%	0.90	-7.9%	2114003%	1.06	-29.9%
rc_203.1.txt	18	180	614230%	2.77	-78.3%	411590%	2.75	-82.7%	869568%	1.86	-57.8%
rc_201.1.txt	19	190	10680%	3.43	-84.2%	255361%	2.90	-82.3%	238801%	3.61	-76.7%
rc_204.3.txt	23	230		8.06	-70.5%		8.05	-42.3%		8.58	-34.3%
rc_206.3.txt	24	240		11.90	-89.5%		5.38	-80.9%		6.59	-70.3%
rc_201.2.txt	25	250		11.89	-86.7%		4.76	-76.5%		7.53	-82.0%
rc_201.4.txt	25	250		8.91	-83.8%		4.87	-73.6%		7.22	-73.9%
rc_205.2.txt	26	260		10.48	-88.1%		5.55	-61.2%		7.73	-62.1%
rc_202.4.txt	27	270		12.30	-82.3%		8.55	-62.8%		9.24	-57.5%
rc_205.4.txt	27	270		10.79	-82.9%		8.89	-89.2%		7.75	-81.1%
rc_202.3.txt	28	280		14.64	-87.3%		8.37	-89.3%		11.63	-84.3%
rc_208.2.txt	28	280		21.19	-75.7%		9.51	-49.6%		14.94	-48.9%
rc_207.2.txt	30	300		17.06	-66.9%		12.63	-90.2%		17.09	-85.8%
rc_201.3.txt	31	310		21.73	-84.6%		9.57	-71.3%		14.21	-64.0%
rc_202.1.txt	32	320		24.36	-86.0%		14.89	-81.5%		19.19	-85.2%
rc_203.2.txt	32	320		20.83	-67.3%		16.69	-36.1%		17.16	-32.5%
rc_204.2.txt	32	320		30.05	-64.3%		17.42	-82.3%		29.51	-82.0%
rc_207.3.txt	32	320		21.59	-80.8%		16.52	-82.9%		21.54	-83.5%
rc_207.1.txt	33	330		26.41	-88.1%		25.58	-91.2%		29.99	-88.8%
rc_205.3.txt	34	340		23.66	-66.1%		28.71	-70.8%		25.41	-57.9%
rc_208.3.txt	35	350		40.79	-79.7%		35.01	-88.7%		35.77	-89.4%
rc_203.3.txt	36	360		43.70	-79.7%		27.84	-81.7%		32.68	-82.2%
rc_206.2.txt	36	360		51.46	-83.5%		27.81	-75.7%		33.62	-72.3%
rc_206.4.txt	37	370		43.46	-83.3%		24.12	-71.9%		30.42	-71.0%
rc_208.1.txt	37	370		50.12	-75.8%		38.22	-87.3%		46.25	-92.8%
rc_204.1.txt	45	450		95.62	-50.2%		87.72	-57.2%		115.32	-54.6%
<i>Prosečno</i>			323607%	22.39	-73.8%	660875%	16.71	-66.2%	1375953%	20.25	-62.1%

**Tabela 5.10.** Prosečne vreme realizacije algoritama za slučaj Langevin-ovog skupa benčmark instanci

Instanca	n	LB	m=2			m=5			m=10		
			$\Delta B\&B$ [%]	VNS [s]	$\Delta RdVNS$ [%]	$\Delta B\&B$ [%]	VNS [s]	$\Delta RdVNS$ [%]	$\Delta B\&B$ [%]	VNS [s]	$\Delta RdVNS$ [%]
N20ft204.dat	19	0	1218%	1.80	-82.8%	489650%	1.30	-74.8%	404670%	1.70	-63.6%
N20ft204new.txt	19	0	1041%	2.10	-86.8%	442839%	1.43	-76.4%	407550%	1.69	-62.5%
N20ft205.dat	19	0	1133%	1.84	-77.2%	492197%	1.26	-53.9%	639436%	1.81	-42.6%
N20ft206.dat	19	0	546%	1.54	-79.6%	476992%	1.36	-74.4%	396057%	2.11	-64.1%
N20ft207.dat	19	0	1446%	2.03	-81.2%	502912%	1.32	-73.0%	412676%	1.91	-65.5%
N20ft208.dat	19	0	306%	1.87	-86.1%	502940%	1.35	-69.9%	436870%	1.81	-54.8%
N20ft209.dat	19	0	925%	1.21	-81.7%	632154%	1.10	-68.9%	479914%	1.57	-59.3%
N20ft210.dat	19	0	353%	1.70	-79.7%	580645%	1.20	-68.6%	509096%	1.62	-45.0%
N20ft301.dat	19	0	935%	2.01	-77.7%	550023%	1.25	-58.9%	484474%	2.02	-47.9%
N20ft302.dat	19	0	1002%	1.81	-83.6%	557844%	1.23	-74.1%	416494%	1.95	-62.7%
N20ft303.dat	19	0	1052%	1.95	-83.6%	699083%	1.17	-70.9%	552777%	1.58	-54.3%
N20ft304.dat	19	0	4042%	2.17	-84.6%	553505%	1.20	-72.5%	90847%	1.90	-62.5%
N20ft305.dat	19	0	912%	2.41	-83.3%	439513%	1.64	-69.9%	462779%	1.70	-38.1%
N20ft306.dat	19	0	340%	1.79	-82.2%	736373%	1.34	-73.6%	469447%	1.66	-53.5%
N20ft307.dat	19	0	703%	2.70	-86.2%	484164%	1.43	-72.9%	346761%	1.84	-57.8%
N20ft308.dat	19	0	341%	1.85	-85.8%	536642%	1.34	-72.9%	455997%	1.80	-58.0%
N20ft309.dat	19	0	1098%	2.01	-84.8%	679230%	1.22	-70.6%	563894%	1.67	-59.7%
N20ft310.dat	19	0	563%	1.85	-83.0%	636977%	1.02	-60.8%	430663%	1.59	-47.5%
N20ft401.dat	19	0	2093%	2.36	-76.7%	493937%	1.54	-56.3%	333190%	2.19	-56.8%
N20ft402.dat	19	0	512%	2.69	-88.2%	502472%	1.29	-75.4%	301811%	1.82	-57.5%
N20ft403.dat	19	0	769%	2.33	-76.5%	883669%	1.01	-67.0%	462245%	1.68	-52.7%
N20ft404.dat	19	0	975%	3.27	-83.8%	442872%	1.54	-76.8%	494046%	1.80	-60.0%
N20ft405.dat	19	0	878%	3.20	-83.3%	470902%	1.45	-69.3%	375915%	2.10	-47.8%
N20ft406.dat	19	0	889%	2.98	-87.1%	570634%	1.12	-68.6%	30369%	1.73	-56.2%
N20ft407.dat	19	0	1202%	2.04	-84.3%	399335%	1.61	-77.2%	351517%	2.19	-61.7%
N20ft408.dat	19	0	881%	2.57	-85.7%	548142%	1.31	-68.1%	448121%	1.76	-46.6%
N20ft409.dat	19	0	677%	2.34	-86.6%	499515%	1.45	-72.0%	467654%	1.84	-53.2%
N20ft410.dat	19	0	1032%	1.79	-80.8%	731481%	1.07	-64.1%	540889%	1.82	-54.8%
N40ft201.dat	39	0		26.01	-76.4%		11.88	-63.9%		13.90	-51.7%
N40ft202.dat	39	0		26.70	-80.7%		11.86	-64.2%		15.89	-52.2%
N40ft203.dat	39	0		26.68	-81.4%		11.74	-70.0%		14.93	-70.5%
N40ft204.dat	39	0		41.06	-83.5%		13.10	-74.1%		15.24	-62.7%
N40ft205.dat	39	0		20.95	-82.0%		16.97	-75.3%		17.87	-63.1%
N40ft206.dat	39	0		21.33	-77.0%		12.58	-67.4%		15.44	-56.4%
N40ft207.dat	39	0		32.67	-77.1%		11.40	-69.1%		15.47	-70.4%
N40ft208.dat	39	0		31.87	-76.7%		12.78	-63.4%		13.12	-50.3%
N40ft209.dat	39	0		29.42	-82.5%		11.14	-51.4%		17.92	-59.8%
N40ft210.dat	39	0		37.33	-79.5%		14.01	-76.6%		16.05	-62.1%
N40ft301.dat	39	0		29.33	-79.2%		11.11	-70.4%		18.74	-69.3%
N40ft302.dat	39	0		33.28	-80.9%		13.27	-68.9%		16.75	-59.4%
N40ft303.dat	39	0		25.25	-77.7%		13.04	-70.6%		13.83	-57.0%
N40ft304.dat	39	0		45.27	-78.8%		16.46	-81.6%		13.90	-66.2%
N40ft305.dat	39	0		26.19	-81.3%		16.30	-71.3%		17.19	-43.6%
N40ft306.dat	39	0		24.50	-81.8%		14.26	-68.5%		14.84	-40.6%

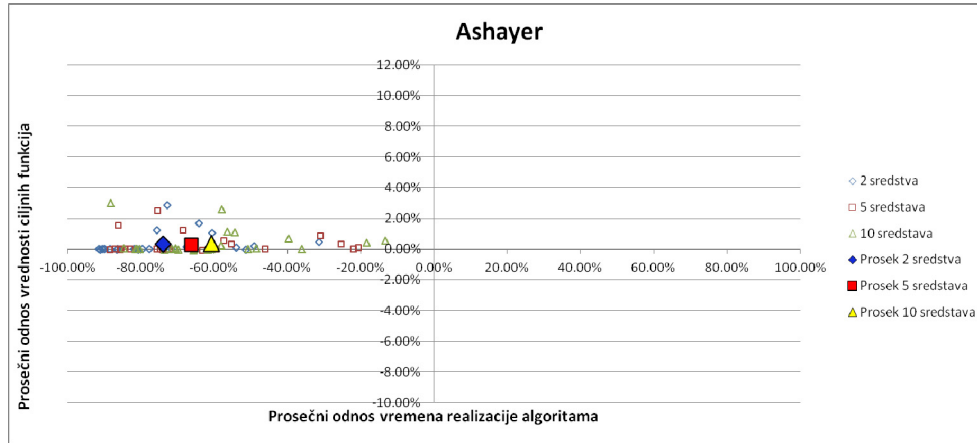
**Tabela 5.10.** Prosečne vreme realizacije algoritama za slučaj Langevin-ovog skupa benčmark instanci (*nastavak*)

