УНИВЕРЗИТЕТ У БЕОГРАДУ

МАТЕМАТИЧКИ ФАКУЛТЕТ

Наташа С. Глишовић

ОПТИМИЗАЦИЈА ПРОБЛЕМА УПРАВЉАЊА ОДНОСИМА КОРИСТИ И ТРОШКОВА ПРИ РАСПОРЕЂИВАЊУ ПРОЈЕКАТА ПРИМЕНОМ МЕТАХЕУРИСТИЧКИХ АЛГОРИТАМА

докторска дисертација

Београд, 2017.

UNIVERSITY OF BELGRADE

FACULTY OF MATHEMATICS

Nataša S. Glišović

THE OPTIMIZATION OF THE BENEFIT/COSTS TRADEOFF DURING THE SCHEDULING OF THE PROJECTS BY APPLYING METAHEURISTIC ALGORITHMS

Doctoral Dissertation

Belgrade, 2017

Ментор:

др Миодраг Рашковић, научни саветник, Математички институт САНУ, Београд

Чланови комисије:

др Миодраг Рашковић, научни саветник, Математички институт САНУ, Београд

др Зоран Огњановић, научни саветник, Математички институт САНУ, Београд

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

др Татјана Давидовић, виши научни саветник, Математички институт САНУ, Београд

Датум одбране:

Захвалнице

Желим да изразим дубоку захвалност својим руководиоцима који су пратили развој и усмеравали аутора у исраживачком раду: проф. др Александру Јовановићу, проф. др Александру Перовићу, проф др Небојши Бојовићу, проф. др Милану Божићу, проф. др Небојши Икодиновићу, доценту др Милошу Миленковићу, као и свом ментору проф. др Миодрагу Рашковићу.

Такође захвалност дугујем и осталим блиским сарадницима, као и свом друштву, који нису овде наведени, али свакако њихов значај није тиме умањен, који су били уз мене и веровали у мој рад и труд и својом подршком и охрабрењима довели до финализације ове докторске дисертације.

Наслов дисертације: Оптимизација проблема управљања односима користи и трошкова при распоређивању пројеката применом метахеуристичких алгоритама.

Резиме: У овој докторској дисертацији разматрано је моделирање процеса одлучивања у присуству неизвесности. Анализирана су два типа проблема један је проблем оптимизације односа корист/трошак при распоређивању пројеката, а други је класификација података описаних атрибутима међу којима неки недостају. Основни проблеми при моделирању процеса одлучивања у присуству неизвесности су избор адекватног третмана неизвесности и избор методе за доношење одлуке.

Један од циљева дисертације је испитивање погодности примене метахеуристичких алгоритама на разматране оптимизационе проблеме. Главна мера евалуације њихових перформанси се састоји од вредности функције циља (било да је у питању однос корист/трошак при распоређивању пројеката или кластеровање). Код проблема распоређивања пројеката може се разматрати и ниво сатисфакције за ограничења која карактеришу проблем (тзв. ниво прага). Друга метода евалуације примењених метахеуристичких метода је време потребно да би се пронашло решење. Дискутује се и о параметрима који контролишу алгоритме метахеуристичких метода и о правилном избору њихових вредности како би се постигле максималне перформансе имплементација на тестираним примерима разматраних проблема.

Код проблема оптимизације односа корист/трошак неизвесност је моделирана применом троугаоних фази бројева, а затим су примењене метахеуристичке методе симулирано каљење и генетски алгоритам за решавање тако добијеног фази оптимизационог проблема. Тестирани проблеми формулисани су методом фазификације која је предложена од стране (Ribeiro et al. 1999.). Представљени експериментални резултати за низ фази проблема показују ефикасност примењених метода симулираног каљења и генетског алгоритма. Незнатно квалитетнија решења добијена су применом генетског алгоритма, а побољшање у односу на претходне резултате из литературе у оба случаја је око 20%.

Други део дисертације се бави проблемом кластеровања података са недостајућим вредностима атрибута и доношења одлука у таквим околностима. Главне фазе у решавању постављеног проблема су налажење најпогоднијег растојања које ће се користити у случајевима када подаци из неког разлога недостају и избор методе за решавање самог проблема кластеровања. У циљу теоријског и практичног доприноса предложена је метрика која се заснива на логици. Применом вероватноће доказана је теорема којом се одређују вредности тежинских коефицијената атрибута који описују објекте за кластеровање. Све предложено је имплементирано кроз метахеуристичку методу променљивих околина, као и неке њене модификације, и примењено на реалне и проблеме из литературе. Класификацијом пацијената који болују од аутоимуних болести, на основу базе

података Клиничког центра Србије, постигнута је прецизност кластеровања од 93.33%. Као други реалан пример, анализирано је седам база европске комисије које садрже податке везане за поштански саобраћај. Постигнута је прецизност кластеровања од 90%-96.96%. Да би се упоредила ефикасност приступа заснованих на методи промењљивих околина коришћено је и девет база доступних на интернету и добијени резултати поређени су са постојећима из литературе. Експерименти су показали велику стабилност методе промењљивих околина, у осам од девет случајева најбоље решење постигнуто је у свих сто извршавања. Осим тога, квалитет добијених решења знатно је премашио резултате из литературе.

Кључне речи: фази логика, мера растојања, вероватноћа, математичко моделирање, оптимизација корист/трошак, недостајући подаци, кластеровање, метахеуристике.

Научна област: Математика

Ужа научна област: Математичка логика

УДК број:

Dissertation title: The optimization of the benefit/costs tradeoff during the scheduling of the projects by applying metaheuristic algorithms

Abstract: In this doctoral dissertation the modelling process has been taken into consideration in the presence of uncertainty. Two types of problems were analyzed: one is the optimization of the benefit/costs tradeoff during the distribution of the projects and the other is the classification of data described by the attributes among which some are missing. The basic problems during the modelling of the decision making in the presence of uncertainty are the choice of the adequate treatment of uncertainty and the choice of the method for making a decision.

One of the aims of the work is investigating the benefits of applying the metaheuristic algorithms on the considered optimization problems. The main measure for the evaluation of their performances is the value of objective function (for both problems: optimization of benefit/costs tradeoff during the project scheduling and clustering of incomplete data). Considering the project scheduling problem the level of satisfaction related to the problem constraints could also be taken into account. The other evaluation criteria of the applied metaheuristic methods is the time required for finding the solution. The influence of the parameters which control the algorithms of the metaheuristic methods is examined, as well as their appropriate values leading to the maximum performances of the implementation could be reached on the tested examples of the considered problems.

As for the optimization problem of the profit/costs tradeoff, the uncertainty is modelled by applying the triangle fuzzy problems and then the metaheuristic methods, simulated annealing and genetic algorithm were applied for solving the obtained fuzzy optimization problem. The tested problems are formulated by the fuzzification method which was suggested by (Ribeiro et al. 1999). The represented experimental results for the set of fuzzy problems show the efficiency of the applied methods: simulated annealing and genetic algorithm. Genetic algorithm seems to produce slightly better solution than the simulated annealing. However, both methods out performed the existing form the literature for about 20%.

The secund part of the work deals with the clustering data problem with the missing values of the attributes and making decisions in such circumstances. The main phases in solving the considered problem are finding the most appropriate distance, which will be used in the cases when the data are missing for some reasons and choosing the method for solving the clustering problem. As the theoretical and practical contribution, the metric, based on the logic principles, was proposed. By applying the probability, the theorem was proved defining the values of the weighting coefficients related to attributes that describe the objects for clustering. The proposed metric was implemented in the variable neighborhood search metaheuristic method as well as in some of its modifications. The implemented methods have been applied on the real life problems from the literature. Classifying the patients who suffer from some auto-immune diseases, stored in the database of Clinical Centre of Serbia, the precision of the clustering of 93.33% was achieved. As another real life example, seven databases of the European Commision (Board), which contain the data for the mail service, have been analyzed. The clustering efficiency of 90% - 96.96% was achieved. In order to compare the efficiency of the approach based on the variable neighborhood search method, nine databases available on the internet were used and the obtained results were compared with the existing ones from the literature. The experiments showed large stability of variable neighborhood search method: in eight out of nine cases the best solution was reached in all hundred repetitions. Besides that, the quality of the obtained solutons have considerably surpassed the results from the literature.

Keywords: fuzzy logic, metrics, probability, mathematical modeling, optimization of the benefit/costs, missing data, clustering, metaheuristics

Research area: Mathematics

Research sub-area: Mathematics logic

UDC number:

Садржај

ПРВО ПОГЛАВЉЕ: Увод

Конкуренција на тржишту је увек присутна. У зависности од економске гране и економског тренутка, некада у већој, а некад у мањој мери, јавља се и присутност на тржишту нових компанија. Када су у питању веће конкуренције да би се стекла конкурентска предност над ривалима, компаније имају за циљ да се смање трошкови ресурса кроз смањење празног хода и радног времена које захтевају прецизно, обазриво и детаљно планирање пројеката.

Менаџери се често суочавају са тешкоћама приликом израде пројектних решења која подразумевају велики број међусобно повезаних активности планирања и распоређивања при управљању пројектима. Већина проблема се јавља у областима као што су развој производа, планирање производње, контрола и постављање производних објеката. Један важан аспект управљања пројектима је убрзање активности, односно смањење времена активности додавањем више ресурса, као што су радници, прековремени рад итд. Важно је донети одлуку о оптималном убрзавању система тако да се пројекат заврши у оквирима жељеног временског периода.

Приликом управљања пројектима скуп активности, са одређеним односима који постоје међу њима, мора да буде тако изведен да се изврши услуга или направи производ. Активност пројекта може бити извршена на неколико различитих начина са различитим временима и/или трошковима. Убрзавање је процес смањења трајања активности. Трајање активности може се смањити додавањем ресурса. Ресурси могу бити људи, машине, алати, извршавање алтернативних опција као што су подуговори. Повећање потрошње ресурса природно повећава трошкове и смањује време. На основу нивоа потрошње ресурса могуће је дефинисати алтернативе. Свака алтернатива дефинише своју комбинацију цене и времена. Број варијаната за сваку активност је обично ограничен. Суштина је у одабиру алтернативних активности које доводе до жељеног резултата тј. које најбоље оптимизују однос корист/трошак пројекта (енг. time/cost tradeoff), јер повећање трајања активности, с друге стране, смањује његову цену. Овакви проблеми се називају проблеми односа користи и трошка. Овај проблем је проучаван од стране истраживача дуги низ година (Siemens, N., 1971). Прве студије почеле су раних 60-

их. Функција циља и ограничења проблема доводе до три различите категорије проблема односа корист/трошак: проблем рока (трајања пројекта), проблем буџета (трошкови пројекта) и проблем односа фукције корист/трошак. Овде се корист односи на скраћење времена трајања пројекта при повећању трошка који се појављује приликом замене неких активности другима.

Проблем рока представља горњу границу трајања пројекта, односно, максимално време за извршење свих активности. Циљ је да се минимизира укупно трајање пројекта. Овај проблем је NP тежак (Dunne, 1997).

Проблем буџета представља горњу границу количине расположивих средстава која су додељена пројекту. Циљ је да се смање трошкови пројекта. Проблем спада у NP тешке.

Проблем односа функције време/трошак представља генерисање свих ефикасних решења у односу на укупан трошак, трајање и циљ пројекта. Ефикасан скуп се генерише коришћењем узастопних решења проблема времена (рока) и буџета (трошкова). Такође, овај проблем је NP тежак.

Планирање пројекта, распоређивање ресурса има велики значај, јер од тих фаза зависи трајање пројекта и његова цена. У циљу стицања конкурентске предности на тржишту, пројекат мора бити извршен у оптималном или скоро оптималном времену. Другим речима, да би се постигла конкурентност на тржишту треба оптимизовати однос корист/трошак, расподелити и ограничити ресурсе на што адекватнији начин у фази планирања пројекта.

Проблем оптимизације корист/трошак при планирању и распоређивању ресурса пројекта, врши се применом генетског алгоритма и симулираног каљења. Дат је предлог алгоритама и показана њихова ефикасност на конкретним пројектима.

Други проблем који се овом дисертацијом решавао је подстакнут основним изазовом у кластеровању (енг. clustering), а то је пронаћи одговарајући алгоритам кластеровања за дати скуп података, јер резултати кластеровања се разликују применом различитих алгоритама. Постојећа истраживања о груписању кластера која предлажу да је различитост чланова кластера кључна за одабир чланова неког кластера (Fern and Brodley, 2003), наглашавајући да већа различитост између чланова кластера има тенденцију да произведе квалитетније груписање објеката.

Насупрот томе, неке од студија су показале да се до најбољих група кластера долази ако је посматрани ниво различитости мањи (Hadjitodorov et al., 2006). Овако, на изглед, контрадикторна посматрања могу се објаснити чињеницом да сваки скуп података има своје карактеристике и као такав може захтевати посебне третмане. Неколико новијих студија бавило се истраживањима како дизајнирати тј. изабрати доброг представника кластера користећи хеуристику која се заснива на различитости (Hadjitodorov et al., 2006; Fern and Lin, 2008). Иако се показало да се перформансе кластера могу побољшати применом хеуристика, оне су направљене да буду универзално примењиве на све скупове података. Зато су Javad и Xiaoli (2009) аутори били мотивисани да посматрају овај проблем другачије. У њиховом истраживању на основу понашања чланова кластера, користећи четири скупа података за обуку, предлаже се формирање чланова кластера на основу карактеристика самих података тако да резултујући представник кластера (центар) буде најпогоднији за одређени скуп података.

Различити аутори користе различите стратегије груписања података. Већина критеријума функције циља кластеровања су нелинеарне и неконевексне тако да проблем може имати локалне минимуме који нису нужно оптимални (Selim et al., 1984). Уз то, они имају експоненцијалну сложеност у смислу броја кластера и постају NP тешки проблеми када број кластера прелази три (Welch et al., 1983). Због стратешког значаја кластеровања на многим пољима овим проблемима су се бавили многи научници и дали неколико алгоритама у предходном периоду (Banfield et al., 1993; Jiang et al., 1997; Szczubialka et al., 1998; Fernández Pierna et al., 2000; Tran et al., 2003). Недавно су истраживања дала предлоге коришћења и еволуционих метахеуристичких алгоритама, табу претраживања (Al-Sultan, 1995), генетских алгоритама (Jiang et al., 1997; Szczubialka et al., 1998; Fernández Pierna et al., 2000; Tran et al., 2003; Murthy et al., 1996) и симулираног каљења (Selim et al., 1991; Sun et al., 1994) као успешних метода при груписању података. У раду Shelokar et al. (2004), предложен је алгоритам оптимизације који користи колонију мрава (енг. Ant colony optimization (ACO)) за потребе кластеровања. Метахеуристике које користе отимизацију засновану на овим алгоритмима предложио је Dorigo са сарадницима (Dorigo et al., 1996 и 1999) за решавање проблема оптимизације дискретног типа.

К-минс (енг. K-means) је алгоритам који је популаран за коришћење одабира центара приликом кластеровања због једноставности и ефикасности, која је линеарне сложености. Међутим, К-минс зависи од почетног одабира центара, а тиме и од конвергенције ка локалном минимуму (Selim et al., 1984). Да би решили овај проблем уведени су алгоритми кластеровања засновани на хеуристици. Генетски алгоритам за проблеме кластеровања предложен је од стране Mualik и Bandyopadhyay (2002) аутора. Krishna и Murty (1999) аутори су предложили нови приступ К-минс алгоритма којим су дефинисали основни оператор мутације специфичан за кластеровање назван мутација заснована на растојању. Симулирано каљење за решавање проблема кластеровања предложено је од стране Selim и Al-Sultan (1991) аутора. Детаљно су дискутовали параметре алгоритма и доказали терорijски да се глобално решење може постићи за проблем кластеровања. Sung и Jin (2000) аутори су предложили хеуристику засновану на табу претраживању за кластеровање. Две комплементарне функционалне процедуре које се називају процедуре паковања и ослобађања су комбиноване са табу претраживањем.

Последњих деценија, моделовање засновано на колективном понашању група (природних или вештачких) агената, као што су птице, мрави, пчеле, бактерије, роботи, у циљу претраживања и оптимизације, примењено на кластеровање показало се успешним.

Алгоритам заснован на колонији мрава за груписање података представили су Shelokar, Jayaraman и Kulkarni (2004) аутори. Алгоритам користи упошљавање агената који имитирају реалну ситуацију у којој мрави ефикасно проналазе најкраћи пут до хране и назад. Перформансе овакво предложеног алгоритма су упоређене са резултатима добијеним генетским алгоритмом, табу претрагом и симулираним каљењем. Оптимизација коришћењем алгоритма роја честица која симулира јата птица коришћена је за кластеровање од стране Kao, Zahara и Kao (2008) аутора. У циљу даљег побољшања перформанси, алгоритам роја пчела је хибридизован са К-минс и Nelder–Mead симплекс методом претраге. Његове перформансе упоређене су са генетским алгоритмом (Murthy et al., 1996) и KGA (K-Means и GA-clustering) (Bandyopadhyay et al., 2002) алгоритмом. Алгоритми засновани на пчелама су најпопуланији. Њихово опонашање учења, памћења и размене информација недавно су међу најинтересантнијим истраживачким областима у интелигенцији роја (енг. swarm intelligence) (Teodorović et al., 2006).

Mladenović и Hansen (1997) аутори су представили алгоритам променљивих околина и применили га за проблем p-медијана. Преглед основа променљивих околина и њихово упоређивање са различитим метахеуристичким алгоритмима представили су у раду Hansen и Mladenović (2001) аутори. Показали су да је метода променљивих околина супериорнија у односу на остале разматране методе.

Зато се проблем у овој дисертацији који се бави кластеровањем решавао управо применом ове методе.

У истраживањима се све чешће користе базе података или скупови података који описују реалне ситуације те с тога долази и до недостајућих података (Allison, PD, 2001; Bose, S., et al., 2012). Иако недостајући подаци можда не изазивају озбиљне проблеме, посебно када њихов број није велик, то не значи да тај проблем не заслужује посебну пажњу. Неки аутори тврде да би такве податке требало одбацити и не узимати их у разматрање. Међутим, оваква мишљења имају очигледне недостатке који су детаљније дати у Chen, SM., et al. 2000, Donders, AR., et al., 2006 и Little, RJA., et al., 2002. У наведеним радовима предлаже се замена недостајућих података неким новим вредностима, што је у литератури названо импутација. Од расположивих метода замене разликују се две категорије: појединачна импутација и вишеструке методе импутације. Нажалост у неким случајевима са великим бројем недостајућих података стратегије импутације предложене у овим радовима постају неизводљиве.

Због актуелности проблема недостајућих података и решавања истог у циљу побољшања кластеровања када подаци недостају предложено је ново растојање (Glišović, N., et al., 2017) које не захтева попуњавање недостајућих података. Оно је имплементирано у оквиру методе променљивих околина и примењено на реалне проблеме кластеровања.

1. 1. Циљ истраживања

Циљ овог истраживања је развој алата за помоћ приликом доношења одлука у присуству неизвесности. Разматрана су два проблема: оптимизација односа корист/трошак при рапоређивању пројеката и проблем кластеровања. У оба проблема присутна је неизвесност тако да је основни циљ ове дисертације проналажење адекватног начина за њено моделирање. Када се разреше проблеми неизвесности проблем се своди на минимизовање функције циља. У вези са тим, циљ ове дисертације је избор адекватних метода за проналажење тог минимума.

Када је први проблем у питању укупна вредност пројекта на основу односа корист/трошак тј. ангажмана ресурса и трајање пројекта, је управо оно што треба минимизовати.

Трајање пројекта се често може скратити убрзавањем неке активности додајући нове ресурсе тој активности, али тиме се повећавају трошкови. Убрзање активности повећава трошкове тако да та активност постаје директан трошак пројекта. Међутим, убрзање активности смањује трајање пројекта и смањује индиректне трошкове пројекта. Циљ је оптимизовати суму директних и индректних трошкова сводећи је на минимум, што је у литератури познато под именом „time-cost tradeoff“ проблем (Hegazy, 1999).

Прекорачење времена захтева ангажовање додатних ресурса (људи и опреме) чиме се смањује продуктивност. Тиме продуктивност активности опада, а укупна радна снага и трошкови опреме се повећавају. Промене коришћења ресурса смањују продуктивност што изазива промену радне снаге и машина. Да би се спречио празан ход одређених ресурса није пожељно упошљавати их по сваку цену, јер и један и други проблем могу проузроковати додатне као што су прекомерни пад производње, проблеми у запошљавању радне снаге у појединим периодима. Повећање цене превоза, проблеми временског коришћења машине, тренутна немогућност изнајмљивања у одређеном периоду су неки од проблема потребе коришћења оваквих ресурса. Као резултат тога потражња ресурса има за циљ да буде што је више могуће опртимизована како би се смањило време у коме се ресурси не користе. Овај проблем је познат као проблем нивелисања ресурса. Њиме се не врши смањење потражње ресурса и не врши њихово ограничавање, али се њиме може постићи минимизација потраживања.

У планирању пројекта не може бити ограничења за ангажовање одређене врсте радова или изнајмљивања опреме. Такође не може бити ограничења за смештај ресурса. Као резултат тога могу постојати само ограничења када је у питању ангажовање максималног броја радне снаге и/или изнајмљивање машина. У већини случајева ресурси који се користе у активностима пројекта имају своје минимално и максимално време коришћења. Као последица тога, кашњења активноси се дешавају ако се те активности налазе на критичном путу. Од значаја су само те активности и њихова оптимизација. Опимизацијом ових активности које се налазе на критичном путу смањује се трајање пројекта.

Један од циљева дисертације је проналажење оптималног решења проблема планирања. Међутим, ако се број активности пројекта повећава, повећава се и димензија проблема. Тиме долази до повећања и времена израчунавања тј. захтева се већи број итерација, што пролонгира време долажења до решења. Често то пролонгирање буде дуже од реалног тако да не служи сврси. Из тог разлога, оптимална решења имају за циљ да, посебно код великих пројеката, смање време трајања.

Дати су предлози метахеуристичких алгоритама у циљу оптимизације проблема управљања односима користи и трошка при распоређивању пројекта. Конкретно, активности које чине пројекат могу имати неколико адекватних замена, при чему свака од њих има одговарајуће трошкове и стохастичко време трајања. Коначна вредност пројекта је резултат времена и трошкова потребних за завршетак сваке активности. Дакле, алгоритми имају за циљ да процене комбинације алтернативних активности и да одаберу оптималну комбинацију која ће смањити време и трошак пројекта.

