УНИВЕРЗИТЕТ У БЕОГРАДУ

САОБРАЋАЈНИ ФАКУЛТЕТ

Милош Љ. Николић

УБЛАЖАВАЊЕ ПОСЛЕДИЦА ПОРЕМЕЋАЈА У ОДВИЈАЊУ САОБРАЋАЈА ПРИМЕНОМ МЕТАХЕУРИСТИКЕ ОПТИМИЗАЦИЈА КОЛОНИЈОМ ПЧЕЛА

докторска дисертација

Београд, 2015

UNIVERSITY OF BELGRADE

FACULTY OF TRANSPORT AND TRAFFIC ENGINEERING

Miloš Lj. Nikolić

DISRUPTION MANAGEMENT IN TRANSPORTATION BY THE BEE COLONY OPTIMIZATION METAHEURISTIC

Doctoral dissertation

Belgrade, 2015

Ментор:

Др Душан Теодоровић, дописни члан САНУ, редовни професор Саобраћајног факултета Универзитета у Београду

Чланови комисије:

Др Јован Поповић, редовни професор Саобраћајног факултета Универзитета у Београду

Др Катарина Вукадиновић, редовни професор Саобраћајног факултета Универзитета у Београду

Др Милорад Видовић, редовни професор Саобраћајног факултета Универзитета у Београду

Др Вера Вујчић, редовни професор ФОН-а Универзитета у Београду

Датум одбране:

УБЛАЖАВАЊЕ ПОСЛЕДИЦА ПОРЕМЕЋАЈА У ОДВИЈАЊУ САОБРАЋАЈА ПРИМЕНОМ МЕТАХЕУРИСТИКЕ ОПТИМИЗАЦИЈА КОЛОНИЈОМ ПЧЕЛА

Резиме

У дисертацији су разматрани проблеми ублажавања последица поремећаја у одвијању планираних редова вожњи, редова летења и планова дистрибуције робе. Разматрани проблеми спадају у тешке проблеме комбинаторне оптимизације. Транспортне мреже на којима долази до поремећаја у одвијању планираних активности карактерише постојање веома великог броја чворова и грана. Комбинаторна природа проблема поремећаја у одвијању планираних транспортних активности, као и велике димензије транспортних мрежа условили су да се назначени проблеми решавају метахеуристиком Оптимизација колонијом пчела.

Од начина доделе паркинг позиција авионима на већим аеродромима, зависе како трошкови авио компанија, тако и квалитет услуга у ваздушном саобраћају. У дисертацији је разматран проблем додељивања паркинг позиција авионима у случајевима кашњења појединих летова. У ситуацијама када долази до кашњења није могуће опслужити авионе на унапред предвиђеним паркинг позицијама. У дисертацији је развијен модел који одређује нови план опслуживања авиона. Посебно су разматрани случајеви када постоји сарадња међу авио компанијама при коришћењу паркинг позиција. Развијени модели су тестирани на примеру *A* терминала аеродрома у Денверу.

Због отказа возила из техничких разлога у јавном градском превозу, диспечери често морају, да на одређеним линијама, обављају саобраћај са бројем возила који је мањи од планираног. У зависности од броја возила која су отказала, може доћи до значајног пада квалитета услуга које се пружају путницима. Ублажавање насталих поремећаја у извршавању планираног реда вожње могуће је постићи: 1) одређивањем новог скупа линија на којима се одвија саобраћај, 2) прерасподелом расположивих возила на већ постојеће линије и 3) прерасподелом

расположивих возила и модификовањем постојећих линија. У докторској дисертацији су развијени модели који омогућавају ублажавање последица поремећаја насталих недостатком планираног броја возила. Развијени модели омогућавају пројектовање нове мреже линија и/или прерасподелу расположивих аутобуса по линијама јавног превоза. Кроз развијене моделе се тежило минимизацији броја неопслужених путника, максимизацији нивоа комфора на линијама, минимизацији укупног времена путовања, минимизацији укупног броја преседања путника, итд.

У дисертацији су разматрани и проблеми поремећаја планираних активности у процесу дистрибуције робе. Компаније које се баве дистрибуцијом робе најчешће дефинишу дневне руте возила којих се придржавају током дужег временског периода. Поремећаји често настају у случајевима када поједини корисници испоставе захтеве за количинама робе које су значајно веће од уобичајених. У оваквим ситуацијама укупна потражња дуж једне или више рута превазилази носивости возила, тако да је потребно пројектовати нови скуп рута возила. У дисертацији је развијен модел за пројектовање новог скупа рута возила којим се тежи минимизирању негативних последица поремећаја (број неопслужених корисника, број измена у односу на постојеће руте, укупна дужина рута).

Поред решавања проблема поремећаја у одвијању планираних редова вожњи, редова летења и планова дистрибуције робе, у дисертацији је извршена и емпиријска студија метахеуристике Оптимизација колонијом пчела. У оквиру студије изналажене су екстремне вредности великог броја функција применом Оптимизације колонијом пчела. Резултати студије, као и резултати решавања проблема ублажавања поремећаја, су показали да коришћена метахеуристика може да пронађе веома квалитетна решења у прихватљивом времену рада рачунара.

Кључне речи: Ублажавање поремећаја, Оптимизација колонијом пчела, Вишекритеријумска оптимизација, Лексикографка оптимизација

Научна област: Операциона истраживања у саобраћају
Ужа научна област: Операциона истраживања у саобраћају

UDK број: 656:519.8(043.3)

DISRUPTION MANAGEMENT IN TRANSPORTATION BY THE BEE COLONY OPTIMIZATION METAHEURISTIC

Abstract

This doctoral dissertation deals with problems caused by disruptions in performing planned schedules in public transit and air transportation, as well as by disruptions in operations in distribution systems. The dissertation presents results of an effort to define new, ad hoc schedules for these situations. These problems arise in transportation networks characterized by a large number of nodes, branches, vehicles and crews. The problems considered here belong to the class of difficult combinatorial optimization problems whose optimal solution is difficult to find. The approach to problems of generating new, ad hoc schedules in cases of disturbances is based on the Bee Colony Optimization (BCO) metaheuristic. The BCO algorithm is a stochastic, random-search technique that belongs to the class of population-based algorithms.

The way in which the gate assignment at the airport is performed highly influences airline operating costs, as well as the level-of-service offered the passengers. The dissertation tackles the problem of generating a new gate assignment plan in the situations when some flights are delayed. In such situations, aircraft cannot be serviced at the earlier planned gates. The dissertation develops a model that generates the new gate assignment plan. The main attention is given to cases when the gates can be used in the system of collaboration among competitive airlines. The models developed are tested on one example at the terminal A at Denver International Airport, Colorado.

In many cases, lower service reliability in public transit can be influenced by the fact that some of the planned buses are out of operation due to different technical reasons. Depending on how many vehicles are out of operations, passengers' level of service can be significantly decreased. Mitigation of disruptions can be done in the

following way: 1) by modification of the existing public transit network (canceling operations at some lines, shortening some lines, etc), 2) by reassignment of available vehicles among the existing lines, 3) by simultaneous reassignment of available vehicles and modification of the existing network of lines. The models that can mitigate disruptions caused by shortage of buses are also developed in this dissertation. The models developed design a new transit network and/or to reassign available number of buses among the bus lines. These models minimize the total number of unserved passengers, maximize the passengers' level of service, minimize the total waiting time, minimize the total number of passenger transfers, etc.

Dissertation also considers the problems of goods distribution to the customers caused by disruptions. Logistic distribution companies normally define daily vehicle routes that could be used during the longer period of time. Disruptions usually arise in cases when demand at some nodes is much higher than usual. In such situations, the total demand along one or more planned routes could be higher than the vehicle capacity. Consequently, a new set of vehicle routes must be generated. The dissertation develops a vehicle routing model that can generate a new set of routes in such a way that the negative consequences of disruptions are minimized (the number of unserved customers, the total number of route changes, and the total route distance) too.

Besides solving the disruption problems in performing planned schedules, the Bee Colony Optimization metaheuristic has been performed. In this study, the Bee Colony Optimization has been used to find extreme values of large number of benchmark functions. The results of the study, as well as the results obtained by solving the problems caused disruptions in planned schedules, show that the BCO metaheuristic can generate high-quality solutions with relatively small CPU time.

Keywords: Disruption management, the Bee Colony Optimization, Multi-criteria optimization, Lexicographic method

Scientific field: Transport and Traffic Engineering

Field of Academic Expertise: Operations research applications in transport and traffic engineering

UDK: 656:519.8(043.3)

САДРЖАЈ

ПРЕГЛЕД СЛИКА

ПРЕГЛЕД ТАБЕЛА

1. УВОДНА РАЗМАТРАЊА

Одређивање најбољих планова рада дуги низ година је био приоритетни задатак истраживача. Укључивањем различитих ограничења у моделе, аутори су тежили развоју модела који су што је могуће више блиски реалности. Међутим, у пракси се често дешава да при реализацији дефинисани „идеални“ планови рада постају недопустиви, а самим тим и планирани трошкови постају далеко већи.

Предмет истраживања у оквиру докторске дисертације су проблеми који настају појавом поремећаја у реализацији планираних активности у саобраћају и транспорту. Проблеми који том приликом настају углавном имају доста заједничких карактеристика са проблемима који су претходно решавани како би се одредили иницијални планови рада. Међутим, поред циљева о којима се морало водити рачуна при одређивању иницијалних планова, за одређивање нових морају се узети у обзир и додатне критеријумске функције којима би се минимизирали негативних ефекати проузроковани насталим поремећајима. Такође, поред нових критеријумских функција, морају се узети у обзир и нова ограничења.

Циљ решавања ових проблема је налажење нових планова рада на начин да се минимизирају негативни ефекти проузроковани појавом поремећаја. С обзиром да се поремећаји углавном јављају током процеса реализације претходно дефинисаних планова, решења је често потребно наћи у што је могуће краћем времену. Због тога је примена хеуристичких алгоритама у односу на егзактне далеко прихватљивија. Из тог разлога као метод за решавање проблема насталих поремећајима корошћена је метахеуристика Оптимизација колонијом пчела.

Поред хеуристичких алгоритама у раду су развијани и модели математичког програмирања чије тестирање је вршено на примерима мањих димензија. У случају математичких модела који имају више критеријумских функција, тестирање је вршено применом Лексикографске методе вишекритеријумске оптимизације.

За решавање проблема линеарног и мешовитог целобројног програмирања коришћен је CPLEX софтвер, а за развој алгоритама заснованих на Оптимизацији колонијом пчела коришћен је NetBeans едитор програмског језика Java. Сва тестирања су урађена на Dell лаптопу који има следеће карактеристике: Intel(R) Core(TM) i5-2430M, 2.40 GHz, RAM 6 GB.

Рад се састоји из седам делова. У другом делу (након увода) детаљније је указано на значај решавања проблема у саобраћају и транспорту насталих поремећајима.

Метахеуристика Оптимизација колонијом пчела је описана у трећем делу дисертације. У овом поглављу је дат кратак преглед литературе у којој је ова метахеуристика коришћена, указано је на верзије алгоритма које постоје, као и на њихове специфичности у имплементацији. Поред тога извршена је емпиријска студија у којој се тежило провери квалитета решења које је могуће постићи Оптимизацијом колонијом пчела.

У четвртом делу описан је проблем који се јавља на паркинг позицијама на аеродромима, а проузрокован је кашњењем авиона. За решавање овог проблема предложена је математичка формулација заснована на вишекритеријумском мешовитом целобројном програмирању. Такође, због великих димензија проблема који се могу јавити предложен је и хеуристички алгоритам заснован на метахеуристици Оптимизација колонијом пчела. Предложени модели су тестирани на примеру аеродрома у Денверу.

У петом делу разматрани су проблеми који настају у случају када је у системима масовног превоза на располагању мањи број возила од планираног. За решавање овог проблема предложено је више модела. У случајевима када је дозвољена само прерасподела расположивих аутобуса између линија, као и када је осим прерасподеле дозвољено и модификовање линија предложене су математичке формулације, као и хеуристички приступи засновани на Оптимизацији колонијом пчела. У овом поглављу је разматран и проблем одређивања новог скупа линија на посматраној мрежи. За решавање овог проблема коришћен је само хеуристички приступ. Предложени модели у овом делу су тестирани на примерима из литературе.

У шестом делу развијена је математичка формулација и хеуристички алгоритам за проблем рутирања возила када због повећане потражње једну или више рута није могуће реализовати. Такође, у овом делу је решаван и проблем рутирања возила са временским интервалима. Тестирања алгоритама су показала ефикасност хеуристичког алгоритма заснованог на метахеуристици Оптимизација колонијом пчела.

У последњем, седмом, поглављу дата су закључна разматрања, као и правци будућих истраживања.

2. УБЛАЖАВАЊЕ ПОСЛЕДИЦА ПОРЕМЕЋАЈА У ОДВИЈАЊУ ПЛАНИРАНИХ РЕДОВА ВОЖЊИ, РЕДОВА ЛЕТЕЊА И ПЛАНОВА ДИСТРИБУЦИЈЕ РОБЕ

Бројни радови у литератури су посвећени проблемима одређивања најбољих могућих планова рада (редова вожњи, редова летења, планова дистрибуције робе). Веома често планиране транспортне активности не могу да буду реализоване због непредвиђених поремећаја. Разлози настанка поремећаја могу да буду: откази возила и/или опреме из техничких разлога (отказ авиона, аутобуса, трамваја), метеоролошки услови (појава магле, велике снежне падавина, ниске температуре и др.), људски фактор (изостанак са посла једног или више радника, грешке у раду, и др.), услови окружења (загушења у саобраћају, затвореност појединих саобраћајница и др.).

Ублажавање могућих последица појаве поремећаја могуће је вршити на два начина:

1. Прављењем таквог плана рада који би у највећој могућој мери био отпоран на појаву поремећа.
2. Деловањем након појаве догађаја у циљу минимизирања негативних ефеката.

У првом случају се на проактиван начин тежи генерисању решења које често није најбоље, али које је мање осетљиво на појаву могућих поремећаја. Оваквим концептом се тежи прављењу робустног решења на које највећи број мањих поремећаја неће оставити значајније последице.

Другим приступом план реализације активности се прави на најбољи могући начин, а уколико дође до појаве поремећаја, приступа се прављењу новог плана како би се предвиђене активности у највећој могућој мери реализовале и при томе сви други негативни ефекти свели на најмању могућу меру.

Рад Теодоровића и Губеринића (1984) представља први рад у коме су аутори покушали да ситуацију која је настала појавом поремећаја превазиђу генерисањем новог плана рада којим би се минимизирали негативни ефекти насталих поремећаја. Теодоровић и Губеринић (1984) су разматрали проблем поремећаја у одвијању реда летења који настаје недостатком авиона. Разматрани проблем решаван је методом гранања и ограничавања (Branch and Bound). Овим радом је отворена нова област истраживања која је позната под називом „Управљање поремећајима“ (*Disruption Management*). Поред проблема из области саобраћаја ова област је обухватила и проблеме из области производње, управљања залихама итд (Yu и Qi, 2004).

У области саобраћаја највећа пажња истраживача посвећена је решавању проблема поремећаја редова летења. Теодоровић и Губеринић (1984), Теодоровић и Стојковић (1990, 1995), Jarrah и остали (1993) и Yan и Young (1996) су током осамдесетих и деведесетих година прошлог века започели истраживачке активности у области управљања поремећајима. У раду Теодоровића и Стојковића (1990) аутори су разматрали проблем поремећаја реда летења који настаје када један или више авиона откаже из техничких разлога. У раду је предложен хеуристички алгоритам за одређивање новог реда летења при чему се тежило минимизирању броја отказаних летова и минимизирању времена чекања свих путника на мрежи. Радом Теодоровића и Стојковића (1995) је проширен претходни модел увођењем додатних ограничења која се односе на захтеве службе техничког одржавања ваздухоплова. У овом случају као излаз из модела се добија и план ротација посада и план ротација авиона. Jarrah и остали (1993) су предложили два модела заснована на токовима у мрежи са минималним трошковима. Предложеним моделима се тежи минимизацији укупних трошкова проузрокованих недостатком авиона. Yan и Young (1996) су такође разматрали проблем који настаје недостатком планираног броја авиона. Аутори су развили модел заснован на Симплекс методу и Лагранжевој релаксацији.

Бабић и остали (2010) су развили систем за подршку одлучивању доносиоцу одлуке у случају када због поремећаја планирани ред летења више није допустив. Систем за подршку одлучивању користи хеуристички алгоритам који доносиоцу одлуке даје више могућих решења. За решавање проблема проузрокованих поремећајима Zegordi и Jafari (2010) су користили Оптимизацију колонијом мрава (ACO). У циљу поправљања поремећаја насталих у реду летења Eggenberg и остали (2010) су правили више „поправљених мрежа“. Такође, аутори су за решавање користили и метод генерисања колона (енгл. *Column Generation*). Sinclair и остали (2014) су у раду користили хеуристику претраживања околина за решавање проблема поправљања реда летења. Њихов метод је базиран на хеуристици предложеној 2009. године која наизменично пролази кроз три фазе: конструисање, поправљање и унапређење. Детаљнији преглед литературе у периоду до 2010. године могуће је наћи у прегледним радовима чији су аутори Kohl и остали (2007) и Clausen и остали (2010).

Вукадиновић и остали (1996), Corman и остали (2010), Nielsen и остали (2012) и Kroon и остали (2014) су се, поред других аутора, бавили проблемима ублажавања и управљања поремећајима у железничком саобраћају. Вукадиновић и остали (1996) су развили систем за подршку одлучивању који је заснован на вештачким неуронским мрежама. У раду су извршена тестирања предложеног система на реалним примерима. У раду Corman и остали (2010) аутори су разматрали кашњења у реализацији редова вожњи проузрокована конфликтима возова. Аутори су предложили побољшања у алгоритму Табу претраживања. Извршена су тестирања на примерима чија је величина блиска проблемима у реалности. Примери су генерисани на основу података са немачких железница. Тестирања су показала да је нови алгоритам ефикаснији од до сада коришћеног ROMA софтвера. Nielsen и остали (2014) су разматрали проблем поремећаја у железничком путничком превозу, у случају када је потребно одредити нови план рада композиција. У циљу решавања проблема аутори су користили приступ котрљајућег хоризонта (*Rolling horizont*) при чему се врши декомпоновање проблема на мање делове. Аутори су радили студију случаја на примеру холандске железнице. Kroon и остали (2014) су разматрали проблем који настаје када дође до отказивања транспортне услуге. Предложени хеуристички алгоритам је тестиран на примеру холандске железнице, где се показало да просечна кашњења путника могу значајно да се смање. Детаљнији преглед литературе могуће је наћи у раду Cacchiani-ја и осталих (2014).

Проблемима поремећаја у одвијању водног саобраћаја бавили су се Ioannou и остали (2011), Zeng и остали (2011), Brouer и остали (2013). Ioannou и остали (2011) су у својој студији разматрали стратегије за ублажавање поремећаја у лукама. Zeng и остали (2011) су предложили више приступа за отклањање поремећаја насталих на пристајалиштима лучких терминала. Поремећајима који настају у линијској пловидби су се бавили аутори Brouer и остали (2013). Аутори су у раду предложили модел који на основу поремећаја који се десио треба да одреди најбољи план пловидбе. Модел је примењен на четири реалне студије случаја компаније *Maersk Line*. Резултати су показали могућа смањења трошкова до 58 %.

Kepaptsoglou и Karlaftis (2009a), Li и остали (2009a), и Zeng и остали (2012) су се бавили проблемима поремећаја у одвијању друмског саобраћаја. Kepaptsoglou и Karlaftis (2009a) су разматрали проблем који настаје отказом метро система. Аутори су у раду предложили приступ за генерисање ефикасне мреже аутобуских линија које повезују станице метроа (*bus bridging*). За проблем прављења нових планова рада возила у случају када дође до отказа једног или више возила Li и остали (2009a) су предложили хеуристику убацивања засновану на Лагранжевим релаксацијама. Аутори су извршили тестирања на генерисаним примерима. Резултати тестирања су показали да хеуристика даје задовољавајућа решења. Zeng и остали (2012) су разматрали проблем који настаје када дође до краткорочних отказа трамваја. Сарадња са такси компанијама је разматрана у раду као једна од могућих стратегија за решавање уоченог проблема.

На основу прегледа релевантне литературе може се закључити да је област управљања и ублажавања поремећаја од изузетног значаја у свим видовима саобраћаја. Као илустрација овој тврдњи могу да послуже следећи подаци:

- У Бразилу је 17,67 % комерцијалних летова имало у 2008. години кашњење веће од 30 минута, при чему је 2,67 % планираних летова отказано (Brazilian Civil Aviation Agency, 2010).
- На холандској железничком мрежи у првој половини 2006. године било је 3992 поремећаја (просечно 22 поремећаја дневно) који су у вези са инфраструктуром (1656 поремећаја из техничких разлога, 172 због времеских услова, итд). Настали поремећаји су просечно трајали 1,7 сати.

(Jespersen-Groth и остали, 2007). Аутори такође наводе да је број поремећаја који је проузрокован оператерима приближно исти као и број поремећаја који је проузрокован инфраструктуром.

При решавању проблема који настају појавом поремећаја, углавном се тежи минимизирању негативних последица које настају. Такође, тежи се и минимизирању одступања од дефинисаног плана реализације активности. До сада развијени модели за отклањање поремећаја у одвијању планираних транспортних активности указују да је приликом отклањања насталих поремећаја потребно водити рачуна како о интересима путника и корисника услуга, тако и о интересима транспортних компанија. Ови интереси су, по правилу, међусобно конфликтни. Другим речима, у већини проблема отклањања поремећаја јавља се већи број критеријумских функција, односно, предмет разматрања су проблеми вишекритеријумске оптимизације. Као једна од метода за решавање проблема вишекритеријумске оптимизације у дисертацији ће бити коришћена Лексикографска метода.

Код методе лексикографске оптимизације потребно је унапред задати редослед критеријумских функција (f_1, f_2, ..., f_k) тако што ће најважнија критеријумска функција бити прва (f_1) , а најмање важна последња (f_k). За сваку критеријумску функцију врши се једно решавање проблема. За *i*-ту критеријумску функцију проблем има следећи облик (Coello и остали, 2002):

Минимизирати

$$f_i(\vec{x}) \tag{2.1}$$

при ограничењима:

$$g_j(\vec{x}) \le 0 \quad \forall\ j = 1, 2, \ldots, m \tag{2.2}$$

$$f_l(\vec{x}) = f_l^* \quad l = 1, 2, \ldots, i-1 \tag{2.3}$$

где f_l^* представља најбољу добијену вредност за критеријумску функцију f_l.

Након решавања првог проблема добија се решење којим се постиже најбоља вредност прве критеријумске функције. Решавањем проблема по другој критеријумској функцији добија се решење које мора да омогући да се вредност прве не погорша. Процес се наставља све док се проблем не реши и по последњој критеријумској функцији. Крајње решење мора да омогући да решења свих претходних критеријумских функција не буду погоршана у односу на њихове претходно добијене вредности.

С обзиром да су у дисертацији разматрани тешки проблеми комбинаторне оптимизације, а да је решење проблема потребно наћи у кратком временском периоду, за решавање оптимизационих задатака коришћена је метахеуристика Оптимизација колонијом пчела. Оптимизација колонијом пчела припада скупу метахеуристичких алгоритама заснованих на интелигенцији групе и инспирисаних природом.

Интелигенција групе (*Swarm Intelligence*) (Beni, 1988; Beni и Wang, 1989; Beni и Hackwood, 1992; Bonabeau и остали, 1999) је грана вештачке интелигенције (Artificial Intelligence) у којој се за решавање проблема користи група вештачких агената. Ови агенти међусобно сарађују приликом решавања проблема. Најчешће је понашање вештачких агената инспирисано понашањем јединки у колонијама социјалних инсеката у природи. Оствареним теоријским резултатима и бројним експеримантима је доказано да се вештачким агентима могу веома успешно решити многи сложени проблеми комбинаторне оптимизације. Међу најпознатије метахеуристичке алгоритма који су засновани на интелигенцији групе могу се сврстати: Оптимизација колонијом мрава, Оптимизација колонијом честица, Оптимизација колонијом вештачких пчела, Оптимизација колонијом пчела, и др.

Основна идеја метахеуристике Оптимизације колонијом пчела је да претрагу поља допустивих решења врше вештачки агенти инспирисани понашањем пчела у природи приликом скупљања нектара. Током процеса претрага пчеле комуницирају међусобно и при томе интензивирају претрагу делова простора у којима су пронађена добра решења. Истовремено, пчеле временом напуштају делове простора у којима пронађена решења нису била задовољавајућа.

3. МЕТАХЕУРИСТИКА ОПТИМИЗАЦИЈА КОЛОНИЈОМ ПЧЕЛА

За решавање сложених проблема комбинаторне оптимизације развијено је више метахеуристичких алгоритама. У овој дисертацији је коришћена метахеуристика Оптимизација колонијом пчела (*Bee Colony Optimization - BCO*). Ова метахеуристика је први пут представљена у радовима Лучића и Теодоровића (2001, 2002, 2003a, 2003b) и припада групи метахеуристика инспирисаних природом.

Након успеха у првим радовима BCO метахеуристика је успешно коришћена за решавање више различитих проблема, као што су:

- Локацијски проблеми: Шелмић и остали (2010), Давидовић и остали (2011), Димитријевић и остали (2011)
- Додељивање таласних дужина: Марковић и остали (2007)
- Распоређивање задатака на процесоре: Давидовић и остали (2012)
- Отклањање загушења у саобраћају: Теодоровић и Dell'Orco (2008)
- Лоцирање сензора на аутопутевима: Шелмић и остали (2010)
- Планирање доза у лечењу рака штитасте жлезде: Теодоровић и остали (2012)
- Распоређивање медицинских сестара у болницама: Тодоровић и Петровић (2013)
- Пројектовање линија јавног превоза: Николић и Теодоровић (2013b, 2014)
- Рутирање возила: Лучић и Теодоровић (2003б), Николић и остали (2013)
- и др.

Детаљнији преглед о примени BCO метахеуристике може се наћи у прегледним радовима чији су аутори Давидовић и остали (2014) и Теодоровић и остали (2014).

Основна идеја BCO метахеуристике је да претраживање поља допустивих решења врше вештачки агенти који опонашају пчеле током процеса сакупљања хране (нектара) у природи (Теодоровић (2003, 2008), Теодоровић и Dell'Orco

(2005)). Током сакупљања хране пчеле лете од цвета до цвета и након одређеног времена враћају се у кошницу. При повратку у кошницу долази до остављања („складиштења“) нектара, након чега пчеле започињу игру. Игром, пчеле које су биле успешне у претходном лету (сакупиле су доста нектара), покушавају да анимирају остале пчеле како би их пратиле. На тај начин, временом, сакупљање нектара се врши на деловима простора где га има највише, док се делови простора у којима има мало нектара временом напуштају.

На сличан начин вештачке пчеле врше претраживање простора допустивих решења. Претраживање простора се врши у итерацијама летовима унапред и уназад. При лету унапред вештачке пчеле врше претрагу поља допустивих решења, док при лету уназад пчеле врше поређења својих решења, након чега доносе одлуке да ли остају лојалне својим решењима или се одлучују да прате неку другу пчелу.

У литератури постоје две верзије BCO алгоритма: 1) конструктивна и 2) верзија заснована на побољшавању решења. Основна разлика између ових верзија BCO алгоритма је што пчеле у конструктивној верзији граде своје решење током итерације, док у верзији са побољшањем пчеле на почетку итерација добијају решења која модификацијама треба да покушају да побољшају. Улазни параметри и код конструктивне и код верзије BCO алгоритма са побољшањима су:

B – број пчела

IT – број итерација

NP – број летова

NC – број промена у једном лету унапред

Као излаз из алгоритма добија се најбоље пронађено решење (*S*).

3.1. Конструктивна верзија BCO алгоритма

Псеудо код за конструктивну верзију BCO метахеуристике се може представити следећом процедуром:

***процедура BCO(улаз** B, IT, NP, NC,* ***излаз** S)*

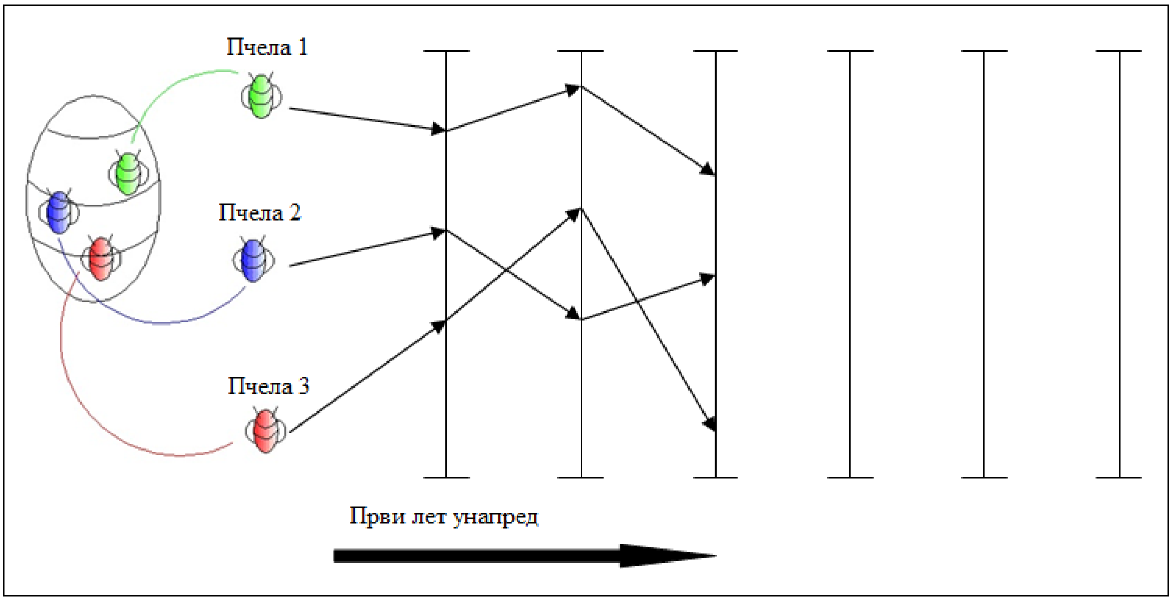
1: S ← Поставити празно решење.

2: ***for*** *j = 1* ***to*** *IT* ***do***

```
3:      for i = 1 to B do
4:          пчела i ← Поставити празно решење
5:      for k = 1 to NP do
6:          for r = 1 to NC do
7:              for i = 1 to B do
8:                  Извредновати могуће кораке пчеле i.
9:                  Коришћењем рулета одабрати један корак.
10:         Нормализовати квалитете решења свих пчела
11:         for i = 1 to B do
12:             Одредити да ли је пчела i лојална.
13:         for i = 1 to B do
14:             if пчела i није лојална then
15:                 Изабрати једну од лојалних пчела коју ће пчела i да прати.
16:     if најбоље решење од свих пчела боље од решења S then
17:             S ← Сачувати најбоље решење пчела као тренутно најбоље.
```

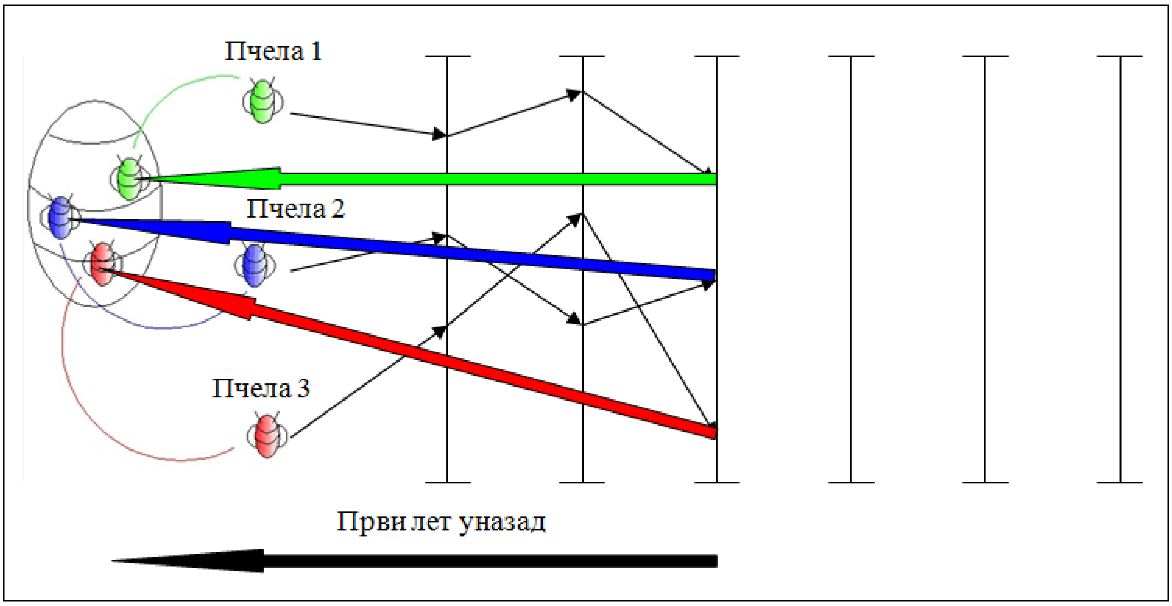
На почетку алгоритма не постоји ниједно решење алгоритма, тако да *S* садржи празно решење (Корак 1). У BCO алгоритму претраживање се врши у итерацијама. Петља итерација је представљена кораком 2. У једној итерацији се извршавају кораци од 3 до 17. На почетку итерација све пчеле добијају празна решења која треба да граде током итерације. Бројач летова је представљен кораком 5. Један лет унапред обухвата кораке 6, 7, 8 и 9. Лет уназад чине кораци од 10 до 15. На крају итерације се врши провера да ли је пронађено боље решење од до тада најбољег познатог.

Пример начина функционисања конструктивне верзије BCO алгоритма може се приказати сликама од 3.1 до 3.5. На Слици 3.1 је приказан први лет унапред у конструктивној варијанти BCO алгоритма. У овом примеру је претпостављено да у сваком лету пчеле треба да направе три корака (NC = 3).



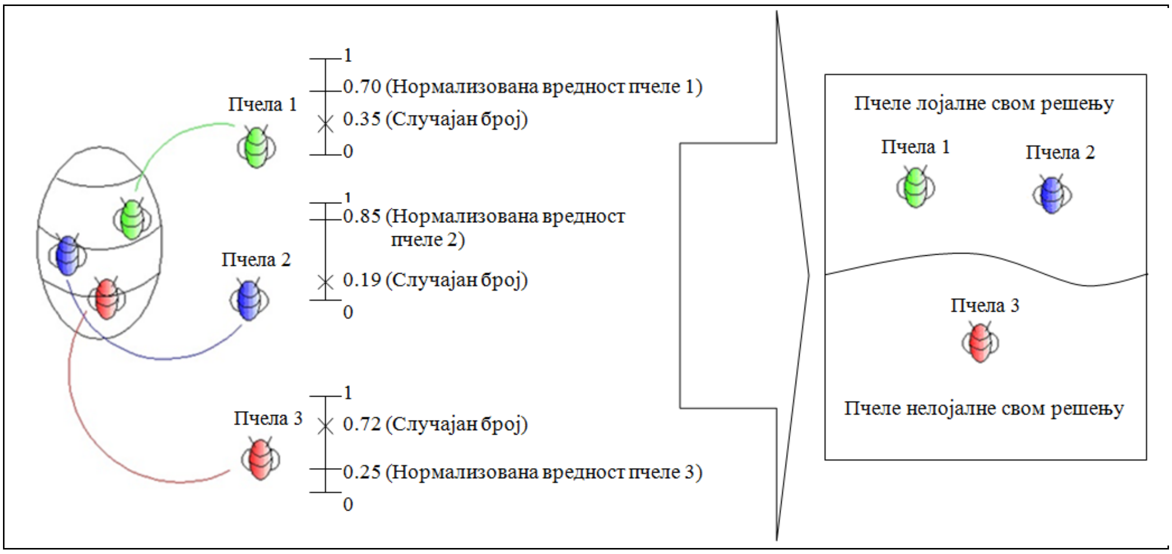
Слика 3.1. Први лет напред

Након што су направиле предвиђен број корака (*NC*) у првом лету, пчеле се враћају у кошницу чиме започиње лет уназад (видети Слику 3.2).



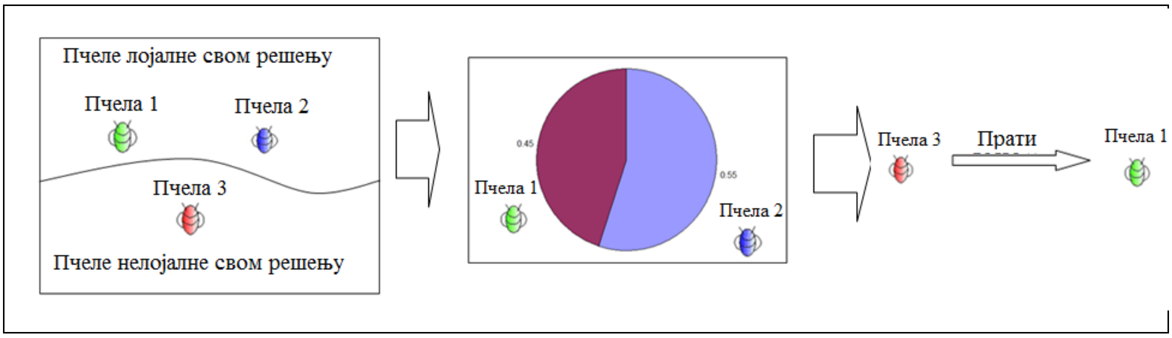
Слика 3.2. Повратак пчела у кошницу након првог лета

У кошници долази до размене информација између вештачких пчела. Пчеле доносе одлуку да ли остају лојалне свом решењу или не (Слика 3.3). Ова одлука се доноси на основу квалитета генерисаних решења. Уколико пчела више не жели да буде лојална свом решењу она постаје неопредељени следбеник.



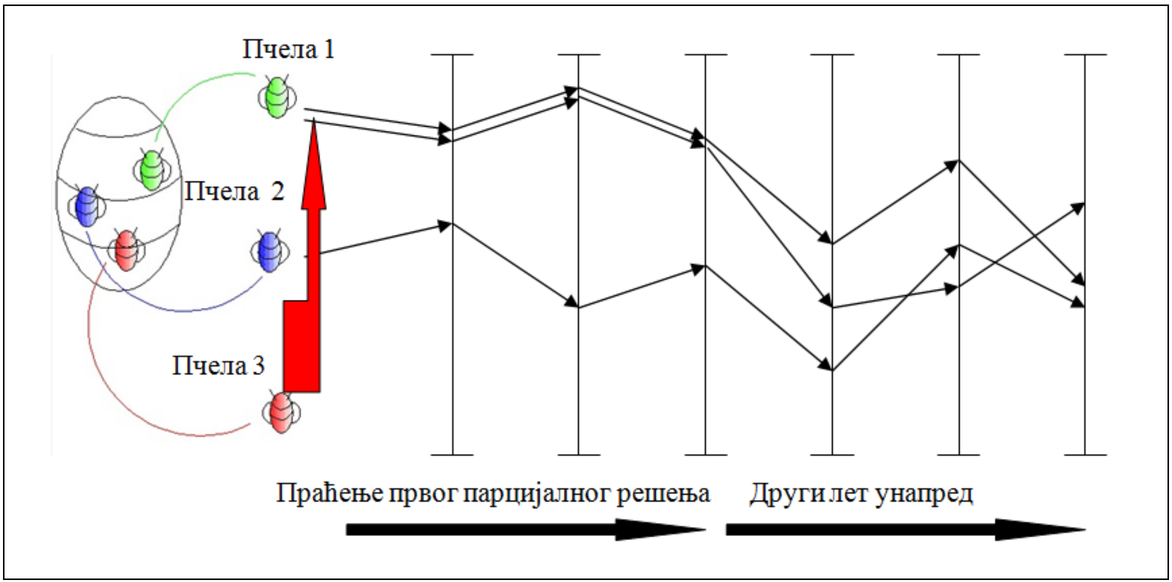
Слика 3.3. Доношење одлука о лојалности генерисаном решењу

Пчеле које нису лојалне свом решењу доносе одлуку коју од лојалних пчела ће да прате на почетку новог лета. На Слици 3.4 се може видети да је пчела 3 донела одлуку да прати пчелу 1 на почетку новог лета.



Слика 3.4. Регрутовање пчела које нису лојалне свом решењу

Након што пчеле 1 и 3 заједно дођу до краја решења генерисаног од стране прве пчеле у првом лету унапред, оне независно настављају да се крећу, тј. да врше претрагу поља допустивих решења (Слика 3.5).



Слика 3.5. Други лет пчела унапред

3.2. Верзија BCO алгоритма заснована на побољшању решења

Верзија која се заснива на побољшању решења први пут је коришћена у раду чији су аутори Давидовић и остали (2011). Псеудо код ове верзије BCO метахеуристике гласи:

***процедура BCOi(улаз** B, IT, NP, NC, **излаз** S)*

```
Одреди почетно решење.
Одредити квалитет почетног решења.
S ← Сачувати почетно решење као најбоље.
for j = 1 to IT do
    for i = 1 to B do
        пчела i ← Пчели доделити почетно решење.
    for k = 1 to NP do
        for r = 1 to NC do
            for i = 1 to B do
                Направити једну модификацију решења пчеле i.
                Одредити квалитет решења пчеле i.
            if најбоље решење од свих пчела боље од решења S then
                S ← Сачувати најбоље решење пчела као тренутно најбоље.
        Нормализовати квалитете решења свих пчела
```

15: **for** *i* = 1 **to** *B* **do**

16: *Одредити да ли је пчела i лојална.*

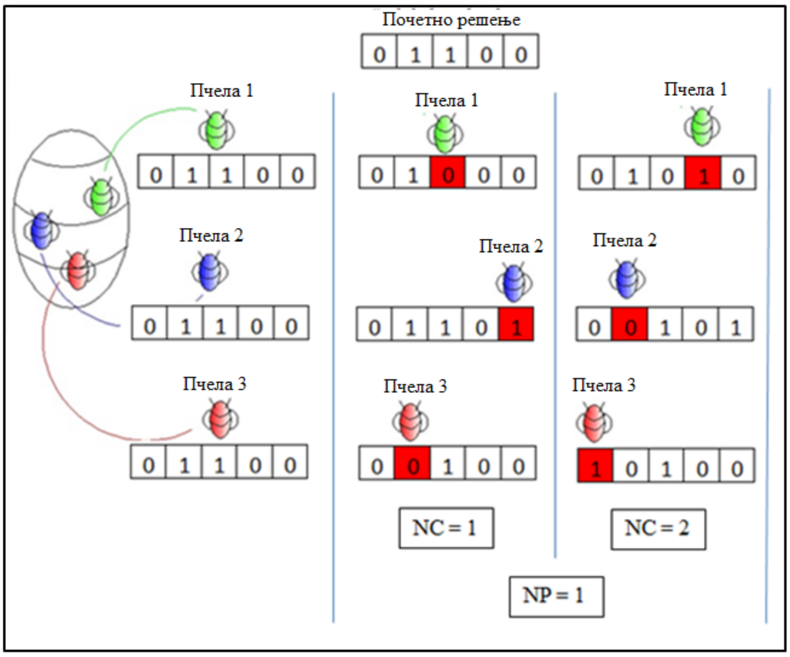
17: **for** *i* = 1 **to** *B* **do**

18: **if** *пчела i није лојална* **then**

19: *Одредити једну од лојалних пчела коју ће пчела i да прати.*

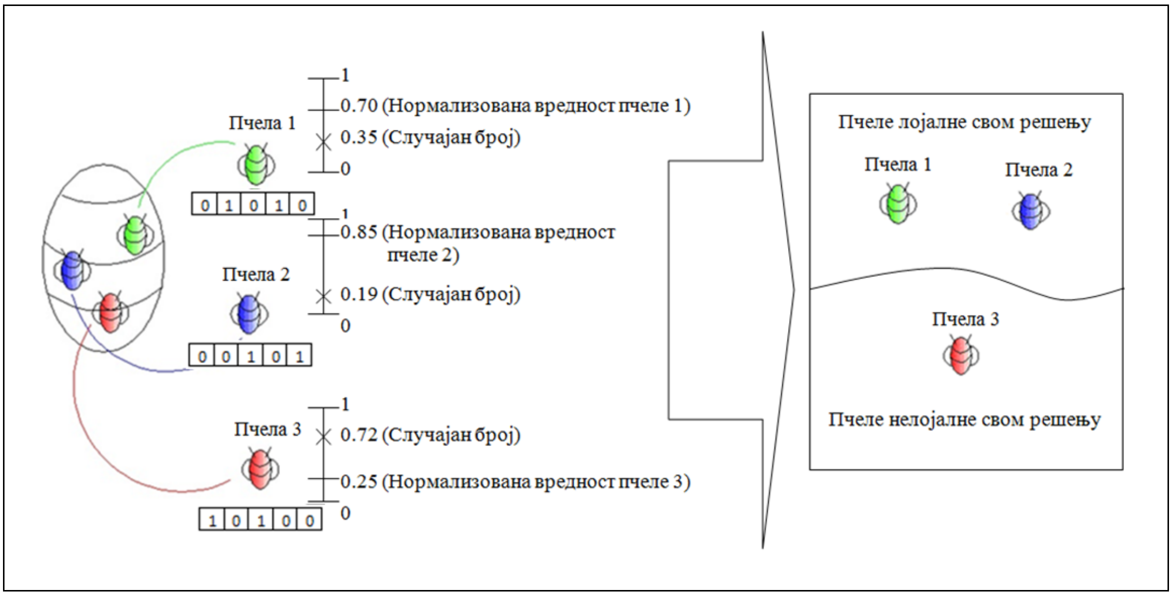
На почетку алгоритма, у корацима 1 и 2 се одређује почетно решење, које се у кораку 3 чува као тренуто најбоље пронађено. У кораку 4 се улази у петљу за итерације. У корацима 5 и 6 се пчелама додељује решење које ће се током итерације модификовати. Од корака 8 до корака 11 се реализује лет унапред. У оквиру лета унапред свака пчела треба да направи *NC* промена свог решења. Када све пчеле направе једну модификацију решења врши се провера да ли је пронађено боље решење од до тада најбољег познатог (кораци 12 и 13). Од корака 14 до 19 се реализује лет уназад.