Instanca	n	LB	m=2			m=5			m=10		
			$\Delta$ B&B [%]	VNS [s]	$\Delta$ RdVNS [%]	$\Delta$ B&B [%]	VNS [s]	$\Delta$ RdVNS [%]	$\Delta$ B&B [%]	VNS [s]	$\Delta$ RdVNS [%]
N40ft307.dat	39	0		42.94	-78.2%		14.11	-74.3%		15.01	-67.0%
N40ft308.dat	39	0		36.23	-79.4%		14.93	-62.2%		14.55	-49.7%
N40ft309.dat	39	0		33.87	-79.4%		15.22	-73.1%		16.10	-61.8%
N40ft401.dat	39	0		28.22	-76.8%		14.99	-74.4%		13.21	-54.0%
N40ft402.dat	39	0		25.83	-77.3%		14.23	-66.9%		15.15	-53.0%
N40ft403.dat	39	0		60.02	-90.6%		16.41	-70.3%		15.46	-61.9%
N40ft404.dat	39	0		46.17	-81.9%		14.57	-81.7%		16.35	-73.7%
N40ft405.dat	39	0		38.13	-86.5%		16.97	-74.7%		21.55	-67.0%
N40ft406.dat	39	0		32.69	-81.3%		10.63	-48.4%		14.20	-49.5%
N40ft407.dat	39	0		56.83	-86.6%		13.98	-78.8%		14.88	-69.4%
N40ft408.dat	39	0		35.78	-70.6%		14.40	-66.5%		15.38	-45.7%
N40ft409.dat	39	0		43.94	-84.7%		13.74	-68.7%		14.94	-62.7%
N40ft410.dat	39	0		40.88	-79.7%		13.22	-75.7%		12.98	-66.7%
N60ft201.dat	59	0		171.00	-71.8%		44.53	-65.5%		47.66	-43.3%
N60ft202.dat	59	0		113.77	-65.1%		55.37	-60.0%		51.83	-51.3%
N60ft203.dat	59	0		128.88	-63.3%		49.20	-56.6%		50.14	-58.1%
N60ft204.dat	59	0		122.62	-75.4%		58.86	-63.2%		57.68	-53.2%
N60ft205.dat	59	0		169.19	-82.2%		43.93	-62.0%		51.89	-58.3%
N60ft206.dat	59	0		142.84	-77.3%		55.04	-67.4%		58.77	-59.6%
N60ft207.dat	59	0		151.43	-74.0%		57.12	-60.3%		54.09	-49.6%
N60ft208.dat	59	0		125.10	-70.1%		45.82	-50.6%		46.43	-35.7%
N60ft209.dat	59	0		114.27	-68.4%		49.63	-59.4%		46.10	-47.9%
N60ft210.dat	59	0		177.66	-70.9%		50.90	-48.6%		49.55	-34.7%
N60ft301.dat	59	0		191.96	-86.0%		46.75	-49.6%		45.73	-45.6%
N60ft302.dat	59	0		227.44	-75.9%		49.49	-71.0%		56.65	-61.6%
N60ft303.dat	59	0		154.03	-69.4%		54.62	-76.9%		46.67	-57.9%
N60ft304.dat	59	0		128.88	-78.3%		58.43	-70.8%		64.25	-61.3%
N60ft305.dat	59	0		161.35	-82.3%		53.73	-69.2%		54.26	-67.2%
N60ft306.dat	59	0		147.71	-72.1%		62.50	-69.9%		67.48	-65.1%
N60ft307.dat	59	0		134.44	-64.1%		49.06	-54.3%		45.90	-38.8%
N60ft308.dat	59	0		177.52	-65.1%		57.26	-63.6%		58.04	-53.7%
N60ft309.dat	59	0		150.55	-69.7%		56.81	-70.1%		51.06	-60.5%
N60ft310.dat	59	0		210.73	-72.7%		71.15	-64.5%		43.50	-46.3%
N60ft401.dat	59	0		186.79	-80.1%		54.96	-60.4%		48.21	-54.1%
N60ft402.dat	59	0		156.21	-73.0%		61.43	-72.0%		49.49	-65.0%
N60ft403.dat	59	0		191.12	-87.6%		62.29	-62.0%		42.97	-58.8%
N60ft404.dat	59	0		223.05	-84.9%		50.99	-68.0%		68.39	-70.5%
N60ft405.dat	59	0		150.42	-73.5%		47.23	-75.6%		49.83	-57.5%
N60ft406.dat	59	0		167.88	-80.7%		60.64	-68.6%		53.47	-55.4%
N60ft407.dat	59	0		248.63	-81.4%		44.27	-56.8%		48.28	-48.2%
N60ft408.dat	59	0		176.63	-74.1%		67.44	-68.7%		58.53	-51.8%
N60ft409.dat	59	0		136.52	-81.3%		59.11	-72.5%		52.83	-58.9%
N60ft410.dat	59	0		251.04	-82.0%		57.84	-52.3%		52.86	-44.5%
<i>Prosečno</i>			995%	171.37	-75.8%	554880%	55.38	-64.1%	420220%	52.53	-54.0%

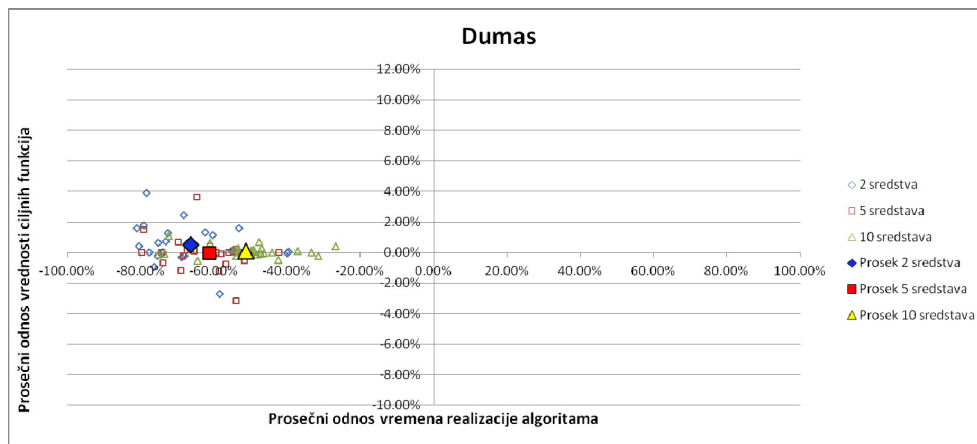
Velika kompleksnost hečTRPTW problema je uzrok što je od 153 instance malih dimenzija, koje su rešavane do optimalnosti, B&B algoritma uspeo da optimalno reši samo 45 instanci. Sve vrednosti ciljnih funkcija optimalno rešenih instanci su prikazane podebljanim ciframa u tabelama. Treba napomenuti da u onim instancama u kojima B&B ne uspeva da reši problem do optimalnosti, vrednost u tabeli predstavlja vrednost ciljne funkcije najboljeg rešenja koje je pronađeno do trenutka dostizanja limita od 1GB RAM-a za formiranje drveta rešenja. Iz tog razloga, od odnosa vrednosti ovog rešena i vrednosti rešenja VNS algoritma zavisi da li će vrednost biti pozitivna ili negativna.

Na osnovu rezultata iz tabela 5.3 do 5.6 uočljivo je da je od 153 instance malih dimenzija B&B algoritma nadmašio VNS algoritma, u odnosu na kriterijum vrednosti ciljne funkcije, u samo šest instanci, pri čemu su najveća odstupanja 1.16% i 0.76% za instance N20ft305.dat i N20ft401.dat, respektivno, iz Langavin-ovog skupa instanci, dok za ostale instance ne prelazi 0.4%. Za sve ostale instance VNS algoritam je dao rešenja istog, ili boljeg, kvaliteta u odnosu na B&B algoritam, pri čemu je primetno da sa porastom kompleksnosti problema, tačnije sa povećanjem broja servisera u floti, raste i broj rešenja u kojima je VNS nadmašio B&B algoritam.

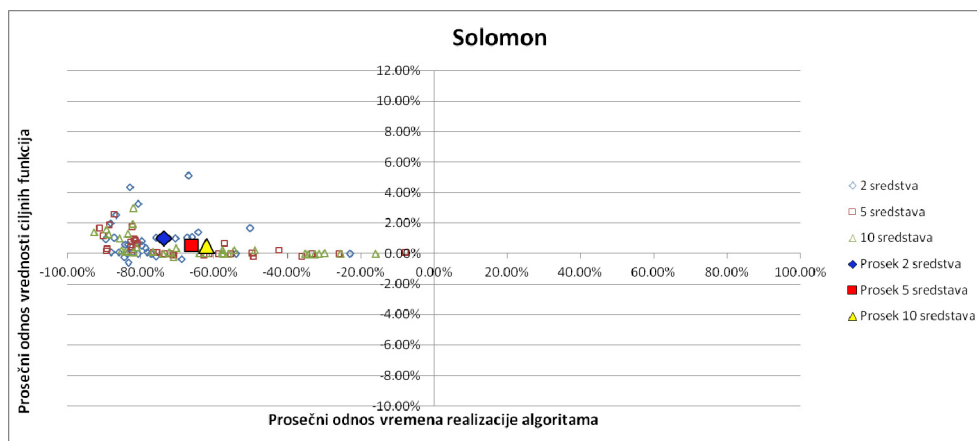
Slikoviti relativni odnos performansi VNS i RdVNS algoritma za slučajeve Ashayer-ovog, Dumas-ovog, Solomon-ovog i Langevin-ovog skupa benčmark instanci je prikazan na slikama 5.13-5.16, respektivno. Svaka tačka na grafiku predstavlja odnos vrednosti parametara performansi algoritma tako što je na X-osi grafika predstavljena relativna razlika vremena realizacije RdVNS algoritma u odnosu na VNS algoritma, a na Y-osi je predstavljena ova razlika za srednju vrednost ciljne funkcije. Na osnovu podataka iz tabela 5.3 do 5.6, ali i na osnovu slika 5.13-5.16, može se uočiti da se sa povećanjem kompleksnosti problema relativna razlika RdVNS i VNS algoritma u odnosu na vrednost ciljne funkcije smanjuje. Razlog za ovakav šablon leži u činjenici da se sa povećanjem kompleksnosti problema povećava i broj rešenja koja su na relativno malom rastojanju jedno od drugog po pitanju vrednosti ciljne funkcije. Samim tim povećava se verovatnoća da će RdVNS algoritam u toku realizacije naići na neko od rešenja koja su, po vrednosti ciljne funkcije, bliska VNS algoritmu.



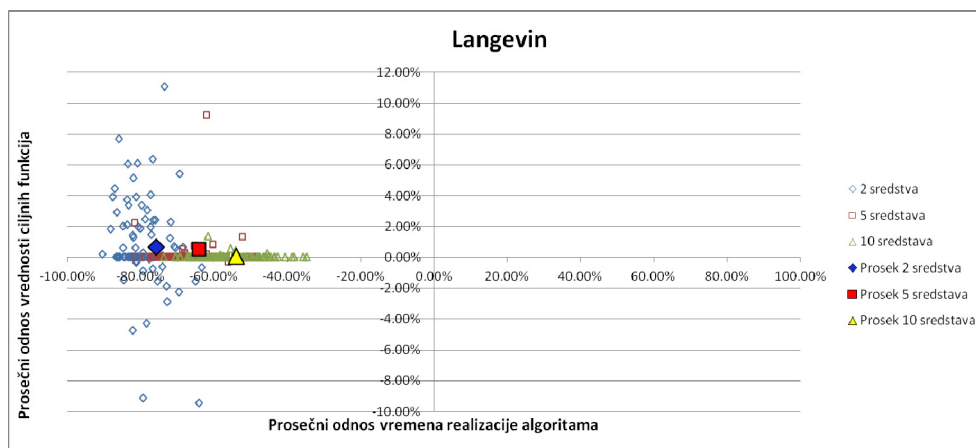
Slika 5.13. Relativna razlika parametara VNS i RdVNS algoritma za slučaj Ashayer-ovog skupa benčmark instanci



Slika 5.14. Relativna razlika parametara VNS i RdVNS algoritma za slučaj Dumas-ovog skupa benčmark instanci



Slika 5.15. Relativna razlika parametara VNS i RdVNS algoritma za slučaj Solomon-ovog skupa benčmark instanci



**Slika 5.16.** *Relativna razlika parametara VNS i RdVNS algoritma za slučaj Langevin-ovog skupa benčmark instanci*

Sa slika 5.13-5.16 se može primetiti da je relativno odstupanje RdVNS algoritma u slučaju Langevin-ovog skupa instanci veće nego u ostalim. Međutim, ovo je samo posledica toga što su vrednosti ciljnih funkcija, za rešenja u slučaju ovog skupa problema, manje nego u slučaju ostalih skupova problema. Ovo znači da će se razlika u vrednosti ciljnih funkcija deliti sa manjom vrednošću, te je krajnji rezultat veći.