Тешко је формулисати и приказати промене и одобравања додавања ресурса преко полиномијаланих функција или линеарних једначина. Слично, хеуристички алгоритми могу имати проблем "заглављивања" у локалном минимуму, зато коришћење метахеуристичких алгоритама представља кориснији метод за решавање ових проблема.

Други проблем који је решаван у овој дисертацији је проблем кластеровања. Кластеровање је техника истраживања података која открива објекте (који се описују атрибутима) сличних особина и дели их у групе (кластере), чинећи их прегледнијим и кориснијим. Оно заправо представља проналажење група објеката таквих да су објекти у групи међусобно слични (или повезани) и да су објекти у различитим групама међусобно раличити (или неповезани).

Краће, проблем кластеровања се може дефинисати на следећи начин: дат је жељени број кластера k, скуп података од n објеката и функција за мерење растојања. Потребно је пронаћи партције скупа тачака тако да се минимизује вредност функције циља која зависи од растојања међу објектима.

У кластер анализи циљ је утврђивање хомогених група или кластера при чему групна припадност објеката није позната, као и коначни број група. Неке од важних одлука које треба донети приликом кластеровања је наћи меру сличности која ће најбоље, у смислу мин/мах функције циља, разврстати објекте у одређене кластере, одредити број кластера на основу података тј. њихових особина, како третирати податке када су непотпуни. Управо ови проблеми су анализирани и дати су предлози њиховог решавања.

Да би се уопште спровела анализа груписања, неопходно је дефинисати мере блискости два објекта на основу њихових карактеристика. Концепт „сличности” се одређује у зависности од самих података. С обзиром да су подаци у већини случајева вектори стварних вредности, Еуклидска удаљеност између података може послужити као мера те различитости. Међутим, ограничења настају када су подаци непотпуни тј. када постоје недостајући елементи вектора. Управо овај проблем је разматран у овом докторату. Како превазићи овај пролбем и да ли постоји бољи начин превазилажења овог проблема од већ постојећих који се користе у пракси. Овим делом истраживања дали смо предлог растојања и тестирали његову успешност са постојећим методама. Имплементирали смо предложено растојање у методи променљивих околина и показали успешност имплементиране методе.

1. 2. Значај дисертације

У дисертацији постоје двојаки резултати, теоријски који поткрепљују примењену методологију и обезбеђују коректност добијених резултата и експериментални који указују на практичну примену изабране методологије, тј. имплементацију система за подршку одлучивању који може бити од велике користи менаџерима и руководиоцима приликом доношења одлука важних за реализације пројеката.

Добијање оптималног или скоро оптималног решења проблема планирања и распоређивања пројекта састоји се од проблема односа корист/трошак нивелације ресурса и њихове алокације који су главни циљ овог истраживања. Развијени су метахеуристички алгоритми ради примене у претрази оптималних или скоро оптималних решења. Поступак оптимизације, имплементиран кроз ове алгоритме је одрађен софтверски, у програмском језику C#, и његови резултати су приказани у поглављу 4. Симулирано каљење као и генетски алгоритам су имплементирани и анализирана су решења при распоређивању пројекта рестаурације дела „Поште Србије" добијена овим алгоритмима.

Поред проблема планирања и распоређивања пројеката у дисертацији се разматра и проблем кластеровања у ситуацијама када подаци који описују објекте за разврставање нису потпуни.

У вези са тим је предложена нова функција мерења растојања која је показала супериорност у поређењу са постојећим техникама превазилажења проблема недостајућих података. Примењена је у кластеровању медицинских података Клиничког центра одељења за алерголигију и имунологију, бази европске комисије која садржи податке везане за пошрански саобраћај и на разне друге базе података доступне на интернету. У овом делу истраживања од метахеуристика користили смо RVNS (енг. Reduced Variable Neighborhood Search) и BVNS (енг. Basic Variable Neighborhood Search).

Ова дисертација се састоји од решења различитих врста проблема планирања пројекта и кластеровања када подаци недостају. У првом делу истраживања је задатак оптимизације корист/трошак проблема. Други задатак је нивелација и распоређивање ресурса, трећи задатак подразумева решавање проблема расподеле ресурса метахеуристичким техникама, линераним програмирањем. У другом делу истраживања су проблеми недостајућих података, кластеровања и метод промењљивих околина као ефикасна метода за решавање оваквих проблема.

1. 3. Основни доприноси дисертације

Ова докторска дисертација бави се моделирањем проблема код којих постоји неизвесност и у вези са тим постигнути су следећи доприноси:

- теоријски
- практични

Код проблема оптимизације односа корист/трошак неизвесност је моделирана применом троугаоних фази бројева. То је омогућило да се побољша модел линеарног програмирања из литературе у смислу да се неизвесност, која се појављује како у функцији циља тако и у ограничењима, третира за сваки параметар ограничења и/или за сваки коефицијент функције циља и/или за сваки коефицијент ограничења. Тим приступом моделирања дат је како теоријски тако и практични допринос јер у стварности постоји низ непредвиђених ситуација које могу да утичу на времена активности пројекта. Дакле, да би се приказала стварна ситуација током реализације пројекта треба релаксирати захтеве за смањење времена и мрежу ограничења. Предложени модел је имплементиран кроз метахеуристичке методе: симулирано каљење и генетски алгоритам. Добијени резултати показују да су предложене методе врло сличних перформанси, а да су супериорне у односу на постојеће методе из литературе.

За проблем кластеровања када су подаци непотпуни, од теријских резултата дато је ново растојање за проблеме где је потребно рачунати растојања код непотпуних података. Растојање је дефинисано преко формула исказне логике и не захтева попуњавање вредности недостајућих података. Као лема у поглављу 5 доказано је да ово растојање задовољава услове да буде метрика. Такође, у овом поглављу, формулисана је тероема за рачунање тежина атрибута којима су описани елементи базе. Теоријски доприноси су потврђени кроз праксу. Екпериментални резултати који су спроведени, применом предложеног растојања и тежина, на проблем кластеровања коришћењем методе променљивих околина показали су висок степен успешности како на конкретним проблемима тако и на проблемима којима су се бавили други научници са којима смо се поредили и постигли боље резултате.

1. 4. Структура дисертације

У другом поглављу ове дисертације је укратко објашњено шта је проблем оптимизације, дат је кратак увод у фази логику и приказан је фази проблем оптимизације, као и проблем линерног програмирања. Објашњење реализованих метахеуристичких алгоритама развијених за проблем оптимизације ових проблема биће приказани у трећем поглављу са својим карактеристикама и детаљима. У четвртом поглављу објашњена је метода критичног пута (енг. Critical Path Method (CPM)), праћење пројекта и типови тендера. Приказани су резултати добијени применом симулираног каљења и генетског алгоритма. Проблем кластеровања, недостајућих података дат је у петом поглављу. У овом поглављу разматрани су и резултати добијени тестирањем описаних метода на конкретним примерима. Закључна разматрања и смернице за будући рад дате су у поглављу шест.

ДРУГО ПОГЛАВЉЕ: ПРОБЛЕМ ОПТИМИЗАЦИЈЕ

2. 1. Математичка оптимизација

Оптимизација је свуда, од бизниса до техничког дизајна, од планирања одмора до дневне рутине. Пословне организације морају да повећају свој профит, а да смање трошкове. Технички дизајн мора да појача рад дизајнираног производа док, наравно, истовремено смањује трошкове. Чак и када планирамо одмор желимо да повећамо уживање, а смањимо трошкове. Према томе, проучавање оптимизације је и од научног интереса, а има и практичне импликације те с тога и методологија оптимизације има много примена.

Који год да је проблем у питању, обично је могуће формулисати проблем оптимизације у општем облику. Сви проблеми оптимизације са јасним циљевима могу уопштено бити изражени као нелинеарно огранчени проблеми оптимизације следећим општим обликом:

$$minimize/maximize\, f(x), x = (x_1, x_2, \dots, x_n)^T \in \mathbb{R}^n,$$

$$\text{у зависности од } \phi_j(x) = 0, (j = 1, 2, \dots, M),$$

$$\psi_k(x) \geq 0, (k = 1, \dots, N),$$

где су $f(x), \phi_j(x)$ и $\psi_k(x)$ скаларне функције реланог вектора колоне x. Компоненте x_i, $i = 1, \dots, n$, вектора $x = (x_1, x_2, \dots, x_n)^T$ се називају промењљивама одлучивања и оне могу бити или непрекидне или дискретне или мешавина ове две врсте. Вектор x се често назива вектором одлуке који се мења у n-димензионом реалном простору $\mathbb{R}^n$. Функција $f(x)$ се зове функција циља или функција трошка, $\phi_j(x)$ су ограничења у облику M једнакости, а $\psi_k(x)$ су ограничења написана као N неједнакости. Тако да постоје $M + N$ ограничења укупно. Формулисани проблем оптимизације је овде представљен као нелинеарни ограничени проблем.

Простор промењљивих се назива простором тражења, $\mathbb{R}^n$, док се простор вредности функције циља назива простором решења. Функције $\phi_j(x)$ и $\psi_k(x)$ дефинишу ограничења која промењљиве одлучивања треба да задовоље, тј.

дефинишу простор допустивих решења који је подскуп од $\mathbb{R}^n$. Функција циља $f(x)$ може бити или линеарна или нелинеарна. Уколико су ограничења $\phi_j(x)$ и $\psi_k(x)$ линеарна, то постаје линеарно ограничени проблем који се решава линеарним програмирањем. Проблем линеарног програмирања је важан и добро проучен.

Најједноставнија оптимизација без било каквих ограничења је тражење максимума или минимума неке функције. На пример, налажење максимума функције $f(x)$, $f(x) = xe^{-x^2}, -\infty < x < \infty$, је једноставни неограничени проблем. Док је следећи проблем једноставно ограничени проблем минимизације:

$f(x_1, x_2) = x_1^2 + x_1x_2 + x_2^2, (x_1, x_2) \in \mathbb{R}^2, x_1 \geq 0, x_2 - 2 = 0.$

Код реалних проблема тешко је измерити шта желимо да постигнемо, али ипак покушавамо да оптимизујемо одређене ствари као што је степен уживања или квалитет услуге на одмору. У осталим случајевима можда буде немогуће написати функцију циља у јасном математичком облику.

Који год били циљеви морамо их проценити више пута. У већини случајева процене функције циља троше много енергије (што кошта) и време дизајнирања. Било који ефикасни алгоритам који може смањити број процена функције циља ће уштедети и време и новац. У овој дисертације су приказани неки од алгоритама који успешно решавају овај проблем уштеде и времена и новца.

2. 2. NP комплетни проблеми

Дефиниција 2.1. Проблем припада класи NP, и називамо га проблемом недетерминистичке полиномске сложености, ако се решење датог проблема може верификовати алгоритмом полиномске сложености. Прецизније, за унапред дато решење се утврђује да ли су сва ограничења задатог проблема. При томе је потребно да он буде формулисан као проблем одлучивања, да би одговор био и формално (математички) коректан.

Дефиниција 2.2. За проблем Q_1 кажемо да је сводив у полиномском времену на проблем Q_2 ако постоји алгоритам полиномске сложености који претвара сваку интерпретацију проблема Q_1 у интерпретацију проблема Q_2 тако да имају аналогно заједничко решење.

Свођењем у полиномском времену показујемо да сложеност полазног проблема није већа од сложености другог проблема, не узимајући у обзир време извршавања алгоритма за свођење, које је полиномске сложености. Овакав метод представља врло моћан апарат за класификацију сложености NP проблема за које није до сада познат алгоритам за решавање у полиномском времену.

Дефиниција 2.3. Проблем одлучивања називамо NP-комплетним ако:

• припада класи NP;

• сви остали NP проблеми се могу алгоритмом полиномске сложености свести на дати проблем.

Помоћу претходне дефиниције, унапред прихватајући за неки полазни проблем да је NP-комплетан, релативно лако налазимо остале NP-комплетне проблеме, коришћењем метода свођења у полиномском времену. Обично се у литератури за полазни NP-комплетан проблем узима проблем задовољивости Boole-овог израза (енг. satisfiability problem) (Кратица, 2000.).

Метахеуристички алгоритми су добри за решавање проблема оптимизације који су NP-комплетни (Kirkpatrick, S., et al., 1983).

Проблеми решавани у овој дисертцији спадају у групу NP-комплетних проблема (Garey et al., 1979.).

2. 3. Фази логика

Данас је употреба фази логике у наглом порасту у разноврсним комерцијалним апликацијама и индустријским системима. Неки од примера као што су веш машине, клима уређаји, усисивачи, навигациони уређаји и многи други, довољан су доказ велике распрострањености и примењивости ове технике. Фази логика је нашла примену и у информационим технологијама и експертским системима, где се користи као подршка при одлучивању.

Фази скупове дефинисао је 1965. године Lotfi Zadeh као математички формализован начин представљања и моделирања неодређености у лингвистици. У теорији класичних скупова, неки одређени елементи или припадају или не припадају неком дефинисаном скупу. Фази скуп је, у том смислу, генерализација класничног скупа, будући да се припадност (тј. степен припадности) елемената фази скупа може окарактерисати бројем из интервала [0, 1]. Другим речима, функција припадности (енг. membeship function) фази скупа пресликава сваки елемент универзалног скупа у поменути интервал реалних бројева. Једна од највећих разлика између класичних и фази скупова јесте у томе што класични скупови увек имају јединствену функцију припадности, док се за фази скуп може дефинисати бесконачно много различитих функција припадности којима се он може описати (сам израз фази (енг. fuzzy) представља нешто нејасно, замагљено, лепршаво).

Фази скуп је скуп објеката у којем није дефинисана јасна „граница“ између објеката који припадају скупу и оних који му не припадају. Много прецизнија дефиниција се може формулисати на следећи начин (Bellman, R., et al. 1970):

Дефиниција 2.4. Нека $X = \{x \in S |\ S \ \neq \emptyset\}$ представља скуп објеката (тачака). Фази скуп A у X је скуп уређених парова, $A = \{(x, \mu A(x))\}$, $x \in X$, где $\mu A(x)$ представља степен чланства x у A. Углавном претпостављамо да $\mu A(x)$ припада интервалу [0, 1] где 0 и 1 представљају нивое чланства. Тако да претпостављамо да се фази скуп може дефинисати прецизно, придруживањем сваком елементу x вредности између 0 и 1, што представља његов ниво чланства у подскупу A.

Основна разлика између фази логике и теорије вероватноће састоји се у томе да фази логика оперише са детерминистичким неодређеностима и недореченостима, док се вероватноћа бави веродостојношћу стохастичких догађаја и иза ње суштински стоји експеримент. Фази логика просто има за циљ превазилажења проблема у комуникацији везаних за разлике између правила која намећу формалне теорије и начина размишљања који описују понашање људског ума, док се вероватноћа генерално бави феноменом понављања који се симболизује случајношћу (случајним променљивама и случајним процесима). Другим речима, фази и случајни су два атрибута која се разликују у својој природи, односно, они описују другачији аспект неодређености. Дакле, фази логика покрива субјективности људског мишљења, осећања, језика, док вероватноћа покрива објективну статистику у природним наукама. Аналогно, фази модели и модели формирани на бази вероватноће носе другачији вид информација: фази функција припадности представља сличност објеката у контексту непрецизне дефиниције особина, док вероватноћа даје информацију о фреквенцији понављања.

Уопштено можемо рећи да доношење фази одлуке интегрише две фундаменталне теме, а то су „вишеструки фази атрибути“ и „фази оптимизација“ (Pires, F., et al. 1998.; Zimmermann, H., 1991.). У овом раду фокусирамо се на фази оптимизацију посебно када су коефицијенти и/или ресурси или/и циљеви непрецизни/фази (Pires, F., et al., 1996.).

2. 3. 1. Основни појмови

Да би се фази скупови увели на начин који омогућава њихово лакше разумевање, дата је паралела са класичним скуповима (енг. crisp sets).

Да би се представили фази скупови, карактеристична функција μ_{A} клсичног скупа A се може генерализовати на функцију припадности $\mu_{\widetilde{\mathrm{A}}}$ фази скупа $\widetilde{\mathrm{A}}$ (Hanss, 2005; Pap, 1999.).

Дефиниција 2.5. Функција припадности $\mu_{\widetilde{\mathrm{A}}}$ фази скупа $\widetilde{\mathrm{A}}$ је пресликавање

$$\mu_{\widetilde{\mathrm{A}}}: X \to [0,1],$$

где је $\widetilde{\mathrm{A}}$ подскуп универзалног скупа X.

Дефиниција 2.6. Фази скуп $\widetilde{\mathrm{A}}$ је скуп уређених парова који се састоји од елемената *x* универзалног скупа *X* и одређеног степена припадности $\mu_{\widetilde{\mathrm{A}}}(x)$, облика

$$\widetilde{\mathrm{A}} = \{(x, \mu_{\tilde{A}}(x) | x \in X, \mu_{\tilde{A}}(x) \in [0,1])\}.$$

Дефиниција 2.7. *n*-арна фази релација $\tilde{R}$ је уређен пар који се састоји од елемената $(x_1, x_2, \dots, x_n) \in X_1 \times X_2 \times \dots \times X_n$ и одређеног степена припадности $\mu_{\tilde{R}}(x_1, x_2, \dots, x_n)$ облика

$$\tilde{R} = \big\{\big((x_1, x_2, \dots, x_n), \mu_{\tilde{R}}(x_1, x_2, \dots, x_n)\big) \big| (x_1, x_2, \dots, x_n) \\ \in X_1 \times X_2 \times \dots \times X_n, \mu_{\tilde{R}}(x_1, x_2, \dots, x_n) \in [0,1]\big\}.$$

Дефиниција 2.8. Фази партитивни скуп $\tilde{\mathcal{P}}(\mathrm{A})$ класичног скупа A је скуп свих могућих фази подскупова $\widetilde{\mathrm{T}}$ од *A*, односно

$$\tilde{\mathcal{P}}(\mathrm{A}) = \{\tilde{T} | \tilde{T} \subseteq A\}.$$

Дефиниција 2.9. (Bezdek, JC, 1980) Висина $hgt(\tilde{A})$ фази скупа $\tilde{A} \in \tilde{\mathcal{P}}(X)$ је супремум (максимум ако је универзални скуп X коначан) функције припадности $\mu_{\tilde{A}}(x)$:

$$hgt(\tilde{A}) = sup_{x \in X} \mu_{\tilde{A}}(x).$$

ако је $hgt(\tilde{A}) = 1$. $\tilde{A}$ се зове нормализован скуп, иначе је субнормализован.

Дефиниција 2.10. Језгро $core(\tilde{A})$ фази скупа $\tilde{A} \in \tilde{\mathcal{P}}(X)$ је класичан скуп свих елемената $x \in X$ чији је степен припадности једнак јединици тј.

$$core(\tilde{A}) = \{x \in X | \mu_{\tilde{A}}(x) = 1\}.$$

Дефиниција 2.11. Носач $supp(\tilde{A})$ фази скупа $\tilde{A} \in \tilde{\mathcal{P}}(X)$ је класични скуп свих елемената $x \in X$ који имају ненула степен припадности, односно

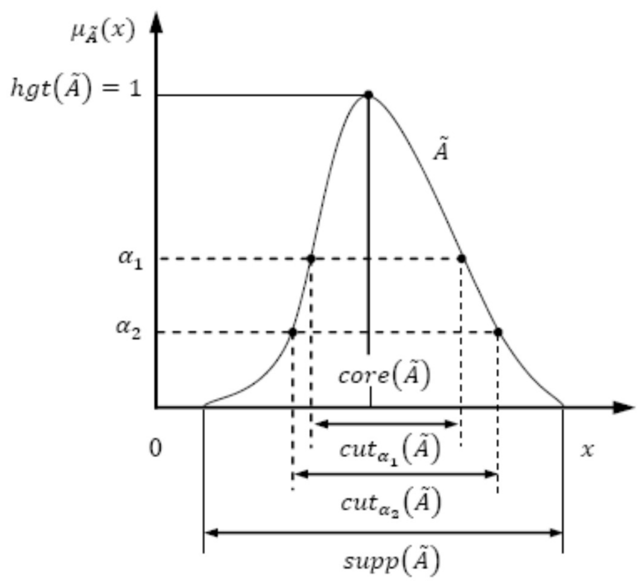
$$supp(\tilde{A}) = \{x \in X | \mu_{\tilde{A}}(x) > 0\}.$$

Дефиниција 2.12. $\alpha -$ пресек $cut_{\alpha}(\tilde{A})$ фази скупа $\tilde{A} \in \tilde{\mathcal{P}}(X)$ је класичан скуп свих елемената $x \in X$ који припадају фази скупу $\tilde{A}$ са степеном најмање $\alpha \in [0,1]$:

$$cut_{\alpha}(\tilde{A}) = \{x \in X | \mu_{\tilde{A}}(x) \geq \alpha\}.$$

Јаки $\alpha -$ пресек $cut_{\alpha+}(\tilde{A})$ фази скупа $\tilde{A} \in \tilde{\mathcal{P}}(X)$ се дефинише са:

$$cut_{\alpha+}(\tilde{A}) = \{x \in X | \mu_{\tilde{A}}(x) > \alpha\}.$$



Слика 2. 1. Функција припадности фази скупа, висина, језгро, носач скупа и $\alpha -$ пресеци.