Пример за верзију BCO метахеуристике са побољшањима могуће је видети на сликама од 3.6 до 3.9. Почетно решење проблема гласи 0 1 1 0 0 (Слика 3.6). Почетно решење је додељено свим пчелама. Један лет се састоји од две промене (*NC* = 2) које треба да направе пчеле. Након извршених промена пчеле се враћају у кошницу.



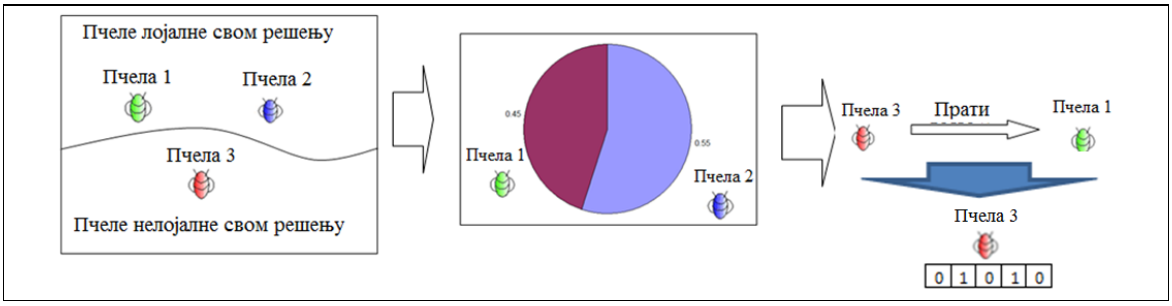
Слика 3.6. Први лет унапред друге верзије BCO метахеуристике

У кошници долази до нормализације квалитета решења пчела и доношења одлука које пчеле остају лојалне својим решењима, а које не (Слика 3.7).



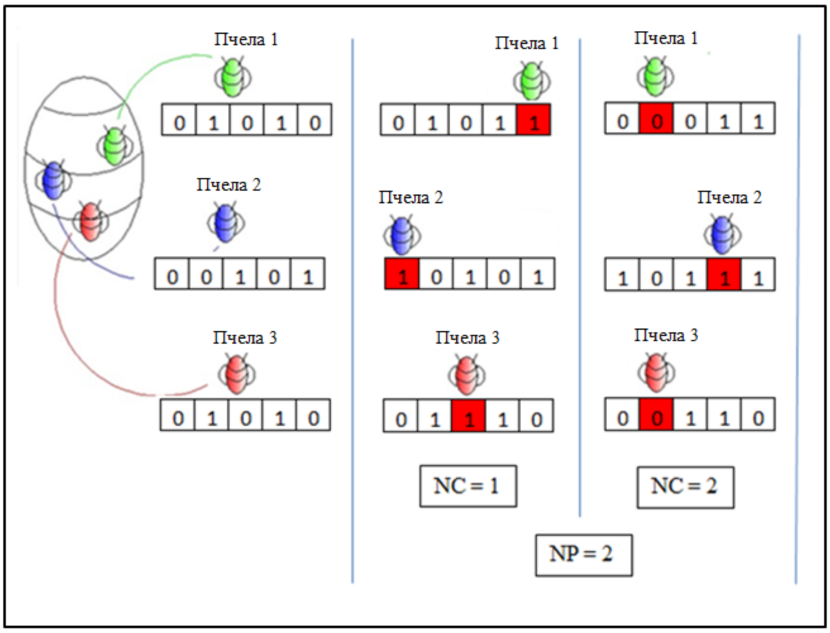
Слика 3.7. Доношење одлука о лојалности пчела својим решењима

Лојалне пчеле регрутују неопредељене пчеле (пчеле које нису остале лојалне својим решењима). У процесу регрутације су успешније пчеле које су генерисале квалитетнија решења. Неопредељеној пчели се додељује решење лојалне пчеле коју је изабрала. У разматраном примеру, као што се може видети са слике, пчела 3 је неопредељена и одабрала је да ће пратити пчелу 1. Због тога се пчели 3 додељује решење пчеле 1.



Слика 3.8. Регрутовање неопредељених пчела

При другом лету унапред (Слика 3.9), пчела 1 наставља са модификовањем свог решења које је имала на крају претходног лета унапред (0 1 0 1 0). Пчела 2 такође наставља са модификовањем свог решења (0 0 1 0 1), с обзиром да је у претходном лету уназад остала лојална свом решењу. Пчела 3 у другом лету унапред ће модификовати решење које је добила од пчеле 1, с обзиром да у лету уназад није била лојална свом решењу и да је одабрала да ће пратити пчелу 1.



Слика 3.9. Други лет унапред

3.3. Имплементација BCO метахеуристике

У зависности од проблема који се решава одређене делове алгоритма BCO метахеуристике је неопходно прилагодити. Ти делови алгоритма су углавном следећи:

- Одређивање почетног решења (код верзије BCO метахеуристике засноване на концепту побољшавања решења)
- Постављање почетног решења пчелама на почетку итерације (код верзије BCO метахеуристике засноване на концепту побољшавања решења)
- Прављење једног корака, односно модификације решења, при лету унапред (у обе верзије BCO метахеуристике)
- Одређивање квалитета решења (у обе верзије BCO метахеуристике) и
- Поређење решења (у обе верзије BCO метахеуристике)

У верзији BCO метахеуристике засноване на побољшавању решења, пре започињања итерација неопходно је одредити почетно решење. За генерисање почетног решења је могуће користити различите приступе. Један од могућих приступа је да се за посматрани проблем користи неки већ постојећи хеуристички алгоритам. Уколико такав алгоритам не постоји неопходно је осмислити алгоритам који би у сваком случају гарантовао добијање допустивог решења.

Код конструктивне верзије BCO алгоритма пчелама се на почетку итерација увек додељују празна решења. Међутим, код верзије BCO метахеуристике засноване на побољшањима то није случај. Један од приступа који се користи у овој верзији је да се пчелама на почетку итерација увек додељује најбоље пронађено решење. На овај начин се повећава брзина конвергенције. Негативна страна овог приступа је реална могућност заробљавања алгоритма у локалном оптимуму. Ово се најчешће дешава када је дефинисани број летова у итерацији мали.

Модификовање решења се у конструктивној верзији BCO метахеуристике углавном реализује избором корака на основу вероватноће добијене рачунањем користи. На тај начин пчеле имају могућност претраживања широког простора допустивих решења, али се ипак води рачуна да се фаворизују решења за која се локално зна да су добра.

Модификовање решења при лету унапред, у верзији BCO метахеуристике заснованој на побољшавању решења, такође је могуће вршити на више начине. Углавном заједнички принцип свим начинима модификовања решења је да већи део решења остане непромењен, односно, да се праве мале локалне промене. Као приступ који заслужује пажњу може се указати на могућност да се као део овог корака може користити и одређени алгоритам за локално претраживање који би након модификовања, решење тежио да доведе у неки од локалних оптимума. На пример, ако се решава проблем трговачког путника, 2-OPT или 3-OPT алгоритам се може користити за побољшавање решења након што се изврши модификација.

Уколико проблем који се решава има више критеријумских функција неопходно је дефинисати како се одређује квалитет решења. Ово је неопходно с обзиром да се при лету уназад за доношење одлука о лојалности пчела и слеђења користи само једна вредност. Добијање једне вредности као мере квалитета решења пчеле могуће је извршити на више начина и при решавању проблема неопходно је тачно дефинисати како се то ради. Један од могућих начина да се више критеријумских функција сведе на једну је њиховим сабирањем, при чему је извршено њихово претходно пондерисање одговарајућим тежинама. Такође, могући су и приступи засновани на методама вишекритеријумске анализе.

И поређење решења, као и одређивање квалитета решења, је потребно дефинисати само ако посматрани проблем има више критеријумских функција. И у овом случају аналитичар мора на јасан начин да дефинише како пореди решења. Као један од начина да се ово уради може да се користи принцип заснован на лексикографској оптимизацији (овај приступ је коришћен више пута у дисертацији).

У случају лета уназад после повратка у кошницу и после *NC* корака/промена, врши се нормализација квалитета решења које су генерисале пчеле. Ако је T_i функција циља i-те пчеле и при томе се тежи њеној максимизацији, нормализована вредност квалитета решења i-те пчеле се одређује као (Теодоровић и Шелмић, 2012):

$$O_i = \frac{T_i - T_{\min}}{T_{\max} - T_{\min}} \tag{3.1}$$

Уколико се тежи минимизацији функције циља, нормализована вредност квалитета решења i-те пчеле се одређује као (Теодоровић и Шелмић, 2012):

$$O_i = \frac{T_{\max} - T_i}{T_{\max} - T_{\min}}, \tag{3.2}$$

где су:

T_i –вредност критеријумске функције i –те пчеле,

T_{max} - највећа вредност критеријумске функције решења пронађеног од стране свих пчела ($T_{\max} = \max\limits_{i=\overline{1,n}}\{T_i\}$),

T_{min}- најмања вредност критеријумске функције решења пронађеног од стране свих пчела ($T_{\min} = \min\limits_{i=\overline{1,n}}\{T_i\}$).

Вештачка пчела доноси одлуку да ли ће остати лојална свом решењу на основу вероватноће која се израчунава као (Николић и Теодоровић, 2013а):

$$p_i = e^{-(O_{\max} - O_i)} \tag{3.3}$$

где су:

O_i – нормализована вредност критеријумске функције i – те пчеле,

O_{max} – максимална нормализована вредност генерисаних решења узимајући у обзир све пчеле.

Када пчела одлучи да неће да остане лојална свом решењу, она треба да се определи коју лојалну пчелу ће да прати. Вероватноћа да ће нелојална пчела да прати пчелу i једнака је:

$$p_i = \frac{O_i}{\sum_{k \in L} O_k} \tag{3.4}$$

где је:

L скуп лојалних пчела.

3.4. Емпиријска студија Оптимизације колонијом пчела

У циљу евалуације метахеуристике Оптимизација колонијом пчела у дисертацији је извршена и емпиријска студија ове метахеуристике. У оквиру студије изналажене су екстремне вредности великог броја функција применом Оптимизације колонијом пчела. За оптимизацију функција коришћена је верзија BCO метахеуристика заснована на концепту побољшања решења. У псеудо коду алгоритма је коришћена следећа нотација:

B – број пчела

TNC – укупан број промена у једној итерацији

NC – број промена промењљивих које треба обавити у једном лету унапред

IT – укупан број итерација

LB – лева граница (минимална вредност промењљиве)

RB – десна граница (максимална вредност промењљиве)

d - опсег

Алгоритам за оптимизацију функција може се представити следећим псеудо кодом:

процедура оптимизацијаФункција(улаз B, IT, TNC, NC, излаз S)

1: ***for*** *b = 1* ***to*** *B*

2: Одредити почетно решење пчеле b (На случајан начин одредити вредности свих променљивих водећи рачуна о опсезима у којима могу да се крећу).

3: S ← Најбоље решење од свих пчела сачувати као тренутно најбоље познато.

4: d = RB - LB

5: ***for*** *it = 1 to IT*

6: Доделити почетно решење свакој пчели.

7: Доделити вредност величини NC на случајан начин на интервалу од 1 до 5.

8: ***for*** *к = 1* ***to*** *TNC*

9: ***for*** *b = 1 to* ***B***

10: Променљивим одабраним на случајан начин одредити нову вредност.

11: Одредити квалитет решења пчеле b.

12: Проверити да ли је пронађено ново најбоље познато решење.

13: ***if*** *k mod NC = 0* ***then***

14: Нормализовати квалитете решења свих пчела.

15: ***for*** *b = 1* ***to*** *B*

16: Одредити да ли је пчела b лојална решењу или не.

17: ***for*** *b = 1* ***to*** *B*

18: ***if*** *пчела b није лојална* ***then***

19: Одабрати једну пчелу која је лојална, а коју ће пчела b да прати.

20: $d = d \cdot 0{,}998$

21: ***if*** *d < 0,001* ***then***

22: d = RB - LB

Пре почетка прве итерације свакој пчели се генерише једно решење. Вредности које се додељују променљивима одређују се на случајан начин, водећи рачуна о дозвољеним опсезима. На тај начин креира се *B* решења. Најбоље међу њима се чува као тренутно најбоље пронађено.

На почетку итерације, узима се у обзир најбоље познато решење, као и *B* решења пчела (укупно *B*+1 решење). Користећи рулет бира се једно решење и додељује се свим пчелама као почетно. Добијено почетно решење пчеле током итерације се модификује, са циљем да пронађу још боља.

Укупан број променљивих чија вредност ће да се промени у сваком лету се одређује на случајан начин на интервалу од 1 до 5. Променљиве чије вредности ће бити промењене одређује се на случајан начин. Одлука о томе да ли се вредност променљиве повећава или смањује се такође доноси на случајан начин. Величина опсега *d* се смањује у свакој итерацији множењем вредношћу 0,998. При смањивању опсега се води рачуна да његова вредност не буде мања од 0,001.

Нова вредност променљиве (у случају повећања вредности) је одређивана на следећи начин:

$$x_i = x_i + random(0,1) \cdot \min\left(d, RB - x_i\right) \tag{3.5}$$

Нова вредност променљиве (у случају смањења вредности) је одређивана на следећи начин:

$$x_i = x_i - random(0,1) \cdot \min\left(d, x_i - LB\right) \tag{3.6}$$

где је:

random (0,1) – функција која генерише случајан број на интервалу од 0 до 1.

За тестирање предложеног алгоритма коришћена је 51 функција. Овај скуп функција је релативно велики и укључује различите типове (унимодална, мултимодална, мултидимензионална, итд.). У Табели 3.1 су приказане функције које су решаване у овом експерименту.

Табела 3.1. Функције коришћене за тестирање Hedar и Fukushima (2006), Karaboga и Bahriye (2009)

Бр.	Опсег	Д	К	Функција	
1	[-5.12, 5.12]	5	US	Stepint	$f(x) = 25 + \sum_{i=1}^{n} \lfloor x_i \rfloor$
2	[-100, 100]	30	US	Step	$f(x) = \sum_{i=1}^{n} (\lfloor x_i + 0.5 \rfloor)^2$
3	[-100, 100]	30	US	Sphere	$f(x) = \sum_{i=1}^{n} x_i^2$
4	[-10, 10]	30	US	SumSquares	$f(x) = \sum_{i=1}^{n} i\, x_i^2$
5	[-1.28, 1.28]	30	US	Quartic	$f(x) = \sum_{i=1}^{n} i\, x_i^4 + random[0,1)$
6	[-4.5, 4.5]	5	UN	Beale	$f(x) = (1.5 - x_1 + x_1 x_2)^2 + (2.25 - x_1 + x_1 x_2^2)^2 + (2.625 - x_1 + x_1 x_2^3)^2$
7	[-100, 100]	2	UN	Eason	$f(x) = -\cos(x_1)\cos(x_2)\exp(-(x_1 - \pi)^2 - (x_2 - \pi)^2)$
8	[-10, 10]	2	UN	Matyas	$f(x) = 0.26(x_1^2 + x_2^2) - 0.48 x_1 x_2$
9	[-10, 10]	4	UN	Colville	$f_{20}(x) = 100(x_1^2 + x_2^2)^2 + (x_1 - 1)^2 + (x_3 - 1)^2 + 90(x_3^2 - x_4)^2 + 10.1((x_2 - 1)^2 + (x_4 - 1)^2) + 19.8(x_2 - 1)(x_4 - 1)$
10	[-D^2, D^2]	6	UN	Trid6	$f(x) = \sum_{i=1}^{n} (x_i - 1)^2 - \sum_{i=2}^{n} x_i x_{i-1}$
11	[-D^2, D^2]	10	UN	Trid10	$f(x) = \sum_{i=1}^{n} (x_i - 1)^2 - \sum_{i=2}^{n} x_i x_{i-1}$
12	[-5, 10]	10	UN	Zakharov	$f(x) = \sum_{i=1}^{n} x_i^2 + \left(\sum_{i=1}^{n} 0.5 i x_i\right)^2 + \left(\sum_{i=1}^{n} 0.5 i x_i\right)^4$
13	[-4, 5]	24	UN	Powell	$f(x) = \sum_{i=1}^{n/k} (x_{4i-3} + 10 x_{4i-2})^2 + 5(x_{4i-1} - x_{4i})^2 + (x_{4i-2} - x_{4i-1})^4 + 10(x_{4i-3} - x_{4i})^4$
14	[-10, 10]	30	UN	Schwefel 2.22	$f(x) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$
15	[-100, 100]	30	UN	Schwefel 1.2	$f(x) = \sum_{i=1}^{n} \left(\sum_{j=1}^{i} x_j\right)^2$
16	[-30, 30]	30	UN	Rosenbrock	$f(x) = \sum_{i=1}^{n} \left[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2\right]$
17	[-10, 10]	30	UN	Dixon - Price	$f(x) = (x_i - 1)^2 + \sum_{i=2}^{n} i(2x_i^2 - x_{i-1})^2$
18	[-65.536, 65.536]	2	MS	Foxholes	$f(x) = \left[\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^{2} (x_i - a_{ij})^6}\right]^{-1}$
19	[-5, 10] x [0, 15]	2	MS	Branin	$f(x) = \left(x_2 - \frac{5.1}{4\pi^2} x_1^2 + \frac{5}{\pi} x_1 - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos x_1 + 10$
20	[-100, 100]	2	MS	Bohachevsky1	$f(x) = x_1^2 + 2x_2^2 - 0.3\cos(3\pi x_1) - 0.4\cos(4\pi x_2) + 0.7$
21	[-10, 10]	2	MS	Booth	$f(x) = (x_1 + 2x_2 - 7)^2 + (2x_1 + x_2 - 5)^2$
22	[-5.12, 5.12]	30	MS	Rastrigin	$f(x) = \sum_{i=1}^{n} \left[x_i^2 - 10\cos(2\pi x_i) + 10\right]$

23	[-500, 500]	30	MS	Schwefel	$f(x)=\sum_{i=1}^{n}-x_i\sin\left(\sqrt{\lvert x_i\rvert}\right)$
24	[0, π]	2	MS	Michalewicz2	$f(x)=-\sum_{i=1}^{n}\sin(x_i)\left(\sin\left(i x_i^2/\pi\right)\right)^{2m}$ $m=10$
25	[0, π]	5	MS	Michalewicz5	$f(x)=-\sum_{i=1}^{n}\sin(x_i)\left(\sin\left(i x_i^2/\pi\right)\right)^{2m}$ $m=10$
26	[0, π]	10	MS	Michalewicz10	$f(x)=-\sum_{i=1}^{n}\sin(x_i)\left(\sin\left(i x_i^2/\pi\right)\right)^{2m}$ $m=10$
27	[-100, 100]	2	MN	Schaffer	$f(x)=0.5+\frac{\sin^2\left(\sqrt{x_1^2+x_2^2}\right)-0.5}{\left(1+0.001\left(x_1^2+x_2^2\right)\right)^2}$
28	[-5, 5]	2	MN	Six Hump Camel Back	$f(x)=4x_1^2-2.1x_1^4+\frac{1}{3}x_1^6+x_1x_2-4x_2^2+4x_2^4$
29	[-100, 100]	2	MN	Bohachevsky2	$f(x)=x_1^2+2x_2^2-0.3\cos(3\pi x_1)(4\pi x_2)+0.3$
30	[-100, 100]	2	MN	Bohachevsky3	$f(x)=x_1^2+2x_2^2-0.3\cos(3\pi x_1+4\pi x_2)+0.3$
31	[-10, 10]	2	MN	Shubert	$f(x)=\left(\sum_{i=1}^{5}i\cos((i+1)x_1+i)\right)((i+1)x_2+i)$
32	[-2, 2]	2	MN	GoldStein-Price	$f(x)=\left[1+(x_1+x_2+1)^2\left(19-14x_1+3x_1^2-14x_2+6x_1x_2+3x_2^2\right)\right]$ $\left[30+(2x_1-3x_2)^2\left(18-32x_1+12x_1^2+48x_2-36x_1x_2+27x_2^2\right)\right]$
33	[-5, 5]	4	MN	Kowalik	$f(x)=\sum_{i=1}^{11}\left(a_i-\frac{x_1\left(b_i^2+b_ix_2\right)}{b_i^2+b_ix_3+x_4}\right)^2$
34	[0, 10]	4	MN	Shekel5	$f(x)=-\sum_{i=1}^{5}\left[(x-a_i)(x-a_i)^T+c_i\right]^{-1}$
35	[0, 10]	4	MN	Shekel7	$f(x)=-\sum_{i=1}^{7}\left[(x-a_i)(x-a_i)^T+c_i\right]^{-1}$
36	[0, 10]	4	MN	Shekel10	$f(x)=-\sum_{i=1}^{10}\left[(x-a_i)(x-a_i)^T+c_i\right]^{-1}$
37	[-D, D]	4	MN	Perm	$f(x)=\sum_{k=1}^{n}\left[\sum_{i=1}^{n}\left(i^k+\beta\right)\left((x_i/i)^k-1\right)\right]^2$
38	[0, D]	4	MN	PowerSum	$f(x)=\sum_{k=1}^{n}\left[\left(\sum_{i=1}^{n}x_i^k\right)-b_k\right]^2$
39	[0, 1]	3	MN	Hartman3	$f(x)=-\sum_{i=1}^{4}c_i\exp\left[-\sum_{j=1}^{3}a_{ij}\left(x_j-p_{ij}\right)^2\right]$
40	[0, 1]	3	MN	Hartman6	$f(x)=-\sum_{i=1}^{4}c_i\exp\left[-\sum_{j=1}^{6}a_{ij}\left(x_j-p_{ij}\right)^2\right]$
41	[-600, 600]	30	MN	Griewank	$f(x)=\frac{1}{4000}\sum_{i=1}^{n}x_i^2-\prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right)+1$
42	[-32, 32]	30	MN	Ackley	$f(x)=-20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n}x_i^2}\right)-\exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right)+$ $20+e$

43	[-50, 50]	30	MN	Penalized	$f(x)=\frac{\pi}{n}\left\{10\sin^2(\pi y_1)+\sum_{i=1}^{n-1}(y_i-1)^2\left[1+10\sin^2(\pi y_{i+1})\right]+(y_n-1)^2\right\}$ $+\sum_{i=1}^{n}u(x_i,10,100,4)$ $y_i=1+\frac{1}{4}(x_i+1)$ $u(x_i,a,k,m)=\begin{cases}k(x_i-a)^m, & x_i>a\\ 0, & -a\le x_i\le a\\ k(-x_i-a)^m, & x_i<-a\end{cases}$
44	[-50, 50]	30	MN	Penalized2	$f(x)=0.1\left\{\sin^2(\pi x_1)+\sum_{i=1}^{n-1}(x_i-1)^2\left[1+\sin^2(3\pi x_{i+1})\right]+(x_n-1)^2\left[1+\sin^2(2\pi x_n)\right]\right\}+$ $\sum_{i=1}^{n}u(x_i,5,100,4)$
45	[0, 10]	2	MN	Langerman2	$f(x)=-\sum_{i=1}^{m}c_i\left(\exp\left(-\frac{1}{\pi}\sum_{j=1}^{n}(x_j-a_{ij})^2\right)\cos\left(\pi\sum_{j=1}^{n}(x_j-a_{ij})^2\right)\right)$
46	[0, 10]	5	MN	Langerman5	$f(x)=-\sum_{i=1}^{m}c_i\left(\exp\left(-\frac{1}{\pi}\sum_{j=1}^{n}(x_j-a_{ij})^2\right)\cos\left(\pi\sum_{j=1}^{n}(x_j-a_{ij})^2\right)\right)$
47	[0, 10]	10	MN	Langerman10	$f(x)=-\sum_{i=1}^{m}c_i\left(\exp\left(-\frac{1}{\pi}\sum_{j=1}^{n}(x_j-a_{ij})^2\right)\cos\left(\pi\sum_{j=1}^{n}(x_j-a_{ij})^2\right)\right)$
48	[-π, π]	2	MN	FletcherPowell2	$f(x)=\sum_{i=1}^{n}(A_i-B_i)^2$ $A_i=\sum_{j=1}^{n}(a_{ij}\sin\alpha_j+b_{ij}\cos\alpha_j)$ $B_i=\sum_{j=1}^{n}(a_{ij}\sin x_j+b_{ij}\cos x_j)$
49	[-π, π]	5	MN	FletcherPowell5	$f(x)=\sum_{i=1}^{n}(A_i-B_i)^2$ $A_i=\sum_{j=1}^{n}(a_{ij}\sin\alpha_j+b_{ij}\cos\alpha_j)$ $B_i=\sum_{j=1}^{n}(a_{ij}\sin x_j+b_{ij}\cos x_j)$
50	[-π, π]	10	MN	FletcherPowell10	$f(x)=\sum_{i=1}^{n}(A_i-B_i)^2$ $A_i=\sum_{j=1}^{n}(a_{ij}\sin\alpha_j+b_{ij}\cos\alpha_j)$ $B_i=\sum_{j=1}^{n}(a_{ij}\sin x_j+b_{ij}\cos x_j)$
51	[-100, 100]	30		Schwefel	$f(x)=\max_{i=\overline{1,n}}\{\lvert x_i\rvert\}$

3.4.1. Први експеримент

У раду Karaboga и Bahriye (2009) аутори су добили решења након 500000 генерација. Аутори су користили 50 јединки у свакој генерацији. Да би имали приближно исте услове експеримента у овој студији за број пчела узета вредност је 50, за број итерација 10000 и за број промена по итерацији 50.

У првом експерименту вршено је 30 решавања сваке функције BCO алгоритмом. Из добијених резултата одређена је средња вредност, стандардна девијација и SEM за сваку функцију. Добијени резултати су приказани у Табели 3.2, с тим што су резултати за метахеуристике GA, DE, PSO и ABC преузети из рада Karaboga и Bahriye (2009).

Табела 3.2. Резултати првог експеримента

Функција	Мин.		GA	PSO	DE	ABC	BCO
1	0	Mean	0	0	0	0	0
		StdDev	0	0	0	0	0
		SEM	0	0	0	0	0
2	0	Mean	1170	0	0	0	0
		StdDev	76.56145	0	0	0	0
		SEM	13.978144	0	0	0	0
3	0	Mean	1110	0	0	0	7.58145E-11
		StdDev	74.214474	0	0	0	2.65698E-11
		SEM	13.549647	0	0	0	4.85097E-12
4	0	Mean	148	0	0	0	2.92851E-10
		StdDev	12.4092893	0	0	0	1.25473E-10
		SEM	2.265616	0	0	0	2.29082E-11
5	0	Mean	0.1807	0.00115659	0.0013633	0.0300166	5.28384E-05
		StdDev	0.027116	0.000276	0.000417	0.004866	2.03331E-05
		SEM	0.004951	0.0000504	0.0000761	0.000888	3.71231E-06
6	0	Mean	0	0	0	0	1.33701E-12
		StdDev	0	0	0	0	1.7955E-12
		SEM	0	0	0	0	3.27812E-13
7	-1	Mean	-1	-1	-1	-1	-1
		StdDev	0	0	0	0	1.72469E-11
		SEM	0	0	0	0	3.14885E-12
8	0	Mean	0	0	0	0	3.98899E-14
		StdDev	0	0	0	0	4.06916E-14
		SEM	0	0	0	0	7.42924E-15
9	0	Mean	0.014938	0	0.0409122	0.0929674	3.79251E-08
		StdDev	0.007364	0	0.081979	0.066277	4.75467E-08
		SEM	0.001344	0	0.014967	0.0121	8.6808E-09
10	-50	Mean	-49.9999	-50	-50	-50	-49.9997398

		StdDev	0.0000225	0	0	0	0.000400312
		SEM	0.00000411	0	0	0	7.30867E-05
11	-210	Mean	-209.476	-210	-210	-210	-209.959472
		StdDev	0.193417	0	0	0	0.011587643
		SEM	0.035313	0	0	0	0.002115605
12	0	Mean	0.013355	0	0	0.0002476	2.28926E-09
		StdDev	0.004532	0	0	0.000183	2.1658E-09
		SEM	0.000827	0	0	0.0000334	3.95419E-10
13	0	Mean	9.703771	0.00011004	0.000000217	0.0031344	4.08028E-07
		StdDev	1.547983	0.00016	0.000000136	0.000503	3.84256E-07
		SEM	0.282622	0.0000292	2.48E-08	0.0000918	7.01552E-08
14	0	Mean	0	0	0	0	2.0557E-05
		StdDev	1.386856	0	0	0	3.01608E-06
		SEM	0.253204	0	0	0	5.50658E-07
15	0	Mean	7400	0	0	0	2.87601E-08
		StdDev	1140	0	0	0	2.19555E-08
		SEM	208.1346	0	0	0	4.0085E-09
16	0	Mean	196000	15.088617	18.203938	0.0887707	10.57166275
		StdDev	38500	24.170196	5.036187	0.07739	16.58267899
		SEM	7029.106155	4.412854	0.033333	0.014129	3.027569116
17	0	Mean	1220	0.66666667	0.6666667	0	0.450003633
		StdDev	266	0.00000001	0.000000001	0	0.152565513
		SEM	48564733	1.8257E-09	1.8257E-10	0	0.027854524
18	0.998	Mean	0.998004	0.99800393	0.9980039	0.9980039	0.998003838
		StdDev	0	0	0	0	3.55295E-16
		SEM	0	0	0	0	6.48677E-17
19	0.398	Mean	0.397887	0.39788736	0.3978874	0.3978874	0.397887387
		StdDev	0	0	0	0	6.35409E-08
		SEM	0	0	0	0	1.16009E-08
20	0	Mean	0	0	0	0	6.18629E-06
		StdDev	0	0	0	0	9.13214E-06
		SEM	0	0	0	0	1.66729E-06
21	0	Mean	0	0	0	0	6.44059E-14
		StdDev	0	0	0	0	9.80376E-14
		SEM	0	0	0	0	1.78991E-14
22	0	Mean	52.92259	43.9771369	11.716728	0	3.84426E-09
		StdDev	4.56486	11.728676	2.538172	2.538172	1.14559E-09
		SEM	0.833426	2.141353	0.463405	0.463405	2.09154E-10
23	-12569.5	Mean	-11593.4	-6909.1359	-10266	-12569.487	-12569.4866
		StdDev	93.254224	457.957783	521.849292	0	1.00007E-08
		SEM	17.025816	83.611269	95.276209	0	1.82588E-09
24	-1.8013	Mean	-1.8013	-1.5728692	-1.801303	-1.8013034	-1.80130341
		StdDev	0	0.11986	0	0	3.93507E-11
		SEM	0	0.021883	0	0	7.18443E-12
25	-4.6877	Mean	-4.64483	-2.4908728	-4.683482	-4.6876582	-4.687658179
		StdDev	0.09785	0.256952	0.01529	0	8.04178E-10
		SEM	0.017865	0.046913	0.002287	0	1.46822E-10

26	-9.6602	Mean	-9.49683	-4.007183	-9.591151	-9.6601517	-9.66015171
		StdDev	0.141116	0.502628	0.064205	0	4.23107E-09
		SEM	0.025764	0.091767	0.011722	0	7.72485E-10
27	0	Mean	0.004239	0	0	0	4.86463E-15
		StdDev	0.004763	0	0	0	7.63391E-15
		SEM	0.0087	0	0	0	1.39375E-15
28	-1.03163	Mean	-1.03163	-1.0316285	-1.031628	-1.0316285	-1.03162845
		StdDev	0	0	0	0	1.17607E-11
		SEM	0	0	0	0	2.14719E-12
29	0	Mean	0.06829	0	0	0	1.63206E-13
		StdDev	0.078216	0	0	0	2.27566E-13
		SEM	0.01428	0	0	0	4.15477E-14
30	0	Mean	0	0	0	0	6.73076E-12
		StdDev	0	0	0	0	7.43862E-12
		SEM	0	0	0	0	1.3581E-12
31	-186.73	Mean	-186.731	-186.73091	-186.7309	-186.73091	-186.730909
		StdDev	0	0	0	0	2.00751E-10
		SEM	0	0	0	0	3.6652E-11
32	3	Mean	5.250611	3	3	3	3
		StdDev	5.870093	0	0	0	3.49952E-10
		SEM	1.071727	0	0	0	6.38922E-11
33	0.00031	Mean	0.005615	0.00049062	0.0004266	0.0004266	0.000307486
		StdDev	0.008171	0.000366	0.000273	0.0000604	4.34546E-10
		SEM	0.001492	0.0000668	0.0000498	0.000011	7.93369E-11
34	-10.15	Mean	-5.66052	-2.0870079	-10.1532	-10.1532	-10.1531997
		StdDev	3.866737	1.17846	0	0	3.60383E-10
		SEM	0.705966	0.215156	0	0	6.57967E-11
35	-10.4	Mean	-5.34409	-1.9898713	-10.40294	-10.402941	-10.4029406
		StdDev	3.517134	1.420602	0	0	3.78657E-10
		SEM	0.642138	0.259365	0	0	6.9133E-11
36	-10.53	Mean	-3.82984	-1.8796753	-10.53641	-10.53641	-10.5364098
		StdDev	2.451956	0.432476	0	0	2.50189E-10
		SEM	0.447664	0.078959	0	0	4.56781E-11
37	0	Mean	0.302671	0.03605158	0.0240069	0.0411052	0.000929463
		StdDev	0.193254	0.048927	0.046032	0.023056	0.001642492
		SEM	0.035283	0.008933	0.008404	0.004209	0.000299877
38	0	Mean	0.010405	11.3904479	0.0001425	0.0029468	9.73981E-07
		StdDev	0.009077	7.3558	0.000145	0.0029468	1.09971E-06
		SEM	0.001657	1.342979	0.0000265	0.002289	2.00778E-07
39	-3.86	Mean	-3.86278	-3.6333523	-3.862782	-3.8627821	-3.86278215
		StdDev	0	0.116937	0	0	7.0696E-11
		SEM	0	0.02135	0	0	1.29073E-11
40	-3.32	Mean	-3.29822	-1.8591298	-3.226881	-3.3219952	-3.27777000
		StdDev	0.05013	0.439958	0.047557	0.047557	0.053700841
		SEM	0.009152	0.080325	0.008683	0.008683	0.009804387
41	0	Mean	10.63346	0.01739118	0.0014792	0	0.012713431
		StdDev	1.161455	0.020808	0.002958	0	0.017814131

		SEM	0.212052	0.003799	0.00054	0	0.0032524
42	0	Mean	14.67178	0.16462236	0	0	7.35252E-06
		StdDev	0.178141	0.493867	0	0	1.55191E-06
		SEM	0.032524	0.0207338	0	0	2.83339E-07
43	0	Mean	13.3772	0.0207338	0	0	1.77429E-11
		StdDev	1.448726	0.041468	0	0	2.41315E-11
		SEM	0.2645	0.007571	0	0	4.40579E-12
44	0	Mean	125.0613	0.00767535	0.0021975	0	2.45571E-10
		StdDev	12.001204	0.016288	0.004395	0	1.80035E-10
		SEM	2.19111	0.002974	0.000802	0	3.28697E-11
45	-1.08	Mean	-1.08094	-0.67268	-1.080938	-1.0809384	-3.06774756
		StdDev	0	0.274621	0	0	5.27957E-11
		SEM	0	0.050139	0	0	9.63913E-12
46	-1.5	Mean	-0.96842	-0.5048579	-1.499999	-0.93815	-1.49994381
		StdDev	0.287548	0.213626	0	0.000208	5.73078E-08
		SEM	0.052499	0.039003	0	0.000038	1.04629E-08
47	NA	Mean	-0.63644	-0.0025656	-1.0528	-0.4460925	-1.17242327
		StdDev	0.374682	0.003523	0.302257	0.133958	0.294476459
		SEM	0.068407	0.000643	0.055184	0.024457	0.0537638
48	0	Mean	0	0	0	0	9.44478E-12
		StdDev	0	0	0	0	2.26617E-11
		SEM	0	0	0	0	4.13745E-12
49	0	Mean	0.004303	1457.88344	5.988783	0.1735495	8.11416E-06
		StdDev	0.009469	1269.362389	7.334731	0.068175	1.08904E-05
		SEM	0.001729	231.752805	1.339133	0.012447	1.98831E-06
50	0	Mean	29.57348	1364.45555	781.55028	8.2334401	124.8629713
		StdDev	16.02078	1325.379655	1048.813487	8.092742	286.3093074
		SEM	2.925035	1325.379655	241.980111	1.477526	52.27268869

Да би одредили да ли постоји значајна разлика у резлтатима добијеним ВСО техником у односу на резултате добијене другим метахеуристикама урађен је Студентов *t*-тест. Вредност *t* статистике се може одредити на следећи начин (Вукадиновић и Поповић, 2008):

$$t = \frac{\overline{X}_1 - \overline{X}_2}{\sqrt{\frac{(n_1 - 1)SD_1^2 + (n_2 - 1)SD_2^2}{n_1 + n_2 - 2} \frac{n_1 + n_2}{n_1 n_2}}} \quad (3.7)$$

где је:

$\overline{X}_1$ - средња вредност првог узорка,

SD_1 - стандардна девијација првог узорка,

$\overline{X}_2$ - средња вредност другог узорка,

SD_2 - стандардна девијација другог узорка,

n_1 - величина првог узорка и

n_2 - величина другог узорка.

Вредности $\overline{X}_2$ и SD_2 су добијене BCO алгоритмом, док су вредности $\overline{X}_1$ и SD_1 добијене другим алгоритмима (GA, PSO, DE и ABC). За интервал поверења је узета вредност 95% ($t_{0.05} = 1{,}96$). У случају када је $|t| > 1{,}96$, разлика између две вредности је значајна. Када t има позитивну вредност, решење BCO алгоритма је боље. Када t има негативну вредност то значи да је решење конкурентске метахеуристике боље. Случај када је $|t| < 1.96$ одговара ситуацији када разлика између добијених вредности није значајна. У Табели 3.3 су приказане ситуације када су поједине метахеуристике пронашле значајно боља решења.

Табела 3.3. Метахеуристике које су пронашле значајно боља решења

Функција	BCO - GA	BCO - PSO	BCO - DE	BCO - ABC
1	-	-	-	-
2	BCO	-	-	-
3	BCO	-	-	-
4	BCO	-	-	-
5	BCO	-	-	BCO
6	-	-	-	-
7	-	-	-	-
8	-	-	-	-
9	BCO	-	BCO	BCO
10	-	-	-	-
11	BCO	PSO	DE	-
12	BCO	-	-	BCO
13	BCO	-	-	BCO
14	-	-	-	-
15	BCO	-	-	-
16	BCO	-	BCO	ABC
17	BCO	BCO	BCO	ABC
18	BCO	BCO	BCO	BCO
19	-	-	-	-
20	-	-	-	-
21	-	-	-	-
22	BCO	BCO	BCO	-
23	BCO	BCO	BCO	ABC
24	BCO	BCO	-	-
25	BCO	BCO	-	-
26	BCO	BCO	BCO	-
27	BCO	-	-	-
28	GA	-	-	-

29	BCO	-	-	-
30	-	-	-	-
31	GA	-	BCO	-
32	BCO	-	-	-
33	BCO	BCO	BCO	BCO
34	BCO	BCO	-	-
35	BCO	BCO	-	-
36	BCO	BCO	-	-
37	BCO	BCO	BCO	BCO
38	BCO	BCO	-	BCO
39	-	BCO	-	-
40	-	BCO	-	-
41	BCO	-	-	-
42	BCO	-	-	-
43	BCO	BCO	-	-
44	BCO	BCO	BCO	-
45	BCO	BCO	BCO	BCO
46	BCO	BCO	-	BCO
47	BCO	BCO	-	BCO
48	-	-	-	-
49	BCO	BCO	BCO	BCO
50	GA	BCO	BCO	ABC

За 13 функција не постоји значајна разлика између BCO и GA. У случају 34 функције BCO алгоритам је остварио боље резултате од GA. GA је пронашао боља решења од BCO у случају 3 функције.

Између BCO и PSO алгоритама не постоји значајна разлика при решавању 27 функција. BCO је био бољи при решавању 22 функције, док је PSO био бољи при решавању једне функције.

Поређење између BCO и DE је показало да не постоји значајна разлика између BCO и DE у случају 35 функција. За 14 функција BCO је био бољи од DE, док је DE био бољи при решавању једне функције.

На крају, када се пореде BCO и ABC, може се уочити да између тих алгоритама не постоји значајна разлика при решавању 34 функције. BCO је остварио боље резултате од ABC за 12 функција, док је ABC пронашао боља решења за 4 функције.

3.4.2. Други експеримент

У другом алгоритму решаване су 23 функције BCO алгоритмом. У радовима чији су аутори Hedar и Fukushima (2006) и Karaboga и Bahriye (2009) ове функције су решаване кроз 100000 генерација. Број јединки у популацији је био 20. У ABC алгоритму је коришћено 20 вештачких пчела. При решавању функција BCO алгоритмом такође је постављено да је број вештачких пчела 20. У BCO алгоритму је дефинисано да је број итерација 5000, а број промена у свакој итерацији 20.

У Табели 3.4 су дати резултати другог експеримента. Резултати добијени алгоритмима CES, FES, ESLAT, CMA-ES и ABC преузети су из радова чији су аутори Hedar и Fukushima (2006) и Karaboga и Bahriye (2009).