Što se tiče vremenskog aspekta u poređenju RdVNS i VNS algoritma, prezentovani podaci ukazuju da je vreme realizacije RdVNS kraće za više od 50% u proseku. Najveće skraćivanje realizacije je u slučaju instanci sa dva servisera u floti, a najmanje u slučaju 10 servisera. Drugim rečima, najveće vremenske uštede u realizaciji algoritma se postižu u rešenjima čija je struktura takva da je veliki broj zadataka dodeljen malom broju servisera, čime se naglašava uticaj redukcije  $\mathcal{N}_{Or-opt}$  okolina na ukupno smanjenje vremena realizacije. Međutim, kako redukcija  $\mathcal{N}_{Or-opt}$  okolina isključuje iz pretrage najveći deo prostora rešenja, u poređenju sa drugim strategijama redukcije, to je jasno da se njenom primenom iz razmatranja eliminiše najveći broj potencijalno dobrih rešenja. Ovo je i jedan od razloga zašto se sa povećanjem broja sredstava u floti smanjuje razlika u vrednosti ciljnih funkcija. Iz tog razloga, sa slika 5.13-5.16, ali i iz tabela 5.7 do 5.10 može se primetiti da su relativna odstupanja vrednosti ciljnih funkcija veća sa smanjenjem broja servisera u floti.

## **6. Primena predloženih modela u upravljanju pretovarnim sredstvima za istovar šljunka iz potisnica**

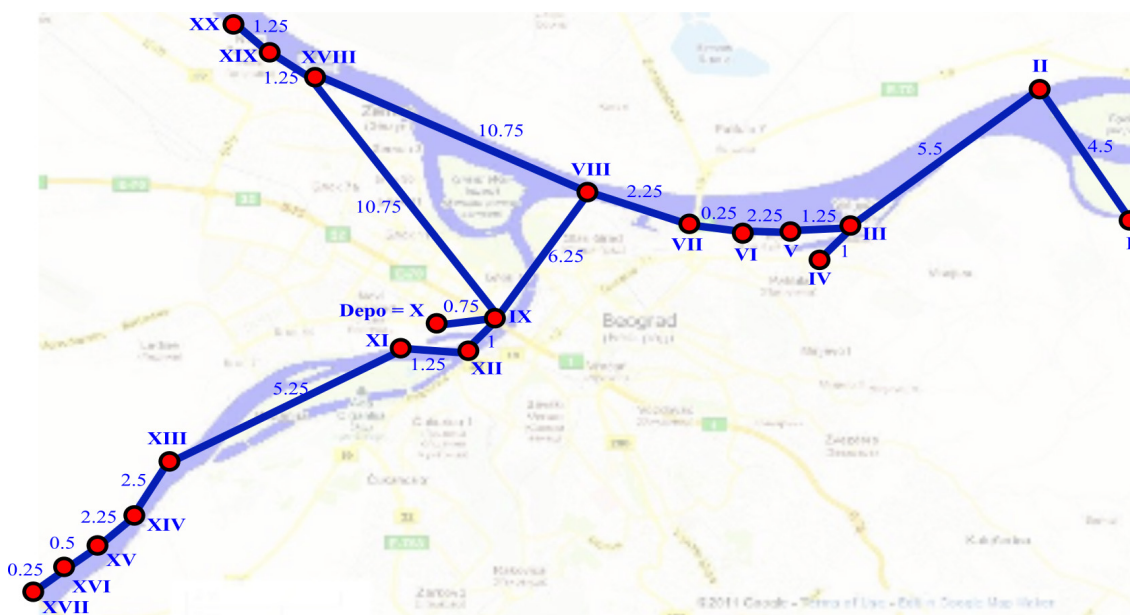
---

Kao što je navedeno u poglavlju 3 modeli za rešavanje heTRPTW-a se mogu koristiti za upravljanje nizom pretovarnih, ili njima veoma bliskih, sistema, pri čemu je za neke potrebno izvršiti izvesni stepen prilagođavanja. Jedan od pretovarnih sistema na koje se predloženi modeli mogu primeniti bez ikakvih modifikacija je i sistem specijalnih plovnih sredstava za pretovar šljunka iz potisnica, predstavljen u poglavlju 3.2.1. Iz tog razloga u ovom poglavlju će na primeru tog sistema biti prikazana primena predloženih modela na operativno upravljanje pretovarnim sistemima sa dinamičkom pojavom zadataka. Treba naglasiti da je statička verzija ovog problema prvi put posmatrana u literaturi u radu Vidović i Vukadinović (2006) kada su ga autori označili kao problem alokacije pretovarnih sredstava - HDAP (eng. Handling Devices Allocation Problem) i predstavili više konceptualnih okvira za njegovu matematičku formulaciju i heurističko rešavanje. Iako se HDAP odnosi na varijantu u kojoj su svi



zadaci prisutni u sistemu na početku planskog perioda, u daljem delu ovog poglavlja HDAP-om će se referencirati na varijantu sa dinamičkom pojavom zadataka, tj. varijanta problema koja je ekvivalentna heTRPTW-u.

Kao primer posmatranog distributivnog područja kompanije koja se bavi proizvodnjom i distribucijom šljunka uzeta je teritorija Beograda na kojoj se na dve reke nalazi oko 20 deponija na kojima se vrši odlaganje šljunka dopremljenog, uglavnom, iz donjeg toka reke Dunav. Približne lokacije deponija su prikazane na slici 6.1, pri čemu je ujedno, u formi grafa, prikazana i njihova prostorna povezanost. Koeficijent uz granu grafa predstavlja prostornu razdaljinu u kilometrima između dve deponije, a deponije su identifikovane na osnovu rimskog broja, koji je prikazan uz svaki čvor grafa. Deponija X je ujedno i depo u kome se pretovarna sredstva nalaze na početku planskog perioda. Na osnovu grafa distributivnog područja primenom Floyd–Warshall-ovog algoritma dobijaju se najkraća rastojanja između svih parova čvorova grafa, koja su prikazana u tabeli 6.1.



**Slika 6.1.** Layout posmatranog distributivnog područja

Kao i u slučaju testiranja heTRPTW-a i u slučaju HDAP-a se posmatraju flote pretovarnih sredstava koje su sačinjene od dva, pet i deset sredstava. Međutim, kako se ovde radi o konkretnim sredstvima to se njihova individualizacija ne vrši primenom korektivnih koeficijenata, već se koriste konkretne vrednosti relevantnih karakteristika. U tabeli 6.2. su prikazane relevantne karakteristike upravljanih

pretovarnih sredstva, pri čemu treba naglasiti da u flotu od dva sredstva ulaze sredstva sa rednim brojevima 1 i 2, u flotu od pet sredstava sredstva sa rednim brojevima 1, 2, 3, 4 i 5, a u flotu od deset sredstava sva sredstva iz tabele. Kolona „R.B.“ tabele 6.1. sadrži informacije o rednom broju sredstva. U koloni „Proizvodnost“ je informacija o količini šljunka koju sredstvo može da pretovari iz potisnice na deponiju na obali u toku jednog sata rada, što, uz količinu šljunka koju je potrebno pretovariti, definiše vreme zadržavanja pretovarnih sredstva u lokacijama deponija. U koloni „Brzina kretanja“ je data vrednost o prosečnoj brzini kretanja pretovarnih sredstava između lokacija deponija.

**Tabela 6.1. Najkraća rastojanja između deponija šljunka[km]**

Od	Do																				
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
0	0	23	18.5	13	14	11.75	9.5	9.25	7	0.75	1.5	3	1.75	8.25	10.75	13	13.5	13.75	11.5	12.75	14
1	23	0	4.5	10	11	11.25	13.5	13.75	16	22.25	23	24.5	23.25	29.75	32.25	34.5	35	35.25	26.75	28	29.25
2	18.5	4.5	0	5.5	6.5	6.75	9	9.25	11.5	17.75	18.5	20	18.75	25.25	27.75	30	30.5	30.75	22.25	23.5	24.75
3	13	10	5.5	0	1	1.25	3.5	3.75	6	12.25	13	14.5	13.25	19.75	22.25	24.5	25	25.25	16.75	18	19.25
4	14	11	6.5	1	0	2.25	4.5	4.75	7	13.25	14	15.5	14.25	20.75	23.25	25.5	26	26.25	17.75	19	20.25
5	11.75	11.25	6.75	1.25	2.25	0	2.25	2.5	4.75	11	11.75	13.25	12	18.5	21	23.25	23.75	24	15.5	16.75	18
6	9.5	13.5	9	3.5	4.5	2.25	0	0.25	2.5	8.75	9.5	11	9.75	16.25	18.75	21	21.5	21.75	13.25	14.5	15.75
7	9.25	13.75	9.25	3.75	4.75	2.5	0.25	0	2.25	8.5	9.25	10.75	9.5	16	18.5	20.75	21.25	21.5	13	14.25	15.5
8	7	16	11.5	6	7	4.75	2.5	2.25	0	6.25	7	8.5	7.25	13.75	16.25	18.5	19	19.25	10.75	12	13.25
9	0.75	22.25	17.75	12.25	13.25	11	8.75	8.5	6.25	0	0.75	2.25	1	7.5	10	12.25	12.75	13	10.75	12	13.25
10	1.5	23	18.5	13	14	11.75	9.5	9.25	7	0.75	0	3	1.75	8.25	10.75	13	13.5	13.75	11.5	12.75	14
11	3	24.5	20	14.5	15.5	13.25	11	10.75	8.5	2.25	3	0	1.25	5.25	7.75	10	10.5	10.75	13	14.25	15.5
12	1.75	23.25	18.75	13.25	14.25	12	9.75	9.5	7.25	1	1.75	1.25	0	6.5	9	11.25	11.75	12	11.75	13	14.25
13	8.25	29.75	25.25	19.75	20.75	18.5	16.25	16	13.75	7.5	8.25	5.25	6.5	0	2.5	4.75	5.25	5.5	18.25	19.5	20.75
14	10.75	32.25	27.75	22.25	23.25	21	18.75	18.5	16.25	10	10.75	7.75	9	2.5	0	2.25	2.75	3	20.75	22	23.25
15	13	34.5	30	24.5	25.5	23.25	21	20.75	18.5	12.25	13	10	11.25	4.75	2.25	0	0.5	0.75	23	24.25	25.5
16	13.5	35	30.5	25	26	23.75	21.5	21.25	19	12.75	13.5	10.5	11.75	5.25	2.75	0.5	0	0.25	23.5	24.75	26
17	13.75	35.25	30.75	25.25	26.25	24	21.75	21.5	19.25	13	13.75	10.75	12	5.5	3	0.75	0.25	0	23.75	25	26.25
18	11.5	25	20.5	15	16	13.75	11.5	11.25	9	10.75	11.5	13	11.75	18.25	20.75	23	23.5	23.75	0	1.25	2.5
19	12.75	26.25	21.75	16.25	17.25	15	12.75	12.5	10.25	12	12.75	14.25	13	19.5	22	24.25	24.75	25	1.25	0	1.25
20	14	27.5	23	17.5	18.5	16.25	14	13.75	11.5	13.25	14	15.5	14.25	20.75	23.25	25.5	26	26.25	2.5	1.25	0

Treba napomenuti da se u praksi uzvodne i nizvodne brzine kretanja mogu znatno razlikovati, ali je ovde usvojena srednja brzina kretanja ovih sredstava kao merodavna vrednost jer sredstva veći deo vremena provode u realizaciji pretovarnih operacija. Pored toga pretpostavljeno je da su ovo optimalne brzine u smislu minimizovanja potrošnje goriva, odnosno da uvek postoji mogućnost da se intenzivnijim angažovanjem rada pogonskih motora omogući kretanje sredstava brzinama iz tabele. Kao što se može videti iz kolone „Raspoloživost sredstava“ pretpostavlja se da su sva sredstva raspoloživa od početka planskog perioda.