Дефиниција 2.13. Кардиналност $card(\tilde{A}) = |\tilde{A}|$ дискретног фази скупа $\tilde{A} \in \tilde{\mathcal{P}}(X)$ са коначним носачем $supp(\tilde{A})$ се може дефинисати као

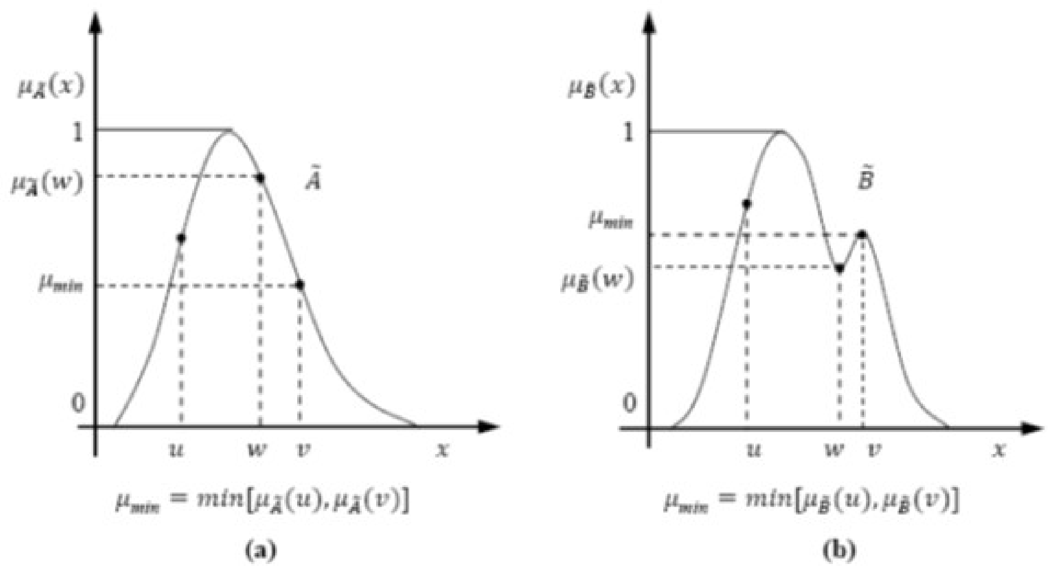
$$card(\tilde{A}) = |\tilde{A}| = \sum_{x \in X} \mu_{\tilde{A}}(x) = \sum_{x \in supp(\tilde{A})} \mu_{\tilde{A}}(x).$$

Дефиниција 2.14. Фази скуп $\tilde{A} \in \tilde{\mathcal{P}}(X)$ је конвексан ако за сваки елемент $u \in cut_\alpha(\tilde{A})$ и $v \in cut_\alpha(\tilde{A})$ и за свако $\alpha \in [0,1]$ важи

$$\lambda u + (1-\lambda)v \in cut_\alpha(\tilde{A}), \quad \forall \lambda \in [0,1].$$

односно, у термину функција припадности, фази скуп $\tilde{A} \in \tilde{\mathcal{P}}(X)$ је конвексан ако за све $u, v, w \in supp(\tilde{A})$ и за $u \leq w \leq v$ важи

$$\mu_{\tilde{A}}(w) \geq \min[\mu_{\tilde{A}}(u), \mu_{\tilde{A}}(v)].$$



Слика 2.2. (a) конвексан фази скуп $\tilde{A} \in \tilde{\mathcal{P}}(\mathbb{R})$, (b) Неконвексан фази скуп $\tilde{B} \in \tilde{\mathcal{P}}(\mathbb{R})$.

Напомена 2.1. Особине које су претходно дефинисане за фази скупове се аналогно дефинишу за фази релације.

2. 3. 3. Операције са фази скуповима и релацијама

Дефиниција 2.15. Инклузија фази скупа $\widetilde{\mathrm{A}}$ у други фази скуп $\tilde{B}$ се може дефинисати преко функција припадности $\mu_{\widetilde{\mathrm{A}}}(x)$ и $\mu_{\tilde{B}}(x)$ као

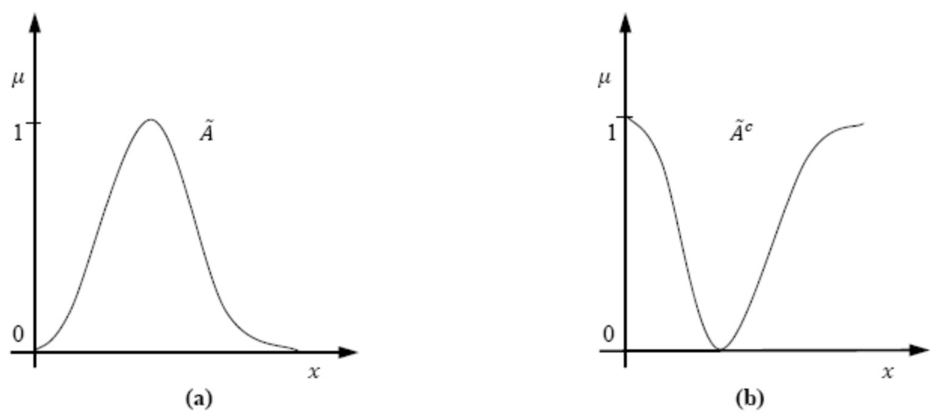
$$\tilde{A} \subseteq \tilde{B} \Leftrightarrow \mu_{\tilde{A}}(x) \leq \mu_{\tilde{B}}(x), \quad \forall x \in X.$$

Дефиниција 2.16. Једнакост два фази скупа $\widetilde{\mathrm{A}}$ и $\tilde{B}$ се дефинише преко функције припадности $\mu_{\widetilde{\mathrm{A}}}(x)$ и $\mu_{\tilde{B}}(x)$ са

$$\tilde{A} = \tilde{B} \Leftrightarrow \mu_{\tilde{A}}(x) = \mu_{\tilde{B}}(x), \quad \forall x \in X.$$

Дефиниција 2.17. За дати фази скуп $\tilde{A} \subseteq X$ са функцијом припадности $\mu_{\tilde{A}}(x)$, функција припадности $\mu_{\tilde{A}^c}(x)$ комплемента $\tilde{A}^c$ се може дефинисати као

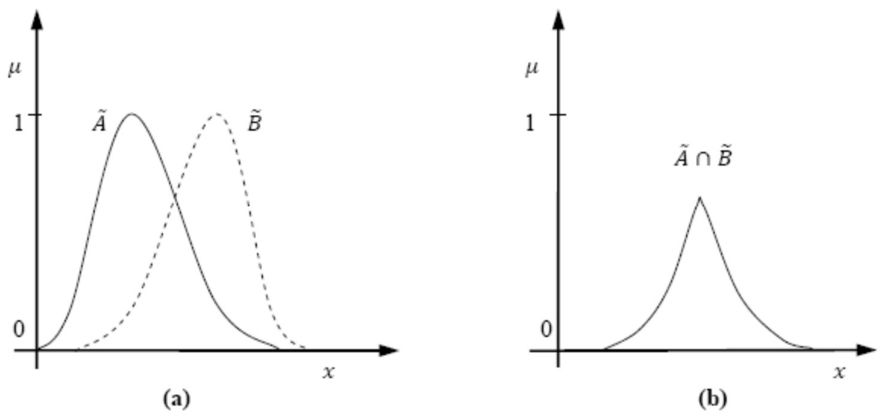
$$\mu_{\tilde{A}^c}(x) = 1 - \mu_{\tilde{A}}(x) \qquad \forall x \in X.$$



Слика 2.3. Функција припадности (а) $\mu_{\tilde{A}}(x)$ фази скупа $\tilde{A}$, (б) $\mu_{\tilde{A}^c}(x)$ фази комплемента $\tilde{A}^c$.

Дефиниција 2.18. За дата два фази скупа, $\tilde{A}$ и $\tilde{B}$, $\tilde{A}, \tilde{B} \subseteq X$ са функцијама припадности $\mu_{\tilde{A}}(x)$ и $\mu_{\tilde{B}}(x)$, функција припадности $\mu_{\widetilde{\mathrm{A}} \cap \tilde{B}}(x)$ пресека $\widetilde{\mathrm{A}} \cap \tilde{B}$ се може дефинисати са

$$\mu_{\widetilde{\mathrm{A}} \cap \tilde{B}}(x) = min[\mu_{\tilde{A}}(x), \mu_{\tilde{B}}(x)], \quad \forall x \in X.$$



Слика 2.4. Функција припадности (а) $\mu_{\tilde{A}}(x)$ и $\mu_{\tilde{B}}(x)$ фази скупова $\tilde{A}$ и $\tilde{B}$, (б) $\mu_{\tilde{A}\cap\tilde{B}}$ фази пресека $\tilde{A} \cap \tilde{B}$.

Дефиниција 2.19. Троугаона норма или t-норма је функција i: $[0,1] \times [0,1] \mapsto [0,1]$ која за све $\mu_0, \mu_1, \mu_2 \in [0,\ 1]$ задовољава следеће услове (Klement et al., 2000):

I1 (рубни услов): $i(\mu_0, 1) = \mu_0$,

I2 (монотоност): $\mu_1 \leq \mu_2 \Longrightarrow i(\mu_0, \mu_1) \leq i(\mu_0, \mu_2)$,

I3 (комутативност): $i(\mu_1, \mu_2) = i(\mu_2, \mu_1)$,

I4 (асоцијативност): $i\big(\mu_0, i(\mu_1, \mu_2)\big) = i(i(\mu_0, \mu_1), \mu_2)$.

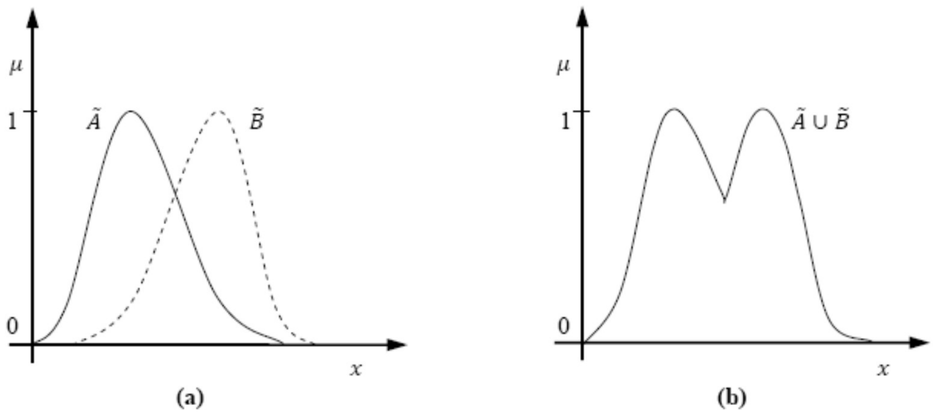
Дефиниција 2.20. Фази пресек $\tilde{A} \cap \tilde{B}$ два фази скупа $\tilde{A}$ и $\tilde{B}$ је нови фази скуп са функцијом припадности облика

$$\mu_{\tilde{A}\cap\tilde{B}}(x) = i[\mu_{\tilde{A}}(x), \mu_{\tilde{B}}(x)], \quad \forall x \in X.$$

где је *i* произвољна t-норма.

Дефиниција 2.21. За дата два фази скупа $\tilde{A}$ и $\tilde{B}$, $\tilde{A}, \tilde{B} \subseteq X$ са функцијом припадности $\mu_{\tilde{A}}(x)$ и $\mu_{\tilde{B}}(x)$, функција припадности $\mu_{\tilde{A}\cup\tilde{B}}(x)$ уније $\tilde{A} \cup \tilde{B}$ је облика

$$\mu_{\tilde{A}\cup\tilde{B}}(x) = \max[\mu_{\tilde{A}}(x), \mu_{\tilde{B}}(x)], \quad \forall x \in X.$$



Слика 2.5. Функција припадности (a) $\mu_{\tilde{A}}(x)$ и $\mu_{\tilde{B}}(x)$ фази скупова $\tilde{A}$ и $\tilde{B}$, (b) $\mu_{\tilde{A}\cup\tilde{B}}(x)$ фази уније $\tilde{A} \cup \tilde{B}$.

Дефиниција 2.22. Троугаона конорма или t-конорма је функција $u: [0,1] \times [0,\ 1] \longmapsto [0,1]$ која за све $\mu_0, \mu_1, \mu_2 \in [0,1]$ задовољава следеће услове:

I1 (рубни услов): $u(\mu_0, 0) = \mu_0$,

I2 (монотоност): $\mu_1 \leq \mu_2 \Longrightarrow u(\mu_0, \mu_1) \leq u(\mu_0, \mu_2)$,

I3 (комутативност): $u(\mu_1, \mu_2) = u(\mu_2, \mu_1)$,

I4 (асоцијативност): $u\big(\mu_0, u(\mu_1, \mu_2)\big) = u(u(\mu_0, \mu_1), \mu_2)$.

Дефиниција 2.23. Фази унија $\tilde{A} \cup \tilde{B}$ два фази скупа $\tilde{A}$ и $\tilde{B}$ је нови фази скуп дат следећом функцијом припадности

$$\mu_{\tilde{A}\cup\tilde{B}}(x) = u[\mu_{\tilde{A}}(x), \mu_{\tilde{B}}(x)], \qquad \forall x \in X,$$

где је *u* прозвољна t-конорма.

Пример: Четири најважније t-конорме су:

- MAX-унија (енг. MAX union (Fredizzi et al., 1991.)): $u_{max}(\mu_1, \mu_2) = \max[\mu_1, \mu_2]$;
- Алгебарска сума (енг. Algebraic sum (Fredizzi et al., 1991)): $u_{alg}(\mu_1, \mu_2) = \mu_1 + \mu_2 - \mu_1\mu_2$;
- Органичена сума (енг. Bounded sum (Chiu, Y.S. P., et al., 2005)): $u_{bnd}(\mu_1, \mu_2) = \max[1, \mu_1 + \mu_2]$;
- Драстична унија (енг. Drastic union (Barber, T. J., et al.,1988; Bellman, R.E., et al., 1970)): $u_{drt} = \begin{cases} \mu_1, \text{ако је } \ \mu_2 = 0, \\ \mu_2, \text{ако је } \ \mu_1 = 0, \\ 1, \quad \text{иначе.} \end{cases}$

Напомена 2.2. Предходно наведене операције са фази скуповима се аналогно дефинишу за фази релације.

Дефиниција 2.24. За дате две бинарне фази релације $\tilde{R} \subseteq X_1 \times X_2$ и $\tilde{S} \subseteq X_2 \times X_3$ са функцијама припадности $\mu_{\tilde{R}}(x_1, x_2)$ и $\mu_{\tilde{S}}(x_2, x_3)$, функција припадности $\mu_{\tilde{R}\circ\tilde{S}}(x_1, x_3)$ композиције $\tilde{R}\circ\tilde{S}$ се дефинише са

$$\mu_{\tilde{R}\circ\tilde{S}}(x_1, x_3) = \sup_{x_2 \in X_2} \min[\mu_{\tilde{R}}(x_1, x_2), \mu_{\tilde{S}}(x_2, x_3)], \quad \forall\, (x_1, x_3) \in\, X_1 \times X_3.$$

Напомена 2.3. У претходној дефиницији за композицију, уколико су фази релазије $\tilde{R}$ и $\tilde{S}$ са коначним носачем, онда се супремум замени максимумом.

2. 3. 4. Основна фази аритметика

Фази бројеви и фази вектори

Од свих типова фази скупова најчешће коришћени при моделовању реалних ситуација су управо они који су дефинисани на универзалном скупу реалних бројева $\mathbb{R}$. У специјалном случају такви скупови се називају фази бројеви. Слично, фази вектори се могу увести као специјална класа фази релација дефинисаних на $\mathbb{R}^n$.

Дефиниција 2.25. Фази скуп $\tilde{P} \in \tilde{\mathcal{P}}(\mathbb{R})$ се зове фази број $\tilde{p}$ ако задовољава следеће услове:

- $\tilde{P}$ је нормализован, тј. $hgt(\tilde{P}) = 1$,
- $\tilde{P}$ је конвексан,
- постоји тачно једно $\bar{x} \in \mathbb{R}$ такво да је $\mu_{\tilde{P}}(\bar{x}) = 1$ тј. $core(\tilde{P}) = \bar{x}$,
- функција припадности $\mu_{\tilde{P}}(x),\ x \in \mathbb{R}$ је бар по деловима непрекидна.

Дефиниција 2.26. Вредност $\bar{x}$ која показује максимални степен припадности, $\mu_{\tilde{P}}(\bar{x}) = 1$, се назива модална вредност фази броја $\tilde{p}$

Напомена 2.4. Скуп свих фази бројева на универзалном скупу $\mathbb{R}$ се означава са $\tilde{\mathcal{P}}'(\mathbb{R})$

Дефиниција 2.27. Фази број $\tilde{p} \in \tilde{\mathcal{P}}'(\mathbb{R})$ је симетричан ако његова функција припадности $\mu_{\tilde{P}}(x)$ задовољава

$$\mu_{\tilde{P}}(\bar{x} + x) = \mu_{\tilde{P}}(\bar{x} - x), \qquad \forall x \in \mathbb{R}.$$

Дефиниција 2.28. За фази број $\tilde{p} \in \tilde{\mathcal{P}}'(\mathbb{R})$ се каже да је (строго) позитиван, у ознаци $\tilde{p} > 0$ ако је $supp(\tilde{p}) \subseteq (0, \infty)$ или строго негативан, у ознаци $\tilde{p} < 0$ ако је $supp(\tilde{p}) \subseteq (-\infty, 0)$.

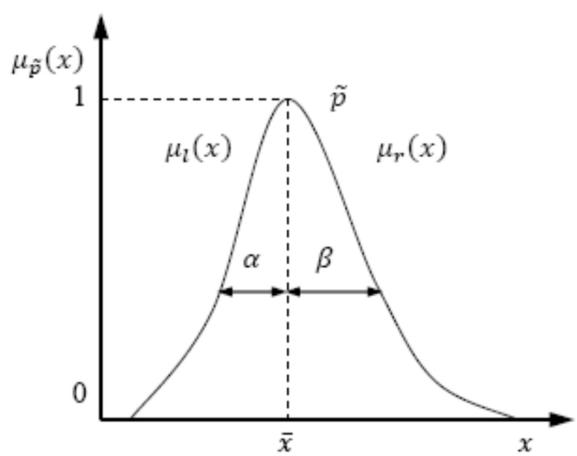
Дефиниција 2.29. Фази број $\tilde{p} \in \tilde{\mathcal{P}}'(\mathbb{R})$ се зове (фази) нула број, у ознаци $sgn(\tilde{p}) = 0$ ако није ни позитиван ни негативан, односно ако $0 \in supp(\tilde{p})$.

L-R фази бројеви

Представљају најважнији тип фази бројева (уведени су у Dubious et al., 1978. и Dubious et al., 1979.). Основна идеја L-R репрезентације фази бројева је да се функција припадности $\mu_{\tilde{p}}(x)$ фази броја $\tilde{p}$ подели на два дела $\mu_l(x)$ и $\mu_r(x)$, лево и десно од модалне вредности $\bar{x}$. Функција припадности $\mu_{\tilde{p}}(x)$ се онда може представити преко такозваних референтних функција (енг. *reference functions*) $L(u)$ и $R(u)$ односно функција припадности се може записати у следећој форми

$$\mu_{\tilde{p}}(x) = \begin{cases} \mu_l(x) = L(u_1), & \text{за } x < \bar{x}, \\ \mu_r(x) = R(u_2), & \text{за } x \geq \bar{x}, \end{cases}$$

где су $u_1 = \frac{\bar{x}-x}{\alpha}$, $u_2 = \frac{x-\bar{x}}{\beta}$, а α и β представљају одступања у леву, доносно, десну страну од $\bar{x}$ (слика 2. 6.). L-R фази број се може у скраћеној форми записати са $\tilde{p} = \langle \bar{x}, \alpha, \beta \rangle_{L.R}$.



Слика 2.6. L-R репрезентација броја $\tilde{p}$.

Да би функције $L(u)$ и $R(u), u \in \mathbb{R}_0^+$, биле референтне функције за L-R фази бројеве, треба да задовољавају следеће услове:

- $L(u) \in [0,1]$ за свако u и $R(u) \in [0,1]$ за свако u,
- $L(0) = R(0) = 1$,

- $L(u)$ и $R(u)$ су опадајуће на интервалу $[0, \infty)$,
- $L(1) = 0$ ако $min_u L(u) = 0$, $\lim_{u \to \infty} L(u) = 0$ ако $L(u) > 0$ за свако u,
- $R(1) = 0$ ако $min_u R(u) = 0$, $\lim_{u \to \infty} R(u) = 0$ ако $R(u) > 0$ за свако u.

Дефиниција 2.30. L-R фази број је семи-симетричан ако су референтне функције L и R индентичне, тј. $L(u) = R(u)$ за свако $u \in \mathbb{R}_0^+$. Уколико су и вредности α и β једнаке, L-R фази број је симетричан.

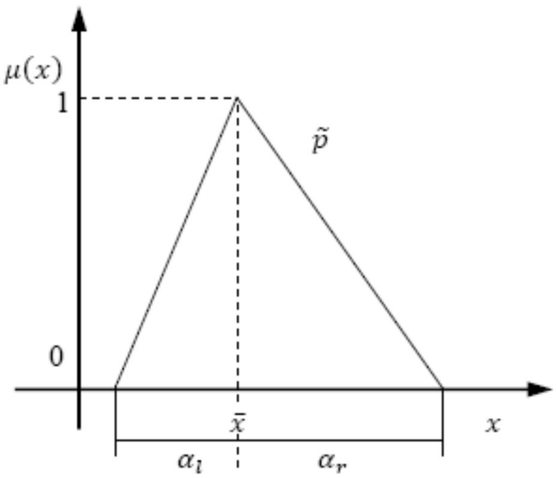
Троугаони (линеарни) фази број

Због своје прилично једноставне функције припадности линеарног типа, троугаони (линеарни) фази број је један од најчешће коришћених фази бројева. Троугаони фази број се у скраћеној форми означава са $\tilde{p} = tfn(\bar{x}, \alpha_l, \alpha_r)$, где $\bar{x}$ представља модалну вредност фази броја, а α_l и α_r одступање са леве, односно, десне стране од модалне вредности, док се функција припадности (слика 2. 7.) дефинише на следећи начин:

$$\mu_{\tilde{p}}(x) = \begin{cases} 1 + \dfrac{x - \bar{x}}{\alpha_l}, & \text{за } \bar{x} - \alpha_l < x < \bar{x}, \\ 1 - \dfrac{x - \bar{x}}{\alpha_r}, & \text{за } \bar{x} \le x < \bar{x} + \alpha_r, \\ 0, & \text{иначе}, \end{cases} \qquad \forall x \in \mathbb{R}.$$

Скуп вредности из универзалног скупа $\mathbb{R}$ које припадају фази броју $\tilde{p}$ се означава као интервал W фази броја $\tilde{p}$ и дефинише се са:

$$W = [w_l, w_r] = [\bar{x} - \alpha_l, \bar{x} + \alpha_r] = supp(\tilde{p}) \cup \{\bar{x} - \alpha_l, \bar{x} + \alpha_r\}.$$



Слика 2.7. Троугаони фази број.

Фази вектори

Дефиниција 2.31. n-арна фази релација $\tilde{P} \in \tilde{\mathcal{P}}(\mathbb{R}^n)$ се зове n-димензионални фази вектор $\tilde{p}$ ако задовољава следеће услове:

- $\tilde{P}$ је нормализована, тј. $hgt(\tilde{P}) = 1$,
- $\tilde{P}$ је конвексна.