Табела 3.4. Резултати другог експеримента

Функција		CES	FES	ESLAT	CMAES	ABC	BCO
3	Mean	1.7E-26	0.00025	2E-17	9.7E-23	0.000757	4.03454E-08
	SD	1.1E-25	0.000068	2.9E-17	3.8E-23	0.000248	1.44103E-08
14	Mean	8.1E-20	0.06	0.000038	4.2E-11	0.000895	0.000121495
	SD	3.6E-19	0.0096	0.000016	7.1E-23	0.000127	2.39181E-05
15	Mean	337.62	0.0014	0.0000061	7.1E-23	0.000701	1.33926E-05
	SD	117.14	0.00053	0.0000075	2.9E-23	0.000278	7.39296E-06
51	Mean	2.41	0.0055	0.78	5.4E-12	2.72	1.38515E-06
	SD	2.15	0.00065	1.64	1.5E-12	1.18	1.34725E-06
16	Mean	27.65	33.28	1.93	0.4	0.936	48.04025793
	SD	0.51	43.13	3.35	1.2	1.76	40.16415276
2	Mean	0	0	0.02	1.44	0	0
	SD	0	0	0.14	1.77	0	0
5	Mean	0.047	0.012	0.39	0.23	0.0906	0.000539556
	SD	0.12	0.0058	0.22	0.087	0.0189	0.00026546
23	Mean	-8E+93	-12556.4	-2.3E+15	-7637.1	-12563.673	-12552.9053
	SD	4.9E+94	32.53	5.7E+15	895.6	23.6	41.51380052
22	Mean	13.38	0.16	4.65	51.78	0.000466	1.67845E-08
	SD	43.15	0.33	5.67	13.56	0.000344	7.51734E-09
42	Mean	6E-13	0.012	0.000000018	6.9E-12	0.000781	6.09131E-05
	SD	1.7E-12	0.0018	5.4E-09	1.3E-12	0.000183	1.41912E-05
41	Mean	6E-14	0.037	0.0014	0.00074	0.000837	0.019122627
	SD	4.2E-13	0.05	0.0047	0.0027	0.00138	0.021086394
43	Mean	1.46	0.0000028	1.5E-12	0.00012	0.000698	2.04747E-09
	SD	3.17	0.00000081	2E-12	0.00034	0.000278	2.15121E-09
44	Mean	2.4	0.000047	0.0064	0.0017	0.000798	4.45677E-08

	SD	0.13	0.000015	0.0089	0.0045	0.000213	3.86834E-08
18	Mean	2.2	1.2	1.77	10.44	0.998	0.998003838
	SD	2.43	0.63	1.37	6.87	0.000321	1.75921E-15
33	Mean	0.0013	0.00097	0.00081	0.0015	0.00118	0.000307487
	SD	0.00063	0.00042	0.00041	0.0042	0.000145	9.04715E-10
28	Mean	-1.031	-1.0316	-1.0316	-1.0316	1.031	-1.03162845
	SD	0.0012	0.0000006	9.7E-14	7.7E-16	0.000304	8.09067E-11
19	Mean	0.401	0.498	0.398	0.398	0.3985	0.397887668
	SD	0.0036	0.00000006	1E-13	1.4E-15	0.000327	5.67035E-07
32	Mean	3.007	3	3	14.34	3	3.000000003
	SD	0.012	0	5.8E-14	25.05	0.000309	3.09352E-09
39	Mean	-38613	-3.86	-3.8628	-3.8628	-3.862	-3.86278215
	SD	0.0012	0.004	2.9E-13	4.8E-16	0.000277	2.19027E-09
40	Mean	-3.24	-3.23	-3.31	-3.28	-3.322	-3.26582313
	SD	0.058	0.12	0.033	0.058	0.000135	0.056160767
34	Mean	-5.72	-5.54	-8.49	-5.86	-10.151	-9.00057412
	SD	2.62	1.82	2.76	3.6	0.0117	2.119259359
35	Mean	-6.09	-6.76	-8.79	-6.58	-10.402	-9.55496662
	SD	2.63	3.01	2.64	3.74	0.000311	1.962694802
36	Mean	-6.42	-7.63	-9.65	-7.03	-10.535	-9.65768072
	SD	2.67	3.27	2.06	3.74	0.00202	1.991488213

Као и у претходном примеру да би се одредило да ли се решења добијена BCO алгоритмом значајно разликују од решења добијених другим приступима урађен је Студентов *t*-тест. У другом експерименту је извршено 50 пропуштања алгоритма за сваку функцију. Коришћен је *t*-тест за случај два велика независна узорка. Вредност *t* статистике се одређује на следећи начин (Вукадиновић и Поповић, 2008):

$$t = \frac{\overline{X}_1 - \overline{X}_2}{\sqrt{\frac{SD_1^2}{n_1 - 1} + \frac{SD_2^2}{n_2 - 1}}} \tag{3.8}$$

За интервал поверења је узето 95% ($t_{0.05} = 1{,}96$). У Табели 3.5 су приказани алгоритми који су пронашли значајно боља решења.

Табела 3.5. Алгоритми који су пронашли значајно боља решења (други експеримент)

Функција	CES	FES	ESLAT	CMA-ES	ABC
3	CES	BCO	ESLAP	CMA-ES	BCO
14	CES	BCO	ESLAP	CMA-ES	BCO
15	BCO	BCO	ESLAP	CMA-ES	BCO
51	BCO	BCO	BCO	CMA-ES	BCO

16	CES	-	ESLAP	CMA-ES	ABC
2	-	-	-	BCO	-
5	BCO	BCO	BCO	BCO	BCO
23	-	-	ESLAP	BCO	-
22	BCO	BCO	BCO	BCO	BCO
42	CES	BCO	ESLAP	CMA-ES	BCO
41	CES	BCO	ESLAP	CMA-ES	ABC
43	BCO	BCO	ESLAP	BCO	BCO
44	BCO	BCO	BCO	BCO	BCO
18	BCO	BCO	BCO	BCO	-
33	BCO	BCO	BCO	BCO	BCO
28	BCO	BCO	BCO	BCO	BCO
19	BCO	BCO	BCO	BCO	BCO
32	BCO	FES	ESLAP	BCO	-
39	BCO	BCO	ESLAP	CMA-ES	BCO
40	BCO	-	ESLAP	-	ABC
34	BCO	BCO	-	BCO	ABC
35	BCO	BCO	-	BCO	ABC
36	BCO	BCO	-	BCO	ABC

Добијени резлтати указују да BCO метахеуристика у великом броју случајева генерише иста или квалитетнија решења у односу на широко распространјене метахеуристике. На основу добијених резултата одлучено је, да се у оквиру ове дисертациије, проблеми комбинаторне оптимизације који се јављају приликом отклањања поремећаја у извршавању планираних редова вожње, редова летења и планова дистрибуције робе решавају применом Оптимизације колонијом пчела.

4. РАСПОРЕЂИВАЊА АВИОНА ПО ПАРКИНГ ПОЗИЦИЈАМА У УСЛОВИМА ПОРЕМЕЋАЈА НАСТАЛИХ КАШЊЕЊЕМ АВИОНА

Повећане захтеве за превозом у ваздушном саобраћају многи аеродроми нису у могућности да задовоље уз одговарајући квалитет услуге. Бољом организацијом и планирањем рада тежи се повећању искоришћења расположивих ресурса аеродрома. Међутим, због различитих случајних утицаја у реализацији великог броја активности на аеродромима често долази до кашњења у реализацији планираних летова. Кашњења у реализацији летова могу да проузрокују велике оперативни трошкове, као и незадовољство путника због смањења квалитета услуге. Процењени укупни трошкови кашњења у Сједињеним Америчким Државама у 2007. години су износили 41 милијарду долара. (Schumer, 2008). Такође је процењено да је неколико хиљада комерцијалних летова у Сједињеним Америчким Државама имало кашњење од најмање 15 минута. Укупно кашњење свих летова је износило 92 милиона минута, узрокујући трошкове од 7,2 милијарди долара.

До кашњења у реализацији планираних летова долази услед поремећаја у реализацији планираних летова, поремећаја у раду посада авиона, поремећаја у коришћењу расположивих ресурса аеродрома, метеоролошких услова, отказа авиона из техничких разлога и др. Планом ротације авиона сваком од авиона из флоте су додељени одређени летови које авион треба да обави током посматраног периода времена. Кашњење при реализацији неког лета директно може проузроковати кашњења на преосталим летовима за које је планирано да их обави посматрани авион. Посаде авиона такође имају планом дефинисане летове на којима обављају летачке дужности. Кашњења једног или више летова могу директно да утичу на кршење одређених ограничења која дефинишу радно време посада, што план рада посада може да учини недопустивим.

Велики обим саобраћаја условљава на већини аеродрома прецизно планирање активности које је потребно реализовати. Планирање рада на аеродромима се, пре свега, врши у циљу повећања искоришћења расположивих ресурса којима аеродром располаже. При великом обиму саобраћаја, кашњења појединих летова могу довести до немогућности проналажења потребних ресурса за опслугом, као и до ометања реализације других активности. Тако се на пример, често дешава да авион који је слетео на аеродром са закашњењем не може да буде опслужен на раније предвиђеној паркинг позицији. Често, планирана паркинг позиција више није доступна, или је доступна током периода времена недовољног да се обави квалитетна опслуга авиона.

У оквиру дисертације је разматран проблем додељивања паркинг позиција авионима у случајевима кашњења појединих авиона. Посебна пажња је посвећена могућности сарадње конкурентских авио компанија при коришћењу паркинг позиција.

4.1. Преглед литературе

Проблем додељивања паркинг позиција авионима (*Gate Assignment Problem*) је тежак проблем комбинаторне оптимизације. Проблем је окарактерисан постојањем више критеријумских функција, као и вишеструким ограничењима. Различите оптимизационе технике (техника гранања и ограничавања (*branch-and-bound*), мешовито целобројно програмирање, модели токова на мрежама), хеуристички и метахеристички алгоритми (Генетски алгоритми, Табу претраживање, Симулирано каљење, Оптимизација колонијом мрава), симулације и експертни системи су коришћени за решавање проблема додељивања паркинг позиција авионима током последњих неколико деценија.

Braaksma и Shortreed (1971) су предложили први модел који минимизира укупна пешачења унутар терминала. Bihr (1980) је разматрао проблем додељивања паркинг позиција у хаб терминалима. Аутори су предложили 0-1 модел целобројног програмирања. Бабић и остали (1984) су предложили модел базиран на техници гранања и ограничавања. Аутори су минимизирали укупно растојање које прелазе путници. Mangoubi и Mathaisel (1985) су користили релаксацију линеарног програмирања и похлепну херистику за решавање проблема додељивања паркинг позиција авионима. Аутори су у предложени модел укључили и различите аспекте проблема које имају транзитни путници на аеродрому. Gosling (1990) је предложио прототип експертног система за додељивање паркинг позиција авионима. Перформансе предложеног експертног система су тестиране у оквиру студије типичног распоређивања летова по паркинг позицијама на великим хаб аеродромима. Srihari и Muthukrishnan (1991) су представили архитектуру система за подршку одлучивању за проблем додељивања паркинг позиција. Hassounah и Steuart (1993) су такође разматрали стохастичке аспекте кашњења летова при решавању проблема додељивања паркинг позиција. Su и Srihari (1993) су предложили експертни систем за проблем додељивања паркинг позиција који помаже при стратешком планирању распореда. Cheng (1997) је развио систем заснован на знању који је повезан са техникама математичког програмирања. Yan и Chang (1998) су предложили модел токова са више врста роба. Vanderstraeten и Bergeron (1998) су аутори хеуристичког алгоритма за дневно коришћење паркинг позиција терминала. Gu и Chung (1999) су за решавање проблема доделе паркинг позиција авионима развили концепт заснован на генетским алгоритмима. Bolat (1999, 2000a, 2000b) је развио математичке моделе који дају релативно робустне распореде авиона по паркинг позицијама. Yan и Huo (2001) су предложили модел целобројног програмирања са две критеријумске функције. Прва критеријумска функција се односи на време пешачења путника. Другом критеријумском функцијом се минимизира укупно време чекања путника. Да би минимизирали кашњења летова Yan и Huo (2001) су у свом моделу користили фиксно резервно време између два узастопна лета која су додељена истој паркинг позицији. Ding и остали (2004а,б) су разматрали проблем додељивања паркинг позиција авионима у случају када је дозвољено кршење задатих ограничења. У овом случају, број летова које треба опслужити је већи од броја паркинг позиција које су на располагању. Аутори су тежили да минимизирају укупан број авиона који нису опслужени на паркинг позицији. Истовремено, аутори су тежили и минимизацији укупног пешачења свих путника. Проблем је решаван применом прождрљивог алгоритма и метахеуристике Табу претраживање. Lim и остали (2005) су разматрали проблем додељивања паркинг позиција са временским интервалима. Аутори су проучавали случај када се времена доласка и одласка авиона на паркинг позиције могу мењати. Приступ за решавање проблема је базиран на Табу претраживању и Memetic алгоритмима. Dorndorf и остали (2007) су извршили преглед развијених модела и техника за решавање проблема распоређивања паркинг позиција. Liu и Wei (2006) су проучавали статичку верзију проблема. Аутори су предложили модел базиран на генетским алгоритмима. Zhang и остали (2007) су се бавили проблемом поновног додељивања паркинг позиција (енг. *gate reassignment problem*). Да би извршили поновно додељивање паркинг позиција у реалном времену, аутори су предложили модел целобројног програмирања. Проблем је решаван применом технике Табу претраживања. Hu и Di Paolo (2007) су третирали проблем додељивања паркинг позиција као вишекритеријумски. Они су предложили Генетски алгоритам са равномерним укрштањем (crossover - ом) као погодним за разматрани проблем. Drexl и Nikulin (2008) су разматрали проблем додељивања паркинг позиција уз коришћење више критеријумских функција. Аутори су тежили да: (а) минимизирају број авиона који немају додељену паркинг позицију; (б) минимизирају укупно растојање пешачења путника и (в) максимизирају укупне приоритете додељивања паркинг позиција летовима. У циљу решавања проблема аутори су користили технике Парето симулираног каљења. Приликом разматрања проблема Dorndorf и остали (2008) су максимизирали укупне приоритете додељивања паркинг позиција летовима, минимизирали број летова којима нису додељене паркинг позиције, минимизирали број померања авиона (у раду је разматрана могућност да долазни и одлазни летови авиона не морају да буду опслужени на истим паркинг позицијама), и максимизирали робустност крајњег распоређивања узимајући у обзир кашњење летова. Pintea и остали (2008) су за решавање проблема додељивања паркинг позиција користили метахеуристику Мрављи систем, при чему су тежили минимизирању времена чекања путника на наредни лет.

Yan и остали (2009) су предложили модел новог додељивања паркинг позиција у случајевима тренутне затворености аеродрома. Аутори су минимизирали укупан број промена паркинг позиција у складу са важећим ограничењима. Предложени модел је тестиран на примеру међународног аеродрома *Chiang Kai-Shek* на Тајвану.

Проблем новог додељивања се јавља када један или више авиона у доласку касне. У случају значајних кашњења, додељене паркинг позиције морају бити модификоване. Gu и Chang (1999) су предложили Генетски алгоритам за решавање проблема новог додељивања паркинг позиција. Аутори су показали да њихов приступ даје боља решења од решења генерисаних од стране искусних диспечера који раде на пословима додељивања паркинг позиција. Tang и остали (2010) су развили GRFRTFD, компјутерски програм за поновно додељивање паркинг позиција. Аутори програма су комбиновали C програмски језик са програмом CPLEX 10.0. Развијени алат је тестиран коришћењем података са аеродрома у Тајвану. Tang (2011) је предложио модел за ново додељивање паркинг позиција који дозвољава кршење ограничења. Перформансе предложеног модела су тестиране на случају аеродрома Taoyuan на Тајвану. Maharjan и Matis (2011) су формулисали математички модел бинарног програмирања за оптимално одређивање новог додељивања авиона паркинг позицијама. Аутори су минимизирали укупно растојање пешачења путника чије су карте издате пре новог додељивања паркинг позиција. Предложени приступ је тестиран на примеру компаније Continental Airlines-а на аеродрому George W. Bush Intercontinental Airport у Хјустону (Тексас).

4.2. Поставка проблема

На многим аеродромима у свету авио компаније изнајмљују паркинг позиције на којима ће да врше опслугу својих авиона. На тај начин доступност паркинг позиција авио компанијама је ограничена на оне које су закупљене. Због тога све авио компаније теже да достигну најбоље искоришћење ресурса са којима располажу. На овај начин се остварују локална оптимална решења која на нивоу аеродрома то нису.

Због кашњења појединих авиона планирани распореди опслуживања на паркинг позицијама могу да постану недопустиви. У оваквим ситуацијама потребно је извршити одређивање нових планова паркирања авиона узимајући у обзир настале околности. Због ограниченог броја паркинг позиција које су закупиле, неке авио компаније могу да буду у позицији да не могу да избегну велика чекања авиона на почетак опслуге. Као могући концепт за смањење временских губитака у опслуживању авиона на аеродромима разматрана је могућа сарадња (колаборација) између авио компанија при коришћењу паркинг позиција (*Colaborative Gate Allocation* (CGA)). Основу овог концепта представља идеја да у одређеним околностима авио компаније могу да користе паркинг позиције које су друге компаније закупиле, како би минимизирале временске губитке који настају уколико у тренутку доласка авиона авио компанија нема слободну паркинг позицију. На тај начин тежи се минимизирању укупних временских губитака свих компанија на посматраном аеродрому.

И поред позитивних страна концепта сарадње (смањење временских губитака, смањење трошкова, повећање нивоа услуге и др.) поједине авио компаније не желе да сарађују и помажу конкурентским. У теорији и пракси, позната је појава да конкурентске фирме сарађују на одређеном нивоу. Тако на пример Gnyavali и Park (2011) дефинишу „*co-opetition*" као „стратегију која оличава истовремену сарадњу и конкуренцију између фирми". У литератури нема много радова који објашњавају зашто и како долази до сарадње између конкурентских фирми (Garud, 1994; Gnyavali и Park, 2009). Gnyavali и Park (2011) су разматрали факторе који узрокују co-opetition, као и утицај co-opetition стратегије на фирме које учествују. Како наводе Harbison и Pekar (1998), више од 50% стратешких савеза је између конкурената унутар исте делатности. Co-opetition је углавном популарна стратегија у високо-технолошкој индустрији (Dagnino и Rocco, 2009; Gnyavali и Park, 2009). На пример, добро је познато заједничко улагање S-LCD, између Samsung Electronics и Sony Corporation у развоју и производњи LCD телевизора са равним екраном. Одлични примери сарадње између конкурената су и алијансе авио превозника. Многе авио компаније сарађују унутар алијансе да би прошириле мрежу, смањиле оперативне трошкове, смањиле цене за путнике, понудиле флексибилнија времена полазака, и више директних летова ка различитим дестинацијама.

Предложени концепт сарадње авио компанија састоји се у међусобној размени информација у реалном времену између авио компанија и оператера аеродрома и заједничког коришћења паркинг позиција. Ова сарадња омогућава авио компанијама да користе паркинг позиције изван оних које оне поседују, као и да уступају своје другим авио компанијама на коришћење. Циљ овог приступа је да се избегне свако кашњење које би настало као последица непоседовања паркинг позиције у одговарајућем тренутку.

Нека је N скуп летова који треба да буду опслужени, а G скуп паркинг позиција на аеродрому. Летови припадају авио компанијама. Сваки авион може да користи само једну паркинг позицију из скупа G. Паркинг позиције могу да буду подељене у следеће три групе: а) паркинг позиције које могу да користе све авио компаније, б) паркинг позиције које могу да користе само неке авио компаније, ц) паркинг позиције које могу да користе само одређене авио компаније (авио компанија има ексклузивно право).

Проблем који се разматра може се дефинисати на следећи начин: доделити летове паркинг позицијама на начин да се минимизирају укупна временска одступања од планираних времена опслуга, укупан број авиона опслужених у систему сарадње (колаборације) и укупно растојање које прелазе путници.

Временска одступања авиона f_i су једнака (видети Слику 4.1):

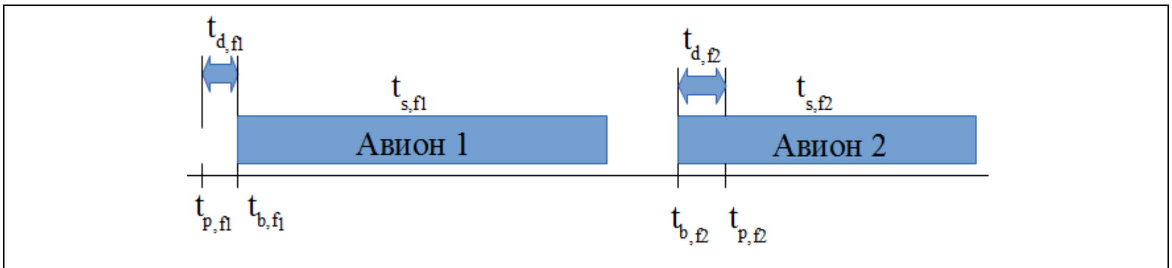
$$t_{d,f_i} = \left| t_{b,f_i} - t_{p,f_i} \right| \tag{4.1}$$

где су:

t_{d,f_i} – временско одступање које се јавља у односу на планирани почетак опслуге авиона f_i,

t_{b,f_i} – време када је почела опслуга авиона f_i,

t_{p,f_i} – планирано време почетка опслуге авиона f_i.



Слика 4.1. Временска одступања при почетку опслуге авиона

У математичкој формулацији проблема су коришћене следеће променљиве:

$$x_{ijk} = \begin{cases} 1, & \text{ако се авион } i \text{ опслужује на паркинг позицији } j \text{ као } k\text{-ти по реду} \\ 0, & \text{у супротном} \end{cases}$$

y_{ijk} - временски тренутак када почиње опслуга лета i на паркинг позицији j као k-тог по реду

$$z_j = \begin{cases} 1, & \text{бинарна променљива која узима вредност 1 ако се паркинг позиција } j \text{ може} \\ & \text{користити за опслуживање у колаборацији} \\ 0, & \text{у супротном} \end{cases}$$

$$u_{ij} = \begin{cases} 1, & \text{ако се авион } i \text{ опслужује на паркинг позицији } j\text{, при чему авион може да} \\ & \text{је користи само ако се она користи у колаборацији} \\ 0, & \text{у супротном} \end{cases}$$

a_i - временско одступање које се јавља ако опслуга авиона i почне након планираног времена почетка опслуге

b_i – временско одступање које се јавља ако опслуга авиона i почне пре планираног времена почетка опслуге

Уведимо у разматрање следеће улазне параметре:

$$f_{ij} = \begin{cases} 1, & \text{ако авион } i \text{ може да буде опслужен на паркинг позицији } j \\ 0, & \text{у супротном} \end{cases}$$

$$h_{ij} = \begin{cases} 1, & \text{ако авио компанија чији је авион } i \text{ може да користи паркинг позицију } j \\ 0, & \text{у супротном} \end{cases}$$

$$r_j = \begin{cases} 1, & \text{ако паркинг позиција } j \text{ може да се користи у колаборацији} \\ 0, & \text{у супротном} \end{cases}$$

A_i - најранији тренутак када може да почне опслуга авиона i

B_j – најранији тренутак када може да почне коришћење паркинг позиције j

T_i - време опслуге авиона i

C_{ij} - укупно растојање пешачења путника авиона i при чему се опслуга врши на паркинг позицији j

D_i - максимално дозвољено кашњење почетка опслуге авиона i

P_i - планирани тренутак почетка опслуге авиона i

w_i - тежински коефицијент значаја авиона i

Математичка формулација проблема гласи:

Минимизирати

$$F_1 = \sum_{i \in N} w_i \left(a_i + b_i\right) \tag{4.2}$$

Минимизирати

$$F_2 = \sum_{i \in N} \sum_{j \in G} u_{ij} \tag{4.3}$$

Минимизирати

$$F_3 = \sum_{i \in N} \sum_{j \in G} \sum_{k=1}^{|N|} C_{ij}\ x_{ijk} \tag{4.4}$$

при ограничењима:

$$\sum_{j \in G} \sum_{k=1}^{|N|} x_{ijk} = 1 \quad \forall\, i \in N \tag{4.5}$$

$$\sum_{i \in N} x_{ijk} \leq 1 \quad \forall\ j \in G, \forall\ k = \overline{1, |N|} \tag{4.6}$$

$$\sum_{k=1}^{|N|} x_{ijk} \leq f_{ij} \quad \forall\, i \in N, \forall\ j \in G \tag{4.7}$$

$$\sum_{k=1}^{|N|} x_{ijk} \leq h_{ij} + u_{ij} \quad \forall\, i \in N, \forall\ j \in G \tag{4.8}$$

$$z_j \leq r_j \quad \forall\ j \in G \tag{4.9}$$

$$u_{ij} \leq z_j \quad \forall\, i \in N, \forall\ j \in G \tag{4.10}$$

$$y_{ijk} \le M\ x_{ijk} \quad \forall\ i \in N, \forall\ j \in G, \forall\ k = \overline{1, |N|} \tag{4.11}$$

$$A_i\ x_{ijk} \le y_{ijk} \quad \forall\ i \in N, \forall\ j \in G, \forall\ k = \overline{1, |N|} \tag{4.12}$$

$$B_j\ x_{ijk} \le y_{ijk} \quad \forall\ i \in N, \forall\ j \in G, \forall\ k = \overline{1, |N|} \tag{4.13}$$

$$y_{ijk} + T_i\ x_{ijk} \le E_j \quad \forall\ i \in N, \forall\ j \in G, \forall\ k = \overline{1, |N|} \tag{4.14}$$

$$\sum_{i \in N} \left(y_{ijk-1} + T_r\ x_{ijk-1}\right) - \sum_{i \in N} \left(y_{ijk} + \left(1 - x_{ijk}\right) M\right) \le 0 \quad \forall\ j \in G, \forall\ k = \overline{2, |N|} \tag{4.15}$$

$$\sum_{i \in N} x_{ijk-1} - \sum_{i \in N} x_{ijk} \ge 0 \quad \forall\ j \in \mathrm{G},\ \forall\ k = \overline{2, |N|} \tag{4.16}$$

$$\sum_{j \in G} \sum_{k=1}^{|N|} y_{ijk} - P_i = a_i - b_i\ \forall \quad i \in N \tag{4.17}$$

$$a_i + b_i \le D_i \quad \forall\ i \in N \tag{4.18}$$

$$x_{ijk} = \{0, 1\} \quad \forall\ i \in N, \forall\ j \in G, \forall\ k = \overline{1, |N|} \tag{4.19}$$

$$z_j = \{0, 1\} \quad \forall\ j \in G \tag{4.20}$$

Проблем дефинисан (4.2)-(4.20) је проблем вишекритеријумског мешовитог целобројног програмирања. Критеријумска функција (4.2) коју желимо да минимизирамо представља укупна временска одступања. Критеријумска функција (4.3) која се такође минимизира репрезентује укупан број авиона који ће да буду опсплужени у систему колаборације. Када се рачуна укупан број авиона који ће бити опслужени у систему колаборације узима се у обзир и значај који имају поједини авиони (летови). Укупно растојање које прелазе путници и које желимо да минимизирамо је дефинисано критеријумском функцијом (4.4).

Ограничење (4.5) гарантује да ће сваки авион бити опслужен на једној од паркинг позиција. У сваком тренутку на паркинг позицији може да се опслужује само један авион. Ово је дефинисано ограничењем (4.6). Ограничење (4.7) дефинише да авион може да се опслужује на паркинг позицији само ако постоји њихова компатибилност. Авион може да се опслужи на паркинг позицији ако авио компанија има право да користи посматрану паркинг позицију или ако се паркинг позиција налази у систему колаборације. Ово је дефинисано ограничењем (4.8). Ограничење (4.9) дефинише да паркинг позиција може да буде коришћена у сврху колаборације само ако је то унапред допуштено. Авион може да буде опслужен у систему колаборације само ако је паркинг позиција у систему колаборације. Ово је дефинисано ограничењем (4.10). Уколико ће авион i бити опслужен на паркинг позицији j као k-ти по реду, променљива y_{ijk} ће узети вредност времена почетка опслуге, у супротном узеће вредност једнаку нули. Ограничење (4.12) дефинише да почетак опслуге не може да се деси пре најранијег тренутка када авион може да дође на паркинг позицију. Слично, ограничење (4.13) означава да авион не може бити опслужен на паркинг позицији пре него што она не почне са радом. Ограничење (4.14) гарантује да опслуга авиона мора да буде завршена пре него што паркинг позиција престане да буде допустива за опслуживање. Ограничењем (4.15) се гарантује да опслуга авиона на паркинг позицији може да почне тек након што је опслуга претходног авиона на истој паркинг позицији завршена. Након опслуге k-тог лета на паркинг позицији може да почне опслуга једино $k+1$ лета. Ово је дефинисано ограничењем (4.16).

Израчунавање укупних временских губитака је омогућено ограничењем (4.17). Ограничење (4.18) гарантује да укупни временски губици по било ком лету не могу бити већи од неких унапред дефинисаних. Ограничењима (4.19) и (4.20) се дефинишу променљиве x_{ijk} и z_j као бинарне.

4.3. Решавање проблема применом Оптимизације колонијом пчела

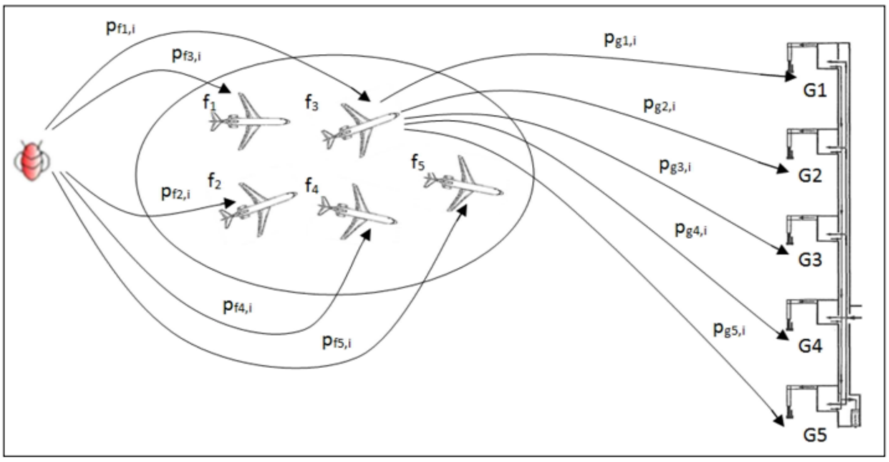
За решавање проблема сарадње авио компанија у додељивању паркинг позиција коришћена је метахеуристика Оптимизација колонијом пчела. Верзија BCO метахеуристике заснована на побољшавању решења је прилагођена разматраном проблему. На почетку алгоритма се генерише почетно решење, које вештачке пчеле кроз итерације побољшавају. Свака итерација се састоји од NP летова унапред у којима пчеле врше модификације решења. При лету уназад пчеле сарађују, при чему доносе одлуке да ли остају лојалне својим решењима или ће да се придруже некој другој пчели.

4.3.1. Генерисање почетног решења

На почетку процеса претраге сви летови су недодељени. Означимо са *F* скуп неопслужених летова и са *G* скуп паркинг позиција. Псеудо код процедуре за генерисање почетног решења гласи:

процедура Додељивање (скуп авиона F, скуп паркинг позиција G)

1: ***for*** *i = 1* ***to*** *број нераспоређених авиона* ***do***
2: *Са вероватноћом P изабрати најранији авион по доласку* f_k*, у супротном за сваки авион* $f_j \in F$ *израчунати вероватноћу да буде изабран. Користећи рулет изабрати један авион:* f_k.
3: *Узимајући у обзир постојећа ограничења одредити скуп паркинг позиција* G_{f_k} .
4: *За сваку паркинх позицију из скупа* G_{f_k} *(* $g_r \in G_{f_k}$ *) израчунати вероватноћу* $p_{g_r,i}$ *да се авион* f_k *опслужује на паркинг позицији* g_r.
5: *Коришћењем рулета одредити паркинг позицију* $g_{r,j}$ *на којој ће да се опслужује авион* f_k*. Доделити авион* f_k *паркинг позицији* $g_{r,j}$*. (Видети Слику 4.2)*
6: ***for*** *сваку паркинг позицију g из скупа G* ***do***
7: *Позвати процедуру: Побољшање (g)*
8: ***if*** *постоје авиони којима нису додељене паркинг позиције* ***then***
9: *означити их као неопслужене.*



Слика 4.2. Распоређивање авиона на паркинг позиције

Када је генерисано почетно решење, вероватноћа избора авиона $p_{f_j,i}$ је једнака:

$$p_{f_j,i} = \frac{\dfrac{1}{t_{a,f_j}}}{\sum\limits_{f_k \in F} \dfrac{1}{t_{a,f_k}}} \tag{4.21}$$

где је:

t_{a,f_k} - време када ће авион f_k доћи на аеродром.

Вероватноћа избора паркинг позиције $p_{g_r,i}$ је једнака:

$$p_{g_r,i} = \frac{\dfrac{1}{\left|t_{a,f_k} - T_{g_r}\right|}}{\sum\limits_{g_s \in G_{f_k}} \dfrac{1}{\left|t_{a,f_k} - T_{g_s}\right|}} \tag{4.22}$$

где је:

T_{g_k} - кумулативно време за паркинг позицију g_k.

$$T_{g_k} = \begin{cases} t_{g_k,b}, & \text{ако је } k = 0 \\ t_{b,f_k} + t_{s,f_k}, & \text{у супротном} \end{cases} \tag{4.23}$$

где су:

f_k - последњи авион додељен паркинг позицији g_k,

k – тренутни број авиона додељених паркинг позицији g_k.

Распоред авиона по паркинг позицијама се унапређује применом процедуре дефинисане следећим псеудо кодом:

процедура Побољшање (Паркинг позиција g)

1: У скуп F додати све авионе додељене паркинг позицији.

2: Очистити листу авиона додељених паркинг позицији.

3: Сортирати авионе из скупа F у растућем поретку према временима долазака.

*4: **for** i = 1 **to** |F| **do***

5: Узети први авион f_i из скупа F.

6: Доделити авион f_i паркинг позицији и одредити време почетка опслуге.

7: Избрисати авион f_i из скупа F.

Време почетка опслуге може се дефинисати у оквиру процедуре побољшања на следећи начин (видети Слику 4.1):

$$t_{b,i} = \begin{cases} T_g, & \text{ако је } \max\{t_{a,f_i}, t_{p,f_i}\} \le T_g \\ T_g + random(0,1)\left(\max\{t_{a,f_i}, t_{p,f_i}\} - T_g\right), & \text{у супротном} \end{cases} \tag{4.24}$$

где је:

k претходник лета *i* који је додељен истој паркинг позицији.

4.3.2. Модификовање решења

Током лета унапред пчеле модификују почетно решење. Процедура за модификовање решења може бити представљена следећим псеудо кодом:

процедура Модификовање (int k, скуп летова F, скуп паркинг позиција G)

1: Означити са F_1 празан скуп авиона.

2: *for* *i = 1* ***to*** *k*

3: Са вероватноћом реципрочном броју додељених авиона изабрати једну паркинг позицију: g.

4: Додати у скуп F_1 све авионе додељене паркинг позицији g.

5: Избрисати све авионе из листе задатака који се опслужују на паркинг позицији g.

6: Позвати процедуру: Додељивање(F_1, G)

где је:

k унапред дефинисано.

За претраживање скупа допустивих решења су коришћена три типа пчела. Типови вештачких пчела се међусобно разликују по начину на који рачунају вероватноћу $p_{g_r,i}$ у процедури за додељивање авиона. Пчела првог типа рачуна веоватноћу на исти начин као и при одређивању почетног решења (израз (4.21)). Пчела другог типа рачуна вероватноћу као:

$$p_{g_r, f_k} = \frac{1 + h_{f_k g_r}}{\sum_{g_s \in G_{f_k}} 1 + h_{f_k g_s}}, \tag{4.25}$$

Трећи тип пчеле рачуна вероватноћу као:

$$p_{g_r, f_k} = \frac{C_{f_k g_r}}{\sum_{g_s \in G_{f_k}} C_{f_k g_s}} \tag{4.26}$$

где су:

h и C дефинисани у поглављу 4.2.

4.3.3. Провера квалитета тренутног најбољег решења и рачунање квалитета решења пчела

Као додатна критеријумска функција (F_0) у моделу је коришћен број неопслужених авиона. Додатна критеријумска функција је уведена у разматрање с обзиром да хеуристички алгоритам може да генерише решење у оквиру кога неки од авиона нису распоређени. Одређивање најбољег решења је вршено лексикографским приступом. Пре покретања програма потребно је одредити редослед критеријумских функција (на пример: F_0, F_1, F_2 и F_3). Након тога, за одређивање најбољег решења коришћена је следећа процедура:

процедура одреди најбољу пчелу и провери најбоље решење ()

1: најбољаПчела = 1

*2: **for** i = 2 **to** B*

*3: **if** ((F₀(пчела i) < F₀(најбољаПчела)) **or** (F₀(пчела i) = F₀(најбољаПчела) **and** F₁(пчела i) < F₁ (најбољаПчела)) **or** (F₀(пчела i)=F₀(најбољаПчела) **and** (F₁(пчела i) = F₁ (најбољаПчела) **and** F₂(пчела i) < F₂(најбољаПчела))) **or** (F₀(пчела i) = F₀(најбољаПчела) **and** F₁(пчела i) = F₁(најбољаПчела) **and** F₂(пчела i) = F₂(најбољаПчела) **and** F₃(пчела i)<F₃(најбољаПчела))) **then***

4: најбољаПчела = i

*5: **if** ((F₀(најбољаПчела) < F₀(најбољеРешење)) **or** (F₀(најбољаПчела) = F₀(најбољеРешење) **and** F₁(најбољаПчела) < F₁(најбољеРешење)) **or** (F₀(најбољаПчела) = F₀(најбољеРешење) **and** (F₁(најбољаПчела) =*

F_1(најбољеРешење) ***and*** *F_2(најбољаПчела) < F_2(најбољеРешење)))* ***or*** *(F_0(најбољаПчела) = F_0(најбољеРешење)* ***and*** *F_1(најбољаПчела) = F_1(најбољеРешење)* ***and*** *F_2(најбољаПчела) = F_2(најбољеРешење)* ***and*** *F_3(најбољаПчела) < F_3(најбољеРешење)))****then***

6: Сачувати решење најбоље пчеле као тренутно најбоље решење.

Свако генерисано решење је окарактерисано бројем неопслужених летова, бројем паркинг позиција које учествују у колаборацији, укупним временским губицима и укупним растојањем које прелазе сви путници (Табела 4.1). За добијање једне вредности као мере квалитета решења на основу дефинисаних улаза је коришћена Метода Тежинских Коефицијената (*Simple Additive Weighting* (SAW)).

Табела 4.1. Карактеристике решења које су генерисале пчеле

	Број неопслужених летова	Број паркинг позиција у колаборацији	Укупни временски губици	Укупно растојање које прелазе путници
Пчела 1	$k_{1,1}$	$k_{1,2}$	$k_{1,3}$	$k_{1,4}$
Пчела 2	$k_{2,1}$	$k_{2,2}$	$k_{2,3}$	$k_{2,4}$
Пчела 3	$k_{3,1}$	$k_{3,2}$	$k_{3,3}$	$k_{3,4}$
⋮				
Пчела *n*	$k_{n,1}$	$k_{n,2}$	$k_{n,3}$	$k_{n,4}$
Тежине	w_1	w_2	w_3	w_4

С обзиром да се тежи минимизацији свих критеријумских функција, вредности дате у Табеле 1 нормализоване су следећим изразом:

$$r_{i,j} = \frac{k_{\max,j} - k_{i,j}}{k_{\max,j} - k_{\min,j}} \tag{4.27}$$

где су:

r_{ij} – нормализована вредност решења пчеле *i* узимајући у обзир критеријум *j*

$k_{max,j}$ – највећа нормализована вредност по критеријуму *j*: $k_{\max,j} = \max_{i=\overline{1,n}} \{k_{i,j}\}$

$k_{min,j}$ – најмања нормализована вредност по критеријуму *j*: $k_{\min,j} = \min_{i=\overline{1,n}} \{k_{i,j}\}$

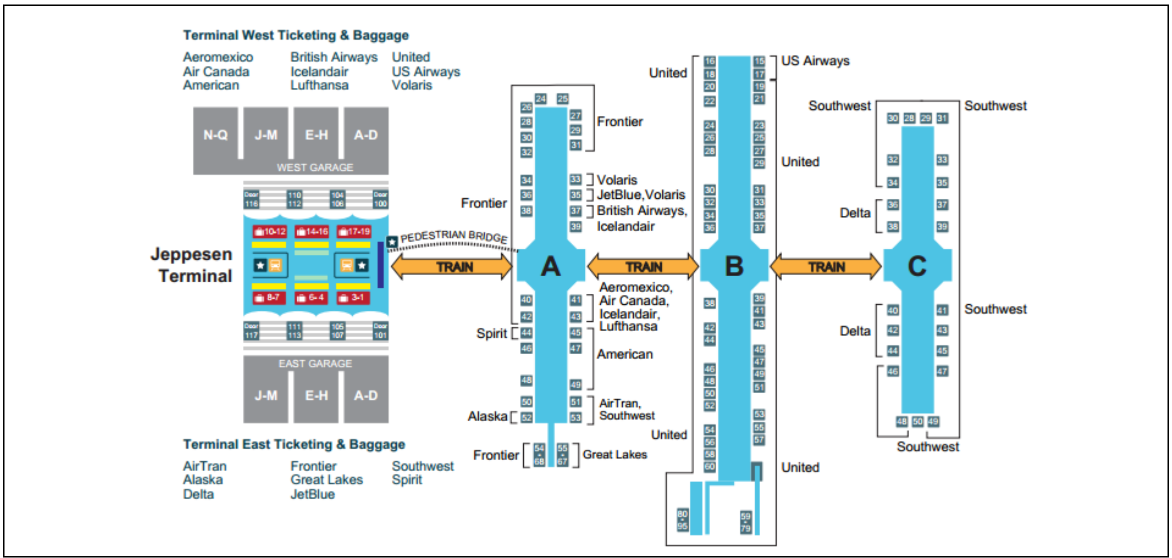
Квалитет *i*-тог решења који је генерисала пчела *b* је једнак:

$$T_{b,i} = \sum_{j=1}^{4} w_j r_{i,j} \tag{4.28}$$

где су w_j дефинисане тежине критеријумских функција.

4.4. Резултати тестирања

Предложена математичка формулација и BCO алгоритам су тестирани на случају Denver International Airport (Слика 4.3). Коришћени су подаци о саобраћају реализованом на дан 16.08.2012. за следеће авио компаније: Air Canada, AirTran Airways, Alaska Airlines, American Airlines, British Airways, Frontier Airlines, Frontier Airlines, Great Lakes Airlines, Icelandair, JetBlue Airways, Lufthansa и Spirit Airlines у оквиру терминала А. На Слици 4.3 је такође приказано које паркинг позиције може да користи свака авио компанија. Укупан број авиона који су разматрани је 315. Претпостављено је да сваки авион може да се опслужује на свакој паркинг позицији. Број путника на сваком лету је симулиран коришћењем равномерне расподеле. Претпостављено је да је број путника по сваком лету између 90 % и 100 % од броја путничких места коришћеног авиона. Коришћењем равномерне расподеле и генератора случајних бројева, за сваки лет је одређен конкретан број путника.



Слика 4.3. Аеродром Денвер (Denver International Airport, 2013)

Урађена су следећа три тестирања. У првом тестирању је коришћена лексикографска метода за решавање математичке формулације (4.2)-(4.20) при чему колаборација није дозвољена. У другом тестирању је такође коришћена лексикографска метода за решавање проблема помоћу математичке формулације (4.2)-(4.20), при чему је колаборација дозвољена. У трећем тесту разматрани проблем је решаван Оптимизацијом колонијом пчела.