**Tabela 6.2.** Karakteristike pretovarnih sredstva relevantne za njihovo upravljanje

R.B.	Proizvodnost [t/h]	Brzina kretanja [km/h]	Raspoloživost sredstva
1	100	6	0.00
2	150	5	0.00
3	200	4	0.00
4	150	5	0.00
5	200	4	0.00
6	150	5	0.00
7	200	4	0.00
8	100	6	0.00
9	150	5	0.00
10	200	4	0.00

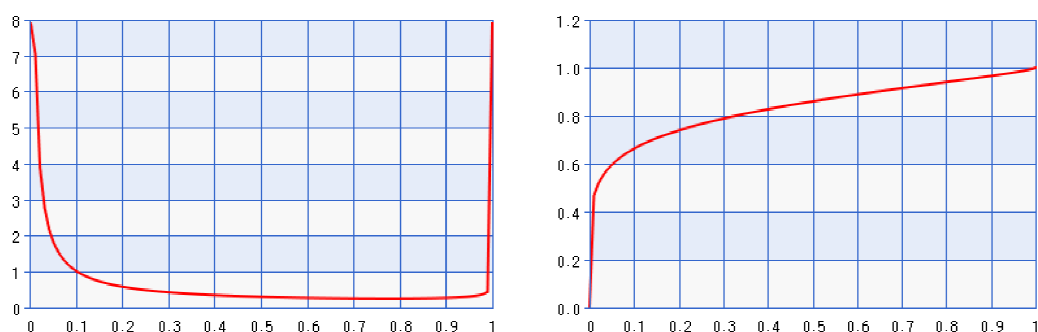
Što se tiče karakteristika pretovarnih zadataka, pretpostavljeno je da se zadaci javljaju sa jednakom vjerojatnošću na svakoj od lokacija deponija i da se po pitanju količine šljunka koji se pretovara javljaju zadaci koji zahtevaju pretovar 750t, 1000t i 1250t šljunka, pri čemu je jednaka vjerojatnoća da se pojavi bilo koja od ove tri količine.

U primerima HDAP problema posmatrani su uslovi malog, srednjeg i velikog opterećenja flote sredstava kojima odgovara korišćenje 2, 5 i 10 pretovarnih sredstava u floti za isti skup zadataka u planskom periodu. Iz tog razloga flota od 5 sredstava je posmatrana kao flota koja je optimalno dimenzionisana za opslugu u planskom periodu, što znači da je prosečno vreme potrebno sredstvima u ovoj floti za realizaciju zadatka jednako očekivanom vremenu između pojave zadatka. U slučaju ovakvog dolaska potisnica na opslugu šanse da će potisnice čekati na pretovarna sredstva su jednake šansama da će pretovarna sredstva čekati na pojavu potisnica. U slučaju velikog opterećenja sistema veća je vjerojatnoća da će potisnice čekati na sredstva, dok je u slučaju malog opterećenja znatno vjerojatnije da će potisnice doći u sistem u kome ih pretovarna sredstva već čekaju u lokaciji deponije.

Na osnovu podataka o najkraćim rastojanjima između lokacija deponija (tabela 6.1) i prosečnim brzinama kretanja sredstava (tabela 6.2) dobija se da je prosečno rastojanje između lokacija 13.31 km, kao i da je približna prosečna brzina kretanja pretovarnog sredstva iz flote od 5 sredstava 4.8 km/h, odnosno da očekivano vreme potrebno sredstvu da pređe od jedne do druge lokacije depoa iznosi 2.77h. Sa druge strane, prosečna proizvodnost sredstava je 160t/h, što uz prosečnu količinu šljunka koja stiže za pretovar od 1000t, znači da je očekivano vreme zadržavanja sredstva u lokaciji deponije 6.25h. Kako realizacija jednog zadatka uključuje i realizaciju kretanja između lokacija deponija i sam pretovar šljunka, to je očekivano vreme realizacije zadatka jednim „prosečnim“ sredstvom 9.02h. Kako u prosečnom opterećenju flota broji 5

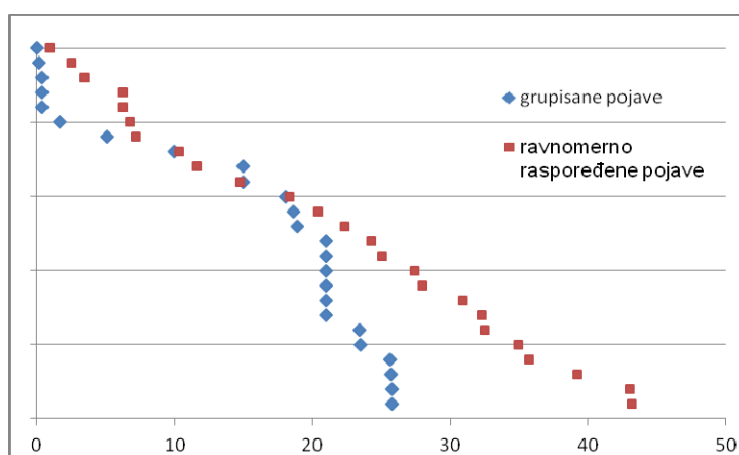
sredstva to je jasno da je očekivano vreme između pojave zadatka jednako  $9.02/5=1.8h$ .

Pored različitog opterećenja upravljanih pretovarnih sistema modeli za rešavanje HDAP-a su testirani i na različite šablone pojave zadatka. Naime, vreme između pojave zadatka je generisano u skladu sa dve raspodele: ravnomernom i beta raspodelom sa parametrima  $\alpha = 0.15$  i  $\beta = 0.75$ . Izgled beta raspodele sa korišćenim parametrima i odgovarajuće funkcije verovatnoća je predstavljena na slici 6.2.



**Slika 6.2.** Izgled raspodele i funkcije verovatnoća korišćene beta raspodela

U slučaju dolaska zahteva po ravnomernoj raspodeli dobija se negrupisani, tj. ravnomerno raspoređen, dolazak zadatka, dok se u slučaju beta raspodele zadaci javljaju u grupama u vrlo kratkom vremenskom roku, nakon čega sledi duži vremenski period u kome se zadaci ne javljaju. Primer dolaska 25 zadatka po ravnomernoj i beta raspodeli je dat na slici 6.3 gde se jasno može uočiti razlika u šablonima pojave zadatka u smislu grupisane pojave zadatka.



**Slika 6.3.** Primer ravnomerno raspoređene i grupisane pojave zadatka

Obe korišćene raspodele za opisivanje vremena između pojave zadatka se primenjuju na intervalima<sup>14</sup>, pri čemu su i jedan i drugi interval sa leve strane ograničeni vrednošću 0.00h, jer je moguće da se zadaci jave istovremeno. Kako bi ravnomerno raspodeljena slučajna promenljiva ( $Y$ ), kojoj je desna granica ( $a$ ) 0.00h, imala očekivanu vrednost ( $E(Y)$ ) 1.8h, njena leva granica ( $b$ ) mora imati vrednost 3.6h, jer je  $E(Y) = \frac{a+b}{2}$ . U slučaju slučajne promenljive ( $Z$ ), koja je opisana beta raspodelom, očekivana vrednost je  $E(Z) = c + (d - c) \frac{\alpha}{\alpha + \beta}$ , gde su  $c$  i  $d$ , leva i desna granica vrednosti  $Z$ , respektivno. Kako je  $E(Z) = E(Y) = 1.8h$ , i kako je  $c=0.00h$  to se lako dobija da je gornja granica vrednosti  $Z$ , tj.  $d$ , jednaka 10.84h.

Za ovako definisane raspodele vremena između pojava zadataka, lokacije deponija i količina koje je potrebno pretovariti na slučajan način je definisano tri grupe testnih primera sa različitim brojem zadataka u planskom periodu. Konkretno, generisane su instance od 10, 25 i 50 pretovarnih zadataka. Kako, usled uticaja dinamičkog javljanja zadataka, presudnu ulogu u upravljanju posmatranim sistemom imaju vremenski trenuci pojave zadataka, to je pri formiranju upravljačkih problema za svaki par (lokacija depoa – količina za pretovar) u skupu zadataka instance problema, generisana slučajna promenljiva po ravnomernoj i beta raspodeli za definisanje trenutka pojave zadatka. Na taj način se za jedan prostorno-količinski raspored zadataka po distributivnoj zoni dobijaju dva instance problema. Prva u kome su trenuci pojave ravnomerno raspoređeni, a druga u kome su raspoređeni po beta raspodeli. Primer ovakvog generisanja za slučaj deset zadataka je dat u tabeli 6.3, dok su svi rešavani problemi dostupni na internet adresi [https://drive.google.com/folderview?id=0B\\_xZi\\_96RclBNEppSU1hZlhiYnM](https://drive.google.com/folderview?id=0B_xZi_96RclBNEppSU1hZlhiYnM).

HDAP je rešavan istim modelima upravljanja kao i hetTRPTW, tj. optimalno rešenje je traženo primenom B&B algoritma implementiranog u CPLEX 12.2 softverskom paketu, ali i predloženim VNS i RdVNS algoritmima. Kako redukovani VNS algoritam, tj. redukcija  $\mathcal{N}_{swap}$  okoline, podrazumeva primenu parametra  $\alpha_{crit}$ , to je i u ovom slučaju, pre realizacije algoritama, sprovedena preliminarna analiza za slučajno odabrane 2., 3., 6., i 10. instancu sa 25 zadataka za ravnomernu i beta raspodelu između

---

<sup>14</sup> Ravnomerna raspodela je specijalni slučaj beta raspodele, sa parametrima  $\alpha = \beta = 1$ .

trenutaka dolaska. Analiza dobijenih rezultata o vrednosti ciljne funkcije i vremenu realizacije algoritma dobija se da je i u ovom slučaju njihov najbolji odnos za vrednost parametra od 0,3, tj. 30% širine planskog perioda.