Постоји тачно један $(\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n) \in \mathbb{R}^n$ са особином $\mu_{\tilde{P}} = (\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n) = 1$ тј. $core(\tilde{P}) = (\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n)$.

Функција припадности $\mu_{\tilde{P}}: \mathbb{R}^n \to [0, 1]$ је по деловима непрекидна.

Дефиниција 2.32. Класичан вектор $\overline{\boldsymbol{x}}$ дефинисан са $\overline{\boldsymbol{x}} = [\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n]^T = core(\widetilde{\boldsymbol{p}})$ и $\mu_{\widetilde{\boldsymbol{p}}}(\overline{\boldsymbol{x}}) = 1$ се зове моделани вектор фази вектора $\widetilde{\boldsymbol{p}}$.

Напомена 2.5. Скуп свих фази вектора на универзалном скупу $\mathbb{R}^n$ се означава са $\tilde{\mathcal{P}}'(\mathbb{R}^n)$.

2. 3. 5. Фази проблем оптимизације

Класичан проблем оптимизације се састоји од максимизирања или минимизирања извесне функције циља која је подложна низу ограничења, што изражава, на пример, ограничење ресурса. Формално, $max\, f(x)$ или $min\, f(x)$, где је $x \in X$.

Први модел решавања проблема одлучивања у фази окружењу је предложен од стране Bellman, R., et al. 1970. Ови аутори сматрају да одлука у вези сваке могућности резултује из пресека циљева и ограничења, а да унија одлука за сваку могућност даје „најбољу" одлуку. Фази одлуку можемо видети као пресек циљева и ограничења, што значи да је то један симетрични модел. Максимизирање фази одлуке се дефинише као нека тачка у простору, у којој функција чланства фази одлуке постиже своју максималну вредност (Bellman et al., 1970).

Главни циљ фази оптимизације је проналажење „најбољег" решења у присуству некомплетних информација тј. непрецизних информација и /или у присуству „нејасних" ограничења у информацијама. Постоји много облика непрецизности у проблемима фази оптимизације, као на пример, променљиви коефицијенти чије вредности нам нису прецизно познате (на пример, „време обраде" у трајању од сат времена, склапања једног дела) и нивоа сатисфакције ограничења са непрецизним ограничењима (на пример, „укупно расположиво време обраде је око сто сати"). Тренутно, изазов је на допуштању конструкције модела који дозвољавају интерепретирање нејасних и непрецизних појмова, што се може превести у фази квантитативним методама (Ribeiro, R. A., et al. 1999).

Класичан проблем линеране оптимизације се може формализовати као

$$max/\min Z = Cx,\ Ax \leq B,\ x \geq 0. \quad (2.\ 1)$$

Фази верзија овог проблема је опште формулације

$$\max/\min \tilde{Z} = \tilde{C}x,\ \tilde{A}x\{\geq, \leq, =\}\ \tilde{B},\ x \geq 0, \quad (2.\ 2)$$

где $\tilde{Z}$ представља фази функцију циља, $\tilde{C}$ је вектор фази трошкова, $\tilde{A}$ је матрица која садржи фази коефицијенте циља(ева) и ограничења, а $\tilde{B}$ је одговарајући вектор ограничења. “Тилда” на врху параметара значи да су они дефинисани као фази скупови.

У општем случају, процес решавања фази оптимизације се састоји од следећих корака:

- зависно од типа проблема, потребно га је формално представити као проблем линеарног програмирања или као вишециљни проблем или чак као проблем нелинеарног програмирања;
- дефинисање циљева;
- одаберу се функције чланства ради представљања фазификације било ког параметра и ограничења, тј.његовог представљања троугаони, синусоидни, трапезни или остали;
- дефинисање функције чланства са неопходним параметрима због вредности избора и толеранције;
- дефинисање прагова за дозвољени степен девијација/нарушавања задовољења ограничења;
- дефинисање оператора скупова ради комбиновања ограничења и циљева, тј. t-норме минимума оператора;
- решавање проблема (у овом раду помоћу генетског алгоритма и алгоритма симулираног каљења).

Први предлог формулисања проблема фази линеарног програмирања је дао Zimmerman 1976. Овај аутор сматра да постоји флексибилност у нарушавању и ограничења и циљева. Овај модел се базира на симетричном моделу Bellman и Zadeh (тј. не постоји разлика између ограничења и циљева). Међутим, тренутно постоји неколико метода фазификације које су предложене у литератури за ресурсе и фазификацију циљева исто као и за фазификацију коефицијената (погледати Lai, Y., et al., 1994).

Фазификација једног или више објеката модела линеарног програмирања обично обухвата четири облика непрецизности (Fredizzi et. al., 1991) (опширније о овоме може се наћи у Lai and Hwang, 1994):

- проблем фази ограничења;
- проблем фази циљева (циљеви наметнути објектном функцијом);
- проблем фази коефицијената промењливих;

- комбинација горе наведених проблема.

У овом истраживању се разматра фазификација модела линераног програмирања која се бави проблемом распореда пројекта у „несигурној средини“, где се непрецизности, тј. фазификација, јавља у границама ограничења и у функцији циља (комбинација горе наведеног првог и другог проблема). Дакле, ово фазификовање циља, као и скупа фази ограничења типа „мање или једнако“ или „веће или једнако“, формулисано је као:

$min\tilde{Z} = c^T x,$

(2. 3)

$Ax\{\lesseqqgtr, \gtreqqless\}\tilde{B},$

(2. 4)

$x \geq 0,$

(2. 5)

где $\tilde{Z}$ представља фази функцију циља, c^T је вектор трошкова, A је матрица која садржи фази коефицијенте циља(ева) и ограничења, а $\tilde{B}$ је одговарајући вектор ограничења.

Поред моделирања неизвесности и непрецизности, у овој дисетацији смо заинтересовани за тестирање хибридизације фазификације зато што желимо да прикажемо флексибилност са метахеуристичким методама, конкретно симулираним каљењем и генетским алгоритмом да бисмо решили линеране или нелинеарне проблеме у фази окружењу.

2. 4. Комплетан модел „флексибилизације“ неког фази проблема

Неколико аутора је предложило различите формализације решавања случаја фазификације као на пример Carlsson и Korhonen (видети детаљније у Lai, Y. et al., 1992). У овом раду само дискутујемо детаљно о методи коју су предложили Ribeiro и Moura-Pires (1996, 1998, 1999) зато што је то флексибилни приступ који се може бавити једноструким или вишеструким циљевима као и линеарним или нелинеарним проблемима програмирања.

Ribeiro и Moura-Pires (1996, 1998, 1999) предложили су методу за проблеме фази оптимизације који обухватају фази коефицијенте или у функцији циља или у ограничењима. Цео процес се означава као „флексибилни приступ“ за проблеме фази оптимизације.

Дефинисана формулација проблема фази оптимизације у (2. 3)-(2. 5) трансформише се у следећи истем нелинераних једначина

$$\min_{x} \tilde{Z} = c^T x, \quad (2.\ 6)$$

$$maxM = \cap \left(\mu_0(c^T x), \mu_i(A_i x), \mu_k(A_k x)\right), \quad (2.\ 7)$$

где су ограничења:

$$c^T x \leq Z_0 + \left(1 - \mu_0(c^T x)\right) p_0, \quad (2.\ \ 8)$$

$$A_i x \leq b_i + \left(1 - \mu_i(A_i x)\right) p_i \ (i = 1, \dots, m), \quad (2.\ 9)$$

$$A_k x \geq b_k - \left(1 - \mu_k(A_k x)\right) p_k \ (k = 1, \dots, n), \quad (2.10)$$

$$\mu_0(c^T x), \mu_i(A_i x), \mu_k(A_k x) \in [0, 1], \quad (2\ .11)$$

$$x \geq 0. \quad (2.\ 12)$$

$\cap$ представља пресек свих вредности функција чланства. Други циљ представља максимизацију пресека свих вредности чланства, тј. овај циљ даје најбољу вредност минимума (пресек) нарушавања ограничења.

Функције чланства које се користе у овом раду су троугаоне функције. Са овим типом функција можемо лако представити било који фазификовани параметар проблема оптимизације. Узевши у обзир девијацију p_i ради флексибилизације, три типа функција који се најчешће користе ради представљања ограничења типа $\leq, \geq$ или $=$ су:

$$\mu_i(A_i x) = \begin{cases} 1, & A_i x \leq b_i, \\ 1 - \frac{A_i x - b_i}{p_i}, & b_i < A_i x \leq b_i + p_i, \\ 0, & A_i x > b_i - p_i. \end{cases} \quad (2.\ 13)$$

$$\mu_i(A_i x) = \begin{cases} 1, & A_i x \geq b_i, \\ 1 + \frac{A_i x - b_i}{p_i}, & b_i - p_i \leq A_i x < b_i, \\ 0, & A_i x < b_i - p_i \end{cases} \quad (2.\ 14)$$

$$\mu_0(c^T x) = \begin{cases} 1, & c^T x < Z_0, \\ 1 + \frac{Z_0 - c^T x}{p_0}, & Z_0 \le c^T x \le Z_0 + p_0, \\ 0 & c^T x > Z_0 + p_0. \end{cases} \quad (2.\ 15)$$

Функција (2. 13) одговара функцији чланства за ограничења мање или једнако; функција (2. 14) је функција чланства за ограничења типа веће или једнако, а функција (2. 15) је функција чланства за функцију циља. Код Зимермановог метода (Zimmermann, H., (1991)) само функције веће или једнако од или мање или једнако од се користе да би се дефинисали или представили било циљеви функције циља, било ограничења.

Метод решења предложен у овом приступу се ослања на два фундаментална корака:

- наћи најбоље вредности за x_i, a_{ij} и c_j, независне променљиве које максимизирају минималну вредност (M) узевши у обзир вредност прага који је претходно дефинисан од стране доносиоца одлуке као минимално прихватљиво нарушавање ограничавања,
- наћи оптималну вредност $\tilde{Z}$ која задовољава сва ограничења исто као и код првог корака.

Можемо рећи да овај модел фазификације обезбеђује однос између задовољавања ограничења и вредности постизања циља. Даље, овај метод дозвољава манипулисање било ког типа линеарног или нелинеарног проблема фази оптимизације. Друга предност овог метода је могућност решавања традиционалног проблема који због тежине проблема не могу дати решење укључујући постојање непрецизности коефицијената или параметара.

2. 5. Линеарни проблем оптимизације

Стандардни облик математичког модела линеарног проблема за налажење минимума (или максимума) линеарне функције је:

$$f(x_1, x_2, \dots, x_n) = c_1x_1 + c_2x_2 + \cdots + c_nx_n, (c_1, c_2, \dots, c_n \in \mathbb{R}), \qquad (2.\ 16)$$

под ограничењима која представљају скуп решења неког система линеарних неједначина и једначина:

$$\begin{aligned} a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n &\geq b_i, \quad i = 1,2, \dots, m, \\ d_{j1}x_1 + d_{j2}x_2 + \cdots + d_{jn}x_n &= e_j, \quad j = 1, 2, \dots, k. \end{aligned} \qquad (2.\ 17)$$

У случају минимизације проблем линеарног програмирања (LP) симболички записујемо као

(LP) $(min)\ f$ при ограничењима (2. 17).

Решења система (2. 17) зову се допустива решења проблема. Скуп свих допустивих решења зовемо кратко допустив скуп.

Функција f је функција циља. Њен минимум на допустивом скупу, зове се оптимална вредност.

За више детаља о теорији оптималности, као и неким њеним применама погледати (Dugošija, 2011).

У овом раду за решавање овог проблема је коришћена симплекс метода. Она представља једну од основних метода за решавање проблема линераног програмирања (опширније погледати у John C. Nash (2000) као и у Dugošija (2011)).

ТРЕЋЕ ПОГЛАВЉЕ: МЕТАХЕУРИСТИЧКЕ МЕТОДЕ

У овом поглављу представљене су укратко методе за решавање оптимизационих проблема, посебно хеуристике и метахеуристике. Након тога детаљније су описане метахеуристичке методе које се користе и развијају за решавање оптимизационих проблема разматраних у овој дисертацији. Неки од постојећих алгоритама се користе без икаквих модификација, неки алгоритми се мењају у сврху побољшања. У овом поглављу ћемо представити и објаснити коришћење алгоритама који су коришћени у истраживању.

Хеуристички приступ решавању проблема обухвата приближне методе за налажење „задовољавајућих“ решења проблема (енг. soft computing) код којих „тврде“ (егзактне) квантитативне методе не могу да нађу оптимално решење.

Појам хеуристике потиче од грчке речи *heuriskein* што значи наћи, открити. Познато је оно чувено Архимедово „еурека“ (пронашао сам) када је открио главни закон хидростатике.

Дефиниција 3. 1. Хеуристика је техника која тражи добра (тзв. субоптимална) решења у разумном времену извршавања без гаранције да ће нађена решења бити оптимална и без знања о блискости ових решења оптималном решењу.

Под „добрим“ решењима подразумевамо да се може очекивати да добијена решења буду блиска оптималном. Док под „разумним временом“ се подразумева да метода буде полиномијалне рачунске сложености.

Хеуристике се интензивно развијају, нарочито у последњих 40ак година са развојем комбинаторне оптимизације, теорије алгоритама, а посебно рачунара. Широку примену нашле су у решавању, како стандардних теоријских проблема, тако и великог броја тешких реалних проблема комбинаторне и глобалне оптимизације (некада је хеуристика једини начин да се реше овакви проблеми). Хеуристике често полазе од реланијег модела проблема, него егзактне методе, јер не захтевају строги математички модел.

Зашто и када се употребљавају хеуристике?

Хеуристике се употребљавају за решавање слабо структуираних проблема за које не постоје егзактни алгоритми. Код ових проблема:

- постоје елементи неизвесности, неодређености и субјективне процене,
- постоје елементи нелинеарности и појава више критеријума, итд.

Пример оваквих проблема су:

- проблеми распоређивања послова на машине, распоређивање возила, распоред часова, итд. (већина реалних проблема).
- Такође се употребљавају за решавање проблема за које постоје егзактни алгоритми, али нису ефикасни за велике димензије проблема.
- За NP тешке проблеме код којих је мала вероватноћа да постоје егзактни алгоритми полиномијалне сложености који их решавају.
  - У ове проблеме спадају проблеми нелинеарног програмирања, проблем целобројног програмирања, трговачког путника, бојења графова, проблем ранца, проблем разбијања (покривања) скупа, итд.

Хеуристика се употребљава за тражење „задовољавајућег" решења проблема у случају ограниченог времена, простора и новчаних средстава.

Хеуристике се деле на (Sorensen, K., 2012):

- Конструктивне: генеришу само једно допустиво решење проблема.
- Методе локалног претраживања: генеришу низ допустивих решења, где је свако следеће решење добијено „побољшавањем" претходног по неком локалном критеријуму.
- Еволутивне методе: у свакој итерацији генеришу „популацију" допустивих решења, где је свака следећа популација „боља" од претходне, итд.
- Специјалне: поштују својства и специфичности проблема.
- Опште или метахеуристике: опште методологије новијег датума (од 1985. године). Већина њих симулира спонтане оптимизационе процесе у физичким или биолошким системима. Неке од њих су:
  - Симулирано каљење (Simulating anniling) - симулира термодинамичке процесе каљења.

- Табу претраживање (Tabu search) – имитира логику „интелигентног“ људског претраживања са памћењем.
- Метода променљивих околина (Variable neighborhood search) – током процеса претраживања мења структуру околине у којој тражи решење.
- Процедура адаптивног похлепног стохастичког претраживања (GRASP - Greedy Randomized Adaptive Search Procedure) - Састоји се у поновљеним применама адаптивних стохастичких конструктивних хеуристика за налажење почетних решења која се затим поправљају локалним претраживањем.
- Генетски алгоритам (Genetic algorithms) – симулира процесе природне селекције врста.
- Мрављи алгоритми, пчелињи алгоритми, алгоритми оптимизације ројем честица, гравитациониа алгоритми, алгоритми засновани на принципима електромагнетизма, и многи други (Schneider, J. and Kirkpatrick, S., 2007.; Talbi, E.G., 2009.; Luke, S., 2009.; Rothlauf, F., 2011.; Xing, B. and Gao, W.J., 2014.).

Добар хеуристички алгоритам треба да поседује следећа својства (Martini, R., et al., (2011)): решења треба да се генеришу у разумном временском периоду, хеуристика треба да произведе решење које је блиско оптималном са великом вероватноћом и вероватноћа добијеног лошег решења би требала да буде минимална.

Напор да се робусност хеуристичких метода и алгоритама за велики број проблема унапреди и развије резултирао је настанак метахеуристика. Основни принципи ових алгоритама базирају се на општим алгоритмима оптимизације који користе итеративне механизме да би поправили постојеће решење.

Постоји велики број критеријума за класификацију метахеуристика. Према подели (Talbi, E.-G., 2009), метахеуристике се деле на:

- детерминистичке (енг. deterministic) и стохастичке (енг. stohastic) метахеуристике. Детерминистичке методе манипулишу решењима проблема оптимизације тако што доносе детерминистичке одлуке (примери су табу претраживање и метода променљивог спуста). То значи да иста почетна решења увек генеришу исто коначно решење. Стохастичке методе користе случајност у процесу претраге (примери су симулирано каљење и генетски алгоритам). Случајност за последицу има да стохатичке методе могу да генеришу различита коначна решења помоћу истих почетних. Зато се код ових метода увек врши понављање извршавања и испитује њихова стабилност под различитим почетним условима.

- метахеуристике које су базиране на популацији решења (енг. population-based) и метахеристике које користе само једно потенцијално решење (енг. single solution based). Метахеуристике које раде са популацијом решења претрагу врше трансформацијом текуће популације у нову која садржи потенцијално боља решења. У ову групу спадају генетски алгоритам, оптимизација колонијом мрава, диференцијална еволуција и многе друге методе. Оне су најчешће инспирисане природом и бројније су у односу на методе које користе само једно решење. У другу групу спадају методе које процес претраге заснивају на једном, тренутно најбољем решењу. Његовим трансформацијама и локалним побољшањима трага се за жељеним оптимумом. Примери ових метода су табу претраживање, симулирано каљење, метода променљивих околина.

- метахеуристике које користе меморију, тј. које имају могућност памћења претходних решења (енг. memory usage methods) и на оне које не користе меморију, тј. које немају могућност памћења претходних решења (енг. memoryless methods). Алгоритми који не користе меморију немају могућност динамичког естраховања информација током процеса претраге. Један од представника ове врсте је симулираног каљења. С друге стране, многи алгоритми користе систем online екстракције знања током претраге која се складишти у меморији, чему служе на пример краткорочна и дугорочна меморија табу претраге;

- природом инспирисане метахеуристике (енг. nature-inspired) и на оне које нису природом инспирисане (енг. non nature inspired). Настанак многих метахеуристичких метода је инспирисан природним процесима, као што су на пример еволутивни алгоритми и вештачки имуни системи. Такође су и многи системи организама у природи послужили или као инспирација за развој ових метода, као што су на пример колоније мрава и пчела, јата птица и риба, итд. Осим наведеног, у литератури се проналазе и метахеуристике које су инспирсане физичким процесима, као што је метахеуристика симулираног каљења. Насупрот методама инспирисаним природним процесима постоје метахеуристике које се заснивају на математичким принципима. У основи њихове имплементације налази се метрика којом се дефинише растојање између различитих допустивих решења разматраног проблема. Ове методе најчешће користе локлано претраживање за поправљање текућег решења.

Методе оптимизације углавном користе једну структуру околине, али постоје и алгоритми, као што је метода променљивих околина, која користи скуп структура околине.

Детаљније се описују три методе које се примењују у овој дисертацији и то: симулирано каљење, генетски алгоритам и метода променљивих околина.

3. 1. Симулирано каљење

Симулирано каљење (Metropolis, 1953; van Laarhoven, 1987) базира се на симулирању хлађења растопљеног материјала. Материјал се брзо загрева, а затим се постепено хлади до чврстог стања. Ако се довољно полако хлади, материјал ће на крају достићи правилну кристалну структуру, тј. стање минималне енергије.

Простор допустивих решења се претражује на случајан начин коришћењем принципа локалног претраживања. При томе се контролисано дозвољава прелазак у „гора“ решења да би се избегле замке локалних минимума.

Вероватноћа прихватања горих решења се смањује када се повећава разлика између квалитета тренутног и наредног решења и смањује параметар температуре. У почетку претраживања често се прихватају гора решења, што омогућава ширу претрагу по допустивом скупу. Међутим, када је температура довољно мала, прихватају се само боља решења и претраживање се локализује на једну област.

Симулирано каљење је техника претраге која припада општој класи случајних техника претраге (Filho et al., 1994), које су засноване на техникама набрајања, али користе додатне информације ради вођења претраге. Технике рандом претраге су прилично уопштене и могу решавати сложене проблеме. Овај тип алгоритама има значајну улогу међу техникама локалне претраге због два главна разлога (Aarts et al. 1997):

- прво зато што су се показали успешним када су примењивани на великом броју практичних проблема;
- друго, алгоритми симулираног каљења имају стохастичку компоненту која олакшава теоретску анализу асимптотске конвергенције и чини их прилично популарним међу математичарима.

Циљ неког алгоритма овакве природе је да се пронађе најбоље решење међу коначним бројем могућих решења. Техника симулираног каљења је нарочито привлачна зато што дозвољава проналажење готово оптималних решења уз разуман рачунски напор. Код овог алгоритма није могуће знати да ли је најбоље пронађено решење заиста глобално оптимално. Ова карактеристика ограничава његову употребу на случајеве где није потребно бити сигуран да је решење глобално оптимално (Blazewics et al., 1996; Varela et. al., 2002).

Код процедуре симулираног каљења, низ решења нема линеарни тренд локалног оптимума, као што се догађа код других локалних техника претраге. Обрнуто, можемо потврдити (верификовати) да решења утврђују ток променљивих кроз скуп могућих решења, и он има тренд вођења у „повољном“ правцу.

Можемо рећи да је алгоритам симулираног каљења добра процедура коришћења у ситуацијама у којима је знање непрецизно. Чак и за сложене проблеме, ова техника је релативно лака за примену и обично спроводи процедуру успињања са вишеструким рестартовањима.