Редослед критеријумских функција који је коришћен за решавање овог проблема је F_1, F_2, F_3. У прва два тестирања због димензија проблема није било могуће да се цео проблем реши без декомпозиције. Због тога је проблем решаван из више делова, тј. уочени проблем је решен као више мањих проблема. Сви авиони су сортирани у растућем поретку у односу на време доласка. Сваки мањи проблем који је решаван се састојао од 10 авиона (последњи проблем се састојао од 5 авиона) које је требало распоредити на посматрани број паркинг позиција. Укупан број мањих задатака које је у овом примеру требало решити је 32. Ако се узме у обзир да проблем има три критеријумске функције и да је у лексикографској методи потребно исто толоко пута решити проблем (за сваку критеријумску функцију по једном) може се уочити да је за добијање решења овим приступом било потребно извршити 96 решавања задатака мешовитог целобројног програмирања како у првом тако и у другом тестирању. Такође, једна од мана овог приступа је што се за добијено решење не може гарантовати да је оптимално. У Табели 4.2 дати су резултати прва два тестирања. Као што се може видети примена колаборације омогућила је да се у потпуности редукују временски губици. Укупан број авиона који би био опслужен на паркинг позицијама на којима, да не постоји сарадња не би имао право да се опслужује, је 28, односно 8,89 % од укупног броја авиона.

Табела 4.2. Поређење добијених резултата

Авиони	Без сарадње авио компанија			Са сарадњом авио компанија		
	Временски губици (min)	Број авиона опслужених у систему колаборације	Укупно растојање пешачења	Временски губици (min)	Број летова опслужених у систему колаборације	Укупно растојање пешачења
0 - 9	0	0	465361	0	0	465361
10 - 19	0	0	380833	0	0	380833
20 - 29	32	0	405060	0	1	390874
30 - 39	4	0	646430	0	1	638474
40 - 49	0	0	459431	0	0	459431
50 - 59	9	0	227577	0	1	219153
60 - 69	136	0	244963	0	5	196256
70 - 79	0	0	590476	0	0	590476
80 - 89	5	0	580268	0	0	580268
90 - 99	6	0	329124	0	1	320700
100 - 109	64	0	410997	0	2	387708
110 - 119	0	0	671405	0	0	671405
120 - 129	0	0	516236	0	0	516236
130 - 139	9	0	484133	0	1	475709
140 - 149	0	0	233250	0	0	233250
150 - 159	67	0	244241	0	3	210738
160 - 169	80	0	376894	0	1	369159
170 - 179	164	0	378204	0	4	341836
180 - 189	51	0	548251	0	0	548251
190 - 199	62	0	519062	0	0	519062
200 - 209	0	0	464224	0	0	464224
210 - 219	0	0	344101	0	0	344101
220 - 229	0	0	657463	0	0	657463
230 - 239	56	0	284065	0	4	248299
240 - 249	106	0	488641	0	2	472191
250 - 259	84	0	528151	0	0	528151
260 - 269	62	0	539665	0	0	539665
270 - 279	42	0	365259	0	1	369159
280 - 289	0	0	457566	0	0	457566
290 - 299	1	0	604687	0	1	550127
300 - 309	0	0	450887	0	0	450887
310 - 314	0	0	275418	0	0	275418
Укупно	1040	0	14172323	0	28	13872431

На слици 4.4 је дат распоред авиона на паркинг позицијама добијен након другог тестирања.

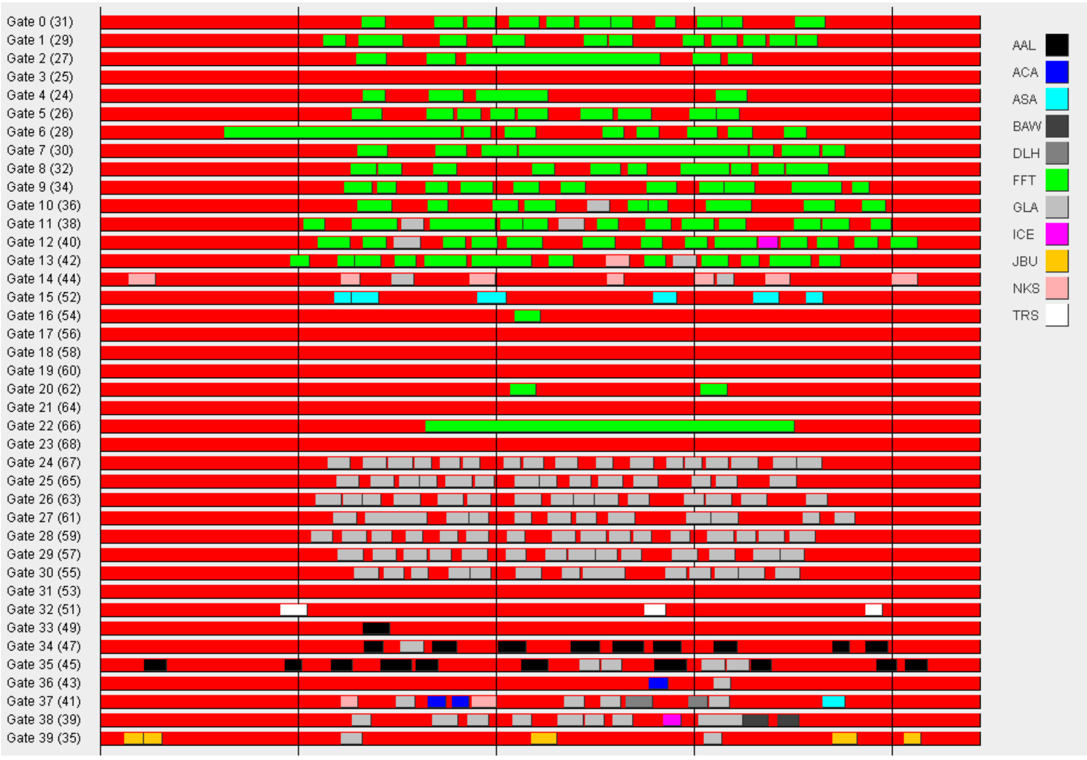
Слика 4.4. План додељивања паркинг позиција добијен лексикографском методом

У трећем тестирању проблем је решен BCO алгоритмом. У овом тестирању било је могуће решавати цео проблем у једном пуштању програма. Коришћене су следеће вредности улазних параметара BCO алгоритма: B = 45 (15 за сваки тип пчеле), IT = 2000, NP = 20, NC =1.

Карактеристике решења добијеног BCO алгоритмом су:

- Укупни временски губици: 0 min
- Број летова опслужених у систему колаборације: 33 (10,48 %)
- Укупно растојање пешачења: 13928785 m

Као што се може приметити BCO алгоритам је такође пронашао решење без кашњења. Поређењем са резултатима у другом тесту, BCO алгоритам је пронашао решење где би 5 авиона више било опслужено коришћењем концепта колаборације (1,58 % у односу на укупан број). Укупно растојање пешачења је такође нешто веће у овом случају (0,4 %). План додељивања паркинг позиција применом BCO алгоритма приказан је на Слици 4.5.



Слика 4.5. План додељивања паркинг позиција добијен BCO алгоритмом

5. УБЛАЖАВАЊЕ ПОРЕМЕЋАЈА НАСТАЛИХ НЕДОСТАТКОМ ВОЗИЛА У СИСТЕМИМА МАСОВНОГ ПРЕВОЗА

У многим градовима света је током протеклих деценија дошло до значајног повећања броја путовања која се обављају путничким аутомобилима. Мреже градских саобраћајница није могуће значајније проширивати како због високих инфраструкурних трошкова, тако и због негативног утицаја на околину. Системи масовног превоза путника у градовима могу да играју значајну улогу у покушајима да се смање просечна времена путовања, укупан број саобраћајних незгода и ниво аеро загађења и буке у градовима. Јавни градски превоз постаје атрактивнији за потенцијалне путнике са снижавањем цене превоза и повећањем нивоа транспортних услуга. Путници у јавном превозу су заинтересовани, пре свега, за високи квалитет пружених транспортних услуга, смањена времена путовања, мали број преседања приликом обављања путовања, смањена просечна времена чекања на станицама, повећану вероватноћу налажења седишта у возилу итд. Поремећаји у одвијању планираних редова вожње у системима масовног превоза путника доводе у одређеним ситуацијама до немогућности да се опслуже сви путници који захтевају превоз. Такође се приликом поремећаја у одвијању редова вожње често повећавају и времена чекања путника на станицама, укупна времена путовања и број преседања приликом обављања путовања. Другим речима, поремећаји у одвијању планираних редова вожње у значајној мери утичу на погоршање квалитета транспортних услуга које се пружају путницима и чине системе масовног превоза мање атрактивним.

У многим јавним транспортним системима аутобуси крећу са терминала у дефинисаним тренуцима времена. Током кретања возила на линијама долази до промена интервала слеђења услед поремећаја узрокованих различитим факторима. Накупљање аутобуса (*Bus bunch*) је уобичајени назив за случај када се два или више аутобуса крећу дуж линије један за другим. Накупљање аутобуса се дешава најчешће током вршних часова. Barnett (1974) је приметио да „постоји много узрока поремећаја у раду аутобуса, на пример, број станица на линији, број светлосних сигнала на којима возило мора да се заустави, степен гужве у саобраћају, карактеристике возача при вожњи, при чему су то све случајне величине“. У пионирске радове из ове области такође се могу уврстити радови чији су аутори Vuchic (1969), Osuna и Newell (1972), Newell (1974), Barnett (1974) и Chapman и Michel (1978). У многим случајевима низак ниво транспортне услуге у јавном превозу путника може бити узрокован и чињеницом да су неки аутобуси који су планирани, ван употребе из различитих техничких разлога (Теодоровић и Rallis (1988)). Мањи број аутобуса на раду, него што је планиран, такође може у одређеним ситуацијама бити проузрокован и недостатком одређеног броја возача. Диспечери у многим компанијама се на дневном нивоу сусрећу са проблемом расподеле аутобуса на линије у условима смањеног укупног броја расположивих аутобуса. Метеоролошки услови, одсуства и закашњења возача, саобраћајне незгоде, загушења у саобраћају, и други фактори такође свакодневно доводе до већих или мањих поремећаја у одвијању планираних редова вожње возила масовног превоза.

Логично се намећу следећа питања: Како треба да се понашају диспечери задужени за вођење саобраћаја у ситуацијама када је дошло до поремећаја? Да ли је могуће расположивим бројем возила превести све путнике који захтевају превоз? Да ли треба, због недостатка одређеног броја возила, скратити неке линије? Како доделити расположиве аутобусе линијама да би се минимизирало укупно време чекања путника?

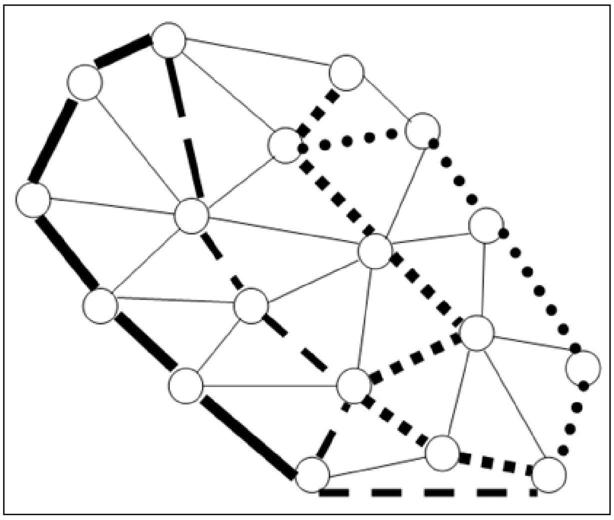
Поремећаји често могу бити изузетно велики тако да је неопходно извршити ново планирање рада возила и возача. При одређивању нових планова рада морају се узети у обзир интереси како компаније која обавља превоз, тако и интереси путника. Ови интереси су међусобно конфликтни што условљава да се проблем распоређивања аутобуса на линије, као проблем могућег модификовања мреже анализирају техникама вишекритеријумске оптимизације.

Диспечери најчешће користе своје искуство и интуицију при решавању проблема у одвијању саобраћаја насталих као последица поремећаја. Због комбинаторне природе проблема, диспечери често нису у могућности да пронађу оптимално решење. Коришћењем одговарајућег система за подршку одлучивању могуће је у знатној мери повећати квалитет одлука диспечера.

Проблемима поремећаја редова вожње у јавном градском превозу није се бавио велики број аутора у свету. У раду чији су аутори Теодоровић и Rallis (1988) развијен је модел којим се тежи минимизацији укупног времена чекања путника. У овом раду је претпостављено да путници, за обављање свог путовања, користе само једну аутобуску линију. Аутори су за решавање назначеног проблема користили динамичко програмирање. Chang и остали (1998) су предложили систем за подршку одлучивању заснован на фази логици. Li и остали (2007а, 2007б) су разматрали проблем који настаје отказом возила при реализацији планираног реда вожње. Аутори су развили систем за подршку одлучирању као помоћ у раду диспечера. Тестирање предложеног система је извршено у студији случаја за град Порто Алегре у Бразилу. Kepaptsoglou и остали (2009а) су пручавали проблем поремећаја саобраћаја у оквиру метро система (*bus bridging system*). У раду је предложена примена различитих хеуристичких алгоритама за решавање проблема поремећаја. Zeng и остали (2012) су предложили модел за отклањање поремећаја у функционисању трамвајског саобраћаја. У раду је предложено ублажавање поремећаја ангажовањем такси возила. Николић и остали (2015) су користили метахеуристику за вршење прерасподеле аутобуса у случају њиховог недостатка. У том раду аутори су тежили минимизацији броја неопслужених путника и њиховог укупног времена чекања. За доношење одлука о кретању путника, од почетне до крајње станице, коришћен је фази логички систем. Сва тестирања су извршена на генерисаним хипотетичким примерима.

5.1. Опис проблема

Нека је са $G = (N, A)$ дата транспортна мрежа где N представља скуп станица, а A скуп грана, тј. путева који повезују станице. Такође, нека на посматраној мрежи постоји k аутобуских линија (Слика 5.1). Претпоставимо да је у редовним условима одвијања саобраћаја, свакој линији додељено N_i $(i = 1,2,\ldots, k)$ аутобуса.



Слика 5.1. Мрежа линија масовног превоза

У случају отказа неких аутобуса на линијама диспечер мора да изврши прерасподелу расположивог броја аутобуса. У раду чији су аутори Теодоровић и Rallis (1988) развијена је математичка формулација и модел динамичког програмирања за решавање посматраног проблема. Математичка формулација ових аутора може се представити на следећи начин:

Минимизирати

$$T_w = \sum_{i=1}^{k} \lambda_i T^{\circ} \frac{T_i}{2M_i} \tag{5.1}$$

при ограничењима:

$$\max\left\{ \left\lceil \frac{T_i}{h_i^{\circ}} \right\rceil, \left\lceil \frac{q_i^{\circ} T_i}{C_v} \right\rceil \right\} \le M_i \le N_i \quad \forall\, i = \overline{1,k} \tag{5.2}$$

$$\sum_{i=1}^{k} M_i = M \tag{5.3}$$

$$M_i \quad \text{је целобројно} \quad \forall\, \mathrm{i} = \overline{1,\mathrm{k}} \tag{5.4}$$

где су:

T_w – укупно време чекања свих путника,

T_i – време обрта аутобуса на линији i,

λ_i – укупан интензитет наиласка путника [број путника/h] на i-тој линији,

T^o–посматрани временски интервал,

h_i^o – максимално дозвољен интервал слеђења на линији i,

C_v – број путничких места у атобусу,

N_i – планирани број аутобуса додељених линији i,

M – укупан број аутобуса на располагању

M_i – променљива која представља број аутобуса које треба доделити линији i.

Критеријумска функција (1) коју желимо да минимизирамо представља укупно време чекања свих путника на мрежи. Ограничење (2) дефинише минималан и максималан број аутобуса на линији. Ограничење (3) дефинише захтев да укупан број аутобуса који се додељују линијама мора да буде једнак расположивом. Ограничење (4) дефинише променљиве M_i као целобројне.

Теодоровић и Rallis (1998) су као један од улазних података користили константан интензитет наиласка путника за сваку од линија. Претпоставком о константности интензитета наиласка путника је имплицитно уведена претпоставка да је број путника на свакој линији константан и да не зависи од броја аутобуса који су додељени тим линијама.

У моделу развијеном у овој дисертацији нису коришћени интензитети наиласка путника по станицама као улазне величине. Развијени модел користи као основне улазне величине елементе изворно – циљне матрице путовања $D = \{d_{ij} | i, j \in [1, 2, \ldots, |N|]\}$ где d_{ij} представља број путника који путују од станице i до станице j у јединици времена.

Такође је уведена реалнија претпоставка да путници при путовању од почетне до крајње станице могу да користе две или више линија. Циљ који је потребно постићи приликом решавања проблема поремећаја је минимизирање негативних ефеката, као што су: број неопслужених путника, број линија које су модификоване (уколико је то дозвољено), минимизирање времена путовања, минимизирање укупног броја преседања, и др.

Као могући приступи за решавање проблема у раду су развијени модели којим се врши прерасподела расположивих аутобуса, модели којим се врши прерасподела расположивих аутобуса и модификовање линија, као и модел којим се врши пројектовање новог скупа линија на посматраној мрежи. За прерасподелу расположивих аутобуса и прерасподелу аутобуса са истовременим модификовањем постојећих линија предложене су математичке формулације, као и хеуристички алгоритми засновани на метахеуристици Оптимизација колонијом пчела. За пројектовање новог скупа линија за посматрану мрежу предложен је само хеуристички алгоритам.

5.2. Математичке формулације за решавање проблема прерасподелом аутобуса и модификовањем линија

За назначени проблем у дисертацији су предложене две математичке формулације. У првој се не узима у обзир могућност модификовања линија, док је у другој и та могућност укључена. У обе математичке формулације постоји више критеријумских функција које је потребно минимизирати.

Нека је L скуп линија посматране мреже. За сваку линију $l \in L$ може се дефинисати скуп допустивих фреквенција за ту линију:

$$\Theta_l = \left\{ \theta_{lf} \middle| \theta_{fl} = \frac{M_{lf}}{T_l}, M_{lf} = \max\left\{ \left\lceil \frac{T_l}{h_l^\circ} \right\rceil, N_l - N_{out} \right\}, \max\left\{ \left\lceil \frac{T_l}{h_l^\circ} \right\rceil, N_l - N_{out} \right\} + 1, \ldots, N_l \right\} \tag{5.5}$$

где је N_{out} број аутобуса који су отказали.

На овај начин за сваку линију добија се скуп вредности фреквенција. На пример: $\Theta_1 = \{\theta_{11}, \theta_{12}, \theta_{13}, \theta_{14}\}$, $\Theta_2 = \{ \theta_{11}, \theta_{12}, \theta_{13}\}$, $\Theta_3 = \{ \theta_{11}, \theta_{12}, \theta_{13}, \theta_{14}, \theta_{15}\}$, итд.

Математичке формулације које су предложене у дисертацији, као и нотација, базирани су на раду Martinez и остали (2013). У обе математичке формулације је коришћена следећа нотација:

N – скуп чворова

N^P – скуп станица

A – скуп грана мреже

$A_n^+ (A_n^-)$ - скуп излазних (улазних) грана из (у) чвор n

L – скуп линија

Θ – скуп фреквенција

y_{lf} – бинарна променљива која узима вредност 1 ако је фреквенција θ_f додељена линији l, у супротном узима вредност 0

B – горња граница броја возила

c_a – трошак гране a

v_a – број захтева који путују граном a

f_a – вредност фреквенције на линији која одговара грани a

$f(a)$ – индекс у скупу Θ који одговара грани a

$l(a)$ - индекс у L одговарајуће линије гране a

K – скуп изворно-циљних парова чворова

O_k (D_k) – изворни (циљни) чвор пара чворова $\kappa \in K$

δ_k – број путовања изворно-циљног пара чворова k

w_n – време чекања на станици n

z_k – број неопслужених путника изворно-циљног пара чворова k

α – коефицијент попуњености,

C_v - број места у аутобусу

Математичка формулација за проблем одређивања новог додељивања аутобуса линијама може се дефинисати на следећи начин:

Минимизирати

$$F_1 = \sum_{k \in K} z_k \tag{5.6}$$

Минимизирати

$$F_2 = \sum_{k \in K} \left(\sum_{a \in A} c_a \, v_{ak} + \sum_{n \in N^P} w_{nk} \right) \tag{5.7}$$

при ограничењима

$$\sum_{l \in L} T_l \sum_{f=1}^{|\Theta_l|} \theta_f \, y_{lf} \le M \tag{5.8}$$

$$\sum_{f=1}^{|\Theta_l|} y_{lf} = 1 \quad \forall \, l \in L \tag{5.9}$$

$$\sum_{a \in A_n^+} v_{ak} + z_k = \delta_k \quad \forall \, k \in K, n = O_k \tag{5.10}$$

$$\sum_{a \in A_n^+} v_{ak} - \sum_{a \in A_n^-} v_{ak} = 0 \quad \forall \, k \in K, n \in N, n \neq O_k, n \neq D_k \tag{5.11}$$

$$\sum_{a \in A_n^-} v_{ak} + z_k = \delta_k \quad \forall \, k \in K, n = D_k \tag{5.12}$$

$$v_{ak} \le \theta_{f(a)} w_{nk} \quad \forall \, n \in N^P, a \in A_n^+, k \in K \tag{5.13}$$

$$v_{ak} \le \delta_k \, y_{l(a) f(a)} \quad \forall \, a \in A^B, k \in K \tag{5.14}$$

$$\sum_{k \in K} v_{ak} \le \alpha \, \theta_{f(a)} \, C_v \, y_{l(a) f(a)} \quad \forall \, a \in A \tag{5.15}$$

$$v_{ak} \ge 0 \quad \forall \, a \in A, k \in K \tag{5.16}$$

$$w_{nk} \ge 0 \quad \forall \, n \in N, k \in K \tag{5.17}$$

$$y_{lf} \in \{0,1\} \quad \forall \, l \in L, f = \overline{1, |\Theta_l|} \tag{5.18}$$

Функција циља (5.6) коју желимо да минимизирамо представља укупан број неопслужених путника. Функција циља (5.7) коју такође желимо да минимизирамо представља укупно време путовања (време путовања чини време које путник проведе у возилу и време чекања). Ограничење (5.8) гарантује да је укупан број аутобуса који се расподељује мањи или једнак расположивом броју аутобуса. Ограничење (5.9) гарантује да ће свакој линији бити додељена једна фреквенција. Ограничења (5.10), (5.11) и (5.12) су ограничења која се односе на одржање тока на мрежи. Ограничење (5.13) дефинише најмању могућу вредност укупног времена чекања на станици *n* за све путнике изворно-циљног пара чворова *k*. Ограничење (14) гарантује да одређени део путничких токова може да путује дуж гране ако постоји одређена фреквенција возила дуж те гране. Укупан број путника на некој грани не може да буде већи од укупног броја понуђених места на тој грани (ограничење 15). Ограничењима (5.16), (5.17) и (5.18) су дефинисане променљиве.

У случају када је могуће вршити промену облика мреже (модификовањем једне или више линија) потребно је надградити претходну математичку формулацију (5.6) – (5.18). С обзиром да је модификација сваке линија нешто што је непожељно (како са аспекта даваоца услуге, тако и са аспекта путника), као нова критеријумска функција може се дефинисати минимизација броја модификованих линија, односно, максимизација броја оригиналних линија које ће се користити.

Нека је са L_0 означен скуп оригиналних аутобуских линија. За сваку оригиналну аутобуску линију $(j = 1, 2, \ldots, |L_0|)$ може се дефинисати скуп линија L_j чији су чланови оригинална аутобуска линија j и потенцијалне линије које настају модификацијом оригиналне линије j. Такође, може се дефинисати скуп L као скуп свих линија $(L = L_1 \cup L_2 \cup \ldots L_{|L_0|})$. Након тога могуће је за сваку линију $l \in L$ одредити могуће фреквенције коришћењем израза (5.5).

Математичка формулација сада може бити дата у следећем облику:

Минимизирати

$$F_1 = \sum_{k \in K} z_k \tag{5.6}$$

Максимизирати

$$F_2 = \sum_{l \in L_0} \sum_{f=1}^{|\Theta_l|} y_{lf} \tag{5.19}$$

Минимизирати

$$F_3 = \sum_{k \in K} \left(\sum_{a \in A} c_a \, v_{ak} + \sum_{n \in N^P} w_{nk} \right) \tag{5.7}$$

при ограничењима:

$$\sum_{l \in L} T_l \sum_{f=1}^{|\Theta_l|} \theta_f \ y_{lf} \leq M \tag{5.8}$$

$$\sum_{l \in L_j} \sum_{f=1}^{|\Theta_l|} y_{lf} = 1 \quad \forall \ j = \overline{1, |L_0|} \tag{5.20}$$

$$\sum_{a \in A_n^+} v_{ak} - \sum_{a \in A_n^-} v_{ak} = 0 \quad \forall \ k \in K, n \in N, n \neq O_k, n \neq D_k \tag{5.10}$$

$$\sum_{a \in A_n^+} v_{ak} + z_k = \delta_k \quad \forall \ k \in K, n = O_k \tag{5.11}$$

$$\sum_{a \in A_n^-} v_{ak} + z_k = \delta_k \quad \forall \ k \in K, n = D_k \tag{5.12}$$

$$v_{ak} \leq \theta_{f(a)} w_{nk} \quad \forall \ n \in N^P, a \in A_n^+, k \in K \tag{5.13}$$

$$v_{ak} \leq \delta_k \, y_{l(a) f(a)} \quad \forall \ a \in A^B, k \in K \tag{5.14}$$

$$\sum_{k \in K} v_{ak} \leq \alpha \, \theta_{f(a)} \, C_v \, y_{l(a) f(a)} \quad \forall \ a \in A \tag{5.15}$$

$$v_{ak} \geq 0 \quad \forall \ a \in A, k \in K \tag{5.16}$$

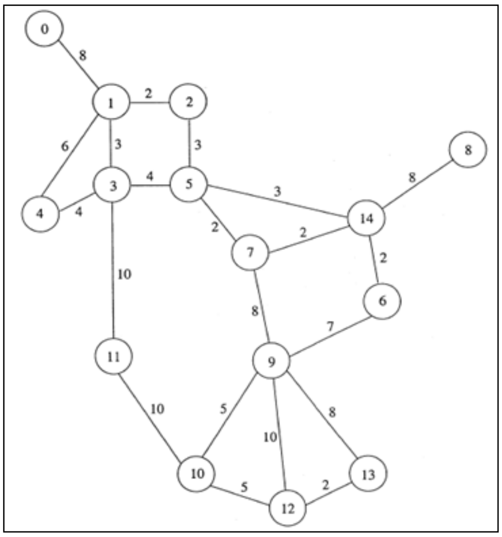
$$w_{nk} \geq 0 \quad \forall \ n \in N, k \in K \tag{5.17}$$

$$y_{lf} \in \{0,1\} \quad \forall \ l \in L, \ f = \overline{1, |\Theta_l|} \tag{5.18}$$

Функција циља (5.19) коју желимо да максимизирамо представља број линија које неће бити модификоване. Ограничење (5.20) гарантује да ће свака оригинална линија бити задржана или ће бити одабрана једна од њених могућих модификација. Такође, овим ограничењем се гарантује да ће свакој линији бити постављена једна фреквенција.

Предложене математичке формулације су тестиране на *Mandl*-овој мрежи (Mandl, 1979) која представља бенчмарк мрежу за тестирање техника које се баве проблемом пројектовања мреже линија. Ову мрежу чини 15 чворова и 21 грана

(Слика 9). Изворно – циљна матрица путовања за овај пример дата је у Прилогу 1. Захтевани број путовања на овој мрежи је 15570. *Mandl* - ова мрежа је приказана на Слици 5.2.



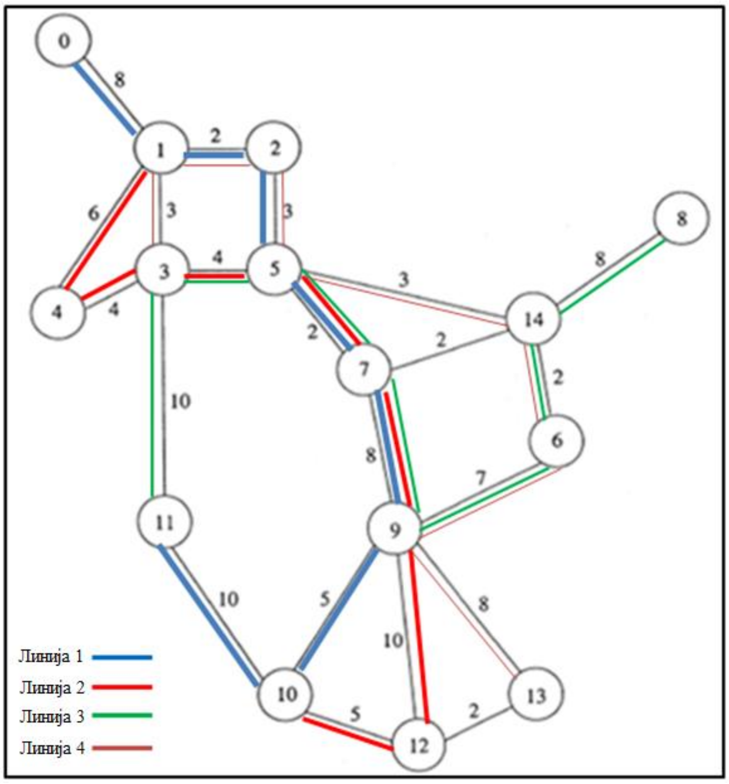
Слика 5.2. *Mandl* - ова транспортна мрежа

За сва тестирања у овом делу коришћена је мрежа која садржи четири линије (за више детаља видети поглавље 5.3.4). За расподелу путника на мрежи и одређивање броја аутобуса на линијама коришћена је процедура описана у поглављу 5.4. Подаци о аутобуским линијама су дати у Табели 5.1. Почетне линије су приказане на Слици 5.3.

Табела 5.1. Карактеристике аутобуских линија

Линија	Станице	Време обрта	Број аутобуса
1	0-1-2-5-7-9-10-11	76	37
2	1-4-3-5-7-9-12-10	78	24
3	8-14-6-9-7-5-3-11	82	17
4	3-1-2-5-14-6-9-13	56	10

Као што се може уочити линије имају респективно следеће интервале слеђења: 2,05; 3,25; 4,82 и 5,6 минута. За максималне дозвољене интервале слеђења су узете вредности добијене додавањем 0,5 минута израчунатим интервалима слеђења. За вредности степена попуњености и расположивог броја места у возилу су респективно узете следеће вредности: α = 1,25 и C_v = 40.



Слика 5.3. Аутобуске линије добијене BCO алгоритмом

За решавање проблема коришћена је лексикографска метода вишекритеријумске оптимизације. Сваки подпроблем у оквиру процедуре решавања лексикографском методом решаван је CPLEX софтвером. У Табели 5.2 су дати резултати за случајеве када је број возила ван употребе од 1 до 11. На пример, ако је број возила ван употребе једнак 5, онда 36 аутобуса треба да буде распоређено на прву линију (за један мање од планираног броја), 21 аутобус треба да буде распоређен на другу линију (три аутобуса мање него што је планом предвиђено), 16 аутобуса треба да буде додељено трећој линији (за један мање од предвиђеног плана). Четвртој линији треба да буде додељено планираних 10 аутобуса.

Табела 5.2. Решење добијено за први предложени модел

Број отказалих аутобуса	Нови распоред аутобуса				Неопслужени путници	Укупно време путовања
	Линија 1	Линија 2	Линија 3	Линија 4		
	Број аутобуса	Број аутобуса	Број аутобуса	Број аутобуса		
1	37	23	17	10	0	219758,72
2	37	22	17	10	0	220368,73
3	37	21	17	10	0	221021,97

4	37	21	16	10	0	221748,31
5	36	21	16	10	0	222491,24
6	35	21	16	10	0	223465,09
7	34	21	16	10	0	224607,79
8	33	21	16	10	35	224239,33
9	32	21	16	10	114	221737,28
10	31	21	16	10	193	219494,86
11	30	21	16	10	272	217735,78

Нумерички експерименти показују да првим предложеним моделом не може да се нађе допустиво решење када постоји више од 11 аутобуса ван употребе. Други модел је тестиран за случајеве када је укупан број аутобуса ван употребе већи или једнак од 8. У Табели 5.3 су приказане оригиналне аутобуске линије и линије које могу настати њиховим скраћивањем (по једна за сваку оригиналну аутобуску линију). Добијени резултати су дати у Табели 5.4. Други модел не може да нађе допустиво решење за случајеве када је укупан број аутобуса који су ван употребе већи или једнак од 27.

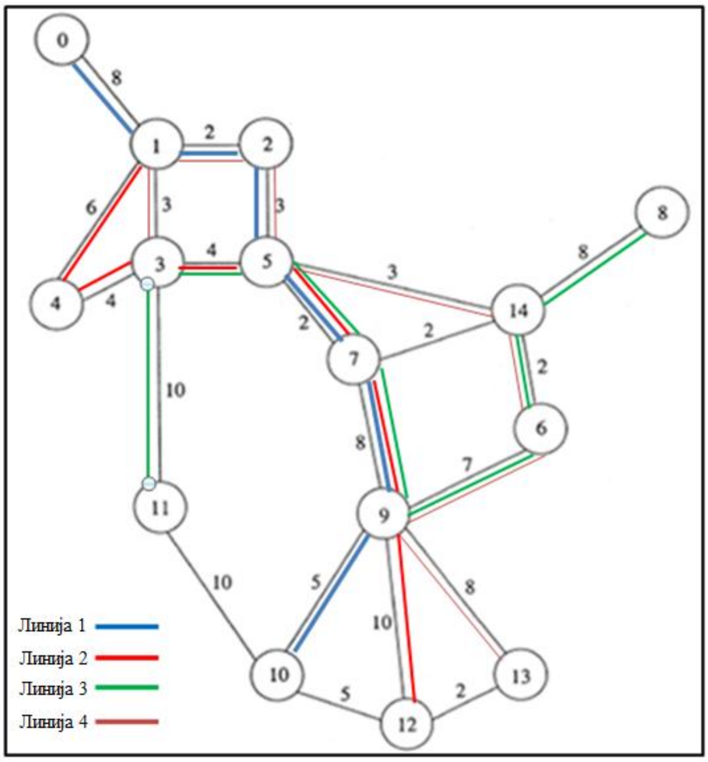
Табела 5.3. Нови скуп аутобуских линије

Оригиналне линије	Нове атобуске линије	Станице које припадају аутобуским линијама	Време обрта
1	$Л_1$	0 – 1 – 2 – 5 – 7 – 9 – 10 – 11	76
	$Л_2$	0 – 1 – 2 – 5 – 7 – 9 – 10	56
2	$Л_3$	1 – 4 – 3 – 5 – 7 – 9 – 12 – 10	78
	$Л_4$	1 – 4 – 3 – 5 – 7 – 9 – 12	68
3	$Л_5$	8 – 14 – 6 – 9 – 7 – 5 – 3 – 11	82
	$Л_6$	8 – 14 – 6 – 9 – 7 – 5 – 3	62
4	$Л_7$	3 – 1 – 2 – 5 – 14 – 6 – 9 – 13	56
	$Л_8$	1 – 2 – 5 – 14 – 6 – 9 – 13	50

Из Табеле 5.4 може се видети које линије треба да буду модификоване (које линије треба да буду краће) и колико аутобуса треба да ради на свакој од њих. На пример, ако има 17 аутобуса ван употребе, онда линије 1 и 2 треба да буду скраћене, док линије 3 и 4 остају непромењене (Слика 5.4). На првој линији треба да ради 26 аутобуса (11 мање него на оригиналној првој линији), 19 аутобуса на модификованој другој линији (5 мање него што је планирано), 16 аутобуса на трећој (један аутобус мање него што је планирано) и 10 аутобуса на четвртој линији.

Табела 5.4. Решења добијена другим предложеним моделом

Број аутобуса који су отказали	Нови распоред аутобуса				Број аутобуса који су отказали	Број неопслу-жених путника	Број измење-них линија	Укупно време путовања
	Линија 1	Линија 2	Линија 3	Линија 4				
	Линија / број аутобуса	Линија / број аутобуса	Линија / број аутобуса	Линија / број аутобуса				
8	$Л_1$ / 37	$Л_3$ / 21	$Л_6$ / 12	$Л_7$ / 10	8	0	1	224,623,165
9	$Л_1$ / 36	$Л_3$ / 21	$Л_6$ / 12	$Л_7$ / 10	9	0	1	225,504,395
10	$Л_1$ / 35	$Л_3$ / 21	$Л_6$ / 12	$Л_7$ / 10	10	0	1	226620,1
11	$Л_1$ / 34	$Л_3$ / 21	$Л_6$ / 12	$Л_7$ / 10	11	0	1	227,842,207
12	$Л_2$ / 28	$Л_3$ / 21	$Л_5$ / 17	$Л_7$ / 10	12	0	1	235,033,321
13	$Л_2$ / 27	$Л_3$ / 21	$Л_5$ / 17	$Л_7$ / 10	13	0	1	235858,53
14	$Л_2$ / 27	$Л_3$ / 21	$Л_5$ / 16	$Л_7$ / 10	14	0	1	236889,64
15	$Л_2$ / 26	$Л_3$ / 21	$Л_5$ / 16	$Л_7$ / 10	15	0	1	237951,4
16	$Л_2$ / 25	$Л_3$ / 21	$Л_5$ / 16	$Л_7$ / 10	16	0	1	239383,56
17	$Л_2$ / 26	$Л_4$ / 19	$Л_5$ / 16	$Л_7$ / 10	17	0	2	241162,28
18	$Л_2$ / 25	$Л_4$ / 19	$Л_5$ / 16	$Л_7$ / 10	18	0	2	242427,06
19	$Л_2$ / 25	$Л_4$ / 19	$Л_5$ / 16	$Л_8$ / 9	19	0	3	244,727,496
20	$Л_2$ / 24	$Л_4$ / 19	$Л_5$ / 16	$Л_8$ / 9	20	69	3	242883,67
21	$Л_2$ / 23	$Л_4$ / 19	$Л_5$ / 16	$Л_8$ / 9	21	176	3	239665,47
22	$Л_2$ / 22	$Л_4$ / 19	$Л_5$ / 16	$Л_8$ / 9	22	283	3	236962,54
23	$Л_2$ / 25	$Л_4$ / 19	$Л_6$ / 12	$Л_8$ / 9	23	1040	4	208498,97
24	$Л_2$ / 24	$Л_4$ / 19	$Л_6$ / 12	$Л_8$ / 9	24	1059	4	209026,08
25	$Л_2$ / 23	$Л_4$ / 19	$Л_6$ / 12	$Л_8$ / 9	25	1166	4	205485,67
26	$Л_2$ / 22	$Л_4$ / 19	$Л_6$ / 12	$Л_8$ / 9	26	1273	4	202654,78



Слика 5.4. Аутобуске линије када има 17 аутобуса мање него обично

5.3. Пројектовање линија применом Оптимизације колонијом пчела

Као што је већ раније истакнуто, мреже урбаних путева у градским подручјима у многим земљама су веома загушене. Последице тога су дуга времена путовања, непредвиђена кашњења, повећани трошкови путовања, повећана загађења ваздуха, ниво буке, број саобраћајних незгода и др. Саобраћајни инжењери и градске власти развијају и имплементирају различите технике за управљање саобраћајним захтевима (Travel Demand Management (*TDM*) techniques) које могу побољшавати одлуке путника ("Park-and-Ride facilities", "High Occupancy Vehicle (HOV) facilities", "Ride-sharing programs", "Telecommuting", "Congestion Pricing"). Даље, раст учешћа јавног превоза у градовима је један од главних задатака планера саобраћаја и градских власти. Ово може бити постигнуто одговарајућим пројектовањем мреже линија превоза, повећањем могућности директног превоза од почетних до крајњих станица, повећањем фреквенција, развојем аутобуских система који би били одвојени од остале саобраћајне мреже, побољшањем услуга током ноћи и викендима, бољим информационим системима за путнике, итд.

Одговарајућа пројектована мрежа линија може значајно повећати коришћење јавног превоза. Проблем пројектовања мреже линија је један од најважнијих проблема са којима се суочавају даваоци услуге и градске власти. Овај проблем планирања транспорта припада класи тешких проблема комбинаторне оптимизације, чије оптимално решење је тешко наћи (Magnanti и Wong (1984)). Облик мреже линија, као и фреквенције, јако зависе од путничких захтева, броја и типова аутобуса (возног парка), као и од расположивог буџета. Лоше испројектована мрежа линија може показати велику неподударност између линија и захтеваних путева великог броја корисника, као и појаву веома дугих времена чекања.

Многи фактори које треба узети у обзир при пројектовању мреже линија су међусобно у конфликту (на пример, краћа времена чекања захтевају већи број аутобуса, итд.). При пројектовању мреже линија интереси и путника и давалаца услуга морају се узети у разматрање. Узимајући у обзир супротстављену природу интереса, проблем пројектовања мреже линија може се третирати као проблем вишекритеријумске оптимизације. При пројектовању мреже линија у оквиру дисертације тежи се максимизирању броја опслужених путника, минимизирању укупног броја преседања и минимизирању укупног времена путовања свих опслужених путника.

5.3.1. Преглед литературе

У литератури је предложен већи број модела за решавање проблема пројектовања мреже линија. Lampkin и Saalmans (1967) су предложили први хеуристички алгоритам за решавање проблема пројектовања мреже линија. У првом кораку, предложени алгоритам прави почетни скуп линија. У следећем кораку, остали чворови се један по један убацују у једну од генерисаних линија. Студија случаја на малом граду на северу Енглеске је такође укључена у рад. Silman и остали (1974) су предложили приступ решавања проблема пројектовања мреже линија у два корака.У првом кораку се генерише скуп линија кандидата кроз неколико итерација. У другом кораку аутори су одређивали оптималне фреквенције аутобуса за сваку од линија. Silman и остали (1974) су покушали да минимизирају време путовања путника, док истовремено воде рачуна о путницима који не могу да нађу седиште. Byrne (1975) је разматрао случај када је област која се опслужује јавним превозом део круга који се може описати поларним координатама. Mandl (1979) је предложио хеуристички алгоритам за одређивање скупа најбољих линија.У раду су такође приказана искуства при решаваљу реалних проблема. Newell (1979) је извршио теоријску анализу проблема пројектовања мреже линија. Ceder и Wilson (1986) су описали проблем пројектовања мреже аутобуских линија, анализирали претходне концепте и предложили методолошки приступ у два нивоа за решавање проблема пројектовања мреже аутобуских линија. Baaj и Mahmassani (1995) су предложили алгоритам генерисања линија (*Route Generation Algorithm* - RGA) за генерисање мреже линија. Предложени приступ је комбинован са експертским знањем и техникама претраживања заснованим на вештачкој интелигенцији (Artificial Intelligence). Ceder и Israeli (1998) дефинишу критеријумску функцију која узима у обзир како интересе путника тако и интересе превозиоца. За решавање проблема пројектовања линија Pattnaik и остали (1998) су предложили процедуру у два корака. У првом кораку се дефинише скуп линија кандидата, док се у другом кораку бира коначан скуп линија Генетским алгоритмима. Bielli и остали (2002) су применили генетски алгоритам за оптимизацију мреже линија. Они су тестирали свој приступ на примеру италијанског града Парме. Chakroborty (2003) је такође предложио приступ за решавање проблема пројектовања мрежа линија базиран на Генетским алгоритмима. Lee и Vuchic (2005) су разматрали проблем пројектовања мреже линија у случају променљиве потражње за захтевима. Аутори су предложили итеративни приступ који води рачуна о вези између промена у захтевима за путовањима и облика транспортне мреже. Предложени приступ је тестиран на релативно малој транспортној мрежи. Guan и остали (2006) су предложили модел за истовремену оптимизацију облика линија и расподеле путника на линијама. Предложени модел је решаван методом гранања и ограничавања. Fan и Machemehl (2006) су користили Симулирано каљење за решавање проблема одређивања аутобуских линија. Предложен концепт је тестиран на три експерименталне мреже. Zhao и Zeng (2006) су комбиновали

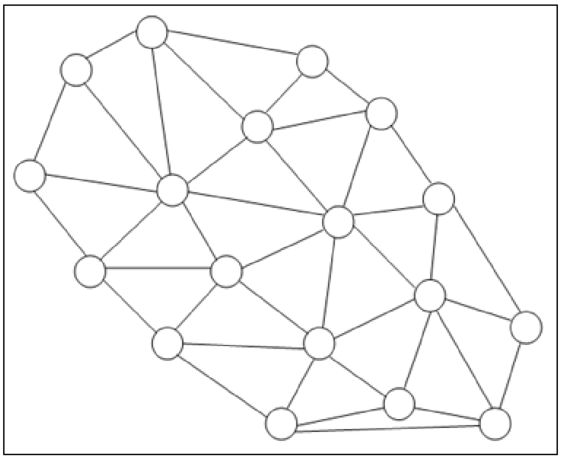
Генетске алгоритме и Симулирано каљење при тражењу оптималних линија и интервала слеђења. Аутори су тежили минимизирању преседања и укупних трошкова корисника, као и максимизирању покривености услуга. Zhao и Zeng (2007) су развили модел за одређивање мреже линија, интервала слеђења возила и прављења реда вожње. Предложени приступ комбинује Симулирано каљење и Табу претраживање. Desaulniers и Hickman (2007) су дали преглед модела и приступа у решавању проблема у јавном превозу путника. Fan и Machemehl (2008) су разматрали проблем одређивања мреже линија у случају променљивих захтева. Аутори су предложили вишекритеријумски модел, при чему је методологија за решавање заснована на Табу претраживању. Guihaire и Hao (2008) су класификовали 69 различитих приступа који се односе на пројектовање мреже линија и одређивање фреквенција. Они такође указују на трендове будућих истраживања. За решавање проблема пројектовања линија и распоређивања аутобуса Pacheco и остали (2009) су развили алгоритам заснован на локалном претраживању као и алгоритам заснован на Табу претраживању. Аутори су показали робусност њиховог приступа узимајући у обзир промене у захтевима. У раду је урађена студија случаја на граду Бургос у Шпанији. Mauttone и Urquhart (2009) су развили алгоритам убацивања парова (Pair Insertion Algorithm - PIA) који може бити коришћен за генерисање почетних решења која би била поправљана алгоритмима локалног претраживања или еволуционим алгоритмима. Алгоритам је инспирисан алгоритмом за генерисање линија (Route Generation Algorithm - RGA) који су предложили Baaj и Mahmassani (1995). Kepaptsoglou и Karlaftis (2009b) су урадили преглед резултата истраживања који су у вези проблема пројектовања линија. Fun и Mumford (2010) су предложили модел за проблем пројектовања градских линија који вреднује потенцијалне скупове линија.