**Tabela 6.3.** Podaci o zadacima u toku planskog perioda za slučaj 10 zadataka

Instanca	Zadatak	Trenutak pojave zadataka		Čvor	Količina	Instanca	Zadatak	Trenutak pojave zadataka		Čvor	Količina
		beta	ravnomerna					beta	ravnomerna		
1	1	1.69	3.31	XVI	750	6	1	0	0.26	XIX	1000
	2	9.35	6.6	XII	750		2	0	3.28	IV	1000
	3	15.85	7.1	XVII	1250		3	0.71	4.81	XVII	750
	4	15.85	10.69	XX	750		4	12.29	6.06	XX	1000
	5	15.85	11.6	I	1000		5	12.29	9.8	X	1250
	6	15.85	15.24	XI	1250		6	12.3	10.34	VI	1000
	7	15.99	16.18	XX	1000		7	12.3	10.65	XX	750
	8	15.99	16.93	XIII	1000		8	12.3	11.86	XI	1000
	9	26.81	17.05	XIX	750		9	12.51	12.82	III	750
	10	37.24	17.63	II	750		10	12.51	16.82	XVII	1250
2	1	11.67	2.57	V	1250	7	1	0.8	2.57	VII	1250
	2	11.67	5.65	I	1250		2	0.82	6.19	IV	1000
	3	12	5.95	XI	1000		3	0.83	8.33	XIII	750
	4	12	6.11	IV	1250		4	3.27	9.44	VIII	1000
	5	21.38	6.27	I	750		5	15.12	11.05	XVII	750
	6	21.38	9.25	X	750		6	15.56	11.29	XIII	750
	7	21.85	10.48	XII	1000		7	15.56	14.98	XX	1250
	8	24.88	13.39	XI	1250		8	15.56	18.39	XIV	1250
	9	24.92	13.57	XX	1000		9	15.56	18.68	IV	1000
	10	25.04	17.46	III	750		10	15.56	20.73	I	1000
3	1	3.3	2.7	XVIII	1250	8	1	2.34	3.16	II	750
	2	3.5	2.75	X	750		2	2.34	3.86	III	1250
	3	4.24	3.02	XIV	1000		3	2.34	6.82	XIII	1000
	4	14.88	5.88	XVIII	750		4	2.55	9.02	I	750
	5	14.88	6.61	XI	750		5	2.55	9.97	I	1000
	6	14.88	8.62	III	1000		6	7.06	12.24	XX	1250
	7	14.98	9.6	XI	1250		7	7.34	13.88	XII	1000
	8	21.37	10.88	VIII	1000		8	7.47	17.43	XVII	1250
	9	21.37	13.87	XII	1250		9	7.47	17.44	III	1250
	10	28.98	15	III	1000		10	18.76	19.27	IV	1000
4	1	0	3.76	XX	1250	9	1	2.41	2.19	XIII	1000
	2	0.29	6.35	XI	1000		2	2.73	5.12	X	1250
	3	0.29	7.7	XVIII	750		3	4.77	6.42	XV	750
	4	11.05	7.77	XVIII	750		4	4.97	9.39	XX	1250
	5	11.05	9.57	IV	1000		5	5.35	10.8	XX	750
	6	22.1	12.43	I	1000		6	5.36	11.34	XVIII	1250
	7	22.23	15.25	VI	750		7	6.1	14.92	V	1000
	8	22.25	18.99	IV	750		8	6.29	15.67	XVI	750
	9	22.25	21.16	VI	1250		9	6.33	16.33	III	1000
	10	22.25	22.74	III	1250		10	6.33	19.37	IV	1250
5	1	0.49	1.59	V	1250	10	1	1.63	0.64	XX	1250
	2	0.49	2.58	XVII	750		2	1.63	4.32	V	1250
	3	3.14	6.46	X	1250		3	1.63	8.23	II	1250
	4	3.14	7.82	XV	1250		4	1.63	9.79	VIII	1250
	5	3.73	8.5	XVI	1000		5	1.66	12.39	VII	1250
	6	3.76	9.33	VIII	750		6	1.74	14.73	VI	1000
	7	6.56	10.36	IX	1000		7	11.99	18.03	XIX	1000
	8	6.56	12.9	XVII	1250		8	11.99	18.91	XVI	1000
	9	13.98	13.85	XVI	1250		9	23.1	19.82	XVI	750
	10	25.75	17.25	XV	1000		10	23.1	21.72	XX	750

Kao i u slučaju hetTRPTW-a i HDAP je rešavan na računaru koji radi na Windows XP SP3 operativnom sistemu sa AMD Phenom II 2.61GHz CPU-om sa 2GB RAM. Kodiranje svih algoritma realizovano u programskom jeziku Python 2.5, a svaka instanca problema je rešavana po jednom za slučaj determinističkog B&B algoritama, dok je u slučajevima stohastičkih algoritama (VNS i RdVNS) rešavana po pet puta.

Rezultati realizacije upravljačkih algoritma su prikazani u nastavku poglavlja, pri čemu su rezultati za instance sa ravnomernim vremenom između pojave zadataka dati u tabelama 6.4 i 6.6, a za slučaj beta raspodele 6.5 i 6.7. Tabele 6.4 i 6.5 sadrže podatke o efikasnosti primenjenih algoritma u odnosu na vrednost ciljne funkcije, dok tabele 6.6 i 6.7 sadrže podatke o vremenskom aspektu efikasnosti algoritama.

**Tabela 6.4.** Referentne veličine HDAP ciljne funkcije za slučaj vremena između pojava pretovarnih zadataka prema ravnomernoj raspodeli

n	Inst	m=2				m=5				m=10			
		LB	$\Delta B\&B$ [%]	VNS	$\Delta RdVNS$ [%]	LB	$\Delta B\&B$ [%]	VNS	$\Delta RdVNS$ [%]	LB	$\Delta B\&B$ [%]	VNS	$\Delta RdVNS$ [%]
10	1	61.67	<b>0.00%</b>	165.88	0.00%	46.25	1.45%	56.70	0.36%	46.25	1.85%	47.75	0.00%
	2	68.33	<b>0.00%</b>	210.30	0.08%	51.25	0.00%	71.35	0.35%	51.25	0.37%	56.00	0.00%
	3	66.67	<b>0.00%</b>	195.35	1.75%	50.00	0.46%	63.83	0.00%	50.00	0.79%	52.92	0.00%
	4	65.00	<b>-0.22%</b>	169.97	0.82%	48.75	0.00%	58.95	0.00%	48.75	0.00%	51.25	0.00%
	5	71.67	<b>0.00%</b>	188.31	0.09%	53.75	1.84%	70.29	0.00%	53.75	0.59%	59.67	0.00%
	6	65.00	<b>0.00%</b>	186.49	0.00%	48.75	0.18%	67.26	-0.12%	48.75	-0.30%	55.81	-0.02%
	7	66.67	<b>0.00%</b>	179.19	1.26%	50.00	0.00%	59.96	0.00%	50.00	0.88%	52.04	0.00%
	8	70.00	<b>0.00%</b>	188.78	0.00%	52.50	1.29%	66.75	0.00%	52.50	0.00%	56.05	0.00%
	9	68.33	<b>0.00%</b>	189.31	0.66%	51.25	0.41%	66.60	0.10%	51.25	-0.35%	55.19	-0.35%
	10	71.67	<b>0.00%</b>	199.81	0.00%	53.75	0.00%	71.73	0.00%	53.75	0.00%	60.54	0.00%
25	1	163.33		770.23	0.36%	122.50		149.45	-0.08%	122.50		128.31	-0.11%
	2	163.33		744.80	0.55%	122.50		152.56	0.00%	122.50		130.29	0.00%
	3	183.33		996.67	0.68%	137.50		176.00	0.05%	137.50		146.30	0.11%
	4	160.00		845.90	-0.56%	120.00		151.31	-0.09%	120.00		125.33	0.23%
	5	163.33		809.10	0.76%	122.50		157.71	0.15%	122.50		129.75	0.01%
	6	166.67		837.53	1.88%	125.00		155.86	0.45%	125.00		130.70	0.15%
	7	161.67		765.09	0.12%	121.25		148.61	0.27%	121.25		128.10	0.01%
	8	160.00		886.78	1.11%	120.00		165.01	-0.23%	120.00		130.76	-0.14%
	9	166.67		852.92	0.35%	125.00		153.48	0.07%	125.00		131.84	-0.14%
	10	165.00		870.21	0.50%	123.75		160.03	-0.64%	123.75		131.85	0.00%
50	1	333.33		2717.25	0.98%	250.00		357.27	0.42%	250.00		275.38	-0.10%
	2	331.67		2852.43	0.44%	248.75		317.15	0.18%	248.75		261.14	-0.08%
	3	351.67		3358.43	1.25%	263.75		365.18	0.22%	263.75		290.57	-0.20%
	4	321.67		2647.23	0.01%	241.25		319.36	0.07%	241.25		259.01	0.06%
	5	346.67		3307.83	0.18%	260.00		355.34	-0.13%	260.00		278.37	-0.39%
	6	328.33		2881.84	-0.42%	246.25		313.07	-0.07%	246.25		257.12	-0.04%
	7	343.33		2828.05	1.50%	257.50		313.38	0.10%	257.50		270.75	-0.01%
	8	343.33		3277.49	0.68%	257.50		354.60	-0.02%	257.50		276.12	-0.02%
	9	321.67		2831.30	0.45%	241.25		300.86	-0.32%	241.25		252.96	0.05%
	10	341.67		2856.02	0.83%	256.25		342.74	-0.57%	256.25		271.82	0.15%
Prosečno		189.72	-0.02%	1327.02	0.54%	142.29	0.56%	185.41	0.02%	142.29	0.38%	151.79	-0.03%



**Tabela 6.5.** Referentne veličine HDAP ciljne funkcije za slučaj vremena između pojava pretovarnih zadataka prema beta raspodeli

n	Inst	m=2				m=5				m=10			
		LB	$\Delta B\&B$ [%]	VNS	$\Delta RdVNS$ [%]	LB	$\Delta B\&B$ [%]	VNS	$\Delta RdVNS$ [%]	LB	$\Delta B\&B$ [%]	VNS	$\Delta RdVNS$ [%]
10	1	61.67	<b>-0.47%</b>	151.82	0.19%	46.25	0.00%	60.30	0.00%	46.25	0.00%	50.85	0.00%
	2	68.33	<b>0.00%</b>	191.84	0.81%	51.25	0.00%	61.92	0.24%	51.25	0.00%	53.50	0.00%
	3	66.67	<b>0.00%</b>	149.89	0.08%	50.00	0.00%	53.93	0.00%	50.00	0.00%	50.00	0.00%
	4	65.00	<b>0.00%</b>	130.39	0.00%	48.75	0.00%	62.62	0.01%	48.75	0.00%	56.55	0.00%
	5	71.67	<b>0.00%</b>	210.35	0.71%	53.75	1.61%	77.95	-0.07%	53.75	0.00%	64.23	0.00%
	6	65.00	<b>0.00%</b>	194.38	0.00%	48.75	-0.03%	75.90	0.00%	48.75	0.00%	62.33	0.00%
	7	66.67	<b>-0.16%</b>	186.20	0.32%	50.00	0.00%	68.37	0.00%	50.00	0.00%	57.93	0.00%
	8	70.00	0.00%	228.63	0.52%	52.50	-0.16%	84.47	0.00%	52.50	0.00%	65.72	0.00%
	9	68.33	0.00%	238.92	0.07%	51.25	-0.20%	92.27	0.00%	51.25	-0.56%	64.46	-0.45%
	10	71.67	<b>-0.27%</b>	222.98	-0.18%	53.75	0.00%	76.47	0.00%	53.75	0.30%	64.44	0.00%
25	1	163.33		1060.22	0.41%	122.50		278.78	0.50%	122.50		163.33	0.05%
	2	163.33		946.30	0.37%	122.50		225.36	-1.55%	122.50		150.05	0.06%
	3	183.33		979.88	0.33%	137.50		198.10	-2.01%	137.50		152.66	0.03%
	4	160.00		711.02	0.63%	120.00		170.96	0.37%	120.00		132.75	0.00%
	5	163.33		940.07	0.17%	122.50		177.83	0.71%	122.50		133.84	-0.30%
	6	166.67		804.52	0.59%	125.00		224.66	0.00%	125.00		144.58	-0.01%
	7	161.67		890.35	0.38%	121.25		199.11	-0.58%	121.25		141.59	0.26%
	8	160.00		1007.16	-0.21%	120.00		233.88	-0.61%	120.00		157.96	0.00%
	9	166.67		1027.20	-0.13%	125.00		211.43	0.00%	125.00		148.72	-0.40%
	10	165.00		654.05	-1.20%	123.75		164.31	0.00%	123.75		136.47	0.00%
50	1	333.33		1767.73	1.06%	250.00		386.91	0.25%	250.00		281.33	-0.13%
	2	331.67		1847.61	3.01%	248.75		297.50	-0.10%	248.75		258.63	0.00%
	3	351.67		2925.00	-0.22%	263.75		401.73	0.39%	263.75		297.44	0.00%
	4	321.67		2631.57	0.27%	241.25		326.78	-0.03%	241.25		260.20	0.00%
	5	346.67		3708.27	0.91%	260.00		621.94	0.03%	260.00		330.73	-0.07%
	6	328.33		3340.79	0.68%	246.25		469.16	0.51%	246.25		303.72	0.01%
	7	343.33		2968.46	0.74%	257.50		357.99	0.47%	257.50		288.78	-0.11%
	8	343.33		2905.03	0.16%	257.50		372.45	-0.01%	257.50		278.98	0.01%
	9	321.67		2736.12	0.31%	241.25		408.46	-0.01%	241.25		283.89	-0.06%
	10	341.67		3032.94	0.04%	256.25		409.90	0.33%	256.25		291.93	0.00%
Prosečno		189.72	-0.09%	1292.99	0.36%	142.29	0.12%	228.38	-0.04%	142.29	-0.03%	164.25	-0.04%