Као што је претходно споменуто, алгоритам симулираног каљења је алгоритам који се користи ради решавања проблема оптимизације, где функција циља одговара енергији чврстог тела. Код алгоритма симулираног каљења постоји параметар који означава температуру и допушта разликовање између већих или мањих промена (измена) у функцији циља. Драстичне измене се дешавају при високим температурама, а мале или благе модификације при ниским температурама. То је „еволуциони“ процес који се помера малим корацима (полако), из једног стања у друго, и ту постоји проблем заглављивања методе у локланом оптимуму (Ribeiro et al. 1999). Општи псеудо код ове методе је:

$Let\ s\ =\ s_0;$

$For\ (k\ =\ 0;\ k<\ k_{max}\ ;\ k++)\{$

$\quad T\ =\ temperature(k\ /\ k_{max});$

$\quad s_{new}\ = neighbour(s)\ (pick\ a\ random\ neighbor);$

$\quad If\ P(E(s), E(s_{new}), T)\ \geq\ random(0,1)\ then$

$\quad\quad s\ = s_{new};$

$\}$

$Output\text{: } the\ final\ state\ s.$

s_0 је почетно стање. Циљ је пронаћи стање s, у k_{max} корака које ће имати мању енергију од почетног стања ($E(s)$).

3. 2. Генетски алгоритам (GA)

Еволуционо рачунање је уведено 1960. од стране Rechenberg, I., у раду "Evolution strategies". Ова идеја је касније унапређена истраживањима и коначно, генетски алгоритам је предложен од стране John Holland и развој ове идеје налази се у његовој књизи "Adaptation in natural and artificial systems" из 1975. Holland је предложио генетски алгоритам као хеуристичку методу засновану на принципу „опстанак најздравијих". GA је откривен као корисно средство за претраживања и решавања оптимизационих проблема.

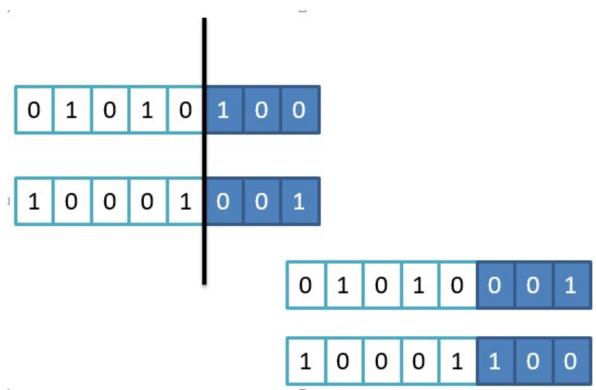
GA је техника претраге која се користи у проблемима оптимизације за проналажење тачних или решења блиских тачним. GA тражи глобални оптимум преко алгоритма заснованог на мејози. Почетна генерација се случајно генерише и нови решења се добијају укрштањем решења из тренутне генерације. Генетске разлике се формирају мутацијом гена и преносе се (шире) оператором природне селекције.

Као што само име каже, сваки кандидат решења у GA је представљен генетским кодом који се назива хромозом. Генетски код може бити представљен целобројно, у покретном зарезу, бинарно или хексадецимално. Обично се представља бинарно, јер тај запис има предност у односу на остале због оператора укрштања и мутације.

Први корак GA је генерисање почетне популације. Одређивање величине популације има велики значај, јер мале популације садрже ризик немогућности покривања простора решења, док велике популације носе ризик сложеног рачуна. Goldberg указује да оптимална величина бинарно кодираног стринга расте експоненцијално са повећањем дужине стринга *n* (Reeves, 1995).

Генерација може да се образује случајно или да настане модификацијама претходне генерације. Друга техника омогућава GA да пронађе боља решења на ефикаснији начин него што је то код настанка генерације на случајан начин. Међутим, у овом случају постоји могућност преране конвергенције ка субоптималним решењима (Revees, 1992; Kapsalis et al., 1993).

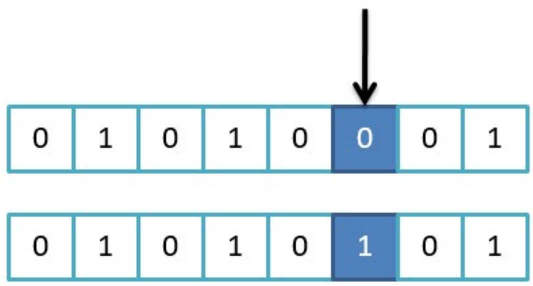
Оператор укрштања (наслеђивања) је неопходан за генетску репродукцију. Нови гени се репродукују из случајно одабраних гена (Слика 3. 1.). Укрштање се примењује у складу са вредностима функције прилагођености. Стандардни приступ сматра да ће боље прилагођене јединке пренети добар генетски материјал на своје потомке, па смањује шансу проласка лоших јединки у наредну генерацију, тако да оне постепено нестају из популације.



Слика 3. 1. Оператор укрштања (наслеђивања).

Eshelman et al. (1989), радио је на вишепозиционом укрштању и испитивао ефекат пристрасности са традиционалним једнопозиционим укрштањем и посматрао низ могућности. Централни аргумент је био да постоје два извора пристрасности која треба истражити у генетским алгоритмима: позиција пристрасности и расподела пристрасности. Eshelman је закључио да једноставно укрштање има значајну позицију пристрасности и да пристрасност може довести до добијања решења која нису добра. Поред тога, оператор укрштања се детаљније проучава од стране Faily (1991).

Код оригиналног GA Holland's, родитељи се бирају стохастичким процедурама из популације решења и комплетно нова популација потомства се генерише да замени своје родитеље. У другој верзији, он је предложио да би сваки потомак требало да замени случајно изабране чланове тренутне популације.

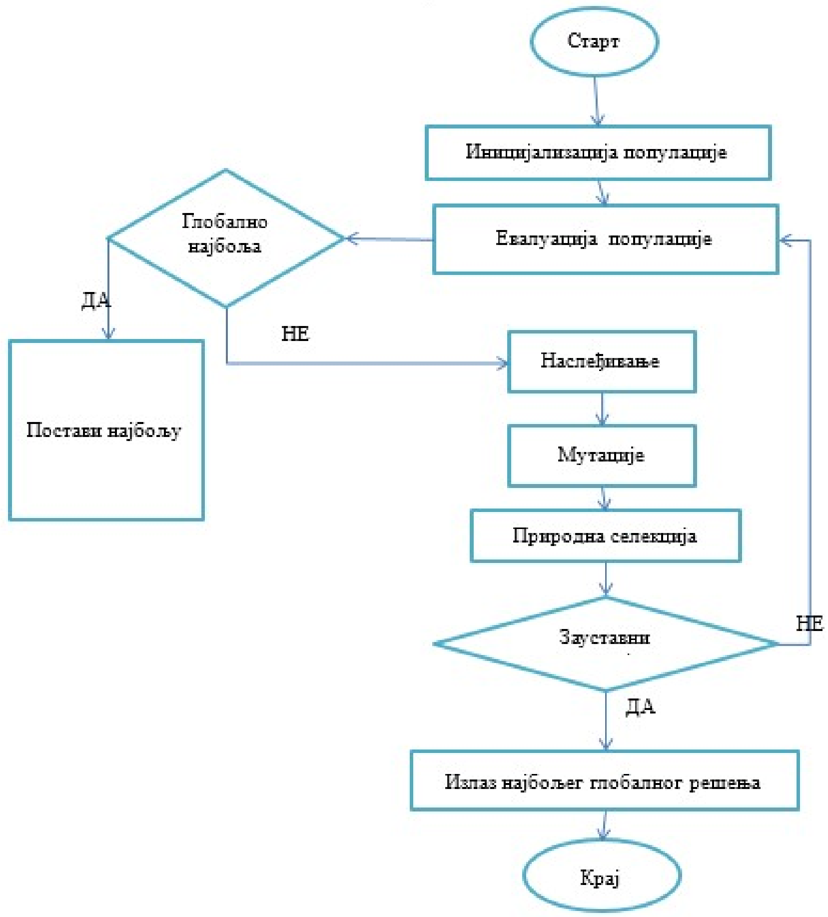


Слика 3. 2. Оператор мутације.

Мутација (слика 3. 2.) спречава доминацију одређеног гена који има велику вероватноћу опстанка. Оператор укрштања може да произведе добро уклопљене гене од постојећих гена, али не може да генерише нове гене за одређени део који не постоји у популацији. Дакле, оператор мутације има велики значај јер може да произведе нове комбинације гена које нису настале у иницијализацији популације или да регенерише одређену комбинацију гена елиминисану природном селекцијом.

Оператор укрштања је значајнији у почетку када је популација разноврсна, али како се претрага приближава неком локално оптималном решењу постаје важније да се повећава шанса за генерисањем разноврснијих решења, где су мутације ефикаснији оператор.

Природна селекција је последњи корак циклуса генетског алгортима. Она је у директној вези са функцијом прилагођености. Користи дистрибуцију у којој је вероватноћа селекције јединке пропорционална њеној прилагођености. Поред тога, побољшава општи квалитет гена популације раскидом лоше упарених гена. С друге стране, овакви гени некада могу да носе веома важне информације на одређеној локацији у циљу очувања ових делова и превентивне иницијализације добрих гена створених за доминацију, зато се неке мере предузимају у фази природне селекције. Правило је да се додели велика вероватноћа раскида када се упарује са лошим генима и мала вероватноћа раскида када су у питању добри гени.



Слика 3. 3. Ток генетског алгоритма.

Једна од евидентних мана оператора једноставне природне селекције је то што не даје гаранције да ће најбољи члан популације преживети у следећој генерацији. Овим проблемом се бавио De Jong's (1975) чији модел је превазишао овај проблем тако што је у њему најбољи члан тренутне популације приморан да буде члан следеће. Други приступ је дат од стане Ackley (1987) где је уведен термин „прекид са предрасудама" у алгоритму. Рачунање прилагођености је од великог значаја, јер проблем може настати на почетку генетског алгоритма када је број лоших јединки велики. Мера прилагођености може довести до брзог опстајања добрих и тиме до преране конвергенције ка локалном оптимуму која постаје неизбежна. Један од приступа који помаже да се ово спречи је игнорисање актуелне функције циља и коришћење процедуре рангирања. Главни аргументи овоме су да је кључ добрих перформанси генетског алгоритма одржавање одговарајућег нивоа селекције путем одговарајуће мере релативне прилагођености.

Оператор природне селекције представља крај једаног циклуса GA. Број генерисаних циклуса зависи од броја улазних параметара и од очекиваног смањења укупних трошкова пројекта. Ток дијаграма GA дат је на слици 3. 3.

Најчешће коришћени начини рачунања функције прилагођености су: директно преузимање, линеарно скалирање, скалирање у јединични интервал и сигма одсецање. Избор одговарајућег начина највише зависи од специфичности проблема, а често је неопходно комбиновање различитих принципа. У предложеном алгоритму GA ове докторске дисертације коришћена је функција прилагођености која јединке скалира.

Функција прилагођености: Јединка се скалира у интервал [0, 1] тако да прилагођеност најбоље јединке буде 1, а најлошије 0. Пошто се тражи минимизација функције циља, коришћена је функција тзв. инверзног скалирања која вредност јединке скалира у јединични интервал

$f(x) = \frac{x_{max}-x}{x_{max}-x_{min}}$,

где су x, x_{max} и x_{min} редом текућа, максимална и минимална вредност јединки у популацији.

GA је ефектан у тражењу оптималних (или близу оптималних) решења проблема где:

- Простор претраге је велики или сложен.
- Нема доступних математичких анализа.
- Традиционални метод претраге „не функционише“.
- Проблем решавања је NP тежак.

3. 3. Метода промењивих околина

Метода промељивих околина је метахеуристика која је представљена деведесетих година прошлог века (Mladenović, N., 1995, Mladenović, N., et al., 1997) након чега је доживела много промена и екстензија (Mladenović, N., et al., 2005; Mladenović, N., et al., 2007). VNS метахеуристика заснована је на три основне чињенице:

1. Локални минимум у односу на једну околину не мора бити и локални минимум у односу на неку другу околину.
2. Глобални минимум је локални минимум у односу на све околине.
3. За већину проблема локални минимум у односу на разне околине су међусобно блиски.

Систематска претрага околина се врши све док се не задовољи неки критеријум заустављања. Најчешће коришћени критеријуми заустављања су: максимално процесорско време рада, максималан број итерација, минимално побољшање у оквиру унапред дефинисаног броја итерација, максималан број итерација између два побољшања и други. Генерално, избор критеријума заустављања би требало да обезбеди компромис између квалитета добијених решења и утрошеног времена.

У литератури се може наћи више варијанти методе промењивих околина:

- Основна метода промењивих околина (енг. Basic VNS - BVNS).
- Метода промењивог спуста (енг. Variable Neighborhood Descent – VND).
- Општа метода промењивих околина (енг. General VNS – GVNS).
- Редукована метода промењивих околина (енг. Reduced VNS – RVNS).
- Адаптивна метода промењивих околина (енг. Skewed VNS – SVNS).
- Метода промењивих околина на бази декомпозиције (енг. Variable Neighborhood Decomposition Search – VNDS).
- Примално - дуална метода промењивих околина (енг. Primal – dual VNS – PD-VNS).
- Паралелна метода промењивих околина (енг. Parallel VNS – PVNS).

- Метода променљивих формулација (енг. Variable Neighborhood Formulation Space Search – VNFSS),

и друге.

Основна метода променљивих околина је најраспрострањенија варијанта методе променљивих околина јер обезбеђује више предуслова за добијање квалитетнијих коначних решења. Код основне VNS методе основни кораци садржани су у петљи у којој мењамо индекс околине k, одређујемо случајно решење из те околине извршавамо процедуру локалног претраживања и проверавамо квалитет добијеног локалног минимума. Ове кораке понављамо док не буде задовољен неки од критеријума заустављања. Приликом сваког одабира околине k почетна решења генеришемо на случајан начин како би обезбедили претраживање различитих региона приликом следећег размрдавања околине k (Davidović, T., 2006). Околине разликујемо по броју трансформација (растојању) или по врсти трансформација (метрици).

Напомена: Околине за избор случајног решења (размрдавање) и локално пртражживање не морају бити истог типа.

Псеудокодом BVNS се може представити на следећи начин:

```
Generate initial solution x;
do{
        k = 1;
        while (k ≤ kmax){
                x' = Shaking(x,k);
                x'' = Local search(x');
                if (f(x'') < f(x)) {
                        x = x'';
                        k = 1;
                }
                else
                        k = k + 1;
        }
} while (! STOP);
```

Метода променљивог спуста је детерминистичка варијанта VNS методе која користи различите типове околина у процесу претраге. У овом случају k_{max} (максимални број околина) представља број различитих околина у којима се покушава поправка неког почетног решења x. За побољшање се користи процедура локалне претраге (енг. Local Search (LS)) у односу на сваку околину. Уколико је

$k_{max} = 1$, реч је о обичном локалном претраживању. Записана у псеудокоду VND има следећи облик:

```
Generate initial solution x.
 x_opt = x;
 f_opt = f(x);
do {
        k = 1;
        do{
                x' = LS(x, k);
                if(f(x') < f( x_opt))){
                        x_opt = x';
                        f( x_opt) = f(x');
                        k = 1;
                }
                else
                        k = k + 1;
        }while(k ≤ k_max);
}while(there are improvements);
```

Општа метода промењљивих околина користи методу промењљивог спуста у фази локалног претраживања. С тога је потребно дефинисати два скупа околина: околине које се користе у фази размрдавања и околине које ће бити претраживане у оквиру методе промењљивог спуста. Могуће је да се исте структуре околина користе у обе фазе али је то ретко зато што се у фази размрдавања најчешће користи једна околина али се повећава растојање. Псеудокод ове методе има следећо облик:

```
Generate initial solution x;
do{
        k = 1;
        while(k ≤ k_max){
                x' = Shaking(x, k);
                x'' = VND(x');
                if(f(x'') < f(x))){
                        x = x'';
                        k = 1;
                }
                else
                        k = k + 1;
        }
}while(stopping criterion satisfied);
```

Редукована метода промењљивих околина не садржи фазу локалне претраге већ је заснована само на размрдавању које се састоји у систематској промени околина и избору једног случајног решења у свакој од околина. Кораци одлучивања базирани су на том једном случајном решењу. Њена предност је брзина извршавања јер се избегава детаљна претрага, често изузетно великих околина. Ова метода је корисна код примера великих димензија или за добијање квалитетних почетних решења за неку другу варијанту VNS методе.

Записана у псеудокоду RVNS има следећи облик:

```
Generate initial solution x;
x_optimal = x;
f_optimal = f(x);
do {
        k = 1;
        do {
        x' = Shaking(x, k);
                if (f(x') < f(x_optimal)) {
                        x_optimal = x';
                        f(x_optimal) = f(x');
                        k = 1;
                }
                else
                        k = k + 1;
        } while (k! = k_max);
} while (! STOP);
```

Уобичајени критеријум заустављања (*STOP*) је максимални број итерација између два побољшања.

ЧЕТВРТО ПОГЛАВЉЕ: ПРОБЛЕМ ПЛАНИРАЊА И РАСПОРЕЂИВАЊА КОРИСТ/ТРОШАК РЕСУРСА ПРОЈЕКТА

У овом поглављу објашњене су основе управљања пројектима, однос корист/трошак и ограничења ресурса пројекта. Укратко је објашњена добро позната метода критичног пута која се користи за распоређивање пројекта.

4. 1. Критични пут (Critical Path Method CPM)

Техника проналажења критичног пута се користи од 1950. године. И тада је највише нашла употребу у грађевинској индустрији за потребе планирања и контроле пројеката, када је у питању планирање комуникације и за обуку менаџера. Нове верзије методе критичног пута су одрађене софтверски тако да олакшају практично коришћење ове методе што доводи до повећања ефикасности управљања пројектима.

Критични пут почива на чврстој процени времена трајања активности које су у саставу анализираног пројекта. Заправо, он представља пут који је састављен од критичних догађаја. Под критичним догађајима сматрамо догађаје код којих кашњење било које активности резултира кашњењем целог пројекта.

Планирање и распоређивање активности пројекта је значајно јер време и трошак сваке активности су углавном прецино познати тек непосредно пре почетка пројекта. Предности коришћења критичног пута могу се укратко објаснити (Suhanic 2001) као:

CPM указује на активности чија трајања (почетак и крај активности) су одговорна тј. Директно утичу на трајање читавог пројекта. Идентификацијом критичних активности обраћа се већа пажња на њихово распоређивање и оптимизацију.

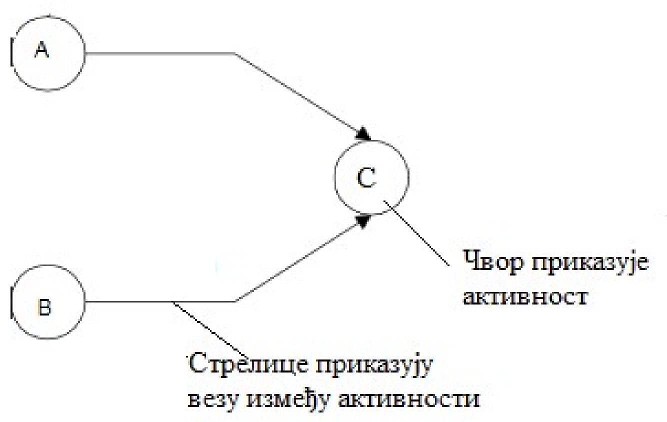
CPM даје квантитативну процену сваке активности. Добија се процена најкраћег и најдуужег времена трајања активности, што нам омогућава нивелисање ресурса.

CPM приказује најекономичније распоређивање сваке активности за свако могуће време завршетка пројекта. Ово омогућава разматрање времена (користи) и трошкова пројекта.

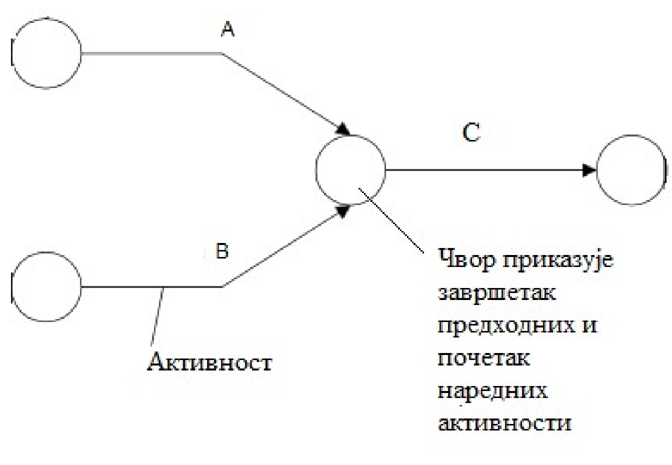
CPM обезбеђује неопходне податке за избор најбољег времена завршетка пројекта који се користи за даље анализе.

Омогућава ефективно праћење промена и начина извршавања активности.

Постоје две реализације мреже критичног пута: активности које приказујемо стрелицама и активности које приказујемо преко чворова (слика 4. 1.и слика 4. 2.).



Слика 4. 1. Мрежни дијаграм са активностима у чворовима.



Слика 4. 2. Мрежни дијаграм са активностима на гранама (стрелицама).

Саставни део критичног пута је математички модел који инкорпорише информације, трајање и трошкове за сваку компоненту пројекта. Анализа решења овог модела омогућава оперативном особљу да одговоре на питања у вези потреба за радом, захтевима буџета, ограничења дизајна, утицаја кашњења и тешкоћа приликом комуникације.

Током протеклих година, дошло је до повећаног интересовања за проблеме распоређивања (управљања) великим пројектима. Литература у вези ове теме се протеже од проблематичних подручија до математичких и статистичких анализа неке префињености. Већина ових каснијих анализа се бави примарно откривањем и истраживањем структуралних карактеристика пројекта са циљем прављења планова, распореда и предвиђања.

Математички модел на којем је заснован метод критичног пута је параметарски линеарни проблем који има за циљ рачунање употребљивости пројекта као функције његовог трајања. За свако могуће трајање пројекта, се добија могући распоред који има максималну употребљивост међу свим распоредима истог трајања пројекта.

У овој дисертацији, за распоређивање активности је коришћен дијаграм активности представљен преко чворова. Разлог за то је лакша констукција овог дијаграма него дијаграма активности представљених стрелицама и чињеница да се мање података захтева од корисника. Поред тога, овим дијаграмом се погодније дефинишу односи међу активностима.

Постоје три фазе анализе методе критичног пута:

1. Утврђивање најранијег почетка и најранијег завршетка активности.
2. Утврђивање најкаснијег почетка и најкаснијег завршетка кативности.
3. Одређивање „временских резерви“ и критичног пута пројекта.

Терминологија која се користи приликом проналажења критичног пута (Suhanic, 2001) изложена је у наставку овог одељка:

Активност: Прецизно дефинисан задатак. У спецификацији активности имамо број активности, приказ особина сваке од њих, тајање и тип. Осим ових спецификација, могу се описати и трошкови и ресурси.