Bagloee и Ceder (2011) су разматрали проблем пројектовања мреже линија при чему њихова величина одговара мрежама данашњих величина. Предложени алгоритам је тестиран на мрежи града Winnipeg у Канади, као и на мрежи коју је предложио Mandl. Прегледни рад Derrible и Kennedy (2011) је посвећен апликацијама теорије графова у проблему пројектовања мреже линија. Szeto и Wu (2011) су разматрали проблем пројектовања мреже линија за Tin Shui Wai, у приградском делу Хонг Конга. Аутори су предложили модел који истовремено врши одређивање линија и постављање фреквенција. Предложени метод комбинује Генетски алгоритам и хеуристику локалног претраживања. Blum и Mathew (2012) су разматрали проблем одређивања нове мреже линија. Предложени приступ је тестиран у студији случаја на граду Mumbai у Индији. Schoebel (2012) је дао преглед различитих модела за планирање аутобуских, железничких и трамвајских линија.

Може се уочити да је већина аутора тежила да минимизира укупно време путовања, или генерално трошкове. Истовремено већи део радова уводи поједностављену претпоставку да су захтеви за опслугом фиксни. Реалнија претпоставка је да захтеви за путничким токовима зависе од понуђене мреже линија и да до коначног облика мреже треба доћи разматрањем проблема еквилибријума. Такође, у литератури постоје различити приступи и у начину на који се путници додељују одређеним путевима на мрежи. Уопштено, постоје два приступа: први, који подразумева да путници могу да користе искључиво један пут при кретању од почетне до крајње станице и други у коме се претпоставља да путници могу да користе већи број путева. У оквиру ове докторске дисертације коришћена су оба приступа.

5.3.2. Опис проблема

Нека је транспортна мрежа приказана на слици 5.5 представљена графом $G = (N, A)$, где је N скуп чворова, а A скуп грана (делова улица). Чворови мреже представљају аутобуске станице (а такође могу представљати и раскрснице или центре зона). Гране мреже представљају делове пута који повезују чворове, односно станице. У дисертацији се подразумева да се користе повезане, неоријентисане, мреже. Пут једног путника може се дефинисати као скуп чворова и грана.



Слика 5.5. Путна мрежа

У оквиру проблема пројектовања мреже линија тежи се налажењу најбољег могућег скупа линија *R*. Другим речима, потребно је одредити такав скуп линија којим би се постигли најбољи показатељи квалитета решења. Као главни показатељ квалитета решења (мреже аутобуских линија) коришћено је укупно време путовања свих путника на мрежи, које се може одредити на следећи начин:

$$T = TT + w_1 TTR + w_2 TU \tag{5.19}$$

где је:

TT – укупно време које путници проводе у возилу при кретању од почетне до крајње станице

TTR – укупан број преседања

TU – укупан број неопслужених путника (путник се третира као неопслужен ако не може да добије опслугу или му је потребно више од два преседања)

w_1 – пенал по једном преседању изражен у времену

w_2 – пенал по једном неопслуженом путнику изражен у времену

Када се вреднује квалитет решења узима се у обзир укупан број преседања путника с обзиром да је преседање непожељна активност при путовању. Очигледно, укупан број преседања путника може бити смањен оптмизацијом мреже аутобуских линија. Такође, при вредновању квалитета решења извршена је пенализација за сваког неопслуженог путника.

Нека је d_{ij} број путовања у јединици времена између чвора i и чвора j, а D изворно – циљна матрица:

$$D = \{d_{ij} | i, j \in [1, 2, \ldots, |N|]\} \tag{5.20}$$

Нека је матрица времена путовања возила између чворова означена са:

$$TR = \{tr_{ij} | i, j \in [1, 2, \ldots |N|]\} \tag{5.21}$$

где елемент матрице tr_{ij} представља потребно време путовања возила од чвора i и чвора j.

Као улазне податке за проблем пројектовања мреже линија потребно је имати: мрежу $G = (N, A)$, изворно-циљну матрицу D и матрицу времена путовања TR.

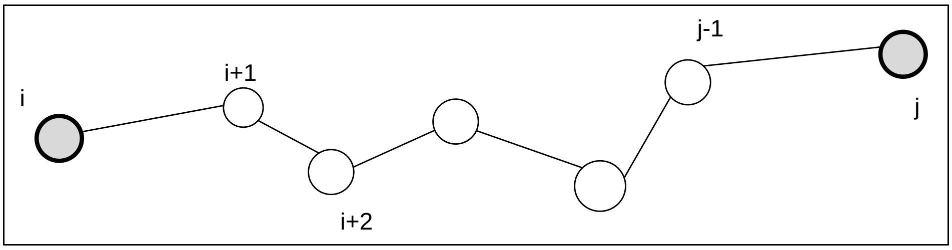
Проблем пројектовања мреже линија који је разматран у овој дисертацији може се дефинисати на следећи начин: за дати скуп чворова, изворно - циљну матрицу путовања (D) и познату матрицу времена путовања (TR) одредити скуп линија на мрежи на начин да се минимизира укупно време путовања (T) свих путника.

5.3.3. Решавање проблема применом Оптимизације колонијом пчела

За решавање овог проблема коришћена је верзија Оптимизације колонијом пчела са побољшањем решења. За одређивање почетног решења коришћен је прождрљиви хеуристички алгоритам, а након тога, за побољшање решења коришћена су два типа пчела. Разлике у типу пчела се огледају у начинима на који врше модификовање решења при лету унапред.

5.3.3.1. Одређивање почетног решења

Нека аутобуска линија l има терминале лоциране у чворовима i и j (Слика 5.6). Аутобуска линија l садржи све чворове који припадају најкраћем путу између чворова i и j. Нека је N_l скуп чворова повезаних линијом l.



Слика 5.6. Аутобуска линија чији су терминали лоцирани у чворовима *i* и *j*

Ова линија може бити коришћена од стране путника који путују без преседања, као и путника који имају највише два преседања. Укупан број путника ds_{ij} који могу да путују без преседања коришћењем линије *l* једнак је:

$$ds_{ij} = \sum_{m \in N_l} \sum_{n \in N_l} d_{mn} \tag{5.22}$$

Нека је са *DS* означена матрица чији су елементи ds_{ij}:

$$DS = \{ds_{ij} | i, j \in [1, 2, \ldots, |N|]\} \tag{5.23}$$

За добијање почетног решења је коришћен једноставан прождрљиви алгоритам. Овим алгоритмом се тежи да се аутобуске линије поставе између парова чворова који имају велику ds_{ij} вредност. На овај начин се тежи повећању броја путника који ће путовати без преседања. Алгоритам се састоји од следећих корака:

Корак 1: Дефинисати број аутобуских линија посматране мреже *NBL*. Нека је *Y* скуп аутобуских линија. Поставити да је *Y* = Ø и *m* = 1.

Корак 2: Пронаћи пар чворова који имају највећу вредност ds_{ij}. Нека је ово пар чворова (*a*, *b*). Чорови *a* и *b* су терминали нове линије *l*. Пронаћи најкраћи пут између ова два чвора. Чворови који припадају најкраћем путу представљају станице нове линије. Додати линију *l* у скуп *Y*.

Корак 3: Одредити поново вредности матрице *DS*, не узимајући у обзир захтеве који већ могу бити реализовани.

Корак 4: Ако је $m = NBL$, крај; у супротном, постави да је $m = m + 1$ и врати се на Корак 2.

5.3.3.2. Модификовање решења

Значајан корак у алгоритму је модификовање решења кроз NP летова унапред у свакој итерацији.

За решавање овог проблема у оквиру дисертације коришћене су хетерогене пчеле. Коришћена два типа пчела се разликују само по начину на који врше модификацију решења при лету унапред. Пчеле оба типа на исти начин доносе одлуку о начину опредељења о лојалности и праћењу.

5.3.3.2.1. Модификовање решења од стране пчела првог типа

На Слици 5.7 може се уочити скуп линија. Пчела првог типа бира једну линију из скупа. Ову одлуку пчела доноси на основу вероватноће. Вероватноћа да ће линија *l* бити одабрана се рачуна као:

$$p_l = \frac{\frac{1}{ds_{ij}}}{\sum_{q \in L} \frac{1}{ds_{rs}}} \tag{5.24}$$

где су:

i и j- терминали линије l,

L - скуп аутобуских линија,

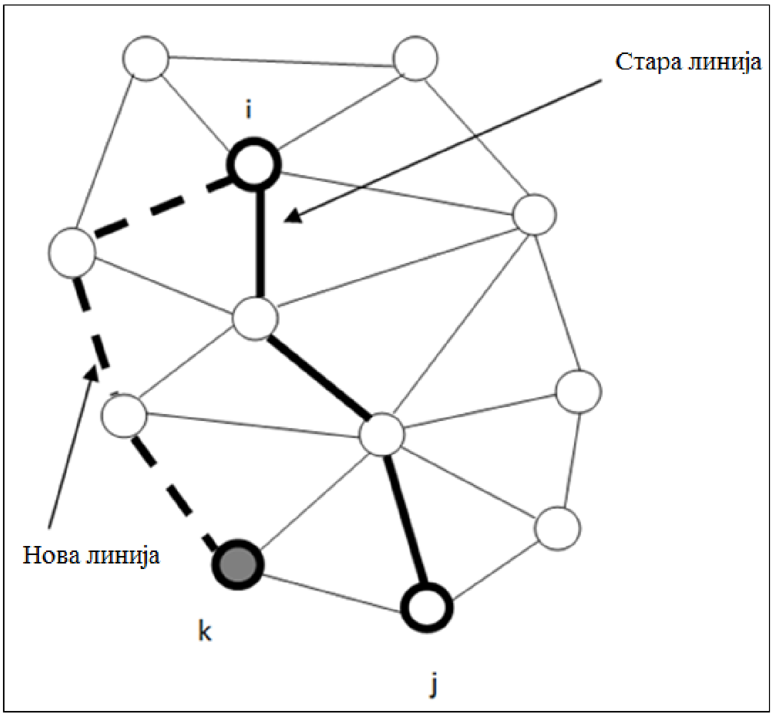
r и s - терминали линије q,

ds_{ij} – укупан број путника који могу путовати без иједног преседања користећи линију чији су терминали i и j.

Нека је пчела првог типа изабрала линију чији су терминали станице i и j. У следећем кораку пчела првог типа бира један од терминала на случајан начин. Претпоставимо да је пчела одабрала терминал i. Преостале станице и терминал који није изабран се искључују из линије. Нови терминал (k) линије се одређује на основу вероватноће:

$$P_k = \frac{ds_{ik}}{\sum_{r \in N} ds_{ir}} \tag{5.25}$$

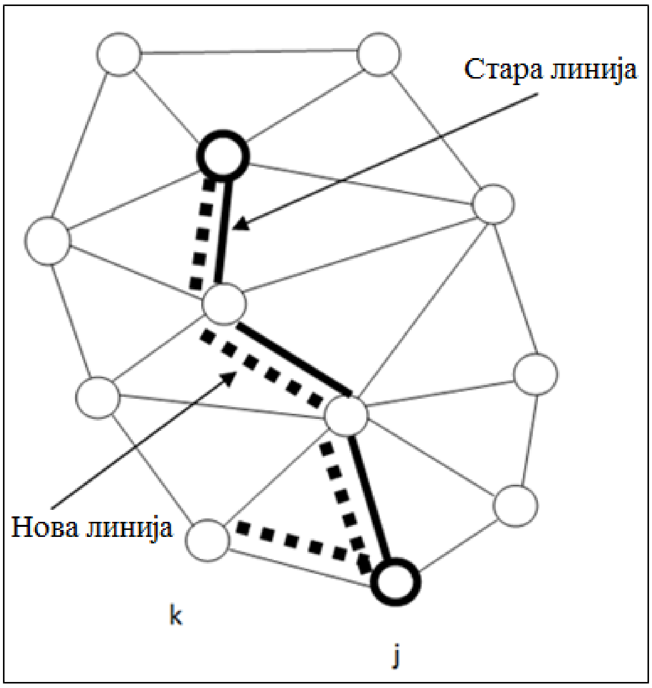
Нека је k нови терминал линије (видети Слику 5.7). Нову линију чине све станице које се налазе на најкраћем путу између два терминала. Нова аутобуска линија се добија налажењем најкраћег пута од чвора i до чвора j (испрекидана линија) и чине је све станице које се налазе на путу између два терминала.



Слика 5.7. Модификовање решења пчеле првог типа

5.3.3.2.2. Модификовање решења од стране пчела другог типа

Нека је скуп линија дат на транспортној мрежи приказаној на Слици 5.8. Пчела другог типа бира једну линију на исти начин као пчела првог типа. Након тога, пчела на случајан начин бира један од два терминала изабране линије. Нека је изабран терминал j. Са вероватноћом P пчела доноси одлуку да искључи један од терминала из линије. Нова аутобуска линија добијена на овај начин садржи све старе станице осим терминала који је избачен (линија је скраћена). У супротном случају, ако се терминал не избацује (терминал неће бити избачен из линије са вероватноћом (1-P)) стара линија биће продужена, тј. биће јој додата још једна станица. Линији се додаје станица која је суседна терминалу. Уколико постоји више суседних станица бира се једна на случајан начин. На Слици 5.8 станица k је изабрана да буде додата линији.

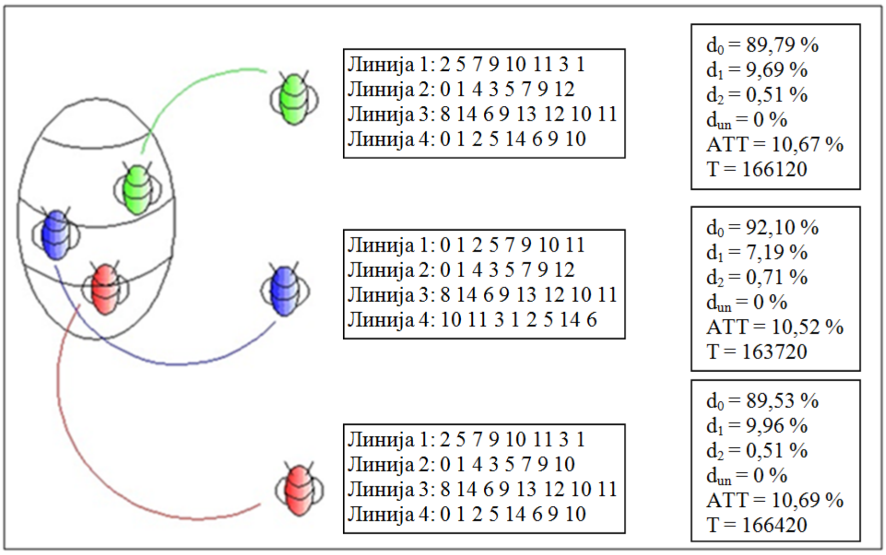


Слика 5.8. Модификовање решења пчеле другог типа

5.3.3.3. Лет уназад

Након модификовања решења пчеле првог и дргог типа се враћају у кошницу. Након повратка у кошницу пчеле врше вредновње својих решења (Слика 5.9).

Као квалитет решења b-те пчеле T_b (b=1, 2,..., B) за овај проблем је коришћено укупно време путовања свих путника, дефинисано изразом (5.19). Нормализација квалитета решења пчела, доношење одлуке о лојалности пчела својим решењима и регрутовање неопредељених следбеника је имплементирано на начин како је то описано у поглављу 3.



Слика 5.9. Вредновање генерисаних решења

5.3.4. Резултати тестирања

Предложен приступ решавања проблема применом *BCO* алгоритма упоређен је са другим приступима у литератури. Поређење је вршено на *Mandl*-овој мрежи предложеној у раду (Mandl, 1979). Тестирање BCO алгоритма је вршено и на мрежи која садржи 110 чворова и 275 грана. Ова мрежа је предложена у раду (Fan и Mumford, 2010).

У предложеном *BCO* алгоритму коришћени су следећи параметри *B* = 10, *IT* = 400, *NP* = 5, *NC* = 2. Као пенал за сваког неопслуженог путника је узето просечно време путовања + 50 минута, а за вероватноћу да ће терминал бити искључен из руте *P* = 0,2.

У првом експерименту поређене су вредности добијене BCO алгоритмом са другим приступима предложеним у радовима Mandl (1979), Baaj и Mahmassani (1991), Kidwai (1998), Chakroborty и Dwivedi (2002), и Fan и Mumford (2010). Поређење је направљено за случај Mandl-ове мреже.

За поређења квалитета решења су коришћени следећи показатељи:

d_0 - проценат захтева задовољених путника без преседања

d_1 - проценат захтева задовољених путника са једним преседањем

d_2 - проценат захтева задовољених путника са два преседања

d_{un} - проценат незадовољених захтева

ATT - просечно време путовања у минутима по путнику. Ово време укључује и пенале преседања. Пенал по једном преседању је 5 минута.

Овај скуп показатеља квалитета решења је предложен у раду Fan и Mumford (2010). Ови аутори су поредили резултате за случајеве када мрежа треба да има: 4, 6, 7 и 8 линија. Поређење резултата је дато у Табелама 5.5 - 5.9.

Почетна решења (скуп аутобуских линија) добијена прождрљивим алгоритмом су приказана у Табели 5.5.

Табела 5.5. Почетно решење

	Број аутобуских линија	Станице линије
1	4	0 1 2 5 7 9 10
		4 3 5 7 9 12
		8 14 6 9 13
		0 1 2 5 14 6
2	6	0 1 2 5 7 9 10
		4 3 5 7 9 12
		8 14 6 9 13
		0 1 2 5 14 6
		9 10 11
		0 1 3 11
3	7	0 1 2 5 7 9 10
		4 3 5 7 9 12
		8 14 6 9 13
		0 1 2 5 14 6
		9 10 11
		0 1 3 11
		11 10 12 13
4	8	0 1 2 5 7 9 10
		4 3 5 7 9 12
		8 14 6 9 13
		0 1 2 5 14 6
		9 10 11
		0 1 3 11
		11 10 12 13
		0 1 4

Поређење почетног решења BCO алгоритма и претходних прилаза датих у литератури дато је у Табели 5.6.

Табела 5.6. Поређења почетних решења добијених прождрљивим хеуристичким алгоритмом и решења добијених у литератури

Број линија	Показатељи квалитета	Mandl (1979)	Baaj and Mahmassani (1991)	Kidwai (1998)	Chakroborty и Dwivedi (2002)	Fan and Mumford (2008)	Прождрљиви алгоритам (почетно решење)
4	d_0	69,94	N	72,95	86,86	93,26	80,47
	d_1	29,93	N	26,92	12	6,74	12,33
	d_2	0,13	N	0,13	1,14	0	0,51
	d_{un}	0	N	0	0	0	6,68
	ATT	12,9	N	12,72	11,9	11,37	10,22
6	d_0	N	78,61	77,92	86,04	91,52	87,73
	d_1	N	21,39	19,68	13,96	8,48	11,75
	d_2	N	0	2,4	0	0	0,51
	d_{un}	N	0	0	0	0	0
	ATT	N	11,86	11,87	10,3	10,48	11,03
7	d_0	N	80,99	93,91	89,15	93,32	90,62
	d_1	N	19,01	6,09	10,85	6,36	8,86
	d_2	N	0	0	0	0,32	0,51
	d_{un}	N	0	0	0	0	0
	ATT	N	12,5	10,69	10,15	10,42	10,57
8	d_0	N	79,96	84,73	90,38	94,54	91,91
	d_1	N	20,04	15,27	9,62	5,46	7,58
	d_2	N	0	0	0	0	0,51
	d_{un}	N	0	0	0	0	0
	ATT	N	11,86	11,22	10,46	10,36	10,47

Може се уочити да је прождрљива хеуристика генерисала добра почетна решења. Вештачке пчеле су побољшавале ова почетна решења у процесу претраге. Коначна решења добијена BCO алгоритмом представљена су у Табели 5.7.

Табела 5.7. Коначно решење добијено BCO алгоритмом

Решење	Број линија	Станице линија
1	4	0 1 2 5 7 9 10 11
		1 4 3 5 7 9 12 10
		8 14 6 9 7 5 3 11
		3 1 2 5 14 6 9 13
2	6	0 1 2 5 7 9 10 12
		0 1 4 3 5 7 9 10
		8 14 6 9 13 12 10 11
		0 1 2 5 14 6 9 10
		14 7 9 10 11 3 1 0
		8 14 5 2 1 4 3 11

		0 1 2 5 7 9 13 12
		0 1 4 3 5 7 9 10
		8 14 6 9 13 12 10 11
3	7	0 1 2 5 14 6 9 10
		5 7 9 10 11 3 1 0
		8 14 7 5 2 1 3 4
		6 14 7 5 3 11 10 12
		0 1 2 5 7 9 10 12
		2 1 4 3 5 7 14 6
		8 14 6 9 10 11 3 5
4	8	0 1 2 5 14 6 9 13
		8 14 5 2 1 3 11
		0 1 3 11 10 12 13 9
		1 4 3 5 7 9 10 12
		0 1 4 3 11 10 12 13

Поређења коначних решења добијених BCO алгоритмом и решења добијених приступима предложеним у литератури дата су у Табели 5.8.

Табела 5.8. Поређења између коначних решења добијених BCO алгоритмом и претходним приступима

Број линија	Показатељи квалитета	Mandl (1979)	Baaj и Mahmassani (1991)	Kidwai (1998)	Chakroborty и Dwivedi (2002)	Fan и Mumford (2008)	BCO алгоритам
4	d_0	69,94	N	72,95	86,86	93,26	92,1
	d_1	29,93	N	26,92	12	6,74	7,19
	d_2	0,13	N	0,13	1,14	0	0,71
	d_{un}	0	N	0	0	0	0
	ATT	12,9	N	12,72	11,9	11,37	10,51
6	d_0	N	78,61	77,92	86,04	91,52	95,63
	d_1	N	21,39	19,68	13,96	8,48	4,37
	d_2	N	0	2,4	0	0	0
	d_{un}	N	0	0	0	0	0
	ATT	N	11,86	11,87	10,3	10,48	10,23
7	d_0	N	80,99	93,91	89,15	93,32	98,52
	d_1	N	19,01	6,09	10,85	6,36	1,48
	d_2	N	0	0	0	0,32	0
	d_{un}	N	0	0	0	0	0
	ATT	N	12,5	10,69	10,15	10,42	10,15
8	d_0	N	79,96	84,73	90,38	94,54	98,97
	d_1	N	20,04	15,27	9,62	5,46	1,03
	d_2	N	0	0	0	0	0
	d_{un}	N	0	0	0	0	0
	ATT	N	11,86	11,22	10,46	10,36	10,09

Резултати добијени BCO алгоритмом имају боље вредности за d_0, и d_1 у 3 од 4 случаја. BCO резултати за просечно време путовања по путнику су бољи у 3 од 4 случаја.

У другом експерименту је коришћена транспортна мрежа која има 110 чворова и 275 линкова. Укупан број дневних путовања у овом примеру је 3603360. Задатак је да се генерише 55 аутобуских линија. Укупан број станица у свакој линији не може бити већи од 29.

Почетна решења добијена прождрљивим хеуристичким алгоритмом дата су у Табели 5.9.

Табела 5.9. Линије добијене прождрљивим алгоритмом (други пример)

Линија	Станице линије
Линија 1	59 48 92 52 23 2 45 40 56 74 50 6 94 68 12 26 57 21 104 18 79 101
Линија 2	38 78 80 61 34 89 4 29 33 6 94 68 12 26 57 21 104 18 32 87
Линија 3	58 55 51 41 36 28 45 40 56 74 50 6 94 68 12 26 57 21 104 18 73
Линија 4	87 32 18 104 21 57 26 12 68 94 6 50 74 56 40 45 90 75 22 86 107
Линија 5	24 37 13 44 4 29 56 40 45 28 72 3 88
Линија 6	10 48 92 52 23 2 45 40 56 74 30 93 64 67 100
Линија 7	54 53 13 44 4 29 33 6 94 68 12 26 57 21 104 18 79 101
Линија 8	8 108 52 91 19 65 89 44 13 31 82 15
Линија 9	82 5 105 95 33 6 94 68 12 26 35 102
Линија 10	58 55 51 41 36 28 45 2 23 52 92 48 47 71
Линија 11	0 70 26 12 68 94 6 50 74 56 40 45 2 23 108 8
Линија 12	47 46 61 34 89 4 29 50 30 93 64 67
Линија 13	58 55 51 41 36 28 45 40 56 29 4 89 76 60
Линија 14	99 78 48 92 52 23 2 45 90 75 22 86 107
Линија 15	17 9 68 94 6 50 74 56 40 45 2 23 52 92 48 25
Линија 16	81 37 13 44 89 65 19 91 23 2 1 84
Линија 17	87 32 18 104 21 57 26 12 68 94 6 50 74 56 40 45 28 72 3 88
Линија 18	27 105 95 4 65 19 91 52 92 48 103
Линија 19	15 5 105 95 29 56 40 45 90 75 22 86 107
Линија 20	11 1 2 45 40 56 74 50 6 94 68 12 26 35 102
Линија 21	7 49 68 94 6 50 74 56 40 45 2 23 52 92 48 103
Линија 22	87 32 18 104 21 57 26 12 68 94 6 33 29 65 19 91
Линија 23	10 59 46 76 44 13 31 5 27
Линија 24	0 70 26 12 68 94 6 33 29 4 89 34 61 46 69
Линија 25	38 10 48 92 52 23 2 45 28 72 3 88
Линија 26	8 1 84 106 36 72 3 86 107
Линија 27	54 53 13 44 4 29 56 40 45 90 75 22 86 107
Линија 28	15 5 105 95 33 6 94 68 12 26 57 21 104 18 32 87
Линија 29	16 9 68 94 6 33 29 4 89 34 61 46 71
Линија 30	84 106 28 45 40 56 74 50 6 94 68 12 26 57 21 104 18 79 101
Линија 31	24 37 13 44 4 29 33 6 94 68 12 26 57 21 104 18 73

Линија 32	15 82 31 13 44 76 46 47 48 103
Линија 33	98 64 7 12 26 57 21 104 18 79 101
Линија 34	58 55 51 41 36 28 45 40 56 74 30 93 64 67 100
Линија 35	62 1 2 45 40 56 74 50 6 94 68 12 26 57 21 104 18 32 87
Линија 36	63 41 36 28 45 40 56 29 4 89 34 66
Линија 37	60 34 61 46 47 48 92 11 62
Линија 38	20 75 90 45 40 56 74 50 6 94 68 12 26 57 21 104 18 79 101
Линија 39	15 5 105 95 33 50 30 93 64 67
Линија 40	99 61 34 89 4 29 33 6 94 68 12 26 35 102
Линија 41	39 75 90 45 2 23 52 92 48 59 69
Линија 42	14 5 31 13 44 76 61 80 78 38
Линија 43	100 67 64 93 30 74 56 40 45 90 75 22 86 107
Линија 44	58 55 51 41 36 28 45 40 56 29 95 105 5 82
Линија 45	47 46 61 34 89 4 29 33 6 94 68 12 26 57 21 104 18 79 101
Линија 46	11 108 52 91 19 65 4 95 105 14
Линија 47	43 9 68 94 6 33 29 4 44 13 97
Линија 48	67 64 93 30 50 29 4 44 13 37 81
Линија 49	77 30 94 68 12 26 57 21 104 18 79 101
Линија 50	60 76 89 4 29 33 6 94 68 12 26 57 21 104 18 73
Линија 51	54 53 13 44 89 65 19 91 52 108 62
Линија 52	14 105 95 33 6 94 68 12 26 57 21 104 18 79 101
Линија 53	54 53 13 44 4 29 56 40 45 28 36 41 51 55 58
Линија 54	88 3 72 28 45 40 56 74 30 93 64 67 100
Линија 55	11 1 84 106 36 41 51 55 58

Карактеристике добијеног почетног решења су:

Укупан број путовања без преседања = 1438572 (d_0 = 39,92%)

Укупан број путовања са једним преседањем = 1607740 (d_1= 44,62%)

Укупан број путовања са два преседања = 224480 (d_2 = 6,23 %)

Укупан број неопслужених путника = 332568 (d_{un} = 9,23 %)

Просечно време путовања по путовању ATT = 34,90 min.

Као и у претходном примеру, BCO алгоритам је значајно повећао квалитет почетног решења. Коначно шерење добијено BCO алгоритмом је дато у Табели 5.10.

Табела 5.10. Линије добијене BCO алгоритмом (коначно решење)

Линија	Станице линије
Линија 1	80 78 46 99 69 38 10 25 48 92 52 23 2 45 40 56 74 50 6 94 68 12 26 57 21 104 18 79 101
Линија 2	103 25 59 47 48 10 38 78 80 61 34 89 4 29 33 6 94 68 12 26 57 21 104 18 32 87 73 79 101
Линија 3	58 55 51 41 36 28 45 40 56 74 50 6 94 68 12 26 57 21 104 18 73 87 32 79 101
Линија 4	79 73 32 18 104 21 57 26 12 68 94 6 50 74 56 40 45 90 75 22 86 107
Линија 5	54 53 81 24 37 13 44 4 29 56 40 45 28 72 3 39 20 88 75 36 106 84 1 62 11 108 52 91 83

Линија 6	99 78 38 25 10 48 92 52 23 2 45 40 56 74 30 93 64 67 100 7 68 94 85 50 29 65 89 76 60
Линија 7	19 65 109 95 96 105 5 97 37 81 54 53 13 44 4 29 33 6 94 68 12 26 57 21 104 18 79 32 87
Линија 8	76 61 99 78 69 59 48 8 108 52 91 19 65 89 44 13 31 82 15 5 105 96 95 109 33 74 85 6 42
Линија 9	54 37 97 82 5 105 95 33 6 94 68 12 26 35 102 0 70 57 16 17 49 100 9 7 67 64 98 30 85
Линија 10	58 55 51 41 36 28 45 2 23 52 92 48 47 71 80 61 76 60 34 89 65 19 91 83 108 11 62 1 84
Линија 11	98 64 67 100 49 7 9 17 43 16 57 102 0 70 26 12 68 94 6 50 74 56 40 45 2 23 108 92 8
Линија 12	38 10 103 48 25 59 69 80 47 46 61 34 89 4 29 50 30 93 64 67 100 49 7 17 35 70 0 102 21
Линија 13	58 55 51 41 36 28 45 40 56 29 4 89 76 60 34 66 61 80 99 46 71 47 48 10 103 25 59 69 78
Линија 14	59 10 103 25 38 69 47 46 99 78 48 92 52 23 2 45 90 75 22 86 107
Линија 15	35 57 16 43 17 9 68 94 6 50 74 56 40 45 2 23 52 92 48 25 38 69 47 71 46 76 44 13 53
Линија 16	14 27 5 15 82 97 81 54 24 37 13 44 89 65 19 91 23 2 1 84 106 36 28 90 75 41 51 55 58
Линија 17	73 87 32 18 104 21 57 26 12 68 94 6 50 74 56 40 45 28 72 3 88 86 22 75 36 106 84 1 8
Линија 18	15 82 5 27 105 95 4 65 19 91 52 92 83 108 11 62 8 48 78 46 61 66 34 89 44 13 31 97 81
Линија 19	10 59 46 61 34 60 76 44 13 37 24 53 54 81 97 82 15 5 105 95 29 56 40 45 90 75 22 86 107
Линија 20	23 83 92 11 1 2 45 40 56 74 50 6 94 68 12 26 35 102 21 70 57 16 43 17 7 64 98 30 77
Линија 21	43 17 26 57 21 0 70 102 35 12 9 7 49 68 94 6 50 74 56 40 45 2 23 52 92 48 103 10 38
Линија 22	73 87 32 18 104 21 57 26 12 68 94 6 33 29 65 19 91 83 92 48 59 47 46 61 66 76 44 4 95
Линија 23	66 61 80 78 48 25 103 10 59 46 76 44 13 31 5 27 105 96 95 4 109 65 19 91 52 23 108 11 62
Линија 24	32 73 18 104 21 102 0 70 26 12 68 94 6 33 29 4 89 34 61 46 69 38 25 59 10 48 92 11 62
Линија 25	82 5 105 96 95 109 65 89 34 61 80 71 69 59 25 38 10 48 92 52 23 2 45 28 72 3 88 75 20
Линија 26	71 47 59 78 38 69 80 61 66 76 89 65 19 91 83 23 52 92 11 108 8 1 84 106 36 72 3 86 107
Линија 27	5 82 31 97 81 24 37 54 53 13 44 4 29 56 40 45 90 75 22 88 20 39 72 28 36 41 3 86 107
Линија 28	37 81 97 24 54 53 13 31 82 15 5 105 95 33 6 94 68 12 26 57 21 104 18 32 87 73 79 101
Линија 29	49 17 26 70 35 12 43 16 9 68 94 6 33 29 4 89 34 61 46 71 80 78 59 25 48 92 108 52 83
Линија 30	69 78 59 48 92 11 8 1 84 106 28 45 40 56 74 50 6 94 68 12 26 57 21 104 18 79 73 87 32
Линија 31	54 53 81 24 37 13 44 4 29 33 6 94 68 12 26 57 21 104 18 73 87 32 79 101
Линија 32	12 68 94 42 50 6 93 98 85 74 77 56 29 95 105 14 5 15 82 31 13 44 76 46 47 48 103 25 10
Линија 33	48 47 46 76 89 65 109 33 74 50 30 94 42 6 93 85 98 64 7 12 26 57 21 104 18 79 73 32 87
Линија 34	58 55 51 41 36 28 45 40 56 74 30 93 64 67 100 49 7 12 17 35 102 0 21 104 18 73 32 79 101
Линија 35	31 13 44 4 65 19 91 52 92 8 62 1 2 45 40 56 74 50 6 94 68 12 26 57 21 104 18 32 87
Линија 36	63 41 36 28 45 40 56 29 4 89 34 66 76 44 13 53 81 97 31 82 14 27 105 95 33 50 30 98 93
Линија 37	23 91 19 65 89 44 76 34 61 46 47 48 92 11 62 8 1 2 45 28 36 75 3 51 63 41 72 39 20
Линија 38	23 92 8 1 84 106 28 36 72 39 20 75 90 45 40 56 74 50 6 94 68 12 26 57 21 104 18 79 101
Линија 39	54 81 53 24 97 82 15 5 105 95 33 50 30 93 64 67 100 9 12 43 16 57 21 104 18 73 87 32
Линија 40	36 106 84 1 8 48 10 25 38 78 69 99 61 34 89 4 29 33 6 94 68 12 26 35 17 49 100 9 7
Линија 41	11 62 1 84 106 36 72 41 63 51 3 39 75 90 45 2 23 52 92 48 59 69 78 99 46 76 44 13 97
Линија 42	68 49 7 64 93 85 77 74 50 33 95 105 14 5 31 13 44 76 61 80 69 99 46 59 47 48 78 38 10
Линија 43	101 79 73 87 32 18 104 21 57 26 17 9 100 67 64 93 30 74 56 40 45 90 75 22 86 107
Линија 44	58 55 51 41 36 28 45 40 56 29 95 105 5 82 97 31 13 44 89 65 19 91 83 92 11 8 62 1 2
Линија 45	69 99 78 48 10 103 25 59 47 46 61 34 89 4 29 33 6 94 68 12 26 57 21 104 18 79 32 87 73
Линија 46	58 55 51 3 39 20 75 36 28 45 2 23 83 92 8 1 62 11 108 52 91 19 65 4 95 105 14 82 15
Линија 47	12 17 43 9 68 94 6 33 29 4 44 13 97 82 14 27 105 96 95 109 65 19 91 83 92 48 103 10 38
Линија 48	26 17 16 9 7 67 64 93 30 50 29 4 44 13 37 81 97 5 31 82 14 105 96 95 109 33 74 6 42
Линија 49	24 37 53 81 97 5 27 105 96 95 29 109 33 42 50 77 30 94 68 12 26 57 21 104 18 79 32 73 87
Линија 50	38 69 71 46 80 99 61 66 34 60 76 89 4 29 33 6 94 68 12 26 57 21 104 18 73 87 32 79 101
Линија 51	5 15 82 97 37 81 54 53 13 44 89 65 19 91 52 108 11 1 84 106 36 72 39 75 20 3 51 55 58
Линија 52	31 13 37 54 53 24 97 82 5 14 105 95 33 6 94 68 12 26 57 21 104 18 79 101
Линија 53	95 96 105 14 27 5 97 37 54 53 13 44 4 29 56 40 45 28 36 41 51 3 86 22 88 39 72 75 90
Линија 54	41 75 20 39 88 3 72 28 45 40 56 74 30 93 64 67 100 7 68 94 42 33 95 96 105 14 82 97 24
Линија 55	27 14 82 15 5 105 95 4 44 76 46 59 48 8 11 1 84 106 36 41 51 55 58

Добијено почетно решење има следеће карактеристике:

d_0 = 59,65 %

d_1 = 40,10 %

d_2 = 0,25 %

d_{un} = 0 %

ATT = 36,16 min

Добијено просечно време путовања, *ATT,* (у минутима по путнику) се састоји од следећих компоненти:

$$36{,}16 = 34{,}13 + 2{,}03$$

Први део представља просечно време путовања, док други део представља просечан пенал по једном преседању путника. Да би се боље оценио квалитет добијеног крајњег решења израчунато је просечно време путовања по путнику у случају када путници искључиво користе најкраће путеве у мрежи при путовању од почетне до крајње станице. Ово просечно време износи 33,84 минута. Уједно, ово је и доња граница просечног времена путовања. Вредност средњег времена путовања (*ATT*) добијена BCO алгоритмом је релативно близу доње границе из чега се може закључити да је квалитет решења добијеног BCO алгоритмом релативно висок.

5.3.5. Пројектовање аутобуских линија са истовременим одређивањем фреквенција

У циљу одређивања фреквенција на линијама масовног превоза претходно је потребно одредити део линије на коме се јавља највећи број путника. При симултаном пројектовању мреже линија и одређивању вредности фреквенција, аутори најчешће не користе принцип да се путници крећу од почетне до крајње станице путем који им омогућава најкраће време путовања.

У овој дисертацији су приликом симултаног пројектовања мреже линија и одређивања вредности фреквенција коришћене следеће претпоставке: (а) Путници бирају линију превоза, пре свега, на основу вредности фреквенције на линији (б) Путници којима је потребно два или више преседања да би дошли до одредишне станице третирају се као неопслужени. Другим речима, расподела путника на мрежи вршена је само за путнике који могу да изврше путовање без преседања или са највише једним преседањем.

Расподела путника на мрежи у овом делу дисертације је базирана на радовима чији су аутори Shih и остали (1994) и Shih и остали (1997).

Процедура расподеле путника на мрежи се може представити следећим псеудо кодом:

процедура РасподелаПутника()

do

    for *i = 1* ***to*** *n*

        for *j = 1* ***to*** *n*

            if *број путника који путују од i до j > 0* ***then***

                Одреди најмањи број преседања при путовању од i до j

                if *могуће путовање од i до j без преседања* ***then***

                    Расподелити путнике на мрежи који путују од i до j користећи једну линију.

                else if *од i до j могће путовање са једним преседањем* ***then***

                    Расподели путнике на мрежи који путују од i до j користећи две линије.

                else

                    Путнике који путују од i до j прогласити неопслуженим.

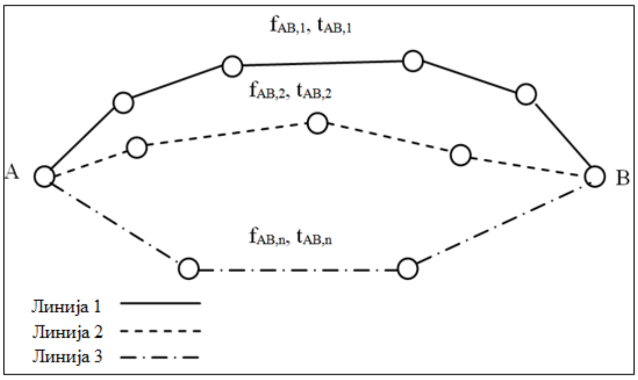
    За сваку линију одреди број аутобуса и фреквенције.

while *(фреквенција конвергира)*

5.3.5.1. Расподела путника на мрежи у случају путовања без преседања

На Слици 5.10 може се уочити транспортна мрежа на којој путници од станице *A* треба да стигну до станице *B*. Нека је *L* скуп аутобуских линија на мрежи. Такође нека је $L_{AB} = \{l \in L \mid A \in l, B \in l\}$ скуп аутобуских линија којима је могуће од станице *A* стићи до станице *B*. Чворови *A* и *B* код неких линија представљају терминале, док код других представљају станице.

Нека су f_l и $t_{AB,l}$ фреквенција и време путовања од станице *A* до станице *B* коришћењем линије $l \in L_{AB}$.



Слика 5.10. Доступне линије путницима који путују од станице *A* до станице *B*

Нека је $t_{AB,\min}$ минимално време путовња од станице *A* до станице *B*, тј.:

$$t_{AB,\min} = \min_{l \in L_{AB}} \{t_{ABl}\} \tag{5.26}$$

Времена путовања одређеним линијама могу бити значајно већа од минимално потребног. Из тог разлога је претпостављено да путници такве линије неће узимати у обзир при планирању путовања. Другим речима при путовању од станице *A* до станице *B* могуће је дефинисати потенцијални скуп линија $L_{AB,t}$ које ће путници узимати у разматрање при путовању. Скуп $L_{AB,t}$ могуће је одредити на следећи начин:

$$L_{AB,t} = \{l \mid l \in L_{AB};\, t_{ABl} \leq c_t\, t_{AB\min}\}, \tag{5.27}$$

где је:

$c_t \geq 1$- коефицијент толеранције (праг) који дефинише аналитичар.