**Tabela 6.6.** Referentne veličine vremena rešavanja HDAP instanci za slučaj vremena između pojava pretovarnih zadataka prema ravnomernoj raspodeli

n	Inst	m=2			m=5			m=10		
		$\Delta B\&B$ [%]	VNS [s]	$\Delta RdVNS$ [%]	$\Delta B\&B$ [%]	VNS [s]	$\Delta RdVNS$ [%]	$\Delta B\&B$ [%]	VNS [s]	$\Delta RdVNS$ [%]
10	1	126784.87%	4.818	-56.81%	52069.12%	6.084	-32.63%	61453.43%	11.573	-29.48%
	2	166437.58%	9.578	-75.13%	41494.26%	8.797	-35.17%	60512.49%	13.12	-26.63%
	3	199959.62%	7.942	-69.89%	85836.93%	5.513	-10.92%	54809.32%	13.482	-45.09%
	4	64729.24%	7.216	-74.57%	39920.67%	8.147	-47.86%	43743.42%	10.955	-25.52%
	5	84422.33%	12.048	-76.32%	37296.46%	9.205	-40.34%	66365.16%	10.541	-28.40%
	6	223774.43%	5.518	-67.92%	79943.34%	6.613	-24.38%	52022.39%	12.215	-29.67%
	7	123893.17%	5.919	-71.33%	47849.51%	8.328	-30.67%	65033.82%	10.873	-31.68%
	8	106164.40%	7.158	-77.06%	59651.59%	7.24	-46.10%	40215.28%	8.98	-26.97%
	9	122235.78%	7.711	-71.83%	36063.52%	11.213	-59.62%	48364.28%	11.434	-21.03%
	10	139496.42%	7.597	-58.76%	46790.88%	6.591	-35.52%	40816.14%	10.868	-38.58%
25	1		242.746	-74.37%		133.656	-57.89%		163.489	-27.98%
	2		257.496	-71.41%		120.254	-48.94%		161.726	-64.21%
	3		262.519	-62.73%		141.547	-50.45%		126.222	-45.64%
	4		278.684	-55.16%		163.837	-51.75%		201.639	-62.06%
	5		317.099	-72.45%		122.363	-52.98%		133.073	-27.36%
	6		224.973	-77.15%		151.148	-59.24%		218.74	-58.56%
	7		250.212	-69.42%		136.266	-54.32%		153.727	-53.40%
	8		188.768	-79.31%		133.641	-27.43%		134.099	-21.79%
	9		315.591	-69.89%		167.899	-62.01%		159.185	-43.43%
	10		171.652	-61.16%		136.877	-49.30%		110.37	-44.09%
50	1		2673.301	-34.95%		1952.13	-70.91%		1295.315	-47.40%
	2		2391.824	-63.39%		1637.465	-49.43%		1243.935	-53.08%
	3		3576.429	-60.10%		1335.966	-56.49%		1333.003	-38.12%
	4		2801.922	-64.01%		1515.381	-36.16%		1851.634	-48.98%
	5		3562.916	-37.40%		1620.503	-52.40%		1203.408	-33.27%
	6		2492.531	-31.76%		1681.167	-57.41%		1412.57	-58.37%
	7		2759.793	-51.10%		1297.736	-63.77%		1064.846	-28.72%
	8		1915.842	-26.91%		1510.337	-64.78%		1307.735	-59.46%
	9		3746.838	-48.84%		1715.888	-50.61%		874.945	-41.95%
	10		3485.722	-56.54%		1318.32	-50.88%		1223.26	-54.92%
Prosečno		135789.78%	1066.412	-62.26%	52691.63%	569.0037	-47.68%	53333.57%	482.8987	-40.53%

**Tabela 6.7.** Referentne veličine vremena rešavanja HDAP instanci za slučaj vremena između pojava pretovarnih zadataka prema beta raspodeli [s]

n	Inst	m=2			m=5			m=10		
		$\Delta B\&B$ [%]	VNS [s]	$\Delta RdVNS$ [%]	$\Delta B\&B$ [%]	VNS [s]	$\Delta RdVNS$ [%]	$\Delta B\&B$ [%]	VNS [s]	$\Delta RdVNS$ [%]
10	1	22094.76%	8.512	-71.97%	41618.23%	6.313	-25.93%	62343.27%	9.392	-20.83%
	2	99068.91%	9.97	-73.85%	42918.57%	6.462	-31.58%	57830.00%	9.907	-5.33%
	3	25427.18%	6.971	-71.63%	56615.72%	6.559	-44.69%	94219.61%	9.461	-27.98%
	4	5737.95%	8.426	-76.23%	60067.81%	7.024	-54.84%	48928.07%	9.62	-31.79%
	5	143307.12%	8.353	-61.27%	60487.78%	7.792	-42.56%	67340.34%	12.267	-24.08%
	6	162232.06%	8.876	-72.32%	66422.12%	8.125	-36.65%	49985.77%	10.766	-14.26%
	7	95499.30%	9.752	-64.96%	61671.64%	6.809	-38.58%	58121.73%	11.848	-37.28%
	8	103508.17%	8.24	-71.65%	78677.11%	9.304	-17.19%	93411.05%	11.09	-11.44%
	9	116606.60%	5.621	-55.68%	90405.59%	7.511	-17.89%	25446.88%	13.317	-20.64%
	10	186863.23%	8.104	-71.79%	88458.55%	8.669	-26.06%	87368.99%	11.689	-22.29%
25	1		145.711	-65.27%		153.334	-33.19%		219.751	-20.03%
	2		325.354	-61.85%		148.307	-19.20%		137.538	-24.27%
	3		191.379	-57.81%		185.199	-51.15%		148.712	-48.11%
	4		319.121	-71.56%		251.836	-61.78%		176.679	-52.98%
	5		228.228	-64.27%		157.01	-64.29%		184.154	-32.87%
	6		196.752	-46.84%		162.104	-51.47%		172.624	-47.83%
	7		216.919	-77.75%		159.833	-29.70%		146.416	-32.22%
	8		170.697	-16.77%		139.981	-26.50%		164.1	-36.18%
	9		198.752	-65.34%		167.899	-47.63%		149.91	-46.08%
	10		187.564	-57.87%		143.576	-60.99%		117.825	-39.86%
50	1		3372.734	-65.16%		1315.201	-35.32%		1065.813	-35.38%
	2		2816.844	-56.22%		1248.771	-64.38%		999.24	-56.52%
	3		3625.432	-57.91%		2601.374	-73.10%		1314.186	-59.58%
	4		3378.369	-61.54%		1542.646	-64.93%		1235.051	-59.90%
	5		4048.135	-55.48%		2096.351	-58.16%		1614.74	-27.48%
	6		2434.296	-51.12%		1216.006	-54.56%		938.062	-36.96%
	7		2995.768	-42.73%		1797.7	-57.24%		1127.791	-57.33%
	8		3602.417	-45.28%		1214.273	-44.34%		1146.303	-55.73%
	9		1885.952	-25.60%		1956.457	-44.77%		1290.186	-47.69%
	10		3025.527	-53.38%		1507.168	-62.86%		839.708	-30.41%
Prosečno		96034.53%	1114.959	-59.70%	64734.31%	607.9865	-44.72%	64499.57%	443.2715	-35.44%

Kolona „n“ u tabelama 6.4 i 6.5 sadrži informaciju u broju zadataka u rešavanoj instanci problema, pri čemu je redni broj instance dat u koloni „instanca“. Kolone „LB“ sadrže nominalne najmanje vrednosti rešenja. Vrednosti LB su računane za sve veličine flote jer u slučaju HDAP-a sredstvo sa najvećom proizvodnošću nije sredstvo sa rednim brojem jedan, kao što je slučaj u heTRPTW numeričkim eksperimentima. Naime, u heTRPTW eksperimentima sredstvo sa najvećom proizvodnošću je uvek sredstvo sa rednim brojem jedan, a kako je ono uvek uključeno u flotu serviseru to je njegovo vreme opsluge zadataka uvek i minimalno vreme, te je LB isti za sve veličine

flote. U slučaju HDAP-a, to nije slučaj jer se povećanjem flote sa 2 na 5 sredstava pojavljuje sredstvo koje je u stanju brže da pretovari šljunak. Naravno, vrednosti LB-a za slučajeve flota od 5 i 10 sredstava su uvek iste, jer ne postoji sredstvo sa proizvodnošću većom od 200t/h.

Što se tiče sadržine ostalih kolona u tabelama 6.4 i 6.5, one su iste kao i u slučaju hetTRPTW-a, tj. za svaku veličinu flote ( $m$ ) je nominalna vrednost ciljne funkcije VNS algoritma data u koloni „VNS“ i predstavlja vrednost u odnosu na koju se meri relativno odstupanje ciljnih funkcija dobijenih primenom B&B i RdVNS algoritama, predstavljeno u kolonama „B&B“ i „RdVNS“, respektivno.

Usled načina računanja LB-a njegovo vreme realizacije nije predmet analize vremenske efikasnosti primanjenih algoritma, te tabele 6.6 i 6.7 ne sadrže informacije o prosečnom vremenu potrebnom za dobijanje LB-a. Vremena realizacije ostalih algoritma su prikazane u odgovarajućim kolonama, za odgovarajući broj sredstava u floti, pri čemu je vreme realizacije VNS algoritma prikazano u nominalnoj vrednosti u sekundama, a vremena realizacije B&B i RdVNS algoritama su prikazana kao relativno odstupanje u odnosu na VNS algoritma.

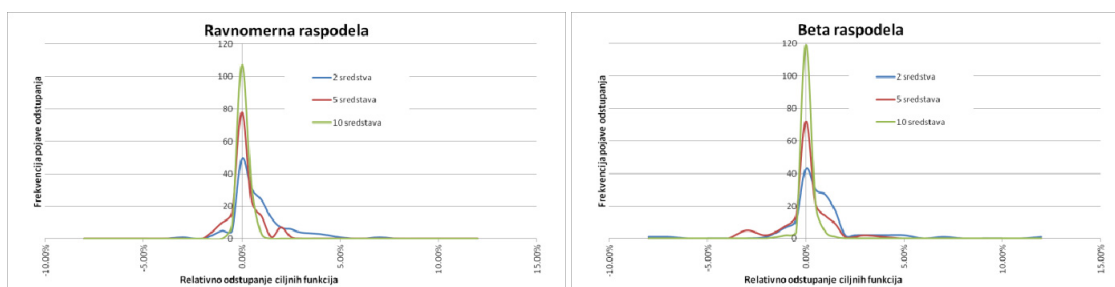
Kao i u slučaju hetTRPTW-a i ovde su optimalno rešavane samo instance malih dimenzija, tj. one kod kojih je  $n=10$ . Od 60 rešavanih instanci 18 ih je rešeno do optimalnosti i vrednosti njihovih ciljnih funkcija su u tabelama 6.4 i 6.5 prikazane podebljanim ciframa. Ono što se primećuje je da su sva optimalna rešenja postignuta za slučaj kada je  $m=2$ . Na ovaj način se potvrđuje konstatacija iz Bjelić i ostali (2013b) da povećanje broja servisera u floti značajno doprinosi povećanju kompleksnosti problema.

Na osnovu informacija iz tabela 6.4 i 6.5 uočava se da je prosečna vrednost ciljne funkcije VNS i RdVNS algoritama veća od rešenja B&B algoritma u 10 i 19 instanci, respektivno. Međutim, u svim instancama su stohastički algoritmi u nekom od 5 prolaza dostizali vrednost B&B rešenja, ali je način reprezentacije rešenja takav da nedostizanje B&B rešenja u jednom prolazu obično znači da je prosečna vrednost algoritma lošija od B&B rešenja.