Критични пут: Најдужи део мреже или ланац или низ активности за које треба највише времена или најдужа несводљива секвенца догађаја. Критични пут одређује трајање пројекта.

Трајање: Период изражен у временској јединици (радним данима) између почетка и краја једне активности.

Најраније време почетка НВП (Early Start Date (ES)): Најраније време када се може отпочети са активношћу (до којег се долази на основу логичких односа између активности и њених претходника).

Најраније време завршетка НВЗ (Early Finish Date (EF)): Најраније време могућег завршетка активности на основу логичких односа између посматране активности и њених претходника.

Кашњење почетка активности НКП (Late Start Date (LS)): Израчунато најкасније време почетка активности која се налази на путу кашњења или од ње креће кашњење. Време кашњења почетка је дозовљени најкаснији почетак активности тако да време трајања целог пројекта не буде промењено (да не доводи до кашњења пројекта).

Кашњење краја активности НКЗ (Late Finish (LF)): Израчунато кашњење завршетка активности која се налази на путу кашњења или од ње креће кашњење. Кашњење краја активности представља дозвољено ново време завршетка активности тако да се не наруши трајање пројека (време трајања пројекта остаје исто као и пре кашњења).

Крај краја (Finish to Finish (FF)): Однос у мрежи где је крај претходне активности услов за крај наредне активности.

Заврши да би почео (Finish to Start (FS)): Однос у мрежи где се претходна активност мора завршити пре него што следећа активност почне.

Почетак краја (Start to Finish (SF)): Однос у мрежи при чему нека активност може да се заврши само ако је почела претходна активност.

Почетак почетка (Start to Start (SS)): Однос у мрежи где активност може да почне само ако је њена претходна активност почела.

Укупан ход (Total Float): Количина времена (простора) коју активност има на располагању да не дође до негативног утицаја на критични пут.

Слободни ход (Free Float): Количина простора којом активност располаже пре него почне негативно да утиче на критичан пут.

Ход унапред (Forward Pass): Рачуна се најранији почетак и крај свих активности. Најдужа секвенца активности одређује критичан пут и време завршетка пројекта.

Ход уназад (Backward Pass): Израчунава се уназад кашњење почетка и кашњење завршетка за све активности пројекта и на основу тога се рачуна почетак пројекта. Активности које имају слична времена превременог или кашњења почетка или завршетка су на критичном путу. Ове активности се називају критичним активностима.

Временске резерве (P)

Активност пројекта (A)

Трајање одређене активности (T)

4. 2. Формулација проблема

Нека је E коначни делимично уређени скуп од n елемента који се називају догађаји. Означимо са $\bar{E} = E \cup \{S, T\}$, где S представља почетни догађај који предходи (не нужно непосредно) свим догађајима из E, а T је крајњи догађај коме сви догађаји из E предходе. Ово су фиктивно уведени догађаји ради лакше манипулације и њихове тежине (дужине) су нула.

Сваки догађај из E карактерише ненегативни број, ознака. Како је E делимично уређен, можемо претпоставити да су догађаји означени тако да за сваки догађај i који претходи догађају j ту чињеницу можемо записати као $i<j$. Конкретно, почетак је обележен са 0, а крај са $n + 1$.

Такође повезан са догађајем i је ненегативни број, t_i, који представља трајање догађаја i. Према томе, ако догађају j следи догађај i онда $t_i \leq t_j$. Увек ћемо поставити $t_o = 0$.

Активност је елемент $(i,\ j)$, скупа ExE, тако да је $i < j$. Свака активност је повезана ненегативним бројем, y_{ij}, својим трајањем. Претпоставља се да се активност мора обавити између појаве догађаја i и појаве догађаја j. С тога мора да важи:

$$y_{ij} + t_i - t_j \leq 0. \tag{4. 1}$$

Пројекат, P, је скуп догађаја и активности са особином да ако је догађајима κ у P онда је или почетни или крајњи догађај или у супротном постоје догађаји i и j у P такви да су обе активности (i, κ) и (κ, j) у P.

Додела трајања, y_{ij}, активностима и времена дешавања, t_i, догађајима у P назива се распоред. Распоред ће бити означен са $\{y, t\}$, где су y и t вектори чије су координате, y_{ij} и t_i, односно, времена која одређују распоред. Ако постоји m активности у P, $\{y, t\}$ може бити представљен као $(m + n + 2)$-димензиони вектор Еуклидског простора.

Понекад је трајање активности одлука менаџмента која подлеже одређеним ограничењима. Најједноставнија ограничења су да y_{ij} мора бити између доње и горње границе (ограничења) за сваку активност у P. Наиме, постоје бројеви d_{ij} и D_{ij} такви да важи:

$$0 \leq d_{ij} \leq y_{ij} \leq D_{ij} < \infty \tag{4. 2}$$

за свако (i, j) у P. D_{ij} се назива нормалним трајањем активности (i, j), а d_{ij} убрзано или смањено трајање.

Напомена 4. 1: Вредност d_{ij} представља апроксимацију најбржег времена коју активност може да достигне узимајући у обзир њену природу и окружење у којем се извршава. С друге стране, D_{ij} обично мора да се утврди неким декретом. Он представља „разумно“ време под „нормалним“ условима извршења активности.

Распоред који задовољава (4. 1) и (4. 2) са $t_o = 0$ назива се изводљивим или допустивим распоредом. Трајање за сваку активност дефинише њену корисност.

Претпоставићемо да је корисност активности линеарна функција њеног трајања на затвореном интервалу дефинисана са (4. 2) и да има облик: $a_{ij}y_{ij} + b_{ij}$, где је $0 \leq a_{ij} < \infty$ и $-\infty < b_{ij} < \infty$.

Корист распореда се дефинише као збир корисности сваке активности из P, тј.

$$\sum_{(i,j)\in P}(a_{ij}y_{ij} + b_{ij}) \tag{4. 3}$$

Трајање распореда је $\lambda = t_n$.

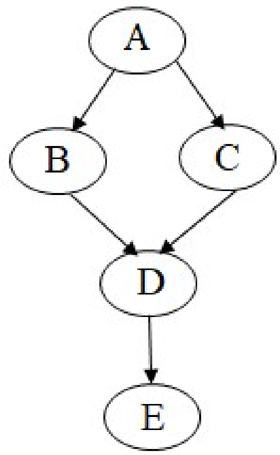
Јасно је да међу свим изводљивим распоредима који имају дато трајање, λ, постоји бар један који има максималну корисност, тј. који максимизира (4. 3). Такав изводљив распоред се назива оптималним. Означићемо ову вредност (4. 3) са овим распоредом са $U(\lambda)$ (уколико је мера користи трошак, губитак итд. који захтевају минимизацију, једноставно се само узме негативна функција и онда се минимизује).

Посматрано као функција од λ, $U(\lambda)$ ћемо звати функцијом корисности пројекта. Главни циљ је проналажење алгоритма за генерисање $U(\lambda)$ и дефинисање оптималног распореда који је изводљив.

Напомена 4. 2: Постоји оправданост коришћења алата које превазилазе ограничења математичких техника упошљавањем функције користи активности које су неопадајуће и линеарне. Претпоставимо да је корист активности $(i,\ j)$ највећа у d_{ij}. То значи да би у било ком могућем распореду би увек имали $y_{ij} = d_{ij}$. Дакле, „ефективна“ корист $(i,\ j)$ на $[d_{ij}, D_{ij}]$ је константно једнака користи у d_{ij}. Напоменимо да, генерално, ово важи на сваком подинтервалу $[p, q]$ од $[d_{ij}, D_{ij}]$где је корист p већа од користи сваке друге тачке у $[p, q]$. С тога, „ефективна корист“ $(i,\ j)$ на $[d_{ij}, D_{ij}]$ је увек неопадајућа. Даље, у пракси је често тешко добити процене користи активности за више од неколико трајања. У таквим околностима, узимање тренда се сматра разумним. Међутим, претпоставка о линеарности може бити замењена претпоставком да су функције користи активности део по део линеарне, неопадајуће и конкавне између смањеног и нормалног трајања.

4. 3. Практична правила за описивање пројекта

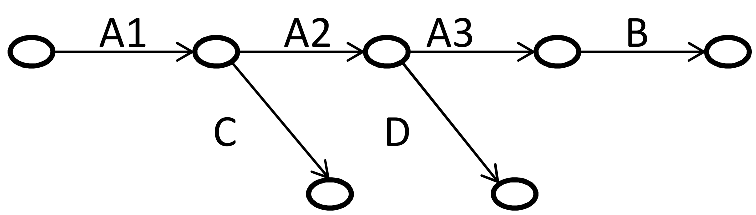
Из свега горе наведеног јасно је да се процес који описује односе између активости пројекта може представити усмереним ацикличним графовима (енг. Direct Acyclic Graphs, DAG). Свака активност пројекта је означена чвором. Стрелице показују односе (релације) између активности и смер протока времена (време тече од репа до главе стрелице). Резултат тога је усмерени ациклични граф. На слици 4. 3. је приказан пример графа пројекта.



Слика 4. 3. Приказ активности пројекта A, B, C, D и E преко усмереног ацикличног графа.

Приликом формирања графа пројекта морају се поштовати следећа правила:

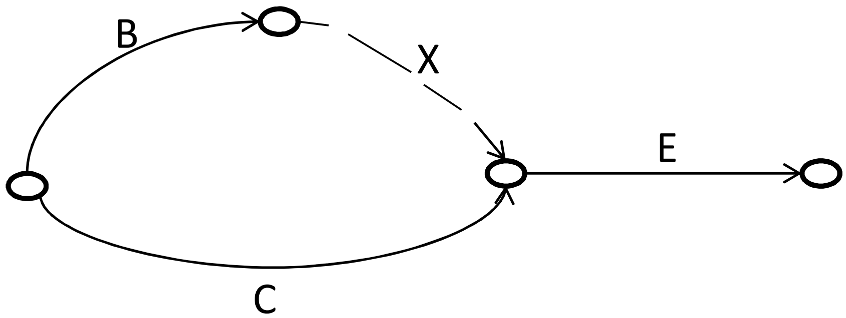
Правило 4. 1: Сложене активности. Нека активност *A* буде претходник активности *B*. У пракси, *A* се често представља на такав начин да *B* може стартовати чим се *A* делимично заврши. Или многе активности могу бити започете чим је *A* у неком проценту при завршетку. У овој ситуацији уводимо да *A* буде склоп многих активности. На пример, уколико активност *C* може бити започета када се *A* на пола заврши, активност *D* када је *A* завршено ¾, а активност *B* када се *A* у потпуности заврши, разматрамо *A* као склоп три различите активности: A_1, A_2 и A_3 као на слици 4. 4.:



Слика 4. 4. Приказ сложених активности.

Коришћењем овог средства предефинисаних активности помоћу њихових компоненти када то захтева прилика, можемо увек претпоставити да је сваки посао у пројекту у потпуности завршен пре него било који од њихових следбеника започне. Наравно, могу се направити одвојене анализе функције користи сваке компоненте.

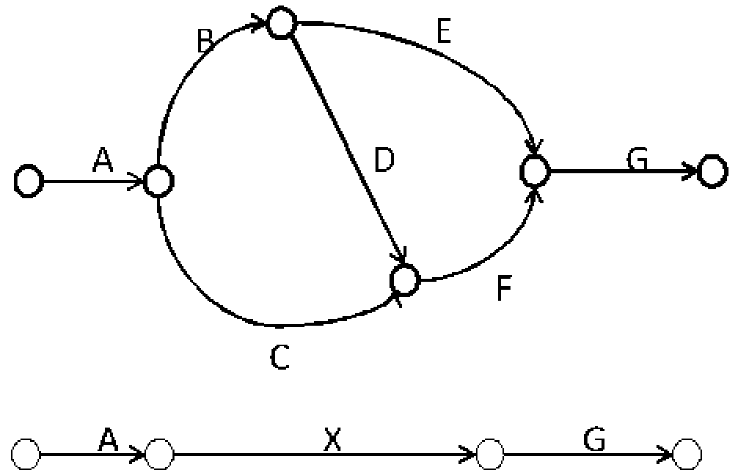
Правило 4. 2: Истовремене активности. Може се догодити да две или више активности у пројекту морају да почну и да се заврше до остварења неког догађаја. Све те активности могу имати различита трајања и да би се то назначило све активности осим најдуже представљају се као склоп две активности стварне активности са својим трајањем за којом следи фиктивна или лажна активности. Ова ситуација је илустована на слици 4. 5.:



Слика 4. 5. Илустрација истовремених активности.

Активности *B* и *C* морају бити завршене пре него *E* може да почне. Лажна активност *X* се уводи да би се раздвојиле *B* и *C* једна од друге али још увек одржава захтевано низање активности.

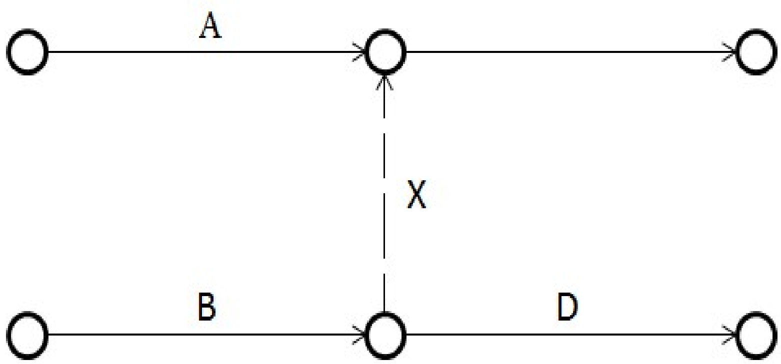
Правило 4. 3: Обједињене активности. Понекад се извесна група активности може сматрати као једна активност. Ова интерпретација може бити сасвим пожељна, посебно када су све активности поређане у групе и могу се разматрати за формирање малог пројекта. Када дође до овог типа ситуације нека лажна активност може да замени целу групу активности. Овим типом обједињавања, главни пројекти могу бити поједностављени у извесне сврхе. Слика 4. 6. илуструје овај случај:



Слика 4. 6. Графички приказ правила обједињавања активности.

Лажни (фиктивни) *X* замењује активности *B, C, D, E* и *F*. Наравно, специјална анализа функције корисности и ограничење трајања за *X* се мора направити.

Правило 4. 4.: Зависне и независне активности. Често се може догодити да је нека одређена активност *C* наследник две истовремене активности *A* и *B*, али *B* може имати налседника *D* који није наследник *A*. Ова ситуација може бити решена увођењем фиктивне активности *X* као што је приказано на слици 4. 7.:



Слика 4. 7. Зависне и независне активности. Илустрација правила 4. 4.

Правило 4. 5: Активност времена почетка. Коначно пројекат мора бити стављен у календар. Тако је неопходно увести барем једну активност која почиње са нултим датумом календара и ограничава се на старту пројекта.

Ова активност представља време почетка целог пројекта. Може се користити да сакупи одређене административне ствари о којима се мора бринути пре стварног почетка пројекта.

Обично нормално трајање времена почетка је време које протекне од неког фиксираног датума до фиктивног почетка пројекта. Смањење трајања је нула. Остала ограничења трајања су могућа у зависности од тумачења времена почетка.

Функцији користи почетка се обично даје нулти степен слободе. Међутим, када је пожељно одложити почетак пројекта што је више могуће, велики позитивни степен слободе се понекад приписује функцији користи.

Правило 4. 6: Услови старта. Покретање неких активности у пројекту може зависити од испоруке одређених ставки или од прикладних услова-материјала, планова, одобрења фондова, времена итд. Када све активности које проистичу из случаја I зависе од истог стартног стања, ограничење $t_i \geq T$ мора бити укључено у модел пројекта. Време које се односи на почетак пројекта, при којем ове активности могу почети се ограничава са Т.

Додатак стартних услова моделу не мења његов облик у основи. Можемо редуковати промењени проблем до стандардног (4. 1) и (4. 2) увођењем фиктивне активности $(0,\ i)$ тако да $D_{oi} = d_{oi} = T$ и $a_{oi} = b_{oi} = 0$. Лако је видети да са овим условима било који могући распоред мора имати $t_i \geq T$. Када постоји нека активност која проистиче из случаја *i* а да не зависи од стартног стања, правило 4. 4 се примењује на очигледан начин да би се обезбедило да она буде независна од стартног услова.

Уколико неколико испорука треба урадити до случаја *i* од почетка, правило 4. 2 мора бити примењено да би се елиминисале њихове нејасноће да сви имају исте ознаке.

4. 4. Корист/трошак

Анализа корист/трошак представља што бољи распоред пројекта у циљу постизања повољнијих исхода трајања пројекта, његових трошкова као и добити. Циљ анализе корист/трошак је проналажење оптималног трајања које минимизира укупни трошак пројекта. Анализа корист/трошак врши процену могућих варијанти убрзања активности.

Укупна вредност пројекта се састоји из директних и индиректних трошкова. Трошкови се могу дефинисати као новчани (вредносни) израз утрошених елемената производње који су настали да би се произвели учинци предузећа и продали на тржишту, а све у циљу остварења добити. Из ове дефиниције произилази да трошкови имају своју количинску и вредносну компоненту. Количинска компонента трошкова представља количински одређену (природно одређену) потрошњу фактора производње и другачије се назива утрошком. Утрошци говоре о томе колико је на пример, килограма материјала потрошено да би се остварио одређени обим производње, или колико је киловат-сати електричне енергије утрошено у процесу производње.

За трошкове неких других елемената производње утврђивање количинске компоненте не представља ни мало лак посао. На пример, трошење алата, пословног инвентара, зграда, машина и слично, није физички видљиво, али и ови трошкови имају своју количинску компоненту која се може изразити индиректно, или преко количине произведених учинака коришћењем тих фактора производње или узимањем у обзир временске компоненте (протекло време коришћења) датог материјалног добра. Највећи проблем у утврђивању количинске компоненте јавља се код трошкова капитала где се као количинска компонента узима износ капитала (износ узетог кредита) и време коришћења тог капитала. Када се количинска компонента трошкова (утрошак) помножи са ценовном компонентом добија се трошак.

Анализа односа корист/трошак је једна од главних метода менаџера, јер оптимална решења, добијена анализом овог проблема, директно повећавају продуктивност и профит пројекта. Постоји неколико техника, хеуристичких алгоритама који су развијени и спроводе се, а који имају за циљ да се њима постигне проналажење оптималног решења проблема односа корист/трошак пројекта. Овај проблем је препознат отприлике пре пола века, готово истовремено када су се развиле технике анализе пројекта по Fulkerson и Kelly (De et al. 1995). Први значајан покушај да се реши проблем односа корист/трошак може да се сматра настанком хеуристичког алгоритма Nicolai Siemens (Siemens, 1971), а касније је тај алгоритам побољшао Goyal (1975, 1996).

Овај хеуристички алгоритам је заснован на оптимизацији времена најдужег/их дела/ова мреже уколико је укупна корист смањивања времена била мања од уштеде добијене смањивањем индиректних трошкова. Недостатак овог алгоритма је потреба за одређивањем свих путања кроз мрежу. Овај захтев је лако било задовољити за мање пројекте, али за веће, било је немогуће да се сачувају сви путеви у меморији рачунара. Број путева у мрежи расте експоненцијално ако нека од активности има више од једног претка. Уколико би се десило да дође до елиминације путева овим хеуристичким алгоритмом, то би довело до могућности заглављивања у локалном минимуму што имплицира немогућност постизања резултата хеуристичком методом. Осим тога, ова хеуристичка метода је погодна за стално убрзавање функција. Ако је оптимизација означена функцијама активности, хеуристички алгоритам преферира најмању цену убрзавања активности. Ако активност треба убрзати више тако да критични пут који настаје овако убрзаном активношћу буде краћи, одбацује се претходни. Остале активности претходног критичног пута могу обезбедити јефтиније алтернативе убразања, а могу остати и са истим дужинама трајања. Да сумирамо, хеуристичке методе се могу сматрати задовољавајућим средством за решавање проблема корист/трошак могу да обезбеде оптимална или њима билска решења са разумним рачунским трајањем. Сходно томе, многи истраживачи су спроводили хеуристичке алгоритме у својим студијама за тражење оптималних решења овог проблема (Panagiotafopoulos, 1977; Schwarze, 1980; Barber et al., 1988; Chiu et al., 2005; Vanhoucke et al., 2007).

Модел Fondahl’s (1961), модел Siemens’s (1971), и модел Moselhi’s (1993) су примери хеуристичког приступа метода које дају добра решења, али не гарантују оптималност.

Многи истраживачи су покушали да реше проблем корист/трошак линеарним програмирањем (Babu et al., 1996; Burns et al., 1996; Khang et al., 1999; Wei et al., 2003; Moussourakis et al., 2004; Vanhoucke, 2005; Yang, 2007a, 2007b; Bidhandi, 2006). Чињеница је да се сваки проблем корист/трошак може пребацити у проблем линераног програмирања и као такав се решавати коришћењем комерцијалних софтвера за овакве проблеме. Ако функција трошка укључује квадратне једначине, оне се могу линеаризовати Тејлоровим развојем и линеарним програмирањем се итеративно решити. Недостатак линеараног програмирања је да број параметра знатно расте са порастом броја активности. Обично је број параметра четири до пет пута већи од броја активности у зависности од проблема корист/трошкова који се решава што овај проблем чини сложенијим за решавање код већих пројекта. Уколико не постоји меморијско ограничење, овим приступом тј. Линеарним програмирањем је загарантовано налажење оптималних решења проблема корист/трошкова. Природа проблема корист/трошак чини је погодном за проналажење решења линераним програмирањем и гарантује конвергенцију решења.

Динамичко програмирање је метода која знатно смањује величину мреже спајањем активности. Динамичко програмирање има за циљ смањивање чворова мреже у циљу решавања проблема корист/трошкова усклађивањем система убрзавањем активности. Алгоритам редукције се може испрограмирати, али задржавање или елиминација убрзавањем активности је тешка и захтева велике капацитете меморије, посебно код већих пројеката. Осим тога, свака мрежа се не може свести на систем једног чвора због сложених логичких веза између активности. Алтернатива спајања није могућа ако постоје сложени односи и ти чворови се задржавају такви какви јесу. На крају процеса спајања, мрежа се решава таксативним набрајањем сложених чворова који не могу да се редукују тј. споје. Време трајања израчунавања динамичким програмирањем је прихватљиво код малих пројеката, међутим за сложеније пројекте и комплексне мреже набрајање није практично. De et al., 1995, аутори и Demeulemeester et al., 1996, аутори су објаснили алгоритам декомпозиције мреже.