Број путника $p_{AB,i}$ који ће користити линију *i* је једнак:

$$p_{AB,i} = \frac{f_{AB,i}}{\sum\limits_{l \in L_{ABt}} f_{ABl}} d_{AB} \tag{5.28}$$

где је:

d_{AB}– укупан број путника који путују од станице *A* до станице *B*.

Релација (5.28) одражава уведену претпоставку да путници бирају линију превоза, пре свега, на основу вредности фреквенција на линији.

5.3.5.2. Расподела путника на мрежи у случају путовања са једним преседањем

У случају путовања са једним преседањем путник бира линије два пута: на почетној станици и на станици преседања. Као и у претходном случају, претпостављено је да путници не разматрају путеве који захтевају дуга времена путовања.

Нека је на транспортној мрежи приказаној на Слици 5.11 потребно расподелити путнике који путују од станице *A* до станице *B*. Такође, нека су могући следећи путеви:

Пут 1: $A \xrightarrow{\text{линија1}} 1 \xrightarrow{\text{линија1}} 4 \xrightarrow{\text{линија1}} 7\,(\text{преседање}) \xrightarrow{\text{линија4}} 11 \xrightarrow{\text{линија4}} B$
Пут 2: $A \xrightarrow{\text{линија2}} 2 \xrightarrow{\text{линија2}} 5\,(\text{преседање}) \xrightarrow{\text{линија5}} 10 \xrightarrow{\text{линија5}} B$
Пут 3: $A \xrightarrow{\text{линија2}} 2 \xrightarrow{\text{линија2}} 5\,(\text{преседање}) \xrightarrow{\text{линија6}} 9 \xrightarrow{\text{линија6}} B$
Пут 4: $A \xrightarrow{\text{линија2}} 2\,(\text{преседање}) \xrightarrow{\text{линија3}} 8 \xrightarrow{\text{линија3}} B$

На станици *A* путници се расподељују на линије 1 и 2 на следећи начин:

$$p_{ABl_1} = \frac{f_{l_1}}{f_{l_1} + f_{l_2}} d_{AB} \tag{5.29}$$

и

$$p_{ABl_2} = \frac{f_{l_2}}{f_{l_1} + f_{l_2}} d_{AB} \tag{5.30}$$

Постоји само један пут (Пут 1) који користи линију 1. Због тога се сви путници који користе линију 1 додељују том путу.

Са друге стране, постоје три пута који садрже линију 2. У овом случају путници са линије 2 ће бити подељени на три једнака дела и сваки додељен једном путу:

$$p_{ABl_2,p_2} = p_{ABl_2,p_3} = p_{ABl_2,p_4} = \frac{p_{ABl_2}}{3} \tag{5.31}$$

Путеви који имају исту почетну линију и станицу преседања чине једну групу. У примеру приказаном на Слици 5.11 постоје следеће 3 групе: прва има један пут: Пут 1, друга група има два пута: Пут 2 и Пут 3, и на крају, трећа група има један пут: Пут 4.

У разматраном примеру, сви путници из прве групе се додељују линији 4. Слично је са трећом групом где сви путници који путују од станице 2 до станице *B* користе линију 3. У случају групе 2, путници који дођу у станицу 5 биће додељени линијама 5 и 6 према правилу поделе на основу фреквенција.

Укупан број путника који долазе у станицу 5 је једнак:

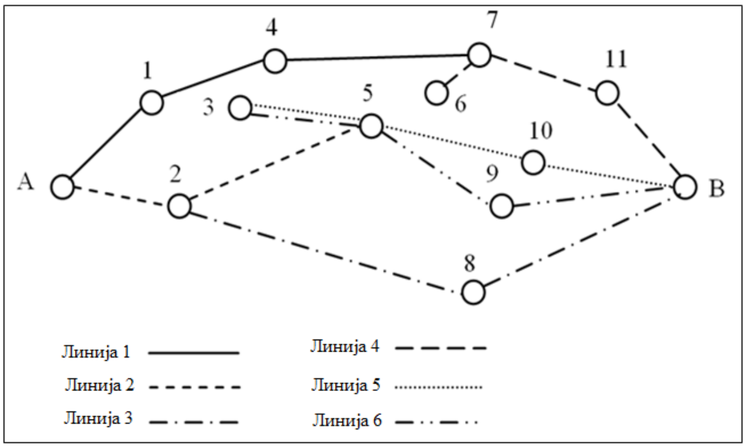
$$p_{ABl_2,5} = p_{ABl_2,p_3} + p_{ABl_2,p_4} \tag{5.32}$$

Број путника који користе линију 5 за путовање од станице 5 до станице *B* је једнак:

$$p_{ABl_2,5,l_5} = \frac{f_{l_5}}{f_{l_5} + f_{l_6}} p_{ABl_2,5} \tag{5.33}$$

Број путника који ће да користе линију 6 за путовање од станице 5 до станице *B* је једнак:

$$p_{ABl_2,5,l_6} = \frac{f_{l_6}}{f_{l_5} + f_{l_6}} p_{ABl_2,5} \tag{5.34}$$



Слика 5.11. Расподела путника при путовању са једним преседањем

5.3.5.3. *Одређивање карактеристика линије*

Након расподеле путника на мрежи, могу се одредити основне карактеристике линија као што су: фреквенција аутобуса на линијама, потребни бројеви аутобуса, интервали слеђења, и др. Фреквенција аутобуса на линији *k* се одређује као (Shih и Mahmassani, 1994):

$$f_k = \frac{Q_{k,\max}}{\alpha\, C_k} \tag{5.35}$$

где је:

$Q_{k,\,max}$ - максималан број путника на линији *k*

α – коефицијент попуњености

C_k – расположиви број места у аутобусу који се користе на линији *k*.

Потребан број аутобуса који треба да буде додељен линији *k* је (Shih и Mahmassani, 1994):

$$N_{sk} = \frac{f_k\, T_k}{60} \tag{5.36}$$

где је:

T_k [h] – време обрта на линији k

Укупно време превоза путника се може изразити као:

$$t_{tt} = t_v + t_w + t_t \tag{5.37}$$

где су:

t_v – време проведено у возилу

t_w – време чекања

t_t – пенал изражен у времену због преседања једног путника

Интервал слеђења аутобуса на линији k је једнак:

$$h = \frac{60}{f} = \frac{60}{\sum_{l \in L_{ABt}} f_l} \tag{5.38}$$

Време чекања путника је једнако:

$$t_w = \frac{1}{2} h = \frac{60}{2 \sum_{l \in L_{ABt}} f_l} = \frac{30}{\sum_{l \in L_{ABt}} f_l} \tag{5.39}$$

5.3.5.4. Резултати тестирања

Предложени алгоритам базиран на BCO метахеристици тестиран је на примеру *Mandl*-ове мреже (Mandl, 1979). Претпостављено је да је максималан укупан број аутобуса који је на располагању 99. За BCO алгоритам коришћени су следећи параметри:

- Број итерација: *IT* = 200
- Број летова напред/унатраг: *NP* = 5
- Број промена у сваком лету унапред: *NC* = random (1, 3)
- Број пчела: *B* = 20 (10 пчела сваког типа)

У Табели 5.11 су приказане генерисане линије и потребан број аутобуса за сваку од линија. Линије су добијене прождрљивом хеуристиком и BCO алгоритмом. У табели су дате две групе линија добијене BCO метахеристиком: 1) скуп линија који је најбољи за путнике и 2) скуп линија који је најбољи могућ за даваоца услуге (превозиоца). У Табели 5.12 су дати показатељи квалитета решења линија датих у Табели 5.11. Може се уочити да је BCO алгоритам значајно побољшао почетно решење.

Табела 5.11. Решења добијена BCO алгоритмом

Број линија	Почетно решење		BCO (Најбоље за путнике)		BCO (Најбоље за превозиоца)	
	Станице линија	Број аутобуса	Станице линија	Број аутобуса	Станице линија	Број аутобуса
4	0 1 2 5 7 9 10	31	0 1 2 5 7 9 10 11	35	0 1 2 5 7 9 10 12	34
	4 3 5 7 9 12	25	0 1 4 3 5 7 9 13 12	30	4 3 5 7 9	11
	8 14 6 9 13	14	8 14 6 9 10 12 13	14	8 14 6 9 13 12	6
	0 1 2 5 14 6	7	6 14 5 2 1 3 11 10 12	15	3 1 2 5 14 6 9 10 11	16
6	0 1 2 5 7 9 10	25	0 1 2 5 7 9 10 11	34	0 1 2 5 7 9 10 12	35
	4 3 5 7 9 12	21	11 10 9 7 5 3	1	4 3 5 7 9	20
	8 14 6 9 13	12	8 14 6 9 10	11	8 14 6 9 13 12	5
	0 1 2 5 14 6	7	0 1 2 5 14 6	7	3 1 2 5 14 6	5
	9 10 11	5	10 9 7 5 3 4 1 2	23	6 9 10 11	8
	0 1 3 11	5	0 1 3 11 10 12 13 9	23	8 14 5 3	1
7	0 1 2 5 7 9 10	26	0 1 2 5 7 9 10 11	32	1 2 5 7 9 10	1
	4 3 5 7 9 12	21	2 1 4 3 5 7 9 10	18	2 1 4 3 5 7 9 10	22
	8 14 6 9 13	12	8 14 6 9 10	6	8 14 6 9 10 11	15
	0 1 2 5 14 6	7	0 1 2 5 14 6	7	0 1 2 5 14 6	7
	9 10 11	4	1 3 5 7 9 13	1	10 9 7 5 2 1 0	24
	0 1 3 11	5	10 9 6 14 8	6	0 1 3 11 10 12 13 9 7 14	29
	11 10 12 13	3	0 1 3 11 10 12 13 9 7 5	29	10 9 7 5	1
8	0 1 2 5 7 9 10	26	0 1 2 5 7 9 10 11	16	0 1 2 5 7 9 10	12
	4 3 5 7 9 12	21	2 1 4 3 5 7 9 10 11	18	0 1 4	1
	8 14 6 9 13	12	10 9 7 5 3 4 1 0	12	8 14 6 9	2
	0 1 2 5 14 6	7	0 1 2 5 14 6 9	10	4 1 2 5 14 6 9	10
	9 10 11	4	0 1 2 5 7 9 10	10	12 13 9 7 5 3 1	18
	0 1 3 11	4	0 1 3 11 10 12	6	10 11 3 5 14 8	5
	11 10 12 13	3	11 10 12 13 9 7 14 5 2 1 4	19	11 10 12 13	3
	0 1 4	1	12 10 9 6 14 8	8	0 1 2 5 7 9 10	12
12	0 1 2 5 7 9 10	20	10 9 7	1	0 1 2 5 7 9 10	21
	4 3 5 7 9 12	1	5 7 9 10	1	4 3 5 14 8	4
	8 14 6 9 13	12	13 12 10 9 6 14 8	14	8 14 6 9	2
	0 1 2 5 14 6	5	0 1 2 5 14 6	7	10 9 7 5 3 4	1
	9 10 11	4	9 10 11 3 5 7	2	11 10 12	2
	0 1 3 11	4	0 1 3 11 10 12	6	1 3 11	2
	11 10 12 13	3	5 2 1 3 11	1	11 10 12	2
	0 1 4	1	7 14 5 3 4	1	4 3 5 7 9 10	1
	0 1 2 5 14 8	2	0 1 2 5 7 9 13 12	26	4 3 5 7 9 10	1
	4 3 5 14 6	1	9 7 5 3 4 1 2	1	3 5 14 6 9	8
	4 3 5 7 9 10	19	2 1 4 3 5 7 9 10 11	37	10 9 7 5 3 4	1

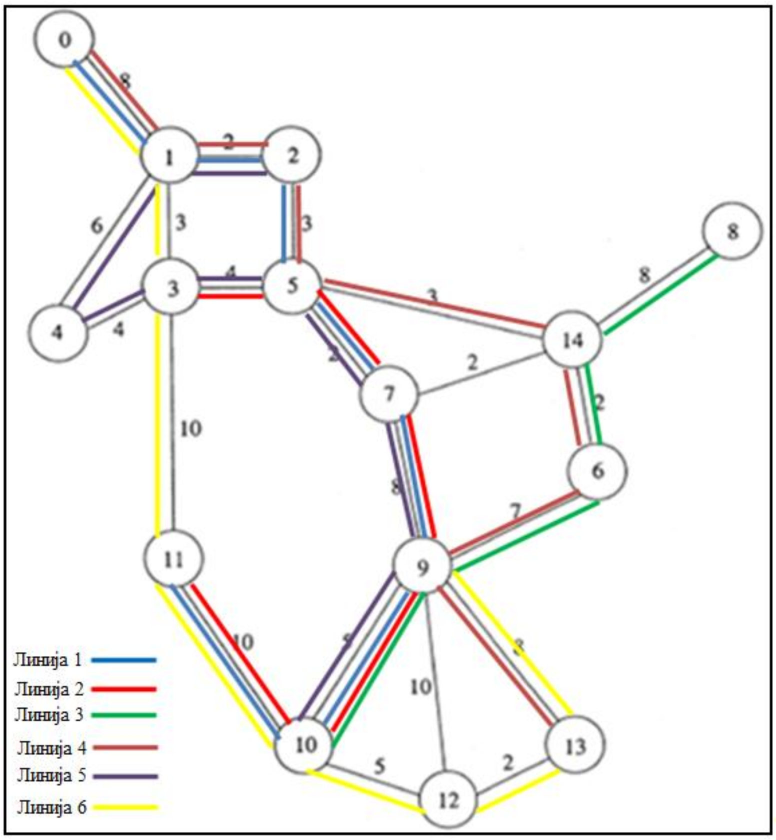
	0 1 2 5 7 9 12	13	5 7 9	1	0 1 2 5 7 9 12 13	20

У Табели 5.12 су дати показатељи квалитета генерисаних скупова линија, као и показатељи квалитета решења добијених другим приступима у литератури. Може се уочити да је хеуристика за одређивање почетног решења пронашла добра решења, при чему само у случају када је потребно одредити четири линије за дату мрежу постоји одређени број неопслужених путника (6,68 %).

Табела 5.12. Поређење решења добијених BCO алгоритмом и решења добијених другим приступима

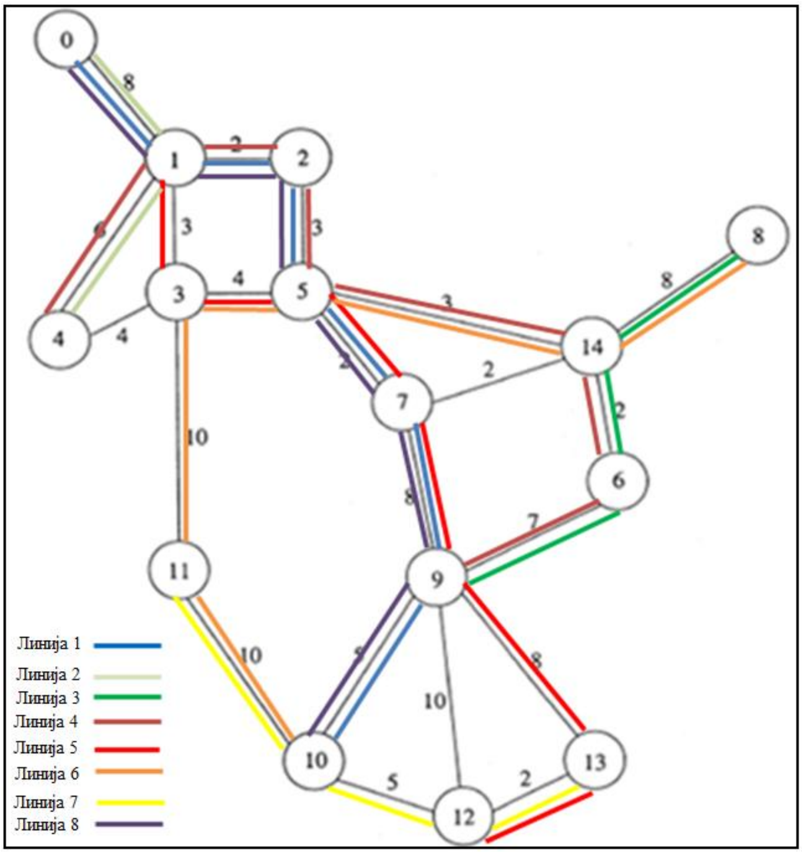
Број линија	Метода решавања	Број возила	Проценат захтева				Времена (min)			
			d_0	d_1	d_2	d_{un}	Време путовања	Време у возилу	Време чекања	Пенали преседања
4	Mandl	99	69,94	29,93	0,13	0	219094	177400	18194	23500
	Прождрљиви алгоритам	70	80,48	12,84	0	6,68	180453	149904	20549	10000
	BCO (путници)	94	95,05	4,95	0	0	186368	161371	21147	3850
	BCO (превозилац)	67	90,69	9,31	0	0	197624	164317	26057	7250
6	Shih и остали Coordinated	87	82,59	17,41	0	0	225102	191826	19726	13550
	Shih и остали	84	82,59	17,41	0	0	203936	170328	20058	13550
	Прождрљиви алгоритам	75	87,73	12,27	0	0	199908	163020	27338	9550
	BCO (путници)	99	94,34	5,65	0	0	185224	159059	21766	4400
	BCO (превозилац)	66	89,98	10,02	0	0	201066	161765	31500	7800
7	Baaj и остали.	82	80,99	19,01	0	0	217954	180356	22804	14800
	Прождрљиви алгоритам	78	90,62	9,38	0	0	195477	158100	30076	7300
	BCO (путници)	99	94,41	5,59	0	0	185405	157899	23157	4350
	BCO (превозилац)	63	87,80	12,20	0	0	204006	159024	35481	9500
8	Shih et al.	68	87,73	12,27	0	0	204028	168023	26455	9550
	Baaj et al.	77	79,96	20,04	0	0	209318	166654	27064	15600
	Прождрљиви алгоритам	78	91,91	8,09	0	0	197516	157950	33266	6300
	BCO (путници)	99	96,40	3,60	0	0	185590	158064	24726	2800
	BCO (превозилац)	63	88,57	11,43	0	0	203975	160144	34931	8900
12	Bagloee	87	83,66	15,21	0.95	0	202255	167198	24591	10465
	Прождрљиви алгоритам	85	95,50	4,50	0	0	200624	156769	40355	3500
	BCO (путници)	98	95,38	4,62	0	0	187919	160452	23867	3600
	BCO (превозилац)	65	85,74	14,26	0	0	208355	161204	36051	11100

ВСО алгоритам је значајно поправио почетно решење. ВСО алгоритам (најбоље за путнике) за мрежу од 6 аутобуских линија (Слика 5.12) има најмање време путовања (укупно време путовања 185224 или просечно 11,9 минута по путнику).



Слика 5.12. Најбоље линије са аспекта путника добијене ВСО алгоритмом

Најмањи број аутобуса (63) је добијен ВСО алгоритмом за мрежу од 7 и 8 аутобуских линија. Међутим, мрежа са 8 аутобуских линија је боља с обзиром да је проценат реализованих захтева без преседања већи и да је укупно време путовања мање. На Слици 5.13 је представљен најбољи скуп линија за пружаоца услуге.



Слика 5.13. Најбоље аутобуске линије за превозиоца добијене BCO алгоритмом

5.4. Ублажавање поремећаја насталих недостатком аутобуса применом Оптимизације колонијом пчела

За решавање проблема ублажавања и отклањања поремећаја насталих недостацима одређеног броја планираних возила поред оптимизационог приступа развијена су и два хеуристичка алгоритма заснована на метахеуристици Оптимизација колонијом пчела. Први алгоритам омогућава прерасподелу расположивог броја аутобуса. Другим алгоритмом је омогућено и модификовање већ постојећих линија.

5.4.1. Генерисање почетног решења

За одређивање почетног решења коришћен је хеуристички алгоритам који се може представити следећим корацима:

Корак 1: Са линија искључити аутобусе који су отказали.

Корак 2: Одредити интервале слеђења на линијама.

Корак 3: Уколико не постоји линија која има интервал слеђења већи од максимално дозвољеног завршити са алгоритмом, у супротном прећи на Корак 4.

Корак 4: Одабрати две линије, прву, која има највећу вредност разлике интервала слеђења и максималног интервала слеђења и другу на којој је дозвољено смањити број аутобуса, а која има најмању вредност разлике интервала слеђења и максималног интервала слеђења.

Корак 5: Један аутобус са друге линије доделити првој линији. Вратити се на Корак 2.

5.4.2. Одређивање квалитета решења пчела

Након формирања решења потребно је извршити одређивање његовог квалитета. Да би се одредио квалитет решења прво је неопходно пропустити путнике кроз мрежу. За расподелу путника на мрежи у дисертацији је коришћен приступ описан у поглављу (5.3.5), при чему се *while* петља извршава само једном. Након распоређивања путника, на појединим линијама може да дође до прекорачења предвиђеног броја места за путнике. Већи број путника на линији од предвиђеног представља пад квалитета услуге која се пружа путницима. Претпоставимо да је на делу линије дужине 0,4 km у возилима присутно 10 путника више у односу на предвиђени број места на линији. У оваквој ситуацији остварује се „непожељни транспортни рад“ који је једнак 0,4 km x 10 путника = 4 pkm. Предложеним моделом тежи се минимизирању укупног „непожељног транспортног рада“.

Након пропуштања свих путника кроз мрежу, од почетне до крајње станице, могу се одредити карактеристике линија. Свако решење је окарактерисано са три критеријумске функције: број неопслужених путника (F_1), укупно прекорачење предвиђеног транспортног рада (F_2) и укупно време путовања свих путника на мрежи (F_3).

Сваки путник који од почетне до крајње станице не може да стигне без преседања, или са највише једним преседањем сматра се неопслуженим. За одређивање прекорачења предвиђеног транспортног рада на линији Z_l је коришћен следећи израз:

$$Z_l = \sum_{i=1}^{n_l-1} q_{l,(i,i+1)}\, d_{(i,i+1)} + q_{l,(i+1,i)}\, d_{(i+1,i)} \tag{5.40}$$

где су (Слика 5.14):

n_l – број станица на линији l

$q_{l,(i,i+1)}$ [put] – прекорачење предвиђеног броја путника на делу линије l, од станице i до станице i+1

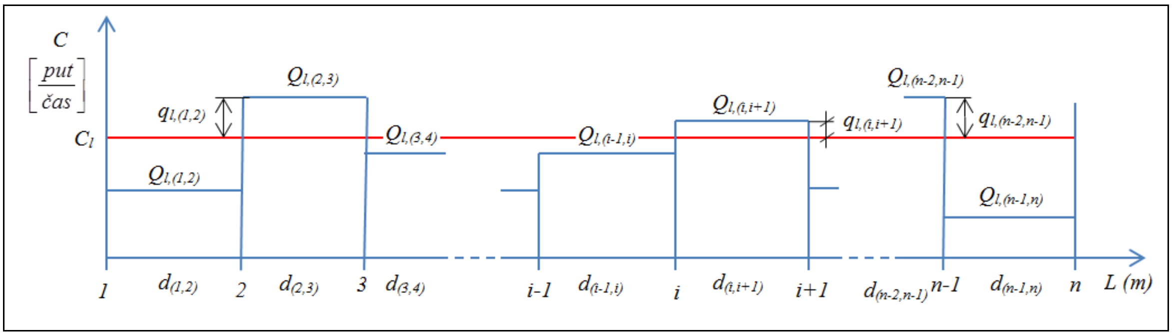
$d_{(i,i+1)}$ – растојање између станице i и станице i+1

Прекорачење броја путника на линији l, између станица i и i+1 се одређује као:

$$q_{l,(i,i+1)} = \begin{cases} Q_{l,(i,i+1)} - C_l, & Q_{l,(i,i+1)} > C_l \\ 0 & , \text{ у супротном} \end{cases} \tag{5.41}$$

где је:

C_l – предвиђени број места у јединици времена на линији l



Слика 5.14. Пример протока путника на једној линији

Начин израчунавања укупног времена путовања свих путника на мрежи је исти као и код проблема пројектовања линија (5.37), односно, укупно време путовања чине: укупно време које путници проведу у возилу, укупно време чекања свих путника и укупан износ пенала за преседање путника.

За одређивање једне вредности као мере квалитета решења се користи следећи израз:

$$T_i = w_1 \cdot \frac{F_{1i}}{F_{1\max}} + w_2 \cdot \frac{F_{2i}}{F_{2\max}} + w_3 \cdot \frac{F_{3i}}{F_{3\max}} \quad (5.42)$$

где су:

T_i – квалитет решења пчеле i,

F_{ji} – вредност критеријумске функције F_j пчеле i,

F_{jmax} – максимална вредност критеријумске функције F_j узимајући у обзир решења генерисана од стране свих пчела $F_{j\max} = \max\limits_{i=\overline{1,B}}\{F_{ji}\}$

w_j- значај j-те критеријумске функције

Из израза 5.40 може се уочити да веће вредности T_i одговарају лошијим решењима, док мање одговарају бољим. Вредност квалитета решења пчела се користи при лету уназад за доношење одлуке о лојалности и за доношење одлука о праћењу пчела.

За проверу тренутног најбољег решења користе се вредности функција циља решења (F_1, F_2, F_3). Провера најбољег решења се врши лексикографским приступом.

5.4.3. Модификовање решења пчела

У зависности од тога да ли је дозвољено модификовање постојећих линија у моделу се користи један или два типа пчела. Када је дозвољено само вршити прерасподелу расположивих аутобуса на већ постојеће линије у моделу се користи само први тип пчела. Два типа пчела се користе када је дозвољено и модификовање постојеће мреже линија.

Први тип пчела врши модификацију мреже тако што одабере две линије (L_1 и L_2) и један аутобус искључи са линије L_1 и додели линији L_2. Као прву линију (L_1) пчела бира линију l из скупа линија на којима је могуће искључити јадан аутобус (Ω), са вероватноћом:

$$p_l = \frac{U_l}{\sum_{k \in \Omega} U_k} \tag{5.43}$$

где је:

$$U_l = \begin{cases} S_l & , \quad \forall\, S_k > 0,\, k \in \Omega \\ S_l + \min\limits_{k \in \Omega}\{S_k\} & + \xi, \quad \text{у супротном} \end{cases} \tag{5.44}$$

при чему је:

$$S_l = \sum_{i=1}^{n_l - 1} \left[\left(C_l - Q_{l,(i,i+1)} \right) + \left(C_l - Q_{l,(i+1,i)} \right) \right] \tag{5.45}$$

а ξ константа која има малу вредност.

На овај начин, изразом (5.43), се фаворизује избор линија које имају већу неискоришћеност понуђеног транспортног рада. Супротно томе, ако је Ω скуп линија којима је дозвољено додати аутобус, тада је вероватноћа избора линије l као друге линије (L_2) једнака:

$$p_l = \frac{\frac{1}{U_l}}{\sum_{k \in \Omega} \frac{1}{U_k}} \tag{5.46}$$

Изразом (5.46) се фаворизује избор једне од линија које имају велико искоришћење понуђеног транспортног рада. Додавањем аутобуса линији са великим искоришћењем смањује се могућност да дође до прекорачења предвиђеног броја путника на тој линији.

Други тип пчела омогућава модификовање постојећих линија. У моделу је претпостављено да је при вршењу модификација допуштено само скраћивање већ постојећих линија. Током једног модификовања решења пчела врши промену на само једној од својих линија. Уколико пчела већ поседује одређени број линија које су скраћене у односу на почетне, она са вероватноћом 0,5 доноси одлуку да ли ће неку од већ скраћених линија продужи или ће се определити за скраћивање једне од линија код које је то дозвољено. Продужавање линије подразумева да ће линији бити додата станица која је у оригиналној линији била у њеном саставу, а која је суседна терминалу тренутне линије. Скраћивање линије подразумева да ће прва или последња станица бити искључена из састава линије.

Уколико пчела донесе одлуку да продужи неку од линија, она треба да изабере једну од већ модификованих. Вероватноћа избора линије *l* која ће да буде продужена за једну станицу је једнака:

$$p_l = \frac{n_l}{\sum_{k \in \Omega} n_k} \tag{5.47}$$

где су:

n_k – број станица које повезује линија k

Ω – скуп линија које су модификоване у односу на почетне

Уколико само једном терминалу може да се дода суседна станица, станицу треба додати линији. На тај начин додата станица постаје терминал. Ако код оба терминала могу да се додају станице, на случајан начин треба одредити која ће бити додата линији.

Ако пчела донесе одлуку да изврши скраћивање једне од линија, избор линије (l) се врши са вероватноћом:

$$p_l = \frac{i_l}{\sum_{k \in \Omega} i_k} \tag{5.48}$$

где је:

i_k – интервал слеђења возила на линији k

Ω – скуп линија које је допуштено скратити

Коришћењем вероватноће дефинисане изразом (5.46) фаворизује се скраћивање линија које имају дужи интервал слеђења. Скраћивањем линије смањује се време обрта возила на линији чиме се и интервал слеђења смањује.

Након избора линије доноси се одлука који се терминал искључује. У дисертацији је коришћен приступ да се то ради на случајан начин, односно да је вероватноћа искључивања сваког терминала једнака 0,5.

5.4.4. Резултати тестирања

Тестирања предложеног алгоритма су извршена на примерима прилагођеним тест примеру града Ривера у Уругвају (Mauttone и Urquhart, 2009). За одређивање почетних линија и броја аутобуса на линијама коришћен је BCO алгоритам описан у поглављу 5.3. За број места у аутобусу и коефицијент попуњености су узете исте вредности као и у *Mandl*-овим примерима (40 и 1,25). Претпостављено је да је расположиви број аутобуса једнак 140. Генерисана су три скупа линија. Први скуп има укупно 13 линија, други 15 и трећи 20. У Табели 5.13 је приказан скуп од 13 линија. У последњој колони је дат број аутобуса који треба да ради на свакој од линија. Може се уочити да је укупан потребан број аутобуса у овом случају једнак 139.

Табела 5.13. Скуп од 13 линија добијен BCO метахеуристиком

Број линије	Линија	Време обрта [min]	Број аутобуса
1	2 3 5 4 7 9 22 23 28 29 31 32 33 66 70 67 68 64 63 59 57	87,702	30
2	15 16 20 19 21 25 33 66 67 68 64 60 57 56 54 53 52 51 50 49	88,588	17
3	77 76 75 72 69 64 60 57 56 54 53 47 48 46	59,502	1
4	0 1 2 3 4 7 9 22 23 28 29 31 61 38 37 36 35	82,754	8
5	15 14 12 16 20 19 21 26 23 28 29 31 61 38 58 56 54 53 52 51 50 49	88,597	21
6	43 42 41 40 39 38 61 31 32 33 66 67 65 68 70 73 74 72	85,034	8
7	83 82 77 76 75 72 69 64 60 57 56 54 53 47 46 48	77,751	2
8	11 12 16 20 19 21 25 33 66 67 68 69 72 71 82 83 78	74,502	6
9	83 78 77 76 75 72 69 64 60 57 56 54 53 47 48 46 45 44	86,834	2
10	12 14 16 20 18 17 21 25 33 66 67 65 62 63 59 58 56 57 60 79 80 81	80,649	14
11	44 45 46 47 53 54 56 57 60 64 68 67 66 33 25 27 26 23 22 34 35	88,015	12
12	0 1 6 8 13 17 21 25 33 66 70 73 74 75 76 77	89,151	8
13	12 16 15 13 17 24 26 27 30 32 65 68 69 72 71 81 79 80 55	89,751	10

У Табели 3.14 је приказано 15 линија које су добијене BCO алгоритмом. Потребан број аутобуса у овом случају је једнак 137.

Табела 5.14. Скуп од 15 линија добијен BCO метахеуристиком

Број линије	Линија	Време обрта [min]	Број аутобуса
1	14 15 13 9 22 23 28 29 31 61 38 58 56 54 53 47 46 45	73,966	2
2	45 46 48 47 53 54 56 58 38 61 31 29 28 23 26 21 19 20 16 14 12	86,603	2
3	71 72 69 68 67 66 33 25 21 17 13 8 6 1 0	86,308	10
4	0 1 2 3 4 7 9 22 23 28 29 31 61 38 37 36 35	82,754	8
5	11 12 16 20 19 21 26 23 28 29 31 61 38 58 56 54 53 52 51 50 49	89,382	16
6	43 42 41 40 39 38 61 31 32 33 66 70 67 65 68 69 79 81	83,215	17

7	46 48 47 53 54 56 58 38 61 31 29 28 23 26	59,003	1
8	24 17 18 20 19 21 25 33 66 67 68 64 60 57 56 54 53 52 51 50 49	87,351	18
9	83 78 77 76 75 74 73 68 65 32 33 66 70 67	55,615	3
10	7 9 22 24 26 21 25 33 66 67 65 62 63 59 58 56 54 53 52 51 50	88,606	17
11	44 45 46 47 53 54 56 58 38 37 36 35 34	61,034	8
12	72 74 73 69 68 70 66 67 32 31 29 28 23 22 9 7 4 5 3 2	89,898	23
13	8 15 14 16 20 18 13 17 24 26 27 30 32 65 68 69 72 81	69,425	10
14	55 54 56 57 60 64 68 67 66 33	33,212	1
15	67 65 62 31 29 28 23 22 9 13 15 14 16	47,317	1

Скуп од 20 линија је приказан у Табели 5.15. Укупан потребан број аутобуса у овом случају је 140.

Табела 5.15. Скуп од 20 линија добијен BCO алгоритмом

Број линије	Линија	Време обрта [min]	Број аутобуса
1	6 3 5 4 7 9 22 23 28 29 31 32 33 66 70 68 69 72 71	83,289	12
2	70 66 33 25 21 17 13 8 6 3	45,452	1
3	0 1 6 8 13 17 21 25 33 66 67 68 69 72 71 82	89,095	13
4	0 1 2 3 4 7 9 22 23 28 29 31 61 38 37 36 35 34	86,843	9
5	80 81 79 64 68 70 66 33 32 31 29 28 23 22 9 7 4 5 3 6	86,022	14
6	43 42 41 40 39 38 61 31 32 33 66 67 65 68 69 81 71 76	89,538	17
7	32 31 61 38 58 56 54 53 52 51 50 49	54,563	16
8	8 11 12 16 20 19 21 26 27 30 32 65 67	49,966	4
9	18 20 19 21 25 33 66 67 68 64 60 57 56 54 53 52 51 50 49	81,609	13
10	55 54 56 57 59 63 62 65 67 66 33 25 21 19 20 18 17	58,652	9
11	73 70 66 67 68 64 60 57 56 54 53 47 48	52,763	1
12	40 39 38 58 56 54	35,834	1
13	17 24 26 27 30 32 65 68 69 72 71 81 79	51,988	4
14	43 42 41 40 39 38 61 31 29 28 23 22 9 7 4 3	77,289	2
15	34 35 36 37 29 31 65 68 73 74 75 76 77 78 83 82 71 81	85,615	5
16	32 30 27 26 21 19 20 16 12 11 8	44,215	4
17	50 51 52 53 54 56 57 60 64 69 72 75	54,738	1
18	70 66 67 68 64 60 57 56 54 53 47 48	45,886	1
19	15 14 16 20 19 21 25 33 66 67 68 64 60 57 56 54 53 47 46 45 44	87,342	12
20	45 46 48 47 53 54 56 57 60 64 68 67 66 70	58,218	1

На сваком од три скупа линија вршена су тестирања предложеног алгоритма у случајевима када је 5, 10 и 15 аутобуса ван употребе. За параметре BCO метахеуристике су коришћене следеће вредности: B = 20, IT = 100, NP = 5; NC = random (1,3). Сваки пример је решаван три пута, а најбољи резултати су представљени у Табели 5.16. У табели су дате вредности добијене за почетно решење које је нађено описаним хеуристичким алгоритмом и вредности добијене као крајње решење применом BCO метахеуристике.

Из добијених резултата се може уочити да нема неопслужених путника, али у свим случајевима постоји одређени „непожељни транспортни рад“ који је реализован при чему је дошло до прекорачења предвиђеног броја места у аутобусу. Такође, може се уочити да је BCO метахеуристика значајно смањила број „непожељних путничких километара“. Смањења времена путовања добијена BCO метахеуристиком су незнатна у односу на почетна решења. Детаљнији подаци о распореду аутобуса по линијама у тестираним случајевима су дати у Прилогу 2 у Табелама од (П.1) до (П.3).

Табела 5.16. Резултати тестирања

Број линија	Број аутобуса ван употребе	Почетно решење			BCO решење		
		Број неопслу-жених путника	„Непоже-љни транспо-ртни рад“	Просечно време [min]	Број неопслу-жених путника	„Непоже-љни транспо-ртни рад“	Просечно време [min]
13	5	0	79,065	19,838	0	26,236	19,806
	10	0	174,654	20,022	0	53,436	19,886
	15	0	620,786	20,316	0	110,715	20,116
15	5	0	56,596	19,749	0	0,282	19,761
	10	0	199,83	19,898	0	35,825	19,887
	15	0	274,93	20,052	0	115,133	20,08
20	5	0	102,431	19,839	0	31,955	19,752
	10	0	238,463	19,932	0	67,041	19,913
	15	0	489,255	20,152	0	160,403	20,104

На истим истанцама је извршено и тестирање у случају када је дозвољено модификовање линија. При тестирању је претпостављено да линија може да садржи све или део станица почетне линије. Такође, претпостављено је да линија, без обзира да ли ће да буде модификована или не, мора да задовољи ограничење максималног интервала слеђења на линији. За параметре BCO метахеуристике узете су следеће вредности: B = 20 (10 пчела сваког типа), IT = 300, NP = 5 и NC = random (1, 3).

У Табели 5.17 су приказани резултати добијени у случају када је допуштено модификовање линија. Може се уочити да у тестираним примерима нема неопслужених путника, као и да не долази до погоршања комфора у аутобусима због присуства већег броја путника него што је то планирано. Из пете колоне табеле се може уочити да је у свим примерима дошло до модификовања једне или више линија. Највећи број модификованих линија је у случају када мрежа има 13 линија, а 15 аутобуса је ван употребе. Такође, може се уочити да када мрежа има више линија, мања је потреба за модификовањем у случају отказа мањег броја аутобуса. Детаљни подати о линијама и броју аутобуса дати су у прилогу 3 у Табелама од (П.4) до (П.12).

Табела 5.17. Резултати тестирања када је дозвољено модификовање линија

Број линија	Број аутобуса ван употребе	Број неопслужених путника	„Непожељни транспортни рад“	Број промењених линија	Просечно време путовања [min]
13	5	0	0	1	19,771
	10	0	0	3	19,792
	15	0	0	8	20,277
15	5	0	0	1	19,716
	10	0	0	2	19,919
	15	0	0	5	20,037
20	5	0	0	1	19,664
	10	0	0	3	19,577
	15	0	0	4	19,736

6. УБЛАЖАВАЊЕ ПОРЕМЕЋАЈА У ДИСТРИБУЦИЈИ РОБЕ

Поремећаји у реализацији планиране дистрибуције робе могу бити проузроковани различитим неочекиваним догађајима као што су: откази возила, загушења у саобраћају, промена захтева корисника, промене локација испоруке, и др. Неочекивани догађаји могу да доведу до повећања трошкова реализације дистрибуције, немогућности реализације, незадовољства корисника, незадовољства запослених који би добили додатни посао, итд. Из наведених разлога у литератури су развијани различити приступи како би се последице нежељених догађаја свеле на најмању могућу меру.

Li и остали (2009a, b) су разматрали проблем рутирања који настаје због отказа једног или више возила. С обзиром да је разматрани проблем NP тежак аутори су предложили хеуристику убацивања засновану на Лагранжевим релаксацијама. Предложени алгоритам је тестиран на генерисаним инстанцама, а резултати су показали његову ефикасност. Проблем који такође настаје при отказу возила у току дистрибуције је разматран и у раду чији су аутори Mu и остали (2011). Ови аутори су за његово решавање предложили алгоритме базиране на Табу претраживању. Wang и остали (2011, 2012) су разматрали проблем који настаје при дистрибуцији робе а проузрокован је појавом већег броја нежељених догађаја, као што су: појава нових захтева, промена захтева корисника, промена допустивих интервала опслуге, промена локације корисника и др. Аутори су за разматрани проблем предложили математичку формулацију, као и хеуристички алгоритам. Hu и Sun (2012) су за решавање проблема у дистрибуцији који настају поремећајима предложили моделирање засновано на знању (енг. *knowledge-based modeling*). Након тога Hu и остали (2013) су предложили приступ који укључује процедуре за отклањање поремећаја, алгоритме локалног претраживања и објектно-орјентисаног моделирања. У раду чији су аутори Mu и Eglese (2013) аутори су разматрали проблем који настаје уколико касни испорука робе дистрибутивном складишту што даље проузрокује кашњења у испоруци корисницима. За решавање проблема аутори су предложили два алгоритма заснована на Табу претраживању. Резултати тестирања су показали да алгоритми могу значајно да смање трошкове дистрибуције.

Компаније које се баве дистрибуцијом робе углавном имају познат број корисника које опслужују. Због своје одганизације рада компаније често унапред дефинишу руте кретања возила на основу искуства о карактеристикама захтева које имају њихови корисници. Уколико нема неких драстичних промена у виду потражње корисника план рутирања остаје допустив.

Временом, ако се руте покажу као добре, компаније их се често придржавају у дужем временском периоду (3, 6 или више месеци). Овакав начин организације дистрибуције робе и рутирања возила има одређене предности као што су: планирање ангажовања возног парка, планирање смена и броја возача, планирање трошкова, возачи временом добро упознају руте које реализују и др. Главни недостатак оваквог начина рутирања возила су нешто већи трошкови реализација рута, у случају када је могуће наћи бољи скуп рута.

Повећање захтева неких корисника може довести до случаја да руте није увек могуће реализовати. Тада је потребно одредити нови скуп рута којим би се процес дистрибуције робе организовао на најбољи могући начин и при томе минимизирали негативни ефекти. У раду чији су аутори Spliet и остали (2014) предложена је математичка формулација и двофазни алгоритам за решавање проблема. Предложена математичка формулација је базирана на двоиндексној математичкој формулацији за капацитивни проблем рутирања возила (*Capacity Vehicle Routing Problem* - CVRP).

У овој дисертацији је предложена математичка формулација за решавање проблема ублажавања последица поремећаја у реализацији планираних рута при чему се узимају у обзир и временски интервали у којима је опслуга корисника дозвољена.

Нека је $G = (V, A)$ оријентисани граф где је V скуп чворова $V = \{0, 1, \ldots, n, n+1\}$, а A скуп грана $(i, j) \in A$. Чворовима 0 и $n+1$ је представљен депо возила, а чворовима *1,2, ..., n* су представљени корисници. Нека скуп N чине корисници који треба да буду опслужени, $N = \{1, 2, ..., n\}$. Узимајући у обзир претходни период током кога је вршена опслуга корисника, компаније могу да процене количине робе које ће корисници потраживати. Нека је q_i процењена количина робе коју ће да потражује корисник i. Такође из искуства у претходном периоду компаније знају у ком временском интервалу је дозвољена опслуга сваког од клијената и колико опслуга временски траје. Решавање овог проблема рутирања возила се своди на проблем рутирања возила са времским интервалима (*Vehicle Routing Problem with Time Windows* - VRPTW).

Рутирање возила са временским интервалима представља проблем где је потребно одредити скуп рута возила на начин да се минимизира њихова укупна дужина, при чему ограничења носивости возила и временских интервала, када је опслуга чворова (корисника) дозвољена, морају бити задовољена. Све руте почињу и завршавају се у депоу, а сва возила имају исту носивост. За решавање овог проблема предложене су математичке формулације које гарантују добијање оптималног решења. Међутим, с обзиром да је доказано да је овај проблем NP тежак (Savelsbergh, 1985) многи аутори су предложили приступе засноване на хеуристичким и метахеуристичким алгоритмима.