U pogledu odstupanja B&B rešenja od VNS metode najveće prosečno odstupanje, od 0.56%, u korist B&B metode je zabeleženo u slučaju 9. instance kod beta raspodele za

vremena između pojave zadatka (tabela 6.5), kada je  $m=10$ . Sa druge strane, najveće prosečno odstupanje RdVNS algoritma (1.75%) od B&B rešenja je zabeleženo u slučaju 3. instance kod ravnomerne raspodele vremena između pojave zadatka (tabela 6.4), kada je  $m=2$ .

Što se tiče odnosa rešenja dobijenih VNS i RdVNS algoritmima uočava se da je, kao i u slučaju heTTRPTW-a, prosečno odstupanje RdVNS rešenja manje sa porastom broja sredstava u sistemu. Razlog ovome je taj što, broj veoma bliskih rešenja u odnosu na vrednost ciljne funkcije, raste sa povećanjem broja sredstava u sistemu. Ovo je naročito izraženo u slučaju strukture flote koja je upotrebljena u HDAP-u, gde postoji po više sredstava iz svake od tri klase sredstava. Tako je rešenja sa istom vrednošću ciljne funkcije moguće dobiti bilo kojom kombinacijom dodeljivanja zadataka sredstvima iz iste klase. Raspodela frekvencija relativnog odstupanja vrednosti ciljne funkcije RdVNS i VNS algoritama u realizovanim eksperimentima je prikazana na slici 6.4.

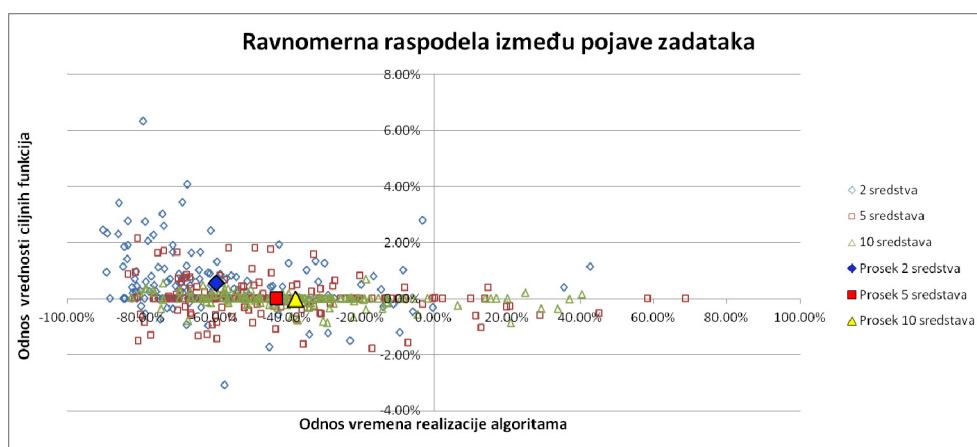


**Slika 6.4.** *Relativno odstupanje vrednosti ciljnih funkcija rešenja RdVNS algoritma od rešenja VNS algoritma*

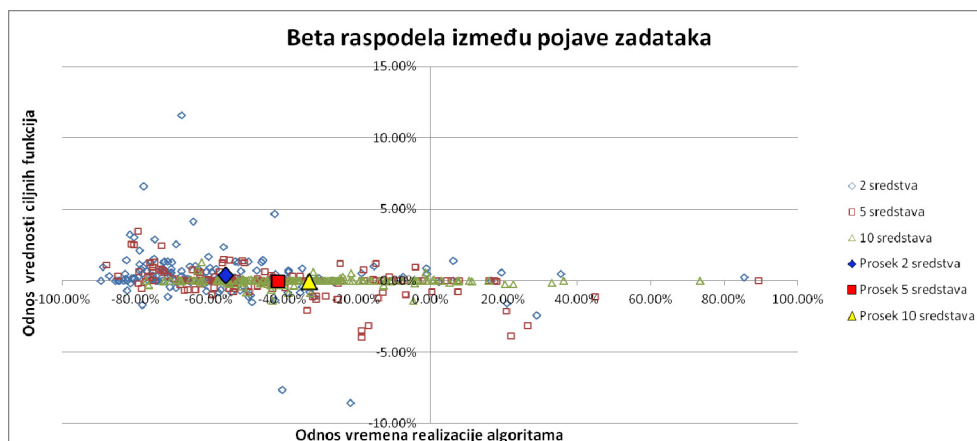
Na osnovu rezultata iz tabela 6.4 i 6.5 uočava se da su najveće prosečne razlike u vrednosti ciljne funkcije 3.01% u korist VNS algoritma i 2.01% u korist RdVNS algoritma. Međutim, na osnovu uvida u podatke o razlikama dobijenih rešenja u svakom ponavljanju instance dobija se da je najveća razlika u korist VNS algoritma 11.61%, a u korist RdVNS algoritma 8.54%. Treba napomenuti, da su obe maksimalne razlike dobijene za slučaj instance 10 kada je  $n=25$ ,  $m=2$ , a vreme između pojave zadatka je u skladu sa beta raspodelom.

U pogledu vremenske efikasnosti predstavljenih upravljačkih algoritma u slučaju HDAP-a može se reći da se sa povećanjem broja sredstava u floti javlja ista situacija kao i u slučaju heTTRPTW-a, tj. smanjenje vremenskih ušteda, ali i razlika u vrednosti

ciljnih funkcija, a kojoj je uzrok manja prisutnost redukcije  $\mathcal{N}_{Or-opt}$  okoline. Uporedni odnosi vremena realizacije i vrednosti ciljnih funkcija svih realizacija HDAP instanci za slučajeve ravnomerne i beta raspodele vremena između pojave zahteva su prikazani na slikama 6.5 i 6.6. Na njima je moguće uočiti da je prosečna ušteda u vremenu realizacije algoritma na nivou od 50% za  $m=2$ , a da se do  $m=10$  smanji na oko 35%. Kada se govori o ekstremnim vrednostima onda se može uočiti da je najveća ušteda vremena zbog primene redukovanih okolina bila 90%, a da je u pojedinim realizacijama VNS brže završavao rešavanje problema, nekada i za oko 90%.



**Slika 6.5.** Efekti primene redukovanih okolina za rešavanje HDPA-a za slučaj ravnomerno raspodeljenih vremena između pojave zadataka



**Slika 6.6.** Efekti primene redukovanih okolina za rešavanje HDPA-a kada su vremena između pojave zadataka raspodeljena prema beta raspodeli

## **7. Zaključak**

---

Praktični problemi čiji se osnovni cilj funkcionisanja može izraziti kumulativnom ciljnom funkcijom su veoma brojni i potreba za njihovim efikasnim rešavanjem je razlog zašto su oni već dugo vremena predmet interesovanja istraživača. Ovi problemi se sreću i u logističkim procesima u vidu minimizacije ukupnih vremena čekanja klijenata (transportnih sredstava, klijenata, mašina) na kraj opsluge. Međutim, detaljnim proučavanjem literature uočava se da su istraživači uglavnom bili usredsređeni na probleme u kojima je prisutno samo jedno sredstvo koje opslužuje zadatke koji su raspoloživi za opslugu od početka planskog perioda. Tek u poslednjih desetak godina pojavili su se rezultati istraživanja koja generalizuju problem respektujući više sredstava za realizaciju zadataka i/ili vremenska ograničenja u pogledu mogućnosti opsluge zadataka. Prvi rad koji respektuje i jedan i drugi aspekt generalizacije je objavljen 2000. godine i vezan je za problem upravljanja sistemom automatski vođenih vozila u kojem su sva vozila istih performansi. Međutim, kako sredstva mogu biti različitih eksploatacionih performansi to je u ovom istraživanju posmatran problem u kome zadatke realizuje heterogena flota sredstava.

Za rešavanje problema upravljanja pretovarnim sistemima sa kumulativnom ciljnom funkcijom i dinamičkom pojavom zadataka predstavljena su tri modela. Matematički MILP model zasnovan na mrežnom protoku koji omogućava dobijanje optimalnih rešenja problema i dva heuristička algoritma zasnovana na metaheurističkom pristupu promene okolina. Prvi heuristički algoritam dosledno primenjuje principe bazične varijante VNS metaheuristike, dok drugi heuristički algoritam, RdVNS, na osnovu karakteristika posmatranog problema pretražuje samo delove okolina čime se smanjuje vreme realizacije algoritma.

Efikasnost predstavljenih metoda za rešavanje predstavljenog upravljačkog algoritma je testirana na, u literaturi, poznatim benčmark instancama, ali i na primeru konkretnog upravljačkog problema upravljanja flotom plovnih pretovarnih sredstava za pretovar šljunka iz potisnica na deponije na obali. Treba napomenuti da, obzirom da posmatrani problem nije bio predmet intenzivnog istraživanja, ne postoji skup benčmark instanci problema generisanih za posmatrani problem, već su korišćene instance sličnih problema iz raspoložive literature koje se prilagođene posmatranom problemu, dok je u slučaju pretovarnog sistema za istovar šljunka korišćen skup slučajno generisanih instanci.

Empirijskim testiranjem se pokazalo da je posmatrani problem izuzetno zahtevan za rešavanje i da je velika verovatnoća da se ni problemi malih dimenzija ne mogu rešiti do optimalnosti u prihvatljivom vremenskom intervalu, pri čemu verovatnoća raste sa porastom broja sredstava u floti i/ili brojem zadataka. Na osnovu rezultata empirijskog testiranja predloženih metoda zapaža se takođe da predloženi heuristički algoritmi daju rešenja problema u prihvatljivom vremenskom intervalu za njihovu praktičnu primenu, pri čemu su, u instancama koje su rešavane do optimalnosti, odstupanja od optimalnih vrednosti minimalna. Pored toga, iz rezultata se uočava da se primenom redukcija korišćenih u RdVNS algoritmu, na račun prihvatljivog pada kvaliteta rešenja od oko 0.5 % prosečno, može skratiti vreme dobijanja upravljačke odluke približno za oko 40%. Treba samo naglasiti da se sa povećanjem broja sredstava u floti smanjuje efekat redukcije u smislu vremena rešavanja i da je stoga u velikim flotama potrebno, ili primeniti strože uslove za redukciju okolina, bilo kroz vrednosti parametara, bilo kroz definisanje modifikovanih redukcija.



Na kraju treba reći da i pored praktične primenljivosti rezultata istraživanja u vidu implementacije u odgovarajuće softverske pakete, rezultate ovog istraživanja treba shvatiti kao polaznu tačku u rešavanju konkretnih problema u smislu proširenja karakteristika koje se uzimaju u obzir prilikom donošenja upravljačkih odluka. Naime, odluke koje se donose u praktičnim uslovima su izuzetno kompleksne, ne samo u pogledu kompleksnosti problema objašnjene u ovom radu, već i u pogledu velikog broja faktora koji utiču na donosioca odluke, a koje nije moguće uvrstiti u procese modeliranja. U tom smislu, uključivanje što većeg broja faktora u rešenja dobijena primenom modela povećava kvalitet odluka krajnjeg donosioca odluke jer je manji broj faktora potrebno sagledavati iskustveno. Konkretno, krajnju odluku o vremenskom raspoređivanju plovnih pretovarnih sredstava na lokacije zadataka ljudski operater (dispečar) donosi na osnovu faktora opisanih u ovom istraživanju, ali i na osnovu trenutnog stanja zaliha šljunka na deponijama, očekivanih nivoa potrošnje šljunka, radnog vremena operatera na sredstvima, ali i faktora koje je praktično nemoguće implementirati u proces modeliranja.