Методе математичког програмирања се углавном користе за решавање проблема корист/трошкова. У њима се користи или линерано или динамичко програмирање. (Kelly, 1961; Meyer et al., 1965; Butcher, 1967; Talbot, 1982). У овим методама однос између трошкова и трајања активности се генерално представља као:

(1)линеарни или нелинеарни,

(2)конкавни или конвенсни или да не буду фиксни,

(3)дискретни или непрекидни, или

(4)хибридни.

Многи истраживачи примењују генетски алгоритам (GA) за решавање проблема корист/трошкова, јер се показао као добар кандидат за проналажење глобалних екстрема. (Zheng et al., 2004; Li et al., 1997; Li et al., 1999; Zheng et al., 2005; Feng et al., 1997; Eshtehardian et al., 2008; Elbeltagi et al., 2005). Предност GA је што може да управља било каквим функцијама, као што су дискренте, линеарне, нелинеарне. Међутим, како се број активности повећава мора да се повећа и број евалуација у циљу добијања бољег решења, чиме се повећава време извршавања.

Метахеуристички алгоритми се широко примењују за решавање једноставног проблема корист/трошак, иако линеарно програмирање такође гарантује решење ових једноставнијих проблема. Један од разлога за примену метахеуристичких алгоритама је тај што је линеарно програмирање тешко применити на инжењерска планирања. Код линеарних једначина се захтева искуство и знање оних који праве план пројекта док код метахеуристичких алгоритама то није случај. Код њих је довољно да се дефинишу варијанте убрзања и критичан пут.

Метахеуристички алгоритми се лако примењују, али не могу увек гарантовати оптимално решење (Elbeltagi et al., 2005). Главни недостатак постојећих метахеуристичких алгоритама је што не могу да побољшају тренутно најбоље решење чак и када се број итерација повећа. То се дешава јер се метахеуристички алгоритми заглаве у локалном минимуму и не могу да изађу. Остали разлози су несавршени алгоритми претраге зато што се оптимална претрага увек заснива на случајним променама појединих алгоритама претраге који треба да обухвате процес претраге екстрема.

У циљу побољшања конвергенције алгоритама ка оптималном решењу, у овој дисертацији развијени су метахеуристички алгоритми засновани на симулираном каљењу и генетском алгоритму.

Знање глобалног оптимума функције је од великог значаја. У супротном само релативна побољшања би могла пратити компарацију резултата са осталим метахеуриситчким методама.

4. 5. Ограничење ресурса у проблему односа користи и трошкова

Овај проблем је веома сличан једноставном проблему односа користи/трошак који се заснива на оптимизацији без ограничавања ресурса, разликује се од њега само што постоји проблем у ограничењу ресурсима при чему се минимизују и трошкови и време трајања пројекта.

У поређењу са једноставним корист/трошак проблемом, овај проблем даје најрентабилнији распоред пројекта. У овом случају, вредност пројекта се своди на минимум узимајући у обзир индиректне трошкове и уговорне казне. С друге стране, минимално трајање пројекта смањује индиректне трошкове пројекта који не могу смањити укупне трошкове пројекта. Решење једноставног корист/трошак проблема може се завршити са неприхватљивим експлоатисањем ресурса, јер се ресурси никада не разматрају овом анализом. Осцилација ресурса може изазвати смањење продуктивности рада. Поред тога, значајан број радника и опреме може бити неискоришћен, а то доводи до повећања директних трошкова пројекта. Као резултат овога, пројекат може коштати више него што се добија простом анализом корист/трошак проблема.

Проблем ограничења ресурса у односу корист/трошак проблема је значајно тежи проблем, не само да се реши, већ и да се дође до података потребних за анализу. Поред захтева времена и ресурса, анализа захтева и трошкове извршења сваког вида активности. Као резултат тога, подаци потребни за анализу траже додатни временски напор.

У циљу припреме подробнијих планова пројекта, фирме би требало да издвоје више новчаних средстава из буџета за софтверску подршку и за припрему тендерске документације. Уз помоћ база података и система за подршку при одлучивању, трошкови активности могу да се процене за кратак временски период тачно. Зато што је тежак проблем и што није лако доћи до података за анализу, проблем ограничења ресурса у односу корист/трошак проблема није привукао пажњу многих истраживача. Pathak et al. (2007) аутори анализирали су овај проблем коришћењем технике генетског алгортима. За тестирање анализирано је 9 активности и 18 активности. Адаптивни ресурси који су утврђени пројектом имали су за циљ да смање максималну потражњу ресурса.

Ораничеље ресурса у проблему односа користи и трошкова се решава тако што се подели у два одвојена проблема где се анализира једноставан проблем корист/трошак проблема без разматрања анализе ресурса, а затим се распореде ресури. Међутим, ова процедура не може увек да помогне, посебно када је у питању велика потражња за неким ресурсом.

4. 6. LP формулација примењена на проблем оптимизације корист/трошка при распоређивању пројекта

У овом одељку представљена је формулација линераног програмирања (LP) за оптимално убрзање пројекта. Пројекат представљамо мрежом $S = (V, P, t, c)$ која се састоји од коначног скупа чворова (догађаја) V и скупа лукова $P \subset V \times V$ са тежинама које су времена активности, а одређене су функцијама $t: P \rightarrow R^+$ и $c: P \rightarrow R^+$, где су чворови повезани луковима. Параметар t_{ij} представља трајање активности $(i,\ j) \in P$, и c_{ij} је цена јединице по времену редукованог трајања активности $(i, j) \in P$. Сврха анализирања односа корист/трошак пројекта је да утврди за коју активност у мрежи може да се скраћује време трајања и за колико може да се убрза пројекат, а да се при томе трошак сведе на минимум и у исто време елиминише кашњење, тј. обезбеди жељено време завршетка пројекта (Chen et al., 2011).

Код анализе односа корист/трошак пројекта мрежа садржи два параметра трошка која се процењују за сваку активност $(i,\ j) \in P$, нормалан трошак c_{ij}^n и трошак убрзања c_{ij}^c, одређени нормалним временом t_{ij}^n и временом убрзања t_{ij}^c за сваку активност $(i,\ j) \in P$. Релација између нормалног трошка и трошка убрзања у односу на време активности у овом раду је представљена линеарно:

$$\Delta C_{ij} = \frac{C_{ij}^{n} - C_{ij}^{c}}{t_{ij}^{n} - t_{ij}^{c}} \tag{4. 3}$$

Нека је време трајања активности t_{ij} одређено скраћењем Δt_{ij}. Коришћењем релације (2. 18), повећање директних трошкова за сваку активност $(i,\ j) \in P$ је $\Delta t_{ij} \Delta \mathrm{C}_{ij}$ чиме је укупна цена трошкова убрзања активности $\sum_{i=1}^{n} \sum_{j=1}^{n} \Delta C_{ij} \Delta t_{\mathrm{ij}}$, где је *n* број активности.

Дакле, циљ анализе односа корист/трошак при распоређивању пројекта је да се минимизује укупно повећање трошкова које се може изразити као:

$$min \sum_{i=1}^{n} \sum_{j=1}^{n} \Delta C_{ij} \Delta t_{ij}. \tag{4. 4}$$

Сходно томе, проблем односа корист/трошак код пројекта мреже са чворовима може да се формулише као следећи LP проблем:

$$Z = \min \sum_{i=1}^{n} \sum_{j=1}^{n} \Delta C_{ij} \Delta \mathrm{t}_{\mathrm{ij}}, \tag{4. 5}$$

одређеног са:

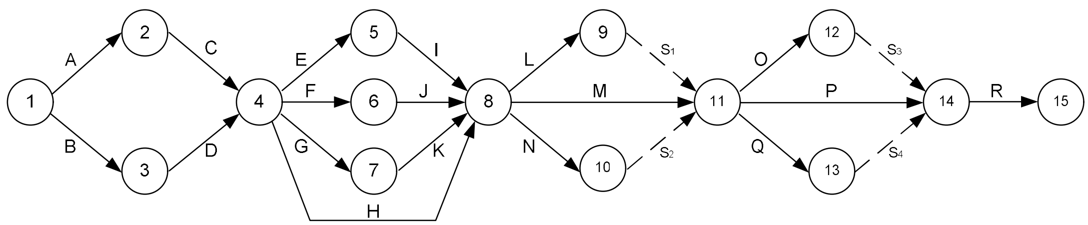
$$\Delta t_{ij} \leq t_{ij}^{n} - t_{ij}^{n}, i = 1, \dots, n, j = 1, \dots, n, i \neq j, \tag{4. 6}$$

$$t_j - \Delta t_{ij} - t_i \geq t_{ij}, i = 1, \dots, n-1, j = 2, \dots n, i \neq j, \tag{4. 7}$$

$$t \leq PCT. \tag{4. 8}$$

где је t_i најкраће време чвора (догађаја) i, а PCT је наведено време завршетка пројекта. У овом моделу ограничења (4. 6) представљају убрзање. Време завршетка активности може се смањити повећањем средстава или побољшањем продуктивности која такође захтева додатне ресурсе. Ово смањење времена није могуће „неограничено“. Оно се може смањивати само до одређене границе. Ограничења (4. 7) описују мрежу. Као што смо поменули, активности пројекта су међусобно повезане. Почетак неких активности зависи од завршетка неких других. Пошто се почиње од чвора 1 са почетком пројекта, његово време постављамо на 0, док је временско ограничење последњег чвора време завршетка пројекта. У реалним случајевима, за већину пројеката, морамо користити људску процену. Начин превазилажења ових непрецизности и бављење њима отвара концепт „замућености“ тј. фазификације.

Да бисмо илустровали важност предложеног приступа, истраживање је спроведено на пројекту „Главни поштански центри на територији Републике Србије“. Табела 4. 1. садржи све активности са временима и одговарајућим параметрима. Моделу проблема морају се дефинисати све активности у смислу тачке почетка и завршетка (Слика 4. 8.). Пројектна мрежа има 18 активности и 15 догађаја.



Слика 4. 8. Дијаграм мреже пројекта.

Табела 4. 1. Активности, време, трошак и релације односа у пројектној мрежи.

Активности	Опис	Зависности	Нормално време (дани)	Убрзано време (дани)	Нормални трошак (EUR)	Убрзани трошак (EUR)
A	Идејни пројекат комплекса Главног Поштанског Центра (ГПЦ) у Београду	-	90	70	3700	5000
B	Техничко-технолошки концепт комплекса ГПЦ Београд	-	30	25	2900	3200
C	Обезбеђивање свих	A	120	100	5000	6500

	неопходних услова и дозвола за комуналне прикључке					
D	Спровођење процедуре отворених јавних набавки за технолошку документацију изградње ГПЦ Београд.	B	60	60	1000	1000
E	Израда инвестиционо-техничке документације за изградњу ГПЦ Београд	C, D	120	110	3522300	3700000
F	Израда инвестиционо-техничке документације за изградњу ГПЦ Ниш	C, D	60	55	23600	29200
G	Израда инвестиционо-техничке документације за изградњу ГПЦ Нови Сад	C, D	120	100	153400	160000

H	Израда технолошког концепта	C, D	90	80	362850	387300
I	Конструкција ГПЦ Београд	E	360	320	15616120	15970000
J	Реконструкција ГПЦ Ниш	F	120	110	1077340	1090750
K	Реконструкција и надоградња ГПЦ Нови Сад.	G	360	320	1437240	1500000
L	Опремање објекта-канцеларијска опрема.	I,J,K,H	90	80	11800	15000
M	Опремање објекта-технолошка опрема.	I,J,K,H	100	80	15300	18000
N	Опремање објекта-ИТ (информационе технологије) опрема.	I,J,K,H	80	60	5900	7000
O	Монтажа опреме за аутоматско сортирање поштанских писама.	L,M,N	120	110	7860000	8000000

P	Монтажа опреме за аутоматско сортирање штампаних ствари.	L,M,N	270	240	8950000	9160000
Q	Монтажа опреме за аутоматско сортирање пакета.	L,M,N	120	100	6774700	6944000
R	Инсталирање ИТ и тлекомуникационе опреме.	O,P,Q	120	90	11800	13200

Према методи критичног пута, за нормалну дужину активности постоје два критична пута, A-C-E-I-M-P-R и A-C-G-K-M-P-R, а дужина ових путева је 1180 временских јединица (дана). У случају убрзања временских активности, ситуација по путањи критичног пута остаје иста, док је дужина дана сада 1010.

Претпоставимо да желимо да подесимо да завршетак пројекта буде убрзан. Према формулацији линераног програмирања (2. 25) - (2. 28) оптимално решење за проблем распоређивања односа корист/трошкова за овај пројекат дат је у табели 4. 3.

Међутим, у стварности постоји низ непредвиђених ситуација које могу да утичу на времена активности пројекта. Дакле, да би се приказала стварна ситуација током реализације пројекта треба релаксирати убрзање времена и мрежу ограничења. Због тога ћемо сада извршити проблем фазификације у односу на интервале толеранције за функцију циља и сва ограничења (табела 4. 2.).

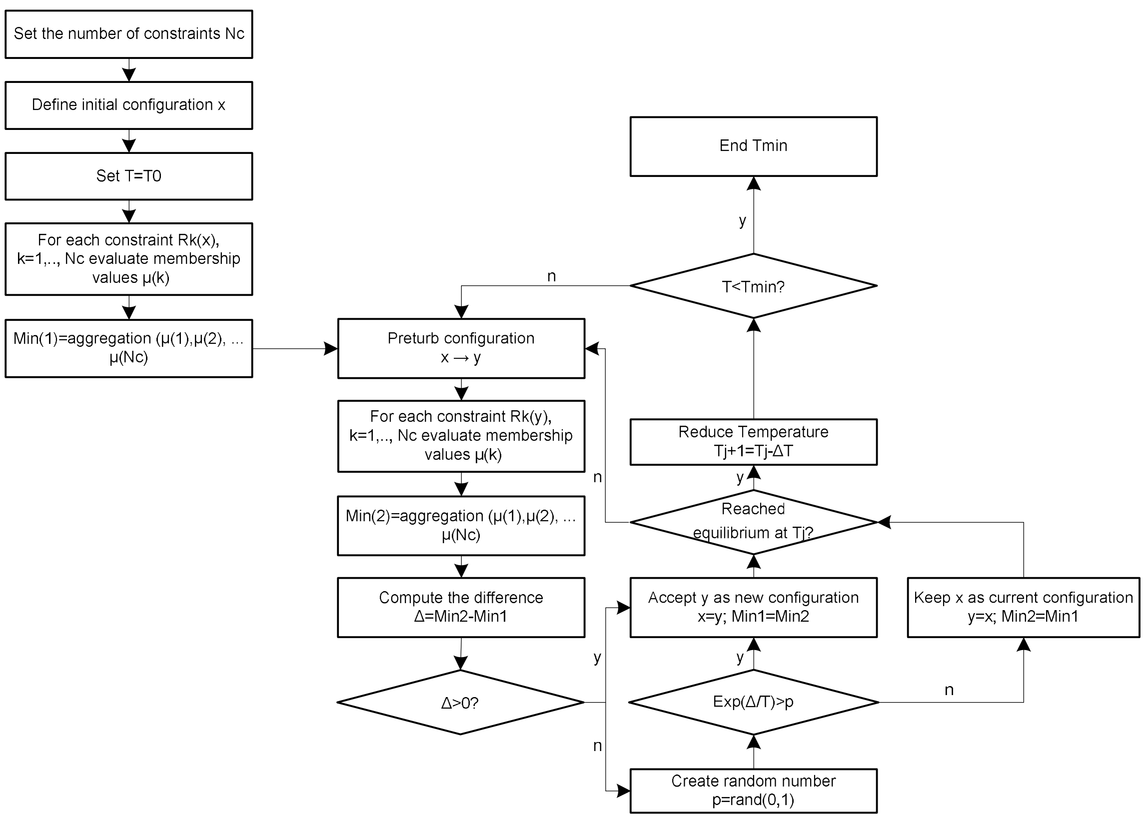
Табела 4. 2. Дозвољена одступања за проблем односа корист/трошак.

Ограничења типа „мање или једнако“			Ограничења типа „веће или једнако“		
Доња граница (c_i)	Девијација (одступање) (d_i)	Горња граница (c_i+d_i)	Доња граница (c_i-d_i)	Девијација (одступање) (d_i)	Горња граница (c_i)
$Z=c_0=750$	220	970	85	$d_{25}=5$	$c_{25}=90$
$c_1=20$	$d_1=2$	22	20	$d_{26}=10$	$c_{26}=30$
$c_2=5$	$d_2=3$	8	110	$d_{27}=10$	$c_{27}=120$
$c_3=20$	$d_3=3$	23	40	$d_{28}=20$	$c_{28}=60$
$c_4=0$	$d_4=5$	5	115	$d_{29}=5$	$c_{29}=120$
$c_5=10$	$d_5=3$	13	55	$d_{30}=5$	$c_{30}=60$
$c_6=5$	$d_6=5$	10	118	$d_{31}=2$	$c_{31}=120$
$c_7=20$	$d_7=5$	25	350	$d_{32}=10$	$c_{32}=360$
$c_8=10$	$d_8=5$	15	115	$d_{33}=5$	$c_{33}=120$
$c_9=40$	$d_9=5$	45	352	$d_{34}=8$	$c_{34}=360$
$c_{10}=10$	$d_{10}=5$	15	85	$d_{35}=5$	$c_{35}=90$
$c_{11}=40$	$d_{11}=5$	45	85	$d_{36}=5$	$c_{36}=90$
$c_{12}=10$	$d_{12}=5$	15	77	$d_{37}=3$	$c_{37}=80$
$c_{13}=10$	$d_{13}=5$	15	98	$d_{38}=2$	$c_{38}=100$
$c_{14}=10$	$d_{14}=5$	15	0	$d_{39}=0$	$c_{39}=0$
$c_{15}=10$	$d_{15}=5$	15	0	$d_{40}=0$	$c_{40}=0$
$c_{16}=40$	$d_{16}=5$	45	115	$d_{41}=5$	$c_{41}=120$
$c_{17}=20$	$d_{17}=5$	25	117	$d_{42}=3$	$c_{42}=120$
$c_{18}=30$	$d_{18}=5$	35	0	$d_{43}=0$	$c_{43}=0$
$c_{19}=30$	$d_{19}=7$	37	0	$d_{44}=0$	$c_{44}=0$
$c_{20}=0$	$d_{20}=5$	5	117	$d_{45}=3$	$c_{45}=120$
$c_{21}=0$	$d_{21}=5$	5			
$c_{22}=0$	$d_{22}=5$	5			

c_{23}=0	d_{23}=5	5			
c_{24}=1010	d_{24}=40	1050			

4. 7. Имплементација алгоритма симулираног каљења на проблем оптимизације односа корист/трошак при распоређивању пројеката

Циљ имплементираног алгоритма симулираног каљења у овој дисертацији је да се реши модел фази оптимизације представљен у овом поглављу тако да се максимизира агрегациона вредност припадности циљева и ограничења. Његова реализација је дефинисана следећим дијаграмом (Milenković et al., 2012):



Променљива k је бројач ограничења. Min представља агрегацију функције припадности фази ограничења. Као оператор агрегације користи се t-норма min. Променљива Δ представља разлику између претходног и новог min. Ово представља разлику старих и нових енергетских стања. Вероватносна функција преласка на мања енергетска стања дата је са $Exp(\frac{\Delta}{T})$. Што је мања температура, то је мања вероватноћа да ће доћи до било каквих промена. Алгоритам укључује, такође, $N(t)$ као број генерисаних суседа. $T(t)$ је опадајућа функција температуре. Алгоритам се зауставља када је температура мања од дефинисане граничне вредности. Као што је приказано у поглављу 2., функције припадности фази циља и ограничења која се користе у овој апликацији су троугаоне функције (Glišović, N., et al., 2012).

Када користимо алгоритам симулираног каљења за решавање проблема фази оптимизације, доносилац одлуке може изабрати нивое (α-прагове) исто као и девијације (толеранције) за сваки параметар ограничења и/или за сваки коефицијент функције циља и/или за сваки коефицијент ограничења. Функције припадности се затим граде уз дефинисане толеранције. Шта више, алгоритам симулираног каљења допушта рад са било којим типом фазификације.

4. 8. Имплементација генетског алгоритма на проблем оптимизације односа корист/трошак при распоређивању пројекта

У овој дисертацији је генетски алгоритам имплементиран за потребе фази модела који су дали (Ribeiro et al., 1999).

За проблеме корист/трошак, низ карактера (генетски код) садржи све могуће активности додељене могућим алтернативама. За проблем нивелисања ресурса генски код садржи време слободног хода активности, а за проблем алокације генски код садржи приоритете активности.

У овом истраживању, јединка (индивидуа) је представљена низом који садржи времена активности које учествују у функцији циља. Иако је избор параметара нивоа наслеђивања, нивоа мутације и броја јединки у популацији врло важан за примену генетског алгоритма, истраживања су показала да ипак није пресудан. Сва истраживања урађена у овом правцу дају закључак да су критични моменти заправо кодирање и функција прилагођености, а да остали параметри мање утичу на перформансе GA.

Када се користи генетски алгоритам за решавање фази оптимизационог проблема, доносилац одлуке може да изабере нивое прага (ограничења α-праг) као и толеранције (одступања) за сваки параметар ограничења и/или коефицијент сваке функције циља и/или за коефицијенте ограничења. Функција припадности се онда генерише на основу тих дефинисаних толеранција. Осим тога, генетски алгоритам омогућава бављење било којом врстом фазификације (Glišović, N., 2014).

Политика замене генерација: Коришћена је елитистичка стратегија. Коришћењем ове стратегије замене генерација, најбољих N_{elite} јединки директно пролази у наредну генерацију без примене генетских оператора селекције, укрштања и мутације. Пошто се на дате јединке не примењује генетски оператор, њихове вредности се преносе из текуће у наредну генерацију без потребе за поновним рачунањем.

Генетски оператори

Селекција. Коришћена је селекција заснована на рангирању (енг. rank based selection) генетских кодова према њиховој прилагођености. Функција прилагођености сваке јединке је вредност ранга дате јединке. Једнинке које имају ранг већи од средње вредности свих јединки се укрштају.