За решавање проблема рутирања возила са временским интервалима у литератури су предложени како оптимизациони тако и хеуристички приступи. Нека је за решавање проблема коришћена следећа математичка формулација мешовитог целобројног програмирања (Desrochers и остали, 1988; Cordeau и остали, 2007):

Минимизирати:

$$F = \sum_{k \in K} \sum_{(i,j) \in A} c_{ij}\, x_{ij}^{k} \tag{6.1}$$

при ограничењима:

$$\sum_{k \in K} \sum_{j \in \delta^{+}(i)} x_{ij}^{k} = 1 \quad \forall\, i \in N \tag{6.2}$$

$$\sum_{j\in\delta^+(0)} x_{0j}^k = 1 \quad \forall\, k \in K \tag{6.3}$$

$$\sum_{i\in\delta^-(j)} x_{ij}^k - \sum_{i\in\delta^+(j)} x_{ji}^k = 0 \quad \forall\, k \in K,\ j \in N \tag{6.4}$$

$$\sum_{i\in\delta^-(n+1)} x_{i,n+1}^k = 1 \quad \forall\, k \in K \tag{6.5}$$

$$w_j^k \ge w_i^k + s_i + t_{ij} - M\left(1 - x_{ij}^k\right) \quad \forall\, k \in K, (i, j) \in A \tag{6.6}$$

$$a_i \le w_i^k \le b_i \quad \forall\, k \in K, i \in V \tag{6.7}$$

$$\sum_{i\in N} q_i \sum_{j\in\delta^+(i)} x_{ij}^k \le Q \quad \forall\, k \in K \tag{6.8}$$

$$x_{ij}^k \in \{0,1\} \quad \forall\, k \in K, (i, j) \in A \tag{6.9}$$

где су:

$$x_{ij}^k = \begin{cases} 1, & \text{ако возило } k \text{ након опслуге чвора } i \text{ иде у чвор } j \\ 0, & \text{у супротном} \end{cases}$$

w_i^k - променљива која означава време почетка опслуге чвора i од стране возила k

Q – носивост возила

c_{ij} – трошак кретања возила од чвора i до чвора j

a_i – најраније време почетка опслуге чвора i

b_i – најкасније време почетка опслуге чвора i

s_i – време трајања опслуге чвора i

q_i – потражња чвора i

K – скуп возила

$\delta^+(i) = \{j : (i, j) \in A\}$

$\delta^-(j) = \{i : (i, j) \in A\}$

Критеријумска функција (6.1) минимизира укупне трошкове реализације рута. Ограничење (6.2) гарантује да ће сви корисници бити опслужени. Ограничења (6.3) и (6.5) гарантују да ће свако возило да напусти, односно, да се врати у депо. Ограничење (6.4). захтева да возило напусти локацију корисника.

Ограничења (6.6) и (6.7) гарантују да ће се опслуга корисника обавити у дефинисаним временским интервалима. Ограничење (6.8) гарантује да носивост возила неће бити нарушена. Ограничење (6.9) дефинише променљиву као бинарну. Решавањем проблема дефинисаног математичком формулацијом (6.1)-(6.9) добија се оптималан скуп рута возила. Због одређених погодности компаније често користе исте руте сваког дана. Проблем у овом приступу се јавља када корисници испоставе такве захтеве да једна или више рута постају недопустиве због носивости возила. У таквим случајевима јавља се следећи проблем: одредити нови скуп рута којим би се минимизирао број корисника који неће добити опслугу, минимизирати број корисника који неће бити опслужени уобичајеном рутом возила и минимизирати трошкове кретања возила, тј. трошак реализација рута. Такође, при решавању овог проблема потребно је водити рачуна и о значају корисника.

С обзиром да посматрани проблем има више циљева може се дефинисати као проблем вишекритеријумске оптимизације. Нека су уобичајене руте возила добијене решавањем VRPTW проблема дефинисаног математичком формулацијом (6.1)-(6.9). У новој математичкој формулацији x_{ij}^k више није променљива него узима вредности које су добијене решавањем формулације (6.1)-(6.9).

Нека су променљиве:

$$\xi_i = \begin{cases} 1, & \text{ако чвор } i \text{ неће бити опслужен} \\ 0, & \text{у супротном} \end{cases}$$

$$z_{ik} = \begin{cases} 1, & \text{ако чвор } i \text{ неће бити опслужен у рути } k \text{ која је његова уобичајена рута} \\ 0, & \text{у супротном} \end{cases}$$

$$y_{ij}^k = \begin{cases} 1, & \text{ако возило } k \text{ након опслуге чвора } i \text{ иде у чвор } j \\ 0, & \text{у супротном} \end{cases}$$

w_i^k - време почетка опслуге чвора i од стране возила k

Улазне вредности:

w_i – значај опслуге корисника i

u_{ik} – значај опслуге корисника у рути која се уобичајено користи за његову опслугу

Остале ознаке су исте као у формулацији (6.1) - (6.9).

Математичка формулација посматраног проблема, базирана на троиндексној формулацији за VRPTW, гласи:

Минимизирати:

$$F_1 = \sum_{i \in N} w_i \, \xi_i \tag{6.10}$$

Минимизирати:

$$F_2 = \sum_{k \in K} \sum_{i \in N} u_{ik} \, z_{ik} \tag{6.11}$$

Минимизирати:

$$F_3 = \sum_{k \in K} \sum_{(i,j) \in A} c_{ij} \, y_{ij}^k \tag{6.12}$$

при ограничењима:

$$\sum_{k \in K} \sum_{j \in \delta^+(i)} y_{ij}^k + \xi_i = 1 \quad \forall\, i \in N \tag{6.13}$$

$$\sum_{j \in \delta^+(0)} y_{0j}^k = 1 \quad \forall\, k \in K \tag{6.14}$$

$$\sum_{i \in \delta^-(j)} y_{ij}^k - \sum_{i \in \delta^+(j)} y_{ji}^k = 0 \quad \forall\, k \in K,\ j \in N \tag{6.15}$$

$$\sum_{i \in \delta^-(n+1)} y_{i,n+1}^k = 1 \quad \forall\, k \in K \tag{6.16}$$

$$w_j^k \geq w_i^k + s_i + t_{ij} - M\left(1 - y_{ij}^k\right) \quad \forall\, k \in K, (i, j) \in A \tag{6.17}$$

$$a_i \leq w_i^k \leq b_i \quad \forall\, k \in K, i \in V \tag{6.18}$$

$$\sum_{i \in N} q_i \sum_{j \in \delta^+(i)} y_{ij}^k \leq Q \quad \forall\, k \in K \tag{6.19}$$

$$\sum_{i \in \delta^{-}(j)} \left(x_{ij}^{k} - y_{ij}^{k} \right) \leq z_{jk} + \xi_j \quad \forall\, k \in K,\, j \in N \tag{6.20}$$

$$\xi_i \in \{0,1\} \quad \forall\, i \in N \tag{6.21}$$

$$y_{ij}^{k} \in \{0,1\} \quad \forall\, k \in K, (i,j) \in A \tag{6.22}$$

$$z_{ik} = \{0,1\} \quad \forall\, k \in K, i \in N \tag{6.23}$$

Критеријумска функција (6.10) коју тежимо да минимизирамо представља број корисника који неће добити опслугу пондерисан значајем корисника. Друга критеријумска функција (6.11), коју такође желимо да минимизирамо, репрезентује пондерисани број корисника који неће бити опслужени у оригиналним рутама. Критеријумска функција (6.12) чијој се минимизацији такође тежи представља укупну дужину свих рута.

Ограничењем (6.13) се гарантује да уколико корисник неће бити опслужен променљива ξ ће имати вредност 1. Ограничења (6.14) и (6.16) гарантују да ће свако возило да изађе из депоа и да се врати у депо. Да возило неће остати у неком од чворова који представљају кориснике је гарантовано ограничењем (6.15). Ограничења (6.17) и (6.18) гарантују да ће временски интервали за опслуживање корисника бити задовољени. Носивост возила неће бити прекорачена због ограничења (6.19). Уколико се врши опслуга корисника $i \in N$ али не у оригиналној рути $k \in K$ због ограничења (6.20) променљива z_{jk} ће имати вредност 1. Ограничењима (6.21), (6.22) и (6.23) се дефинишу променљиве као бинарне.

За тестирања математичке формулације (6.10) – (6.23) коришћена је Лексикографска метода вишекритеријумске оптимизације. Редослед критеријумских функција који је коришћен за тестирање је F_1, F_2 и F_3. Сва тестирања су вршена на Соломоновом бенчмарк примеру C101 за проблем рутирања возила са временским интервалима. Тестирањем су обухваћени случајави када је број чворова (корисника): 25 и 50. Почетна рута добијена је решавањем VRPTW проблема дефинисаног математичком формулацијом (6.1) – (6.9) CPLEX софтвером.

С обзиром да су у случају проблема са 25 чворова довољна 3 возила, вршена су тестирања за величину возног парка од 3 и 4 возила. У Табели 6.1 су приказани резултати добијени за случајеве када сви чворови потражују количину робе која је 10 %, 20 %, 30 %, 40 %, 50 %, 60 %, 70 %, 80 %, 90 % и 100 % већа од уобичајене.

Табела 6.1. Резултати примера са 25 чворова

Повећана количина потражње	C101					
	Број возила: 3			Број возила: 4		
	F1	F2	F3	F1	F2	F3
10 %	0	1,12	248,999	0	1,12	248,999
20 %	0	1,15	239,334	0	1,13	251,921
30 %	1,02	1,15	239,268	0	2,16	271,287
40 %	1,3	2,22	257,763	0	3,27	272,844
50 %	2,34	1,12	243,465	0	3,29	273,3
60 %	3,6	1,09	220,365	0	4,31	265,003
70 %	4,78	0	185,81	1,04	3,37	271,478
80 %	4,78	0	185,81	1,04	3,37	271,478
90 %	6,04	2,21	192,016	2,3	5,55	284,409
100 %	6,04	2,21	192,016	2,3	5,55	284,409

За реализацију рута у примеру са 50 чворова потребан број возила је 5. У тестирањима је проверено каква би се решања добила да је величина возног парка 6 и 7 возила. У Табели 6.2 су дати резултати тестирања.

Табела 6.2. Резултати примера са 50 чворова

Повећана количина потражње	C101					
	Број возила: 6			Број возила: 7		
	F_1	F_2	F_3	F_1	F_2	F_3
10 %	0	2,43	480,792	0	2,43	480,792
20 %	0	3,68	469,916	0	3,66	488,661
30 %	0	6,01	496,304	0	6	505,097
40 %	1,04	7,23	563,661	0	7,16	555,998
50 %	2,62	7,5	492,682	0	8,59	527,251
60 %	4,92	8,99	610,375	1,04	11,16	552,463
70 %	7,54	6,63	516,52	3,46	9,04	561,135
80 %	7,54	6,63	516,52	3,46	9,04	561,135
90 %	10,34	9,16	613,716	5,78	12,18	584,086
100 %	10,34	9,16	613,716	5,78	12,18	584,086

С обзиром да је проблем рутирања возила који настаје појавом поремећаја тежак проблем комбинаторне оптимизације, разматрана је могућност примене метахеуристике Оптимизација колонијом пчела за његово решавање. Поред тога, вршена је провера ефикасности BCO метахеуристике у решавању проблема рутирања возила са временским интервалима. Ово је рађено из два разлога:

1. У одређеним околностима за проблем који је настао након поремећаја (ако ново решење драстично одступа од иницијалног) диспечер може да донесе одлуку да одреди потпуно ново решење. Тада се проблем своди на решавање проблема рутирања возила са временским интервалима.
2. Проблем рутирања возила са временским интервалима је далеко више разматран у литератури од проблема рутирања возила који настаје када се поремећај догоди. С обзиром да је имплементација BCO метахеуристике доста слична за оба проблема, на известан начин може се претпоставити да уколико BCO метахеуристика успешно решава проблем рутирања са временским интервалима, да може успешно да решава и проблеме рутирања који настају појавом поремећаја. Тестирања алгоритма за рутирање возила са временским интервалима је лакше извршити с обзиром да у литератури постоје веома добро познати решени примери овог проблема.

6.1. Решавање проблема рутирања возила са временским интервалима применом Оптимизације колонијом пчела

Проблем рутирања са временским интервалима је веома сложен проблем комбинаторне оптимизације. Добијање оптималног решења наведеног проблема је условљено димензијама проблема. С друге стране, примена VRPTW у реалном животу веома често захтева добијање решења у прихватљивом времену рада рачунара. У досадашњој литератури VRPTW великих димензија је решаван применом бројних хеуристика, метахеуристика и хибридних приступа (комбинација хеуристичких и егзактних техника).

За решавање проблема рутирања возила са временским интервалима у дисертацији је коришћена верзија Оптимизације колонијом пчела са побољшањем решења. Алгоритам за решавање разматраног проблема може се представити следећим псеудо кодом:

```
процедура BCOvrptw (улаз B, IT, NP, NC, излаз S)
S ← формирати иницијално решење (скуп рута возила).
for j = 1 to IT do
    for i = 1 to B do
        пчела i ← S
    for k = 1 to NP do
        for r = 1 to NC do
            for i = 1 to B do
                Направити једну модификацију решења пчеле i.
                Евалуирати решење пчеле i.
        for i = 1 to B do
            побољшати решење пчеле i применом алгоритма за локално
            претраживање
            Евалуирати решење пчеле i.
        if најбоље решење генерисано од стране пчела bolje од решења S then
            S ← усвојити као најбоље решење пчела.
    Нормализовати вредности квалитета решења свих пчела.
    for i = 1 to B do
        Доношење одлуке да ли пчела i остаје лојална свом решењу или не.
    for i = 1 to B do
        if пчела i није лојална then
            Изабрати једну од лојалних пчела коју ће пчела i следити у
            следећем лету унапред.
```

6.1.1. Генерисање почетног решења

У раду је коришћена једноставна хеуристика убацивања за генерисање почетног решења. Хеуристика убацивања представља модификацију хеуристике предложене од стране Bakera и Schaffera (1986). У раду Baker и Schaffer (1986) предложена хеуристика формира руте секвенцијалним приступом (након што се прва рута направи почиње се са прављењем друге) при чему се убацивање чворова врши у два корака. У првом кораку се одређује чвор који ће бити убачен, а у другом његово место у рути. У овом раду хеуристика убацивања формира руте симултаним приступом (у исто време се формира већи број рута), при чему се у сваком кораку одређује чвор који ће да буде укључен, рута и место у рути где ће чвор да буде убачен.

Први чвор који ће бити убачен у руту се одређује на случајан начин. Након тога, сви остали чворови се убацују према тзв. „трошку“ убацивања. Овај „трошак“ убацивања чвора *u* између чворова *i* и *j* рачуна се као:

$$C_{iuj} = d_{iu} + d_{uj} - d_{ij} \tag{6.24}$$

где је d_{ij} најкраће растојање између чворова *i* и j.

Чвор *u* са најмањим „трошком“, C_{iuj}, се укључује између чворова, *i* и *j*. Уколико ниједан од нераспоређених чворова не може бити укључен у неку од постојећих рута, отвара се нова. Први чвор који ће бити додељен новој рути одређује се на случајан начин. Процес убацивања чворова се наставља све док сви чворови не буду укључени у неку руту.

6.1.2. Модификовање решења

Пчела модификује своје тренутно решење у следећа три корака:

Корак 1: Од укупног броја пчела бира *k* рута.

Корак 2: Из сваке одабране руте искључити све или одређени број чворова и прогласити их за нераспоређене.

Корак 3: Укључити нераспоређене чворове у руте применом претходно описане хеуристике убацивања.

Свака пчела одређује број *k* (број рута које ће бити модификоване или укинуте) на случајан начин на интервалу од 1 до 5 или до укупног броја рута уколико је он мањи од 5. Такође, уколико је број корисника који се опслужују у рути мањи или једнак 5 та рута се брише. У супротном, број чворова који се искључују из руте се одређује на случајан начин на интервалу од 1 до 5. У раду је дефинисано да се увек искључују суседни чворови, при чему се први чвор одређује на случајан начин.

У трећем кораку пчела укључује нераспоређене чворове у руте. При укључивању чворова коришћена је описана хеуристика убацивања при чему се „трошак“ рачуна на основу типа пчела.

Baker и Schaffer (1986) су представили три различите хеуристике убацивања у свом раду. Разлике између њих се огледају у начину на који се рачуна “трошак” убацивања. Следећи идеју ових аутора о постојању три типа хеуристика убацивања, у дисертацији се користе три типа вештачких пчела, при чему се оне разликују само по начину на који рачунају „трошак“.

Пчеле првог типа рачунају “трошак” на следећи начин:

$$C_{iuj}^{1} = d_{iu} + d_{uj} - w\, d_{ij} \tag{6.25}$$

где је w параметар. У раду је његова вредност одређивана при сваком убацивању чвора као случајан број на интервалу од 0 до 1.

Други тип пчела “трошак“ убацивања рачуна као:

$$C_{iuj}^{2} = w_1\, C_{iuj}^{1} + w_2\, \left(t(j/u) - t(j)\right) \tag{6.26}$$

где су:

w_1 и w_2 тежински коефицијенти $(w_1 + w_2 = 1)$. У раду су коришћене вредности $w_1 = 0{,}1$ и $w_2 = 0{,}9$

$t(j)$ – време доласка у чвор j пре убацивања чвора u у руту између чворова i и j

$t(j/u)$ – време доласка у чвор j након убацивања чвора u у руту између чворова i и j

Вештачке пчеле трећег типа рачунају "трошак" убацивања на следећи начин:

$$C_{iuj}^{3} = w_1\, C_{iuj}^{1} + w_2\, \left(t(j/u) - t(j)\right) + w3(f(u) - t(u)) \tag{6.27}$$

где су:

$f(u)$ – временски тренутак када почиње опслуга чвора u,

w_1, w_2 и w_3 су тежински коефицијенти $(w_1 + w_2 + w_3 = 1)$. У дисертацији су коришћене вредности $w_1 = 0,1$, $w_2 = 0,45$ и $w_3 = 0,45$.

6.1.3. Лет уназад

При повратку у кошницу вршено је побољшавање решења сваке пчеле. У литератури за проблеме трговачког путника и рутирања возила се углавном користе 2-Opt, 3-Opt и Or-ov алгоритам. За побољшавање решења коришћен је алгоритам локалног претраживања који може бити представљен следећим псеудо-кодом:

процедура алгоритамЛокалногПретраживања ()

1: **boolean** *побољшање← true*

2: ***while(****побољшање****)***

3: Q ← квалитет решења

4: ***for each*** *n из скупа корисника N*

5: r ← пронаћи руту у коју је чвор n укључен

6: уклонити чвор n из руте r

7: ***if*** *чвор може бити убачен у неку од постојећих рута* ***then***

8: убацити чвор n у руту и на место где је то могуће тако да су трошкови минимални

9: ***else***

10: убацити чвор n у нову руту

11: Израчунати квалитет решења.

12: ***if*** *квалитет решења* ***бољи*** *од Q* ***then***

13: побољшање ← true

14: ***else***

15: побољшање ← false

Основна идеја алгоритма локалног претраживања је да покуша да за сваки чвор пронађе боље место (руту и место у рути) уколико је то могуће. Алгоритам се зауставља када ни за једног корисника више није могуће пронаћи побољшање.

Након побољшања решења пчела врше се стандардни кораци BCO алгоритма при лету уназад: нормализација квалитета решења генерисаних од стране пчела, доношење одлуке о лојалности и доношење одлука о слеђењу. Ови кораци се реализују како је то описано у поглављу 3.

6.1.4. Резултати тестирања

Тестирање предложене BCO метахеуристике је извршено на Solomon-овим бенчмарк примерима. Постоји укупно 56 примера подељених у 6 група. За решавање свих примера су коришћени следећи параметри BCO метахеуристике:

- Број итерација: IT = 100,
- Број летова: NP = 20,
- Број модификација у сваком лету унапред: NC = 1,
- Број пчела: B = 60 (по 20 пчела сваког типа).

Сви примери су решавани 5 пута и на основу њих су дати резултати у Табели 1. У првој колони Табеле 6.3 су дати називи примера. Друга колона садржи до сада најбоље познате резултате из литературе (Ribas и остали, 2011). Наредне две колоне представљају најбоље резултате остварене применом BCO, као и релативну грешку у односу на најбоље познато решење. Пета и шеста колона дају просечну вредност критеријумске функције остварене након 5 BCO реализација и одговарајући релативну грешку. Коначно, последње две колоне приказују најлошију вредност критеријумске функције остварене током 5 BCO реализација и одговарајућу релативну грешку у односу на до сада најбоље познато решење.

Из резултата датих у Табели 6.3 може се уочити да најбоља решења добијена BCO алгоритмом просечно одступају од најбољих познатих решења 0,24 %. Просечне вредности критеријумске функције након 5 реализација просечно одступају 0,62 %, док најлошија решења од најбољих познатих одступају 1,1 %.

Такође, из резултата се може уочити да су у пет од укупно 56 примера пронађена боља решења од до сада најбољих познатих.

Табела 6.3. Резултати добијени BCO алгоритмом

Пример	Најбоље познато	BCO најбоље	Релативно одступање (%)	BCO просечно	Релативно одступање (%)	BCO најлошије	Релативно одступање (%)
C101	828,94	828,9369	-0,0004	828,9369	-0,0004	828,9369	-0,0004
C102	828,94	828,9369	-0,0004	828,9369	-0,0004	828,9369	-0,0004
C103	828,06	828,0649	0,0006	828,0649	0,0006	828,0649	0,0006
C104	824,78	824,7767	-0,0004	824,7767	-0,0004	824,7767	-0,0004
C105	828,94	828,9369	-0,0004	828,9369	-0,0004	828,9369	-0,0004
C106	828,94	828,9369	-0,0004	828,9369	-0,0004	828,9369	-0,0004
C107	828,94	828,9369	-0,0004	828,9369	-0,0004	828,9369	-0,0004
C108	828,94	828,9369	-0,0004	828,9369	-0,0004	828,9369	-0,0004
C109	828,94	828,9369	-0,0004	828,9369	-0,0004	828,9369	-0,0004
R101	1642,88	1642,8769	-0,0002	1643,3543	0,0289	1644,0453	0,0709
R102	1472,62	1472,8149	0,0132	1473,5957	0,0663	1475,6934	0,2087
R103	1213,62	1213,6239	0,0003	1214,5836	0,0794	1218,4225	0.3957
R104	**982,3**	**976,6082**	**-0,5794**	**986,7193**	**0,4499**	**996,5559**	**1,4513**
R105	1360,78	1360,7832	0,0002	1364,7972	0,2952	1370,8182	0,7377
R106	1239,37	1239,3719	0,0002	1241,7165	0,1893	1246,2977	0,5590
R107	1075,21	1075,2455	0,0033	1079,1242	0,3640	1083,3961	0,7613
R108	**948,57**	**941,6963**	**-0,7246**	**951,3239**	**0,2903**	**959,9685**	**1,2017**
R109	1151,84	1151,8385	-0,0001	1153,7052	0,1619	1155,4029	0,3093
R110	1072,41	1072,4149	0,0005	1077,1882	0,4456	1088,6522	1,5146
R111	1053,5	1053,4958	-0,0004	1055,8391	0,2220	1061,6185	0,7706
R112	953,63	965,7995	1,2761	969,7337	1,6887	971,3308	1,8561
RC101	1623,58	1643,0432	1,1988	1644,6770	1,2994	1646,5322	1,4137
RC102	1461,23	1477,1971	1,0927	1478,6029	1,1889	1480,9656	1,3506
RC103	1261,67	1273,4214	0,9314	1289,0163	2,1675	1310,0120	3,8316
RC104	1135,48	1138,8686	0,2984	1145,9576	0,9227	1149,2229	1,2103
RC105	1518,58	1521,7196	0,2067	1527,3184	0,5754	1532,6538	0,9268
RC106	1376,99	1376,9937	0,0003	1386,8860	0,7187	1397,3562	1,4790
RC107	1212,83	1216,6472	0,3147	1227,9599	1,2475	1237,4219	2,0276
RC108	1117,53	1117,5265	-0,0003	1130,3566	1,1478	1140,7245	2,0755
C201	591,56	591,5566	-0,0006	591,5566	-0,0006	591,5566	-0,0006
C202	591,56	591,5566	-0,0006	591,5566	-0,0006	591,5566	-0,0006
C203	591,17	591,1734	0,0006	591,1734	0,0006	591,1734	0,0006
C204	590,6	590,5987	-0,0002	590,5987	-0,0002	590,5987	-0,0002
C205	588,88	588,8760	-0,0007	588,8760	-0,0007	588,8760	-0,0007
C206	588,49	588,4928	0,0005	588,4928	0,0005	588,4928	0,0005
C207	588,29	588,2863	-0,0006	588,2863	-0,0006	588,2863	-0,0006
C208	588,32	588,3238	0,0006	588,3238	0,0006	588,3238	0,0006
R201	1147,8	1168,1974	1,7771	1171,9827	2,1069	1173,4062	2,2309
R202	1034,35	1041,1000	0,6526	1042,3670	0,7751	1044,2675	0,9588
R203	874,87	892,9167	2,0628	895,9049	2,4043	900,8101	2,9650
R204	735,8	738,4060	0,3542	747,1529	1,5429	754,0843	2,4850

R205	954,16	954,1617	0,0002	961,6524	0,7852	972,7455	1,9478
R206	884,25	888,2982	0,4578	894,9971	1,2154	904,0753	2,2421
R207	797,99	797,9940	0,0005	809,6659	1,4632	832,9992	4,3872
R208	705,62	706,7401	0,1587	716,9443	1,6049	726,2913	2,9295
R209	**860,11**	**859,3904**	**-0,0837**	**863,9114**	**0,4420**	**870,3809**	**1,1941**
R210	910,98	913,0466	0,2269	920,3318	1,0266	923,2550	1,3474
R211	**755,82**	**754,3129**	**-0,1994**	**755,4893**	**-0,0438**	**755,9490**	**0,0171**
RC201	1265,9	1269,9435	0,3194	1284,2733	1,4514	1300,3312	2,7199
RC202	1095,64	1118,6575	2,1008	1122,5356	2,4548	1138,0478	3,8706
RC203	**926,89**	**926,8193**	**-0,0076**	**931,9844**	**0,5496**	**942,0147**	**1,6318**
RC204	786,38	788,6626	0,2903	794,9777	1,0933	803,3372	2,1564
RC205	1157,55	1157,5504	0,0000	1157,6419	0,0079	1157,6648	0,0099
RC206	1056,21	1063,5284	0,6929	1069,2809	1,2375	1073,3772	1,6254
RC207	966,08	969,9732	0,4030	972,6617	0,6813	980,9952	1,5439
RC208	779,84	782,6962	0,3663	784,3650	0,5802	789,8047	1,2778
Просечно			0,2429		0,6237		1,1016

6.2. Решавање проблема одређивања нових рута при повећаним количинама потражње применом Оптимизације колонијом пчела

Псеудо код метахеуристике Оптимизација колонијом пчела за решавање проблема рутирања возила при захтевима већим од планираних може се представити на следећи начин:

***процедура BCOпоремећајVRPTW(улаз** B, IT, NP, NC, **излаз** S)*

Одредити почетно решење.

Одредити вредности критеријумских функција почетног решења.

S ← сачувати почетно решење као тренутно најбоље.

for** j = 1 **to** IT **do

  for** i = 1 **to** B **do

    пчела i ← доделити тренутно најбоље пронађено решење (S).

  for** k = 1 **to** NP **do

    for** r = 1 **to** NC **do

      for** i = 1 **to** B **do

        Направити једну модификацију решења пчеле i.

        Одредити функције циља решења пчеле i.

        Извршити побољшање решења пчеле i

13: ***if*** *најбоље решење генерисано од стране пчела* ***better*** *од тренутно најбољег познатог решења S* ***then***

14: *S ← доделити најбоље решење пчела.*

15: *Одредити квалитет решења свих пчела.*

16: *Извршити нормализацију квалитета решења генерисаних од стране свих пчела.*

17: ***for*** *i = 1* ***to*** *B* ***do***

18: *Донети одлуку да ли пчела i остаје лојална свом решењу.*

19: ***for*** *i = 1* ***to*** *B* ***do***

20: ***if*** *пчела i није лојална свом решењу* ***then***

21: *Одабрати једну од лојалних пчела коју ће пчела i да прати.*

Као што се може уочити псеудо код је врло сличан општем коду за Оптимизацију колонијом пчела. Разлике међу њима се могу уочити на неколико места:

- С обзиром да у овом проблему постоје три критеријумске функције у кораку 2 треба да буду израчунате све критеријумске функције за почетно решење.
- На почетку сваке итерације пчелама се увек додељује тренутно најбоље откривено решење (кораци 5 и 6).
- У кораку 12 решење пчела се побољшава алгоритмом за локално претраживање.
- За све пчеле се рачуна квалитет решења на основу вредности функција циља.

6.2.1. Вредновање квалитета решења и провера најбољег

При доношењу одлуке да ли остају лојалне свом решењу пчеле користе само једну вредност која представља меру квалитета њиховог решења. С обзиром да је свако решење већ окарактерисано са три критеријумске функције њихове вредности су коришћене како би се квалитет решења изразио једном вредношћу. Тако на пример, квалитет решења пчеле *i* се може одредити на следећи начин:

$$T_i = w_1 \cdot \frac{F_{1i}}{F_{1\max}} + w_2 \cdot \frac{F_{2i}}{F_{2\max}} + w_3 \cdot \frac{F_{3i}}{F_{3\max}} \qquad (6.28)$$

где су:

T_i– квалитет решења пчеле i,

F_{ji} – вредност критеријумске функције F_j пчеле i,
F_{jmax} – максимална вредност критеријумске функције F_j узимајући у обзир решења генерисана од стране свих пчела $F_{j\max} = \max\limits_{i=1,B}\{F_{ji}\}$

w_j- значај j-те критеријумске функције

Вредност T_i се користи само у лету уназад. Провера да ли је пронађено ново најбоље решење се врши поређењем критеријумских функција у складу са принципом Лексикографске оптимизације. Усваја се да је једно решење боље од другог ако има бољу вредност према једној функцији циља и бољу или једнаку вредност према осталим значајнијим функцијама циља. Критеријумска функција F_1 је третирана као најважнија, па затим F_2 и на крају F_3.

6.2.2. Резултати тестирања

Примери решавани у поглављу 6 применом лексикографске методе у овом делу су решавани Оптимизацијом колонијом пчела. За параметре BCO метахеуристике су коришћене следеће вредности:

- Број итерација: IT = 500
- Број летова: NP = 20
- Број промена решења по једном лету унапред: NC = 1
- Број пчела: B = 60 (20 пчела сваког типа)

Резултати за примере са 25 чворова су дати у Табели 6.1. Поредећи резултате добијене BCO метахеуристиком (Табела 6.4) и резултате добијене CPLEX-ом може се уочити да је за све примере BCO метахеуристика пронашла решења истог квалитета као и CPLEX.

Табела 6.4. Резултати добијени BCO метахеуристиком за примере са 25 чворова

Повећана потражња корисника	C101					
	Број возила: 3			Број возила: 4		
	F1	F2	F3	F1	F2	F3
10 %	0	1,12	248,9991	0	1,12	248,9991
20 %	0	1,15	239,3338	0	1,13	251,9211
30 %	1,02	1,15	239,2678	0	2,16	271,2866
40 %	1,3	2,22	257,7629	0	3,27	272,844
50 %	2,34	1,12	243,4646	0	3,29	273,2997
60 %	3,6	1,09	220,3653	0	4,31	265,0025
70 %	4,78	0	185,8104	1,04	3,37	271,4784
80 %	4,78	0	185,8104	1,04	3,37	271,4784
90 %	6,04	2,21	192,0161	2,3	5,55	284,4088
100 %	6,04	2,21	192,0161	2,3	5,55	284,4088

У Табели 6.5 су дати резултати за истанце са 50 чворова. Поредећи резултате са вредностима решења датим у Табели 6.2 може се уочити да је BCO метахеуристика пронашла веома добра решења.

Табела 6.5. Резултати добијени BCO метахеуристиком за примере са 50 чворова

Повећана потражња корисника	C101					
	Број возила: 6			Број возила: 7		
	F1	F2	F3	F1	F2	F3
10 %	0	2,43	480,7925	0	2,43	480,7925
20 %	0	3,68	469,9155	0	3,66	488,6613
30 %	0	6,01	496,304	0	6	505,0974
40 %	1,04	7,23	563,661	0	7,16	555,9984
50 %	2,62	7,5	492,6819	0	8,59	527,2508
60 %	4,92	10,12	515,1527	1,04	11,17	622,149
70 %	7,54	6,63	516,5198	3,46	10,29	567,6405
80 %	7,54	6,69	518,3186	3,46	9,04	561,1351
90 %	10,34	9,24	645,506	5,78	12,21	568,7064
100 %	10,34	9,25	648,6695	5,78	12,32	587,6187

7. ЗАКЉУЧНА РАЗМАТРАЊА И ПРАВЦИ БУДУЋЕГ РАДА

Поремећаји у реализацији планираних активности у саобраћају и транспорту могу да доведу до великог броја неопслужених путника, повећаних времена чекања, повећаних времена путовања, незадовољства корисника, и значајног повећања транспортних трошкова.

У дисертацији су разматрана три проблема која се јављају услед поремећаја на аеродромима, у системима масовног превоза путника у градовима и у системима дистрибуције робе. С обзиром да су у дисертацији разматрани проблеми комбинаторне оптимизације за њихово решавање је коришћена метахеуристика Оптимизација колонијом пчела. Такође, осим хеуристичког приступа, за решавање проблема мањих димензија предложени су модели засновани на математичком програмирању.

Провера квалитета решења која је могуће постићи Оптимизацијом колонијом пчела извршена је на примерима налажења екстремних вредности функција. Извршена су два експеримента, у првом је решавано 50 функција, а у другом 30. У поређењу са другим хеуристичким и метахеуристичким приступима који су коришћени у литератури, показало се да је BCO метахеуристика конкурентна другим приступима.

За решавање проблема паркирања и опслуге авиона који настају на паркинг позицијама на аеродромима, развијена је математичка формулација мешовитог целобројног програмирања. Математичка формулација садржи три критеријумске функције. Тестирања на примеру терминала А на аеродрому у Даласу, коришћењем Лексикографске методе, показала су да је чак и решавање проблема као више мањих веома захтевно са аспекта времена рада рачунара, као и потребног меморијског простора. Из тог разлога предложен је и хеуристички алгоритам заснован на метахеуристици Оптимизација колонијом пчела. Резултати добијени тестирањем хеуристичког алгоритма указују да се у кратким временским интервалима могу пронаћи квалитетна решења.

У случају недостатка возила у јавним системима превоза путника може доћи до значајног пада квалитета услуге која се нуди путницима. Такође, поремећаји већих димензија могу проузроковати да неки путници остану неопслужени. У циљу ублажавања и отклањања поремећаја у дисертацији су предложени модели којима се врши прерасподела расположивих аутобуса на линије, као и модификовање линија. Предложени модели су засновани на математичком програмирању и метахеуристици Оптимизација колонијом пчела. У дисертацији је разматран и приступ заснован на одређивању новог скупа линија, тј. решаван је проблем пројектовања мреже линија. Сва тестирања су вршена на примерима из литературе и добијени резултати указују да предложени хеуристички алгоритми могу да нађу веома квалитетна решења. За проблем прерасподеле расположивих аутобуса на већ постојеће линије, као и за проблем прерасподеле аутобуса и модификовања линија вршена су и тестирања предложених математичких формулација. За тестирања је коришћена Лексикографска метода вишекритеријумске оптимизације. Тестирањима је показано да је овај приступ могуће користити за проблеме малих димензија.

За решавање проблема одређивања нових рута због повећане потражње корисника у дистрибутивним системима у дисертацији је предложена математичка формулација, као и хеуристички алгоритам. Поред тога, предложен је и алгоритам заснован на метахеуристици Оптимизација колонијом пчела за решавање проблема рутирања возила са временским интервалима. Тестирања предложеног хеуристичког алгоритма за решавање овог проблема су вршена на *Solomon*-овим тест примерима. Поређење добијених резултата са најбољим познатим резултатима за те примере је показало да развијени алгоритам може да пронађе изузетно добра решења.

Даља истраживања се могу усмерити у неколико праваца као што су:

- анализа осетљивости појединих параметара у предложеним алгоритмима, као и њихово прецизније одређивање. Анализа осетљивости и прецизније одређивање параметара могу, у извесној мери, да допринесу повећању квалитета решења која се добијају алгоритмима.
- тестирање развијених модела на другим реалним примерима. Већина тестирања у дисертацији је рађена на познатим примерима из литературе.

Додатна тестирања на реалним примерима омогућила би бољи увид у могућности примене развијених модела. Ово би могло да представља значајан подстрек за развој корисничког софтвера који би могао да нађе значајну примену у пракси.

- проширења развијених модела узимајући у обзир додатна ограничења (на пример: могућности опслуживања авиона на појединим паркинг поизицијама, хетерогеност возног парка при решавању проблема рутирања возила, стохастичност у појави захтева за опслугом у системима јавног превоза и др.).
- модификовање развијених модела у циљу решавања других, сродних, проблема. Метахеуристика Оптимизација колонијом пчела представља једну од техника за решавање проблема комбинаторне оптимизације. С обзиром да су проблеми разматрани у дисертацији комбинаторне природе, изнете идеје могу бити у извесној мери искоришћене при решавању других проблема. У оквиру будућих истраживања, значајно би било анализирати, моделирати и решавати следеће проблеме (сличне проблемима разматраним у дисертацији): проблеми који настају поремећајима у реализацији планираних редова летења, проблеми који настају поремећајем планираног рада посада авиона, проблеми који се јављају у јавном превозу због отказа појединих возила из техничких разлога, проблеми који се јављају у дистрибуцији робе при отказу возила, и др.
- развој модела за одређивање решења која су робустна на поремећаје. У дисертацији је пажња посвећена решавању проблема који настају након што се поремећај десио. Интересантно би било проверити у којој мери поремећаји разматрани у дисертацији могу да се спрече генерисањем робусних решења у фази планирања.

Литература

1. Baaj, M.H., Mahmassani, H. 1991. An AI-based approach for transit route system planning and design, Journal of Advanced Transportation, 25, 2, 187-210.

2. Baaj, M.H., Mahmassani, H.S. 1995. Hybrid Route Generation Heuristic Algorithm for the Design of Transit Networks, Transportation Research Part C, 3, 1, 31-50.

3. Babić, O., Kalić, M., Pavković, G., Dožić, S., Čangalović, M. 2010. Heuristic approach to the airline schedule disturbances problem, Transportation Planning and Technology, 33, 3, 257-280.

4. Babić, O., Teodorović, D., Tošić, V., 1984. Aircraft stand assignment to minimize walking, Journal of Transportation Engineering, 110, 1, 56-66.

5. Bagloee, S.A., Ceder, A. 2011. Transit-network design methodology for actual-size road networks, Transportation Research Part B, 45, 10, 1787-1804.

6. Baker, E.K., Schaffer, J.R., 1986. Solution improvement heuristics for the vehicle routing and scheduling problem with time window constraints, Americal Journal of Mathematical and Management Science, 6, 3-4, 261-300.

7. Barnett, A. 1974. On controlling randomness in transit operations, Transportation Science, 8, 2, 102-116.

8. Beni, G. 1988. The concept of cellular robotic system, In: Proceedings of the 1988 IEEE International Symposium on Intelligent Control, IEEE Computer Society Press, Los Alamitos, CA, 57–62.

9. Beni, G., Wang, J., 1989. Swarm intelligence, In: Proceedings of the Seventh Annual Meeting of the Robotics Society of Japan, RSJ Press, Tokyo, 425–428.

10. Beni, G., Hackwood, S. 1992. Stationary waves in cyclic swarms, In: Proceedings of the 1992 International Symposium on Intelligent Control, IEEE Computer Society Press, Los Alamitos, CA, 234–242.

11. Bielli, M., M Caramia, M., Carotenuto, P. 2002. Genetic Algorithms in Bus Network Optimization, Transportation Research Part C: Emerging Technologies, 10, 1, 19-34.

12. Bihr, R. 1980. A conceptual solution to the aircraft gate assignment problem using 0–1 linear programming, Computers & Industrial Engineering, 19, 1-4, 280–284.

13. Blum, J.J., Mathew, T.V. 2012. Implications of the computational complexity of transit route network redesign for metaheuristic optimisation systems, IET Intelligent Transport Systems, 6, 2, 124-131.

14. Bolat, A. 1999. Assigning arriving aircraft flights at an airport to available gates, Journal of the Operational Research Society, 50, 1, 23–34.

15. Bolat, A. 2000a. Procedures for providing robust gate assignments for arriving aircraft, European Journal of Operations Research, 120, 1, 63–80.

16. Bolat, A. 2000b. Models and a genetic algorithm for static aircraft gate assignment problem, Journal of the Operational Research Society, 52, 10, 1107–20.

17. Bonabeau, E., Dorigo, M., Theraulaz, G. 1999. Swarm Intelligence, Oxford University Press, Oxford.

18. Braaksma, J., Shortreed, J. 1971. Improving Airport Gate Usage with Critical Path Method, Transportation Engineering Journal, 97, 2, 187–203.

19. Brazilian Civil Aviation Agency. 2010. Anuário do Transporte Aéreo 2010 (1st ed.). http://www2.anac.gov.br/estatistica/anuarios.asp. Accessed 10 Apr 2010.

20. Brouer, B.D., Dirksen, J., Pisinger, D., Plum, C.E.M., Vaaben, B. 2013. The Vessel Schedule Recovery Problem (VSRP) - A MIP model for handling disruptions in liner shipping, European Journal of Operational Research, 224, 2, 362-374.

21. Byrne, B.F. 1975. Public Transportation Line Positions and headways for Minimum User and System Cost in a Radial Case, Transportation Research, 9, 2-3, 97-102.

22. Ceder, A, Israeli, Y. 1998. User and operator perspectives in transit network design, Transportation Research Record, 1623, 3-7.

23. Ceder, A., Wilson, H.M. 1986. Bus network Design, Transportation Research Part B, 20, 4, 331-334.

24. Chakroborty, P. 2003. Genetic algorithms for optimal urban transit network design, Computer-Aided Civil and Infrastructure Engineering, 18, 3, 184-200.

25. Chakroborty, P., Dwivedi, T. 2002. Optimal route network design for transit systems using genetic algorithms, Engineering Optimization 34, 1, 83–100.

26. Chang, Y.H., Yen, C.H,, Cheng, J.H. 1998. Decision Support for Bus Operations under Uncertainty: a Fuzzy Expert System Approach, Omega, 26, 3, 367-380.

27. Chapman, R.A., Michel J.F. 1978. Modelling the Tendency of Buses to Form Pairs, Transportation Science, 12, 2, 165-175.

28. Cheng, Y., 1997. A knowledge-based airport gate assignment system integrated with mathematical programming, Computers & Industrial Engineering, 32, 4, 837-852.

29. Clausen, J., Larsen, A., Larsen, J., Rezanova, N.J. 2010. Disruption Management in the Airline Industry – Concepts, Models and Methods, Computers & Operations Research, 37, 5, 809–821.

30. Coello, C.A., Van Veldhuizen, D.A., Lamont G.B. 2002. Evolutionary algorithms for solving multi-objective problems, Kluwer.

31. Cordeau, J.F., Laporte, G., Savelsbergh, M. W. P., and Vigo, D. 2007. Vehicle routing. In C. Barnhart & G. Laporte (Eds.),Transportation. Amsterdam: Elsevier.

32. Corman, F., D'Ariano, A., Pacciarelli, D., Pranzo, M. 2010. A tabu search algorithm for rerouting trains during rail operations, Transportation Research Part B: Methodological, 44, 1, 175-192.

33. Dagnino, G., Rocco, E. 2009. Introduction - coopetition strategy, in Dagnino, G., Rocco, E., (Editors), Coopetition Strategy, Routledge, New York, 1-21.

34. Davidović, T., Ramljak, D., Šelmić, M., Teodorović, D. 2011. Bee colony optimization for the p-center problem. Computers & Operations Research, 38, 10, 1367- 1376.

35. Davidović, T., Šelmić, M., Teodorović, D., Ramljak, D. 2012. Bee Colony Optimization for Scheduling Independent Tasks to Identical Processors. Journal of Heuristics, 18, 4, 549-569.