## Literatura

- Aloise Dar., Aloise Dan., Rocha C., Ribeiro C., Ribeiro Filho J., Moura L., Scheduling workover rigs for onshore oil production, *Discrete Applied Mathematics* 154, 695-702
- Angel-Bello F., Alvarez A., García I. (2013) Two improved formulations for the minimum latency problem. *Applied Mathematical Modelling* 37, 2257–2266.
- Ballou, R., (2004) *Business Logistics/Supply Chain Management: Planning, Organizing, and Controlling the Supply Chain*, 5<sup>th</sup> Edition, Pearson Education, New Jersey, USA
- Banks, J., Carson II, J., Nelson, B., Nicol, D. (2005) *Descreet-Event System Simulation*, 4th edition, Pearson Education, Inc. New Jersey, USA
- Bjelić, N., (2009) *Modeli operativnog upravljanja pretovarnim sistemima sa distribuiranim resursima*, Magistarska teza, Univerzitet u Beogradu, Saobraćajni fakultet.
- Bjelić N., Vidović M., Miljuš M., (2010) Two mathematical models of the Handling Devices Allocation Problem, *Zbornik radova XXXVII simpozijuma o operacionim istraživanjima - SYMOPIS 2010*, Tara 21. – 24. septembar 2010, 349.-352.
- Bjelić N., Vidović M., (2011) Memetic algorithm for Dynamic Handling Device Allocation Problem, *Zbornik radova XXXVIII simpozijuma o operacionim istraživanjima - SYMOPIS 2011*, 4.-7. oktobar 2011, Zlatibor, Srbija, str. 359-362.
- Bjelić, N., Vidović, M., Popović, D., Ratković B., (2013a) Rešavanje problema putujućeg servisera sa vremenskim prozorima primenom genetskog algoritma. *Zbornik radova XL simpozijuma o operacionim istraživanjima - SYMOPIS 2013*, 509-514
- Bjelić, N., Vidović, M., Popović, D. (2013b) Variable neighborhood search algorithm for heterogeneous traveling repairmen problem with time windows, *Expert Systems with Applications* 40, 5997–6006

- Chaudhuri K., Godfrey B., Rao S., Talwar K. (2003) Paths, trees, and minimum latency tours. Proceedings of 44th symposium on foundations of computer science (FOCS), 36–45
- Cormen, T., Leiserson, C., Rivest, R., Stein, C., (2001) Introduction to Algorithms 2<sup>nd</sup> edition, MIT Press, USA
- Dasgupta, S., Papadimitriou, C., Vazirani, U., (2008) Algorithms, McGraw-Hill, New York, USA
- Fakcharoenphol J., Harrelson C., Rao S. (2007) The k-traveling repairman problem, ACM Transactions on Algorithms 3 (4)
- Fleischmann, B., Meyr, H., Wagner, M. (2005) Advanced Planning, In: Stadtler, H., Kilger, C., Supply Chain Management and Advanced Planning: Concepts, Models, Software, and Case Studies, Springer, New York, USA
- Gunther, O., Kim, K.H., (eds) (2005) Container terminals and automated transport systems, Springer, Berlin.
- Hansen P., Mladenović N. (2001). Variable neighborhood search: Principles and applications, European Journal of Operational Research 130, 449–467
- Hansen, P., Mladenovic, N., (2003) Variable neighborhood search, In Glover, F., Kochenberger, G., Handbook of metaheuristics, Kluwer Academic Publisher, New York, USA
- Hansen, P., Mladenovic, N., Brimberg, J., Moreno Perez, J. (2010) Variable neighborhood search, In: Gendreau, M., Potvin, J.Y., (Eds.) Handbook of metaheuristics 2<sup>nd</sup> Edition, str. 61-86., Springer, New York, USA
- Heilporn G., Cordeau J., Laporte G. (2010) The delivery man problem with time windows, Discrete Optimization 7, 269–282
- Hendriks, M. P. M., (2009). Multi-step optimization of logistics networks: strategic, tactical, and operational decisions. Doktorska disertacija, Technische Universiteit Eindhoven

- Hetland, M. L., (2010) *Python Algorithms: Mastering Basic Algorithms in the Python Language*, Apress
- Jothi R., Raghavachari B. (2007) Approximating the k-traveling repairman problem with repair times, *Journal of Discrete Algorithms* 5(2), 293-303
- Kara Ā., Kara B.Y., Yetis M.K. (2008) *Cumulative Vehicle Routing Problems*, U knjizi: Carić T., Gold H. , *Vehicle Routing Problem*, I-Tech, Vienna, Austria
- Ke L., Feng Z. (2013) A two-phase metaheuristic for the cumulative capacitated vehicle routing problem, *Computers & Operations Research* 40, 633–638
- Levitin, A. (2012) *Introduction to the design & analysis of algorithms – 3rd ed.*, Pearson Education, Inc., New Jersey, USA
- Lysgaard J., Wøhlk S. (2013) A branch-and-cut-and-price algorithm for the cumulative capacitated vehicle routing problem, *European Journal of Operational Research*, <http://dx.doi.org/10.1016/j.ejor.2013.08.032>
- Meisel, F. (2009) *Seaside Operations Planning in Container Terminals*, Physica-Verlag, Heidelberg, Germany
- Mladenović N., Hansen P. (1997) Variable neighborhood search, *Computers and Operations Research* 24(11), 1097–1100
- Mladenović N., Urošević D., Hanafi S. (2013) Variable neighborhood search for the traveling deliveryman problem, *4OR: A Quarterly Journal of Operations Research* 11(1), 57–73
- Ngueveu U. S., Prins C., Wolfler-Calvo R. (2010) An effective memetic algorithm for the cumulative capacitated vehicle routing problem *Computers and Operations Research* 37, 1877–1885.
- Ohlmann J., Thomas B. (2007) A compressed annealing heuristic for the traveling salesman problem with time windows, *INFORMS Journal on Computing* 19(1), 80–90.
- Papadimitroui, C., Steiglitz, K., (1998) *Combinatorial optimization: Algorithms and Complexity*, Dover publications, New York, USA

- Pisinger, D., Ropke, S. (2010) Large neighborhood search, In: Gendreau, M., Potvin, J.Y., (Eds.) Handbook of metaheuristics 2<sup>nd</sup> Edition, str. 399-420., Springer, New York, USA
- Potvin J., Bengio S. (1996) The vehicle routing problem with time windows Part II: Genetic search, INFORMS Journal on Computing 8(2), 165–172.
- Rushton, A., Croucher, P., Baker, P. (2010) The Handbook of Logistics and Distribution Management, 4<sup>th</sup> edition, Kogan Page Limited, London, UK
- Ribeiro G. M., Laporte G. (2012) An adaptive large neighborhood search heuristic for the cumulative capacitated vehicle routing problem, Computers and Operations Research 39(3), 728–735.
- Salehipour A., Sorensen K., Goos P., Braysy O. (2011) Efficient GRASP+VND and GRASP+VNS metaheuristics for the traveling repairman problem, 4OR: A Quarterly Journal of Operations Research 9, 189–209.
- Silva M. M., Subramanian A., Vidal T., Ochi L. S. (2012). A simple and effective metaheuristic for the minimum latency problem. European Journal of Operational Research 221(3), 513–520
- Sipser, M. (2013) Introduction to the Theory of Computation, 3rd Edition, Cengage Learning, Boston, USA
- Talbi, E.-G. (2009) Metaheuristics: From Design to Implementation, John Wiley & Sons, Inc., Hoboken, New Jersey
- Tsitsikils J. (1992) Special cases of traveling salesman and repairman problem with time windows, Networks 22, 263–282.
- van der Meer, R. (2000) Operational control of internal transport, Doktorska disertacija, University of Rotterdam. <http://repub.eur.nl/res/pub/859/>
- Vidović, M., (1997) Mogućnosti poboljšanja performansi logističkih sistema optimizacijom operativnog planiranja nekih klasa pretovarnih procesa, Doktorska disertacija, Saobraćajni fakultet, Beograd,

- Vidović, M., (2007) Kvantitativna analiza sistema rukovanja materijalom, Univerzitet u Beogradu, Saobraćajni fakultet
- Vidović, M., Vukadinović, K., (2006) Allocation planning of handling devices for barges unloading, EWGT 2006 International Joint Conference Bari 2006 – Proceedings, , str. 740-747
- Yalaoui, A., Chehade, H., Yalaoui, F., Amodeo, L., (2012) Optimization of Logistics, ISTE Ltd and John Wiley and sons,
- Wu B.Y. (2000) Polynomial time algorithms for some minimum latency problems, Information Processing Letters 75, 225–229
- Wu B.Y., Huang Z.-N., Zhan F.-J. (2004) Exact algorithms for the minimum latency problem, Information Processing Letters 92, 303–309
- Zhixing L., Hu Q., Lim A. (2014) Branch-and-price-and-cut for the multiple traveling repairman problem with distance constraints, European Journal of Operational Research 234, 49-60

## Biografija autora

Mr Nenad Bjelić, rođen je 24. oktobra 1977. godine u Kragujevcu od oca Petra i majke Milese. Osnovno i srednje obrazovanje je stekao u Smederevskoj Palanci gde je 1992.god. završio osnovnu školu „Heroj Ivan Muker“, a 1996. god. prirodno-matematički smer gimnazije „Sveta Đorđević“. Školske 1996/97. upisao je Saobraćajni fakultet Univerziteta u Beogradu, gde 2003. god. završava studije odbranom diplomskog rada “Modeli planiranja u sistemima ekspres prenosa pošiljaka”, sa prosečnom ocenom studiranja 8.66 i diplomskim radom ocenjenim ocenom 10.

Postdiplomske studije na Saobraćajnom fakultetu Univerziteta u Beogradu Nenad Bjelić je upisao novembra 2003. godine, na smeru Logistički sistemi. Položio je sve ispite na postdiplomskim studijama sa prosečnom ocenom 10 i u novembru 2009. godine je odbranio magistarsku tezu pod nazivom „Modeli operativnog upravljanja pretovarnim sistemima sa distribuiranim resursima“

Na Odseku za logistiku Saobraćajnog fakulteta Univerziteta u Beogradu je od novembra 2003.god. u zvanjima asistenta-pripravnika, saradnika u nastavi i asistenta angažovan za držanje vežbi na osnovnim akademskim studijama iz predmeta: Mehanizacija pretovara, Roba u logističkim procesima, Objektno orijentisana simulacija logističkih sistema, Simulacija logističkih sistema, Tehnologija sistema rukovanja materijalima, dok je na master studijama angažovan za držanje vežbi na predmetima Operativno planiranje pretovarnih procesa i Softverski alati u logistici.

U dosadašnjem istraživačkom radu je u svojstvu autora ili koautora učestvovao u objavljivanju 3 rada u časopisima sa SCI liste, 5 radova u međunarodnim i domaćim časopisima i 36 radova na međunarodnim i domaćim konferencijama i skupovima. Pored toga, kao član autorskog tima učestvovao je u izradi 9 naučno-istraživačkih i stručnih studija i projekata

Прилог 1.

## Изјава о ауторству

Потписани: Ненад Бјелић

Изјављујем

да је докторска дисертација под насловом

**Модели за оперативно управљање једном класом претоварних средстава  
у условима динамичке појаве задатака**

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

Потпис докторанда

У Београду, 12.11.2014.

Ненад Бјелић



Прилог 2.

## Изјава о истоветности штампане и електронске верзије докторског рада

Име и презиме аутора: Ненад Бјелић

Број индекса: докторанд са звањем магистра

Студијски програм: Саобраћај

Наслов рада: Модели за оперативно управљање једном класом претоварних средстава у условима динамичке појаве задатака

Ментор: Проф. Др. Милорад Видовић, дипл. инж. саоб.

Потписани Ненад Бјелић

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, 12.11.2014.

Nenad Bjelic

Прилог 3.

## Изјава о коришћењу

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

**Модели за оперативно управљање једном класом претоварних средстава у условима динамичке појаве задатака**

која је моје ауторско дело.

Дисертацију са свим прилозима предао 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.

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, 12.11.2014.

Ненад Бјелић

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

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

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

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

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

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