36. Davidović, T., Teodorović, D., Šelmić, M. 2014. Bee Colony Optimization Part I: The algorithm overview, Yugoslav Journal of Operations Research, accepted for publishing, DOI: 10.2298/YJOR131011017D

37. Denver International Airport. *Airport Layout Map*. 14 April 2013 http://flydenver.com/map_airport_layout

38. Derrible, S., Kennedy, C. 2011. Applications of Graph Theory and Network Science to Transit Network Design. Transport reviews, 31, 4, 495-519.

39. Desaulniers, G., Hickman, M.D. 2007. Public Transit. in Handbook in OR & MS, Vol. 14, , C. Barnhart and G. Laporte (Eds.), 69-127.

40. Desrochers, M., Lenstra, J.K., Savelsbergh, M.W.P., Soumis, F. 1988. Vehicle routing with time windows: Optimization and approximation. In: Golden, B.L., Assad, A.A. (Eds), Vehicle Routing: Methods and Studies. North-Holland, Amsterdam, 65-84.

41. Dimitrijević, B., Teodorović, D., Simić, V., Šelmić, M. 2011. A Bee Colony Optimization Approach to Solving the Anti-Covering Location Problem, Journal of Computing in Civil Engineering, 26, 6, 759-768.

42. Ding, H., Lim, A., Rodrigues, B., Zhu, Y. 2004a. New heuristics for the over-constrainted flignt to gate assignments, Journal of the Operational Research Society, 55, 760–768.

43. Ding, H., Lim A., Rodrigues, B., Zhu, Y. 2004b. The over-constrainted airport gate assignment problem, Computers and Operations Research, 32, 7, 1867–1880.

44. Dorndorf, U., Drexl, A., Nikulin, Y., Pesch, E. 2007. Flight gate scheduling: State-of-the-art and recent developments, Omega, 35, 3, 326-334.

45. Dorndorf, U., Jaehn, F., Pesch, E. 2008. Modelling robust flight-gate scheduling as a clique partitioning problem, Transportation Science, 42, 3, 292-301.

46. Drexl, A., Nikulin, Y., 2008. Multicriteria airport gate assignment and Pareto simulated annealing, IIE Transactions, 40, 4, 385-397.

47. Eggenberg, N., Salani, M., Bierlaire, M. 2010. Constraint-specific recovery network for solving airline recovery problems, Computers & Operations Research, 37, 1014-1026.

48. Fan, W., Machemehl, R.B. 2006. Optimal transit route network design problem with variable transit demand: Genetic algorithm approach, Journal of Transportation Engineering-ASCE, 132, 1, 40-51.

49. Fan, W., Machemehl, R.B. 2008. Tabu search strategies for the public transportation network optimizations with variable transit demand, Computer Aided Civil and Infrastructure Engineering, 23, 7, 502-520.

50. Fan, L., Mumford, C.L. 2010. A metaheuristic approach to the urban transit routing problem, Journal of Heuristics, 16, 3, 353-372.

51. Garud, R. 1994. Cooperative and competitive behaviors during the process of creative destruction, Research Policy, 23, 4, 385-394.

52. Gnyavali, D.R., Park, B-J. 2009. Co-opetition and technological innovation in small and medium-sized enterprises: a multilevel conceptual model, Journal of Small Business Management, 47, 3, 308-330.

53. Gnyavali, D.R., Park, B.J. 2011. Co-opetition between giants: Collaboration with competitors for technological innovation, Research Policy, 40, 5, 650- 663.

54. Gosling, G.D., 1990. Design of an expert system for aircraft gate assignment, Transportation Research - A, 24, 1, 59-69.

55. Gu, Y., Chung, C. 1999. Genetic algorithm approach to aircraft gate reassignment problem, ASCE Journal of Transportation Engineering, 125, 5, 384–389.

56. Guan, J. F., Yang, H., Wirasinghe, S. C. 2006. Simultaneous optimization of transit line configuration and passenger line assignment, Transportation Research Part B, 40, 10, 885-902.

57. Guihaire, V., Hao, J.K. 2008. Transit network design and scheduling: A global review, Transportation Research A, 42, 10, 1251-1273.

58. Harbison, J.R., Pekar ,Jr., P. 1998. Smart Alliances, Jossey-Bass, San Francisco.

59. Hassounah, M., Steuart G. 1993. Demand for aircraft gates, Transportation Research Record, 1423, 26–33.

60. Hedar, A., Fukushima, M. 2006. Evolution strategies learned with automatic termination criteria, In: Proceedings of SCIS-ISIS 2006, Tokyo, Japan.

61. Hu, X.B., Di Paolo, E. 2007. An efficient genetic algorithm with uniform crossover for the multi-objective airport gate assignment problem, Proceedings of the IEEE Congress on Evolutionary Computation, September 25-28, 2007 Singapore, Volumes 1-10, 55-62.

62. Hu, X., Sun, L. 2012. Knowledge-based modeling for disruption management in urban distribution, Expert System with Applications, 39, 1, 906-916.

63. Hu, X., Sun, L., Liu, L. 2013. A PAM approach to handling disruptions in real-time vehicle routing problems, Decision Support Systems, 54, 3, 1380-1393.

64. Ioannou, P., Chassiakos, A., Abadi, A., Chang, H., Jula, H., Lestas, M., Saggam, L., Thomas, D., Wang, Y., 2011. Reconfiguration strategies for mitigating the impacts of port disruptions. Final Report. METRANS Project 07-14.

65. Jarrah, A.I.Z., Yu, G., Krishnamurthy, N., Rakshit, A. 1993. A decision support framework for airline flight cancellations and delays, Transportation Science, 27, 3, 266–280.

66. Jespersen-Groth, J., Potthoff, D., Clausen, J., Huisman, D., Kroon, L.G., Maroti, G., Nyhave Nielsen, M. 2007. Disruption Management in Passenger Railway Transportation. Technical Report EI 2007-05, Erasmus University Rotterdam.

67. Karaboga, D., Bahriye, A. 2009. A comparative study of Artificial Bee Colony algorithm, Applied Mathematics and Computation, 214, 1, 108-132.

68. Kepaptsoglou, K., Karlaftis, M.G. 2009a. The bus bridging problem in metro operations: conceptual framework models and algorithms, Public Transport, 1, 4, 275-297.

69. Kepaptsoglou, K., Karlaftis, M. 2009b. Transit Route Network Design Problem: Review, Journal of Transportation Engineering-ASCE, 135, 8, 491-505.

70. Kidwai, F.A. 1998. Optimal design of bus transit network: a genetic algorithm based approach. Phd. dissertation, Indian Institute of Technology, Kanpur, India.

71. Kohl, N., Larsen, A., Larsen, J., Ross, A., Tiourine, S. 2007. Airline disruption management—perspectives, experiences and outlook, Journal of Air Transport Management, 13, 3, 149–162.

72. Kroon, L., Maroti, G., Neilsen, L. 2014. Rescheduling of Railways Rolling Stock with Dynamic Passenger Flows, Transportation Science, accepted for publishing

73. Lampkin, W., Saalmans, P.D. 1967. The Design of Routes Service Frequencies and Schedules for Municipal Bus Undertaking: A Case Study. Operational Research Quarterly, 18, 4, 375-397.

74. Lee, Y.J., Vuchic, V.R. 2005. Transit network design with variable demand, Journal of Transportation Engineering - ASCE, 131, 1, 1-10.

75. Li, J.Q., Borenstein, D., Mirchandani, P.B., 2007a. A decision support system for the single-depot vehicle rescheduling problem, Computers & Operations Research, 34, 4, 1008–1032.

76. Li, J.Q., Mirchandani, P.B., Borenstein, D., 2007b. The vehicle rescheduling problem: Model and algorithms, Networks 50, 3, 211–229.

77. Li, J.Q., Mirchandani, P.B., Borenstein, D. 2009a. A Lagrangian heuristic for the real-time vehicle rescheduling problem, Transportation Research Part E: Logistics and Transportation Review, 45, 3, 419–433.

78. Li, J.Q., Mirchandani, P.B., Borenstein, D. 2009b. Real-time vehicle rerouting problems with time windows, European Journal of Operational Research, 194, 3, 711–727.

79. Lim, A., Rodrigues, B., Zhu, Y. 2005. Airport gate scheduling with time windows, Artificial Intelligence Review, 24, 1, 5-31.

80. Liu, C.Y., Wei, D.X., 2006. Research on the airport gate assignment problem by using GA, Proceedings of 2006 Chinese Control and Decision Conference, 1077-1080, Tianjin, Peoples republic of China.

81. Lučić, P., Teodorović, D., 2001. Bee system: modeling combinatorial optimization transportation engineering problems by swarm intelligence. In: Preprints of the TRISTAN IV Triennial Symposium on Transportation Analysis, Sao Miguel, Azores Islands, Portugal, 441–445.

82. Lučić, P., Teodorović, D. 2002. Transportation modeling: An artificial life approach. In Proceedings of the 14th IEEE "International conference on tools with artificial intelligence", Washington, DC, 216–223.

83. Lučić, P., Teodorović, D. 2003a. Computing with bees: attacking complex transportation engineering problems, International Journal on Artificial Intelligence Tools, 12, 3, 375–394.

84. Lučić, P., Teodorović, D. 2003b. Vehicle routing problem with uncertain demand at nodes: The bee system and fuzzy logic approach. In J. L. Verdegay (Ed.), Fuzzy sets in optimization, 126, 67–82. Heidelberg, Berlin: Springer-Verlag

85. Magnanti, T.L., Wong, R.T. 1984. Network design and transportation planning: Models and algorithms, Transportation Science, 18, 1, 1–55.

86. Maharjan, B., Matis, T. 2011. An optimization model for gate reassignment in response to flight delays, Journal of Air Transportation Management, 17, 4, 256-261.

87. Mandl, C. E. 1979. Evaluation and Optimization of Urban Public Transportation Network, European Journal of Operational Research, 5, 6, 396-404.

88. Mangoubi, R., Mathaisel D. 1985. Optimizing gate assignments at airport terminals, Transportation Science, 19, 2, 173–188.

89. Marković, G., Teodorović, D., Aćimović-Raspopović, V. 2007. Routing and wavelength assignment in all-optical networks based on the bee colony optimization. AI Communications, 20, 4, 273–285.

90. Mauttone, A., Urquhart, M.E. 2009. A route set construction algorithm for the transit network design problem, Computers & Operations Research, 36, 8, 2440-2449.

91. Mu, Q., Fu, Z., Lysgaard, J., Eglese, R., 2011. Disruption management of the vehicle routing problem with vehicle breakdown, Journal of the Operational Research Society, 62, 742-749.

92. Mu, Q., Eglese, R.W. 2013. Disrupted capacitated vehicle routing problem with order release delay, Annals of Operations Research, 207, 1, 201-2016.

93. Newell, G. F. 1974. Control of Pairing of Vehicles on a Public Transportation Route, Two Vehicles, One Control Point, Transportation Science, 8, 3, 248-264.

94. Newell, G. 1979. Some Issues Relating to the Optimal Design of Bus Routes, Transportation Science, 13, 1, 20-35.

95. Nielsen, L.K., Kroon, L.G., Maroti, G. 2012. A rolling horizon approach for disruption management of railway rolling stock, European Journal of Operational Research, 220, 2, 496–509.

96. Nikolić, M., Teodorović, D. 2013a. Empirical study of the bee colony optimization (BCO) algorithm, Expert Systems with Applications, 40, 11, 4609–4620.

97. Nikolić, M., Teodorović D. 2013b. Transit network design by Bee Colony Optimization. Expert Systems with Applications, 40, 15, 5945-5955.

98. Nikolić, M., Teodorović, D., 2014. A simultaneous transit network design and frequency setting: Computing with bees, Expert Systems with Applications, 41, 16, 7200-7209.

99. Nikolić, M., Teodorović, D., Šelmić, M. 2013. Solving the Vehicle Routing Problem With Time Windows by Bee Colony Optimization Metaheuristic, In Proceedings of the 1st Logistics International Conference, Belgrade, 44-48.

100. Nikolić, M., Teodorović, D., Vukadinović, K., 2015. Disruption Management in Public Transit: The Bee Colony Optimization (BCO) approach, Transportation Planning and Technology, 38, 2, 162-180.

101. Osuna, E.E., Newell G.F. 1972. Control strategies for an idealized public transportation system, Transportation Science, 6, 1, 52-72.

102. Pacheco, J. , Alvarez, A., Casado, S., Gonzalez-Velarde, J.L. 2009. A tabu search approach to an urban transport problem in northern Spain, Computers & Operations Research, 36, 3, 967-979.

103. Pattnaik, S., Mohan, S., Tom, V. 1998. Urban Bus Transit Route Network Design Using Genetic Algorithm, Journal of Transportation Engineering, 124, 4, 368–375.

104. Pintea, CM, Pop, PC, Chira, C., et al., 2008. A Hybrid Ant-Based System for Gate Assignment Problem, 3rd International Workshop on Hybrid Artificial Intelligence Systems, SEP 24-26, 2008 Univ Burgos, Burgos, SPAIN, Source: HYBRID ARTIFICIAL INTELLIGENCE SYSTEMS Book Series: LECTURE NOTES IN ARTIFICIAL INTELLIGENCE, Vol: 5271, 273-280.

105. Savelsbergh, M.W.P. 1985. Local search in routing problems with time windows, Annals of Operations Research, 4, 285–305.

106. Schoebel, A. 2012. Line planning in public transportation: models and methods. OR SPECTRUM, 34, 3, 491-510.

107. Schumer, C.E., 2008. Flight Delays Cost Passengers, Airlines and the US Economy Billions, Joint Committee Majority Staff.

108. Šelmić M., Teodorović D., Vukadinović K., 2010. Locating inspection facilities in traffic networks: an artificial intelligence approach, Transportation Planning and Technology, 33, 6, 481-493.

109. Shih, M.C., Mahmassani, H.S. 1994. A Design Methodology for Bus Transit Networks with Coordinated Operations, Research Report SWUTC/94/60016-1. Center for Transportation Research, University of Texas at Austin.

110. Shih, M. C., Mahmassani, H. S., Baaj, M. H. 1997. Trip Assignment Model for Timed-Transfer Transit Systems, Transportation Research Record 1571, TRB, National Research Council, Washington, D. C., 24-30.

111. Silman, L.A., Barzily, Z., Passy, U. 1974. Planning the Route System for Urban Buses, Computers and Operations Research, 1, 2, 201-211.

112. Sinclair, K., Cordeau, J.F., Laporte, G. 2014. Improvements to a large neighborhood search heuristic for an integrated aircraft and passenger recovery problem, European Journal of Operational Research, 233, 1, 234-245.

113. Spliet, R., Gabor, A.F., Dekker, R. 2014. The vehicle rescheduling problem, Computers & Operations Research, 43, 129-136.

114. Srihari, K., Muthukrishnan, R. 1991. An expert system methodology for aircraft-gate assignment, Computers & Industrial Engineering, 21, 1-4, 101-105.

115. Su, Y.Y., Srihari, K. 1993. A knowledge based aircraft-gate assignment advisor, Computers and Industrial Engineering 25, 1-4, 123-126.

116. Szeto, W.V., Wu, Y. 2011. A simultaneous bus route design and frequency setting problem for Tin Shui Wai, Hong Kong, European Journal of Operational Research, 209, 2, 141-155.

117. Tang, C.H., 2011. A Gate Reassignment Model for the Taiwan Taoyuan Airport Under Temporary Gate Shortages and Stochastic Flight Delays, IEEE Transactions on Systems, Man and Cybernetics - Part A: Systems and Humans, 41, 4, 637-650.

118. Tang, C.H., Yan, S., Hou, Y.Z., 2010. A gate reassignment framework for real time flight delays, 4OR, 8, 3, 299–318.

119. Teodorović, D. 2003. Transport Modeling by Multi-Agent Systems: A Swarm Intelligence Approach, Transportation Planning and Technology, 26, 289–312.

120. Teodorović, D. 2008. Swarm Intelligence Systems for Transportation Engineering: Principles and Applications, Transportation Research C, 16, 6, 651-782.

121. Teodorović, D., Dell'Orco, M. 2005. Bee colony optimization - a cooperative learning approach to complex transportation problems. In: Advanced OR and AI Methods in Transportation. Proceedings of the 10th Meeting of the EURO Working Group on Transportation, Poznan, Poland, 51–60.

122. Teodorović, D., Dell'Orco, M. 2008. Mitigating traffic congestion: solving the ride-matching problem by bee colony optimization, Transportation Planning and Technology, 31, 2, 135–152.

123. Teodorović, D., Guberinić, S. 1984. Optimal dispatching strategy on an airline network after a schedule perturbation, European Journal of Operational Research, 15, 2, 178-182.

124. Teodorović, D., Rallis, T. 1988. A Model for Assigning Vehicles to Scheduled Routes when there is a Shortage of Vehicles, Transportation Planning and Technology, 12, 2, 135-150.

125. Teodorović, D., Šelmić, M. 2012. Računarska inteligencija u saobraćaju, Saobraćajni fakultet, Beograd.

126. Teodorović, D., Šelmić, M., Davidović, T. 2014. Bee Colony Optimization Part 2: The Application Survey, Yugoslav Journal Of Operations Research, accepted for publishing, DOI: 10.2298/YJOR131029020T

127. Teodorović, D. Šelmić, M., Mijatović-Teodorović, Lj. 2012. Combining case-based reasoning with Bee Colony Optimization for dose planning in well

differentiated thyroid cancer treatment, Expert Systems with Applications, 40, 6, 2147-2155.

128. Teodorović, D., Stojković, G. 1990. Model for Operational Daily Airline Scheduling, Transportation Planning and Technology, 14, 4, 273-286.

129. Teodorović, D., Stojković, G. 1995. A Model to Reduce Airline Schedule Disturbances, Journal of Transportation Engineering, 121, 4, 324-331.

130. Todorović, N. Petrović, S. 2013. Bee Colony Optimization for Nurse Rostering, IEEE Transactions on Systems Man & Cybernetics: Systems, 43, 2, 467 - 473.

131. Vanderstraeten, G., Bergeron, M. 1998. Automatic assignment of aircrfat to gates at a terminal, Computers & Industrial Engineering, 14, 1, 15-25.

132. Vuchic, V. R. 1969. Propagation of schedule disturbances in line-haul passenger transportation, Revue de L'UITP 28, 281-284.

133. Vukadinović, K., Teodorović, D., Pavković, G., Rosić, S. 1996. A neural network approach to mitigation of vehicle scheduling disturbances, Transportation Planning and Technology, 20,1, 93-102.

134. Вукадиновић, С., Поповић Ј. 2008. Математичка статистика, Саобраћајни факултет, Београд.

135. Yan, S., Chang, C. 1998. A network model for gate assignment, Journal of Advanced Transportation, 32, 2, 176–189.

136. Yan, S., Huo, C. 2001. Optimization of multiple objective gate assignments, Transportation Research A, 35, 5, 413–432.

137. Yan, S., Chen, C.Y., Tang, C.H. 2009. Airport gate reassignment following temporary airport closures, Transportmetrica, 5, 1, 25-41.

138. Yan, S., Young, H. 1996. A decision support framework for multi-fleet routing and multi-stop flight scheduling, Transportation Research A, 30, 5, 379–398.

139. Yu, G., Qi, X. 2004. Disruption Management: Framework, Models and Applications, World Scientific.

140. Wang, X., Ruan, J., Shang, H., Ma, C. 2011. A Combinatorial Disruption Recovery Model for Vehicle Routing Problem with Time Windows, Intelligent Decision Technologies, 10, 3-11.

141. Wang, X., Ruan, J., Shi, Y. 2012. A recovery model for combinatorial disruptions in logistics delivery: Considering the real-world participators, International Journal of Production Economics, 140, 1, 508-520.

142. Zegordi S.H., Jafari, N. 2010. Solving the airline recovery problem by using ant colony optimization. Internation Journal of Industrial Engineering & Production Research, 21, 3,121-128.

143. Zeng, A. Z., Durach, C. F., Fang, Y. 2012. Collaboration decisions on disruption recovery service in urban public tram systems, Transportation Research Part E: Logistics and Transportation Review, 48, 3, 578-590.

144. Zeng, Q., Hu, X., Wang, W., Fang, Y. 2011. Disruption management model and its algorithms for berth allocation problem in container terminals, International Journal of Innovative Computing Information and Control, 7, 5, 2768-2773.

145. Zhang, J.J., Chen, Q.S., Sun, G.H., Zhang, Q. 2007. Disruption scheduling of airport gate based on tabu search algorithm, Proceedings of the 26th Chinese Control Conference, Vol. 6, 84-88 Zhangjiajie, Peoples republic of China.

146. Zhao, F., Zeng, X. 2006. Simulated annealing-genetic algorithm for transit network optimization. Journal of Computing in Civil Engineering 20, 1, 57-68.

147. Zhao, F., Zeng, X. 2007. Optimization of transit route network, vehicle headways and timetables for large-scale transit networks, European Journal of Operational Research, 186, 2, 841-855.

ПРИЛОЗИ

Прилог 1 – Изворно – циљна матрица путовања (Mandl-ов пример)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0	400	200	60	80	150	75	75	30	160	30	25	35	0	0
2	400	0	50	120	20	180	90	90	15	130	20	10	10	5	0
3	200	50	0	40	60	180	90	90	15	45	20	10	10	5	0
4	60	120	40	0	50	100	50	50	15	240	40	25	10	5	0
5	80	20	60	50	0	50	25	25	10	120	20	15	5	0	0
6	150	180	180	100	50	0	100	100	30	880	60	15	15	10	0
7	75	90	90	50	25	100	0	50	15	440	35	10	10	5	0
8	75	90	90	50	25	100	50	0	15	440	35	10	10	5	0
9	30	15	15	15	10	30	15	15	0	140	20	5	0	0	0
10	160	130	45	240	120	880	440	440	140	0	600	250	500	200	0
11	30	20	20	40	20	60	35	35	20	600	0	75	95	15	0
12	25	10	10	25	15	15	10	10	5	250	75	0	70	0	0
13	35	10	10	10	5	15	10	10	0	500	95	70	0	45	0
14	0	5	5	5	0	10	5	5	0	200	15	0	45	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Прилог 2 – Нови распореди аутобуса по линијама (пример – Ривера (Уругвај))

Табела П.1. Распореди аутобуса када има 13 линија

	5 аутобуса ван употребе		10 аутобуса ван употребе		15 аутобуса ван употребе	
	Почетно	ВСО	Почетно	ВСО	Почетно	ВСО
$Л_1$	28	28	28	25	27	25
$Л_2$	17	16	16	17	16	17
$Л_3$	1	1	1	1	1	1
$Л_4$	7	8	7	8	6	8
$Л_5$	20	20	19	16	20	14
$Л_6$	8	8	8	8	7	8
$Л_7$	2	2	2	2	2	2
$Л_8$	6	6	6	6	5	6
$Л_9$	2	2	2	2	2	2
$Л_{10}$	14	14	13	14	13	13
$Л_{11}$	12	12	11	12	10	12
$Л_{12}$	7	7	6	8	6	6
$Л_{13}$	10	10	10	10	9	10

Табела П.2. Распореди аутобуса када има 15 линија

	5 аутобуса ван употребе		10 аутобуса ван употребе		15 аутобуса ван употребе	
	Почетно	ВСО	Почетно	ВСО	Почетно	ВСО
$Л_1$	2	2	2	2	2	2
$Л_2$	2	2	2	2	2	2
$Л_3$	9	7	8	7	7	7
$Л_4$	8	8	7	8	7	8
$Л_5$	15	16	15	16	15	15
$Л_6$	17	15	16	13	15	9
$Л_7$	1	1	1	1	1	1
$Л_8$	18	18	17	17	16	17
$Л_9$	3	3	3	3	3	3
$Л_{10}$	16	17	16	17	15	17
$Л_{11}$	7	8	7	8	7	7
$Л_{12}$	23	23	22	22	21	23
$Л_{13}$	9	10	9	9	9	9
$Л_{14}$	1	1	1	1	1	1
$Л_{15}$	1	1	1	1	1	1

Табела П.3. Распореди аутобуса када има 20 линија

	5 аутобуса ван употребе		10 аутобуса ван употребе		15 аутобуса ван употребе	
	Почетно	ВСО	Почетно	ВСО	Почетно	ВСО
$Л_1$	12	12	11	12	11	11
$Л_2$	1	1	1	1	1	1
$Л_3$	12	13	11	13	10	13
$Л_4$	9	9	9	9	9	9
$Л_5$	14	13	13	14	13	14
$Л_6$	17	16	16	9	16	9
$Л_7$	16	16	15	16	15	16
$Л_8$	4	4	4	4	3	4
$Л_9$	12	13	12	12	12	12
$Л_{10}$	9	8	9	9	8	8
$Л_{11}$	1	1	1	1	1	1
$Л_{12}$	1	1	1	1	1	1
$Л_{13}$	3	4	3	4	3	4
$Л_{14}$	2	2	2	2	2	2
$Л_{15}$	5	5	5	5	4	4
$Л_{16}$	3	3	3	3	3	3
$Л_{17}$	1	1	1	1	1	1
$Л_{18}$	1	1	1	1	1	1
$Л_{19}$	11	11	11	12	10	10
$Л_{20}$	1	1	1	1	1	1

Прилог 3 – Линије и бројеви аутобуса по линијама наком модификовања (пример – Ривера (Уругвај))

Табела П.4. Линије и распореди аутобуса када на мрежи има 13 линија и 5 аутобуса ван употребе

Број линије	Линија	Број аутобуса	Број аутобуса
1	**2 3 5 4 7 9 22 23 28 29 31 32 33 66 70 67 68 64**	30	**26**
2	15 16 20 19 21 25 33 66 67 68 64 60 57 56 54 53 52 51 50 49	17	17
3	77 76 75 72 69 64 60 57 56 54 53 47 48 46	1	1
4	0 1 2 3 4 7 9 22 23 28 29 31 61 38 37 36 35	8	8
5	15 14 12 16 20 19 21 26 23 28 29 31 61 38 58 56 54 53 52 51 50 49	21	**20**
6	43 42 41 40 39 38 61 31 32 33 66 67 65 68 70 73 74 72	8	8
7	83 82 77 76 75 72 69 64 60 57 56 54 53 47 46 48	2	2
8	11 12 16 20 19 21 25 33 66 67 68 69 72 71 82 83 78	6	6
9	83 78 77 76 75 72 69 64 60 57 56 54 53 47 48 46 45 44	2	2
10	12 14 16 20 18 17 21 25 33 66 67 65 62 63 59 58 56 57 60 79 80 81	14	14
11	44 45 46 47 53 54 56 57 60 64 68 67 66 33 25 27 26 23 22 34 35	12	12
12	0 1 6 8 13 17 21 25 33 66 70 73 74 75 76 77	8	8
13	12 16 15 13 17 24 26 27 30 32 65 68 69 72 71 81 79 80 55	10	10

Табела П.5. Линије и распореди аутобуса када на мрежи има 13 линија и 10 аутобуса ван употребе

Број линије	Линија	Број аутобуса	Број аутобуса
1	**5 4 7 9 22 23 28 29 31 32 33 66 70 67 68 64 63 59 57**	30	**25**
2	15 16 20 19 21 25 33 66 67 68 64 60 57 56 54 53 52 51 50 49	17	**16**
3	77 76 75 72 69 64 60 57 56 54 53 47 48 46	1	1
4	**0 1 2 3 4 7 9 22 23 28 29 31 61 38 37**	8	8
5	15 14 12 16 20 19 21 26 23 28 29 31 61 38 58 56 54 53 52 51 50 49	21	**20**
6	43 42 41 40 39 38 61 31 32 33 66 67 65 68 70 73 74 72	8	8
7	83 82 77 76 75 72 69 64 60 57 56 54 53 47 46 48	2	2
8	11 12 16 20 19 21 25 33 66 67 68 69 72 71 82 83 78	6	6
9	83 78 77 76 75 72 69 64 60 57 56 54 53 47 48 46 45 44	2	2
10	**12 14 16 20 18 17 21 25 33 66 67 65 62 63 59 58**	14	**11**
11	44 45 46 47 53 54 56 57 60 64 68 67 66 33 25 27 26 23 22 34 35	12	12
12	0 1 6 8 13 17 21 25 33 66 70 73 74 75 76 77	8	8
13	12 16 15 13 17 24 26 27 30 32 65 68 69 72 71 81 79 80 55	10	10

Табела П.6. Линије и распореди аутобуса када на мрежи има 13 линија и 15 аутобуса ван употребе

Број линије	Линија	Број аутобуса	Број аутобуса
1	**3 5 4 7 9 22 23 28 29 31 32 33 66 70 67 68 64 63 59**	30	**26**
2	**20 19 21 25 33 66 67 68 64 60 57 56 54 53 52 51 50 49**	17	**16**
3	**77 76 75 72 69 64 60 57 56 54**	1	1
4	**0 1 2 3 4 7 9 22 23 28 29 31 61 38 37**	8	8
5	**15 14 12 16 20 19 21 26 23 28 29 31 61 38 58 56 54 53 52 51 50**	21	**18**
6	43 42 41 40 39 38 61 31 32 33 66 67 65 68 70 73 74 72	8	8
7	83 82 77 76 75 72 69 64 60 57 56 54 53 47 46 48	2	2
8	11 12 16 20 19 21 25 33 66 67 68 69 72 71 82 83 78	6	6
9	**64 60 57 56 54 53 47 48 46 45**	2	**1**
10	12 14 16 20 18 17 21 25 33 66 67 65 62 63 59 58 56 57 60 79 80 81	14	**10**
11	44 45 46 47 53 54 56 57 60 64 68 67 66 33 25 27 26 23 22 34 35	12	12
12	**0 1 6 8 13 17 21 25 33 66 70 73 74**	8	**6**
13	**15 13 17 24 26 27 30 32 65 68 69 72 71 81 79 80 55**	10	10

Табела П.7. Линије и распореди аутобуса када на мрежи има 15 линија и 5 аутобуса ван употребе

Број линије	Линија	Број аутобуса	Број аутобуса
1	14 15 13 9 22 23 28 29 31 61 38 58 56 54 53 47 46 45	2	2
2	45 46 48 47 53 54 56 58 38 61 31 29 28 23 26 21 19 20 16 14 12	2	2
3	71 72 69 68 67 66 33 25 21 17 13 8 6 1 0	10	**7**
4	0 1 2 3 4 7 9 22 23 28 29 31 61 38 37 36 35	8	8
5	11 12 16 20 19 21 26 23 28 29 31 61 38 58 56 54 53 52 51 50 49	16	16
6	43 42 41 40 39 38 61 31 32 33 66 70 67 65 68 69 79 81	17	**15**
7	46 48 47 53 54 56 58 38 61 31 29 28 23 26	1	1
8	24 17 18 20 19 21 25 33 66 67 68 64 60 57 56 54 53 52 51 50 49	18	18
9	83 78 77 76 75 74 73 68 65 32 33 66 70 67	3	3
10	7 9 22 24 26 21 25 33 66 67 65 62 63 59 58 56 54 53 52 51 50	17	17
11	44 45 46 47 53 54 56 58 38 37 36 35 34	8	8
12	72 74 73 69 68 70 66 67 32 31 29 28 23 22 9 7 4 5 3 2	23	23
13	**15 14 16 20 18 13 17 24 26 27 30 32 65 68 69 72**	10	10
14	55 54 56 57 60 64 68 67 66 33	1	1
15	67 65 62 31 29 28 23 22 9 13 15 14 16	1	1

Табела П.8. Линије и распореди аутобуса када на мрежи има 15 линија и 10 аутобуса ван употребе

Број линије	Линија	Број аутобуса	Број аутобуса
1	14 15 13 9 22 23 28 29 31 61 38 58 56 54 53 47 46 45	2	2
2	45 46 48 47 53 54 56 58 38 61 31 29 28 23 26 21 19 20 16 14 12	2	2
3	71 72 69 68 67 66 33 25 21 17 13 8 6 1 0	10	**7**
4	0 1 2 3 4 7 9 22 23 28 29 31 61 38 37 36 35	8	8
5	**11 12 16 20 19 21 26 23 28 29 31 61 38 58 56 54 53 52 51**	16	**13**
6	43 42 41 40 39 38 61 31 32 33 66 70 67 65 68 69 79 81	17	**15**
7	46 48 47 53 54 56 58 38 61 31 29 28 23 26	1	1
8	24 17 18 20 19 21 25 33 66 67 68 64 60 57 56 54 53 52 51 50 49	18	**17**
9	83 78 77 76 75 74 73 68 65 32 33 66 70 67	3	3
10	7 9 22 24 26 21 25 33 66 67 65 62 63 59 58 56 54 53 52 51 50	17	17
11	44 45 46 47 53 54 56 58 38 37 36 35 34	8	8
12	**74 73 69 68 70 66 67 32 31 29 28 23 22 9 7 4 5 3 2**	23	**22**
13	8 15 14 16 20 18 13 17 24 26 27 30 32 65 68 69 72 81	10	10
14	55 54 56 57 60 64 68 67 66 33	1	1
15	67 65 62 31 29 28 23 22 9 13 15 14 16	1	1

Табела П.9. Линије и распореди аутобуса када на мрежи има 15 линија и 15 аутобуса ван употребе

Број линије	Линија	Број аутобуса	Број аутобуса
1	14 15 13 9 22 23 28 29 31 61 38 58 56 54 53 47 46 45	2	2
2	45 46 48 47 53 54 56 58 38 61 31 29 28 23 26 21 19 20 16 14 12	2	2
3	71 72 69 68 67 66 33 25 21 17 13 8 6 1 0	10	**7**
4	0 1 2 3 4 7 9 22 23 28 29 31 61 38 37 36 35	8	8
5	**11 12 16 20 19 21 26 23 28 29 31 61 38 58 56 54 53 52 51**	16	**12**
6	43 42 41 40 39 38 61 31 32 33 66 70 67 65 68 69 79 81	17	**14**
7	46 48 47 53 54 56 58 38 61 31 29 28 23 26	1	1
8	24 17 18 20 19 21 25 33 66 67 68 64 60 57 56 54 53 52 51 50 49	18	18
9	**83 78 77 76 75 74 73 68 65 32 33 66**	3	3
10	**7 9 22 24 26 21 25 33 66 67 65 62 63 59 58 56 54 53 52 51**	17	**15**
11	44 45 46 47 53 54 56 58 38 37 36 35 34	8	8
12	**69 68 70 66 67 32 31 29 28 23 22 9 7 4 5 3 2**	23	**20**
13	**15 14 16 20 18 13 17 24 26 27 30 32 65 68 69 72**	10	10
14	55 54 56 57 60 64 68 67 66 33	1	1
15	67 65 62 31 29 28 23 22 9 13 15 14 16	1	1

Табела П.10. Линије и распореди аутобуса када на мрежи има 20 линија и 5 аутобуса ван употребе

Број линије	Линија	Број аутобуса	Број аутобуса
1	6 3 5 4 7 9 22 23 28 29 31 32 33 66 70 68 69 72 71	12	12
2	70 66 33 25 21 17 13 8 6 3	1	1
3	0 1 6 8 13 17 21 25 33 66 67 68 69 72 71 82	13	13
4	0 1 2 3 4 7 9 22 23 28 29 31 61 38 37 36 35 34	9	9
5	**79 64 68 70 66 33 32 31 29 28 23 22 9 7 4 5**	14	**10**
6	43 42 41 40 39 38 61 31 32 33 66 67 65 68 69 81 71 76	17	17
7	32 31 61 38 58 56 54 53 52 51 50 49	16	16
8	8 11 12 16 20 19 21 26 27 30 32 65 67	4	4
9	18 20 19 21 25 33 66 67 68 64 60 57 56 54 53 52 51 50 49	13	13
10	55 54 56 57 59 63 62 65 67 66 33 25 21 19 20 18 17	9	9
11	73 70 66 67 68 64 60 57 56 54 53 47 48	1	1
12	40 39 38 58 56 54	1	1
13	17 24 26 27 30 32 65 68 69 72 71 81 79	4	4
14	43 42 41 40 39 38 61 31 29 28 23 22 9 7 4 3	2	2
15	34 35 36 37 29 31 65 68 73 74 75 76 77 78 83 82 71 81	5	5
16	32 30 27 26 21 19 20 16 12 11 8	4	**3**
17	50 51 52 53 54 56 57 60 64 69 72 75	1	1
18	70 66 67 68 64 60 57 56 54 53 47 48	1	1
19	15 14 16 20 19 21 25 33 66 67 68 64 60 57 56 54 53 47 46 45 44	12	12
20	45 46 48 47 53 54 56 57 60 64 68 67 66 70	1	1

Табела П.11. Линије и распореди аутобуса када на мрежи има 20 линија и 10 аутобуса ван употребе

Број линије	Линија	Број аутобуса	Број аутобуса
1	**5 4 7 9 22 23 28 29 31 32 33 66 70 68 69 72**	12	**10**
2	70 66 33 25 21 17 13 8 6 3	1	1
3	0 1 6 8 13 17 21 25 33 66 67 68 69 72 71 82	13	**12**
4	0 1 2 3 4 7 9 22 23 28 29 31 61 38 37 36 35 34	9	9
5	**79 64 68 70 66 33 32 31 29 28 23 22 9 7 4 5**	14	**9**
6	43 42 41 40 39 38 61 31 32 33 66 67 65 68 69 81 71 76	17	**15**
7	32 31 61 38 58 56 54 53 52 51 50 49	16	16
8	8 11 12 16 20 19 21 26 27 30 32 65 67	4	4
9	18 20 19 21 25 33 66 67 68 64 60 57 56 54 53 52 51 50 49	13	13
10	55 54 56 57 59 63 62 65 67 66 33 25 21 19 20 18 17	9	9
11	73 70 66 67 68 64 60 57 56 54 53 47 48	1	1
12	40 39 38 58 56 54	1	1
13	**17 24 26 27 30 32 65 68 69 72**	4	4
14	43 42 41 40 39 38 61 31 29 28 23 22 9 7 4 3	2	2
15	34 35 36 37 29 31 65 68 73 74 75 76 77 78 83 82 71 81	5	5
16	32 30 27 26 21 19 20 16 12 11 8	4	4
17	50 51 52 53 54 56 57 60 64 69 72 75	1	1
18	70 66 67 68 64 60 57 56 54 53 47 48	1	1
19	15 14 16 20 19 21 25 33 66 67 68 64 60 57 56 54 53 47 46 45 44	12	12
20	45 46 48 47 53 54 56 57 60 64 68 67 66 70	1	1

Табела П.12. Линије и распореди аутобуса када на мрежи има 20 линија и 15 аутобуса ван употребе

Број линије	Линија	Број аутобуса	Број аутобуса
1	**5 4 7 9 22 23 28 29 31 32 33 66 70 68 69 72**	12	12
2	70 66 33 25 21 17 13 8 6 3	1	1
3	0 1 6 8 13 17 21 25 33 66 67 68 69 72 71 82	13	**12**
4	0 1 2 3 4 7 9 22 23 28 29 31 61 38 37 36 35 34	9	9
5	**79 64 68 70 66 33 32 31 29 28 23 22 9 7 4 5 3**	14	**11**
6	43 42 41 40 39 38 61 31 32 33 66 67 65 68 69 81 71 76	17	**9**
7	32 31 61 38 58 56 54 53 52 51 50 49	16	16
8	8 11 12 16 20 19 21 26 27 30 32 65 67	4	4
9	**33 66 67 68 64 60 57 56 54 53 52 51 50 49**	13	**10**
10	55 54 56 57 59 63 62 65 67 66 33 25 21 19 20 18 17	9	9
11	73 70 66 67 68 64 60 57 56 54 53 47 48	1	1
12	40 39 38 58 56 54	1	1
13	**17 24 26 27 30 32 65 68 69 72**	4	4
14	43 42 41 40 39 38 61 31 29 28 23 22 9 7 4 3	2	2
15	34 35 36 37 29 31 65 68 73 74 75 76 77 78 83 82 71 81	5	5
16	32 30 27 26 21 19 20 16 12 11 8	4	4
17	50 51 52 53 54 56 57 60 64 69 72 75	1	1
18	70 66 67 68 64 60 57 56 54 53 47 48	1	1
19	15 14 16 20 19 21 25 33 66 67 68 64 60 57 56 54 53 47 46 45 44	12	12
20	45 46 48 47 53 54 56 57 60 64 68 67 66 70	1	1

Биографски подаци о кандидату

Милош Николић је рођен 12.02.1984. године. После завршене Железничке техничке школе уписао је Вишу железничку школу у Београду на којој је дипломирао 2006. године. На Универзитет у Београду – Саобраћајни факултет, одсек Логистика, уписао се школске 2006/2007. године. Основне академске студије је завршио 2009. године са просечном оценом 8,59. Завршни рад под називом „Одређивање рута возила при сакупљању комуналног отпада у Обреновцу“ оцењен је оценом 10.

Школске 2009/2010 уписао је Дипломске академске студије – Мастер, на одсеку Логистика, које је завршио са просечном оценом 9,86. Дипломски рад под називом „Систем за подршку одлучивању у логистици“ оцењен је оценом 10. Докторске студије на Саобраћајном факултету је уписао школске 2010/2011. године.

Од 1. јула 2011. године ради на Универзитету у Београду - Саобраћајном факултету у звању асистента (ужа научна област „Операциона истраживања у саобраћају“). Кандидат учествује у извођењу наставе на основним студијама на предметима: Операциона истраживања, Анализа транспортних мрежа и Лучки оптимизацини модели. На мастер студијама ангажован је на следећим предметима: Детерминистички модели операциних истраживања, Квантитативне менаџмент методе у транспорту и комуникацијама, Меки рачун и примене у саобраћају, Метахеуристички алгоритми инспирисани природом и примене у саобраћају и Математичко моделирање транспортних мрежа.

Током лета 2013. године провео је три месеца на Универзитету Калифорније у Берклију (*University of California at Berkeley*) радећи као истраживач на пројекту “*Dynamic Collaborative Gate Allocation*“. Истраживачки пројекат је завршен и предат спонзору истраживања (*NASA*). Проблеми којима се кандидат бавио у оквиру истраживачког пројекта су у директној вези са темом докторске дисертације. У досадашњем истраживачком раду је у својству аутора или коаутора учествовао у објављивању више од 20 научних радова објављених у часописима, на домаћим или иностраним конференцијама.

ИЗЈАВА О АУТОРСТВУ

Потписани-а Милош Николић

број индекса 10-д-002

Изјављујем

да је докторска дисертација под насловом

УБЛАЖАВАЊЕ ПОСЛЕДИЦА ПОРЕМЕЋАЈА У ОДВИЈАЊУ САОБРАЋАЈА ПРИМЕНОМ МЕТАХЕУРИСТИКЕ ОПТИМИЗАЦИЈА КОЛОНИЈОМ ПЧЕЛА

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

Потпис докторанда

У Београду, 25.2.2015.

М Николић

Изјава о истоветности штампане и електронске верзије докторског рада

Име и презиме аутора ____Милош Николић________________________________

Број индекса _____10-д-002____

Студијски програм ___Саобраћај_______________________________________

Наслов рада ___Ублажавање последица поремећаја у одвијању саобраћаја применом метахеуристике Оптимизација колонијом пчела_________________

Ментор ___Проф. др Душан Теодоровић, дипл. инж. саоб._________________

Потписани/а _____Милош Николић_________________________

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

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

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

Потпис докторанда

У Београду, 25. 2. 2015.

ИЗЈАВА О КОРИШЋЕЊУ

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

УБЛАЖАВАЊЕ ПОСЛЕДИЦА ПОРЕМЕЋАЈА У ОДВИЈАЊУ САОБРАЋАЈА ПРИМЕНОМ МЕТАХЕУРИСТИКЕ ОПТИМИЗАЦИЈА КОЛОНИЈОМ ПЧЕЛА

која је моје ауторско дело.

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

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

1. Ауторство
2. Ауторство - некомерцијално
3. Ауторство – некомерцијално – без прераде
4. Ауторство – некомерцијално – делити под истим условима
5. Ауторство – без прераде
6. Ауторство – делити под истим условима

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

Потпис докторанда

У Београду, 25. 2. 2015.

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.