Укрштање. Имплементирано је једнопозиционо укрштање (енг. one-point crossover). Претпоставимо да родитељи имају следећа времена извршавања активности која учествују у функцији циља:

Родитељ 1: $r_1 = \{t_1^{r_1}, t_2^{r_1}, \dots, t_n^{r_1}\}$;

Родитељ 2: $r_2 = \{t_1^{r_2}, t_2^{r_2}, \dots, t_n^{r_2}\}$;

Ако је тачка укрштања i на пример, тада су потомци:

Потомак 1: $p_1 = \{t_1^{r_1}, t_2^{r_1}, \dots, t_{i-1}^{r_1}, t_i^{r_1}, t_{i+1}^{r_2}, \dots, t_n^{r_2}\}$;

Потомак 2: $p_2 = \{t_1^{r_2}, t_2^{r_2}, \dots, t_{i-1}^{r_2}, t_i^{r_2}, t_{i+1}^{r_1}, \dots, t_n^{r_1}\}$;

Вероватноћа (ниво) укрштања је $p_{cros} = 0.7$ (у укрштању учествује 70% јединки).

Мутација. У овој дисертацији је примењена проста мутација са задатим нивоом мутације $p_{mut} = 0.2$.

Ако је изабрана јединка $r = \{t_1, t_2, .., t_n\}$ и позиција i за мутацију, резултујућа јединка је $\hat{r} = \{t_1, \dots, t_{i-1}, \hat{t}_i, t_{i+1}, \dots, t_n\}$, при чему је

$\hat{t}_i = \alpha_i t_{i,L} + (1 - \alpha_i) t_{i,U}, 0 \leq \alpha_i \leq 1.$

где је α_i случајан број, а $t_{i,L}$ и $t_{i,U}$ горња и доња граница параметра t_i.

Критеријум завршавања GA је достизање максималног броја генерација N_{gener}. У имплементацији GA развијеној у овој дисертацији, критеријум заустављања постављен је на 300 генерација.

Циљ имплементираног генетског алгоритма у овој дисертацији јесте да се реши модел фази оптимизације односа корист/трошак тако да се максимизира агрегациона вредност чланства циљева и ограничења (Glišović, N., 2013.). Његова реализација је дефинисана следећим псеудо кодом:

```
Algorithm for solving fuzzy optimization model
   /* Initialization */
        set Nc = number constraints
        select an initial state x
    /* Membership determination */
        for i=1 to Nc
                    μ(i) := membership value of the constraint Ri(x)
        min1 := aggregation(μ(1), . . . , μ(Nc))
    /* Genetic Algorithm */
        set numberOfGeneration = 0
        repeat
                    repeat
                            createNextGeneration
                            for i=1 to Nc
                                       μ(i) := membership value of the constraint Ri(x)
                            min2 = aggregation(μ(1),...,μ(Nc))
                            calculate delta = min2 - min1
                            if delta > 0 then
                                        UpdateNextPopulation;
                            min2 = min1;
                    until not terminate(y, evaluate(y), θ(k))
                numberOfGeneration++;
            until stop criteria
```

Процедура createNextGeneration() користи четири екстерне функције:

- x’ = recombine (x, theta(r));
- x’’ = mutate (x’, theta(m));
- pom = evaluate(x’’);
- y = select (x’’, pom, μ, theta(s))

Поступак процедуре updateBestPopulation() пружа проналажење најбоље популације у том тренутку. Ако је evaluate(y) <evaluate(x), онда постављамо x=y.

4. 9. Алгоритам фази симулираног каљења и фази генетски алгоритам примењен на реалан пројекат

Применом алгоритма фази симулираног каљења на реалан проблем представљен у секцији 4. 6. овог поглавља проблем фази линеарног програмирања је решен коришћењем Zimmermann-овог приступа, који сматра да у фази окружењу не постоји разлика између циљева и ограничења.У оквиру овог приступа циљ је померање унутар граница дефинисаних унутар пружања граница као и свако друго фази ограничење. Процес наставља да се трансформише у crisp проблем, максимизирање минимизиране вредности функције чланства, утврђене функацијама (2. 13)-(2. 15). Резултати проблема коришћењем овог приступа дати су у табели 4. 3.

Сада, на основу модела флексибилне фазификације представљеног у Поглављу 2. развијен је одговарајући алгоритам симулираног каљења за решавање модела проблема убрзања пројекта. За реализацију алгоритма симулираног каљења мора се навести: (а) како се генеришу стања y, суседа од x, (b) коју функцију агрегације користити, (в) изабрати број генерисаних суседа, (г) функцију температуре смањења, (д) алгоритам критеријума заустављања.

Избор стања *y*, као суседе од *x*, одрађено је дефинисањем нових стања која су случајне вредности стања *y*, дефинисано са:

$$\eta(t) = \begin{cases} 0.5, & ako\ je\ t < 100 \\ 1.0, & ako\ je\ 100 \leq t < 200 \\ 1.5, & ako\ je\ 200 \leq t < 300 \\ 2.5, & ako\ je\ t \geq 300 \end{cases}$$

Пресек свих вредности функције чланства ограничења је изабран као агрегациона функција, и t-норма минимума оператора се користи. Циљ постаје максимизација минимизације ограничења одступања. Селектовање броја генерисаних суседа одређен је помоћу следеће хеуристике:

$$N(t) = \begin{cases} 250, ako\ je\ t < 300 \\ 300, ako\ je\ t \geq 300 \end{cases}$$

Разматра се ова хеуристика да би се генерисало више наследника када температура опада, како би имали више опција за тестирање. Функција температуре

$T(t)$ је функција која користи фактор смањења од 0.95. Коришћени критеријум заустављања је постигнут када је температура мања од 0.002. Алгоритам фази симулираног каљења за решавање проблема оптимизације односа корист/трошак је имплементиран у C# програмском језику.

Тестирање је извршено за различите граничне вредности. Добијени резултати су упоређени са crisp верзијом (A), Zimmermann-овим приступом (B), и са алгоритмом симулираног каљења заснованом на фази флексибилном моделу (C) за разматран проблем односа корист/трошкова пројекта.

Користећи традиционалну трансформацију Zimmermann-ове релаксације функције циља и ограничења, добили смо вредност од 762.89 EUR, што представља смањење од 171.11 EUR трошкова пројекта убрзаног времена у односу на тадиционални случај.

Праг $\alpha = 0.98$ даје најбоље резултате, као што се може видети из табеле 4. 3. Применом овог прага добили смо вредности 755.31 за функцију циља што представља побољшање у односу на фази Zimmermann-а. Упоређујући променљиве можемо приметити да вредност за променљиву t_{15} представља знатно смањење времена у односу на оба алтернативна приступа, док t_2 и t_3 имају нешто веће вредности (погледати табелу 4. 3.).

Табела 4. 3. Решења Zimmermann-овим приступом и фази флексибилним моделом за праг $\alpha = 0.98$.

Промен љиве	A	B	C	Промен љиве	A	B	C	Промен љиве	A	B	C
Z=	934	762.89	755.31								
M		0.95	0.98								
Δt_{12}	0.00	0.00	0.00	Δt_{1112}	0.00	0.01	0.02	t_7	330	328.97	328.97
Δt_{13}	0.00	0.00	0.00	Δt_{1114}	0.00	0.00	0.01	t_8	690	688.30	688.30

Δt_{24}	0.00	0.10	0.10	Δt_{1113}	0.00	0.00	0.04	t_{9}	790	788.74	784.74
Δt_{34}	0.00	0.23	0.23	Δt_{1415}	20.00	13.86	1.76	t_{10}	790	768.40	768.40
Δt_{45}	0.00	0.00	0.00	Δt_{911}	0.00	0.24	0.24	t_{11}	790	788.51	788.50
Δt_{46}	0.00	0.03	0.03	Δt_{1011}	0.00	0.40	0.40	t_{12}	910	908.37	908.37
Δt_{47}	0.00	0.01	0.01	Δt_{1214}	0.00	0.10	0.10	t_{13}	910	908.37	908.37
Δt_{48}	0.00	0.00	0.00	Δt_{1314}	0.00	0.50	0.50	t_{14}	910	908.14	918.14
Δt_{58}	0.00	0.01	0.02	t_{1}	0.00	0.00	0.00	t_{15}	1010	1021.87	1011.87
Δt_{68}	0.00	0.00	0.00	t_{2}	90.00	89.77	90.77				
Δt_{78}	0.00	0.00	0.00	t_{3}	150.00	29.53	29.54				
Δt_{89}	0.00	0.00	0.00	t_{4}	210.00	209.30	209.30				
Δt_{811}	0.00	0.01	0.01	t_{5}	330.00	329.07	329.07				
Δt_{810}	0.00	0.02	0.03	t_{6}	570.00	568.83	568.83				

Све у свему, може се закључити да је коришћење симулираног каљења у циљу решавања проблема односа корист/трошак пројекта у присуству неизвесности прилично фелксибилно и прилагодљиво него што је то случај са традиционалним математичким трансформацијама дефинисаним методом Zimmermann-a (табела 4. 4.).

У случају када се користе ситуације где прагови (ограничења) могу бити дефинисани, доносилац одлуке може да изабере степен прекорачења за сва ограничења.

Табела 4. 4. Резултати Maxmin код симулираног каљења.

Променљиве	Приступ симулираног каљења	
	Maxmin (α>0.94)	Maxmin (α>0.98)
Z	762.89	755.31
$Ax\tilde{\geq}B$	0.941	0.985
$Ax\tilde{\leq}B$	0.941	0.985
Min	0.941	0.985

Резултати генетског алгоритма примењени на пројекту „Главни поштански центри на територији Републике Србије“. Табела 4. 1. (описан у одељаку 4. 6.) пројектна мрежа има 18 активности и 15 догађаја.

Према формулацији линераног програмирања (2. 25) - (2. 28) оптимално решење за проблем распоређивања односа корист/трошак за овај пројекат дат је у табели 4. 5.

Користећи crisp трансформацију Zimmermann-ове релаксације функције циља и ограничења, добили смо вредност од 753.2 EUR, што представља смањење од 180.8 EUR трошкова пројекта убрзаног времена у односу на crisp случај.

Упоређујући променљиве можемо приметити да вредност за променљиву t_8, t_{10} и t_{11} представља смањење времена у односу на оба алтернативна приступа, док t_2 , t_3 и t_{15} имају нешто веће вредности.

Табела 4. 5. Решења Zimmermann-овим приступом и фази флексибилним моделом за праг $\alpha = 0.98$

Променљиве	A	B	C	Променљиве	A	B	C	Променљиве	A	B	C
Z=	934	753.2	753.03								
M		0.95	0.98								
Δt_{12}	0.00	0.00	0,00	Δt_{1112}	0.00	0.01	0.02	t_7	330	328.97	328.97
Δt_{13}	0.00	0.00	0,00	Δt_{1114}	0.00	0.00	0.01	t_8	690	688.60	688.30
Δt_{24}	0.00	0.00	0,10	Δt_{1113}	0.00	0.00	0.04	t_9	790	788.74	788.74
Δt_{34}	0.00	0.23	0,23	Δt_{1415}	20.00	13.86	1.76	t_{10}	790	788.74	769.40
Δt_{45}	0.00	0.00	0,00	Δt_{911}	0.00	0.24	0.24	t_{11}	790	788.51	788.51
Δt_{46}	0.00	0.00	0,00	Δt_{1011}	0.00	0.40	0.20	t_{12}	910	908.37	908.37
Δt_{47}	0.00	0.03	0,01	Δt_{1214}	0.00	0.10	0.00	t_{13}	910	908.37	908.37
Δt_{48}	0.00	0.01	0,00	Δt_{1314}	0.00	0.50	0.50	t_{14}	910	908.14	918.14
Δt_{58}	0.00	0.00	0,00	t_1	0.00	0.00	0.00	t_{15}	1010	1021,87	1041.87
Δt_{68}	0.00	0.01	0,01	t_2	90.00	89.77	90.77				

Δt_{78}	0. 00	0.0 0	0,0 0	t_3	150 .00	29. 53	29. 53				
Δt_{89}	0. 00	0.0 0	0.0 1	t_4	210 .00	209 .30	209 .30				
Δt_{811}	0. 00	0.0 1	0.0 1	t_5	330 .00	329 .07	329 .07				
Δt_{810}	0. 00	0.0 0	0.0 2	t_6	570 .00	568 .83	568 .83				

Праг $\alpha = 0.98$ даје најбоље резултате, као што се може видети из табеле 4. 6.

Табела 4. 6. Maxmin резултати генетског алгоритма.

Приступ генетски алгоритам		
Промењљиве	Maxmin (α > 0.94)	Maxmin (α>0.98)
Z	753.20	753.10
Ax≥B	0.941	0.985
Ax≤B	0.941	0.985
Min	0.941	0.985

Ако упоредимо резултате добијене применом алгоритама симулираног каљења (табела 4. 3. и табела 4. 4.) и генетског алгоритма (табела 4. 5. и табела 4. 6.), генетски је постигао знатно боље резултате оптимизације. Постигнуто је смањење од 180.8 EUR трошкова пројекта убрзаног времена у односу на crisp случај, а 9.69 EUR ако се упореди са симулираним каљењем.

ПЕТО ПОГЛАВЉЕ: ПРОБЛЕМ КЛАСТЕРОВАЊА. ПРОБЛЕМ НЕДОСТАЈУЋИХ ПОДАТАКА

У овом делу доктората описујемо проблем кластеровања, недостајућих података. Дајемо предлог решавања такве врсте проблема. Предлажемо ново растојање. Дајемо предлог унапређивања постојећих метода кластеровања које се заснивају на тежини атрибута елемената који се распоређују у кластере.

У вези са овим проблемом постоји много радова који су предлагали разне метахеуристике, углавном засноване на популацији решења. Због тога је у овој дисертацији имплементирана метода промењљивих околина и показано је да она може конкурисати осталим методама у погледу квалитета добијених решења.

Прецизност резултата кластеровања се може смањити уколико се у анализу укључе обележја која нису релевантна за поједине кластере. Одабир релевантних обележја углавном представља предкорак у поступку кластеровања. Chan и сарадници (2004) су предложили нови приступ у решавању овог проблема, коришћењем тежинских мера разлика за објекте, при чему алгоритам проналази тежину за свако обележје у сваком кластеру

Нека је X скуп од m објеката описаних помоћу n обележја. Кластер алгоритам за тежинска обележја, који групише скуп X у k кластера, заснива се на минимизацији функције циља:

$$F(W,Z,\Lambda)=\sum_{l=1}^{k}\sum_{j=1}^{m}\sum_{i=1}^{n} w_{l,j}\lambda_{l,i}^{\beta} d(z_{l,i},x_{j,i}) \tag{4. 1}$$

Тако да важи,

$$w_{l,j}\in\{0,1\},\ 1\le l\le k,\ 1\le j\le m \tag{4. 2}$$

$$\sum_{l=1}^{k} w_{l,j}=1, 1\le j\le m$$

(4. 3)

$$0 < \sum_{i=1}^{m} w_{l,j} < m, 1 \le l \le k, \tag{4. 4}$$

$$0 \le \lambda_{l,i}, 1 \le l \le k, 1 \le i \le n \tag{4. 5}$$

$$\sum_{i=1}^{n} \lambda_{l,i} = 1, 1 \le l \le k \tag{4. 6}$$

Где је $k \le m$, бој кластера, $\beta < 1$, $W = \left[w_{ij} \right]$ матрица бинарних бројева реда $k \times m$, $Z = [z_1, z_2, \ldots, z_k] \in \mathbb{R}^{nxk}$, садржи центре кластера $\Lambda = \left[\lambda_{l,i} \right]$ матрица реалних бројева реда $k \times n$, и $0 \le d(z_{l,i}, x_{j,i})$ мера разлике између центра z_l и објекта x_j за i-то обележје. Главна идеја оптимизационог проблема је минимизација мере разлике између центара кластера и објекта. Мера разлике је дефинисана помоћу n тежинских обележја. На овакав начин дефинисана функција циља омогућава разматрање тежина за свако обележје у сваком кластеру.

Минимизација функције $F(\cdot,\cdot,\cdot)$ у јендакости (4. 1), са ограничењима (4. 2)-(4. 6) формира класу нелинеарних једначина са ограничењима, чија су решења непозната. Уобичајен метод оптимизације ове функције $F(\cdot,\cdot,\cdot)$ у (4. 1) је коришћење парцијалне оптимизације за W, Z и Λ. Прво фиксирамо Z и Λ и нађемо потребне услове за W за минимизацију $F(\cdot,\cdot,\cdot)$. После фиксирамо W и Λ и минимизирамо $F(\cdot,\cdot,\cdot)$ у односу на Z. Затим фиксирамо W и Z и минимизирамо $F(\cdot,\cdot,\cdot)$ у односу на Λ. Поступак се понавља све док се потиже побољшање функције циља.

АЛГОРИТАМ: Оптимизациони алгоритам кластеровања са тежинским мерама разлике

Корак 1. Изабрати почетну матрицу $Z^{(1)} \in \mathcal{R}^{nxk}$ и нека је $\Lambda^{(1)}$ матрица реда $k \times n$ са свим елементима једнаким $\frac{1}{n}$. Нека је $t=1$.

Корак 2. Одредити $W^{(t+1)}$ такво да је $F(W^{(t+1)}, Z^{(t)}, \Lambda^{(t)})$ минимизована. Уколико је $F(W^{(t+1)}, Z^{(t)}, \Lambda^{(t)}) = F(W^{(t)}, Z^{(t)}, \Lambda^{(t)})$, зауставити се; у супротном прећи на корак 3.

Корак 3. Одредити $Z^{(t+1)}$ тако да је $F(W^{(t+1)}, Z^{(t+1)}, \Lambda^{(t)})$ минимизована. Уколико је $F(W^{(t+1)}, Z^{(t+1)}, \Lambda^{(t)}) = F(W^{(t+1)}, Z^{(t)}, \Lambda^{(t)})$ зауставити се; у супротном прећи на корак 4.

Корак 4. Одредити $\Lambda^{(t+1)}$ тако да је $F(W^{(t+1)}, Z^{(t+1)}, \Lambda^{(t+1)})$ минимизована. Уколико је $F(W^{(t+1)}, Z^{(t+1)}, \Lambda^{(t+1)}) = F(W^{(t+1)}, Z^{(t+1)}, \Lambda^{(t)})$ зауставити се; у супротном се вратити на корак 2.

Матрице W, Z и Λ се рачунају у складу са следећим теоремама (Chan, EZ, et al., 2004.).

Теорема 5. 1. Нека су $\hat{Z}$ и $\widehat{\Lambda}$ фиксирани. Минимизатор матрице $\widehat{W}$ за оптимизациони проблем $\min_{W} F(W, \hat{Z}, \widehat{\Lambda})$ при чему важи (4. 2)-(4. 6) је дат помоћу

$$\widehat{w}_{ij} = \begin{cases} 1, & \sum_{i-1}^{n} \hat{\lambda}_{li}^{\beta}\left(\hat{z}_{li} - x_{ji}\right)^2 \leq \sum_{i=1}^{n} \hat{\lambda}_{hi}^{\beta}\left(\hat{z}_{hi} - x_{ji}\right)^2, 1 \leq h \leq k, \\ 0, & \text{иначе.} \end{cases}$$

Теорема 5. 2. Нека су $\widehat{W}$ и $\widehat{\Lambda}$ фиксирани. Минимизатор $\hat{Z}$ оптимизационог проблема $\min_{Z} F(\widehat{W}, Z, \widehat{\Lambda})$ је дат са

$\hat{z}_{li} = \frac{\sum_{j=1}^{m} \widehat{w}_{lj} x_{ij}}{\sum_{j=1}^{m} \widehat{w}_{lj}}$, где је $1 \leq l \leq k$,

Када је i–та променљива нумеричка, или

$\hat{z}_{li} = d_i^{(r)} \in DOM(D_i)$, где је

$\left|\left\{w_{i,j} | x_{i,j} = d_i^{(r)}, w_{i,j} = 1\right\}\right| \leq \left|\left\{w_{i,j} | x_{i,j} = d_i^{(t)}, w_{i,j} = 1\right\}\right|, \forall t \in DOM(D_i).$

Када је i–та променљива категоријална. Са $|Y|$ је означена кардиналност скупа Y.

Теорема 5. 3. Нека су $\widehat{W}$ и $\hat{Z}$ фиксирани. Минимизатор матрице $\widehat{\Lambda}$ за оптимизациони проблем $\min_{\Lambda} F(\widehat{W}, \hat{Z}, \Lambda)$, за који важи (4. 2)-(4. 6), је дат помоћу:

$$\widehat{\Lambda}_{l,i} = \begin{cases} \dfrac{1}{n_i}, & \displaystyle\sum_{j=1}^{m} \widehat{w}_{l,j} d(\hat{z}_{l,i}, x_{j,i}) = 0;\ n_i = \left|\left\{t: \sum_{j=1}^{m} \widehat{w}_{l,j} d(\hat{z}_{l,t}, x_{j,t}) = 0\right\}\right|, \\ 0, & \displaystyle\sum_{j=1}^{m} \widehat{w}_{l,j} d(\hat{z}_{l,i}, x_{j,i}) \neq 0;\ (\exists t) \sum_{j=1}^{m} \widehat{w}_{l,j} d(\hat{z}_{l,t}, x_{j,t}) = 0, \\ \dfrac{1}{\sum_{t=1}^{n} \left[\dfrac{\sum_{j=1}^{m} \widehat{w}_{l,j} d(\hat{z}_{l,i}, x_{j,i})}{\sum_{j=1}^{m} \widehat{w}_{l,t} d(\hat{z}_{l,t}, x_{j,t})}\right]^{\frac{1}{(\beta-1)}}}, & \displaystyle\sum_{j=1}^{m} \widehat{w}_{l,j} d(\hat{z}_{l,t}, x_{j,t}) = 0, \forall t, 1 \leq t \leq n. \end{cases}$$

Алгоритам представља кластер алгоритам за тежинска обележја, у коме се матрица W одређује у складу са Теоремом 1, центри кластера Z у свакој итерацији у складу са Теоремом 5. 2, а матрице тежина за обележја Λ у складу са Теоремом 5. 3.

Теорема 5. 4. Оптимизациони кластер алгоритам (АЛГОРИТАМ) конвергира у коначном броју итерација.

На основу теореме 5. 4., овај алгоритам за пондерисана обележја конвегира. Међутим, зауставља се у локалном минимуму (Chan, EZ, et al., 2004). Временска сложеност алгоритма износи $O(l \times m \times n \times k)$, а просторна сложеност $O(m \times (n+k) + 2k \times n)$ где је l укупан број итерација, k број кластера, n број обележја и m број објеката у посматраном скупу. Дакле, предложени алгоритам је прилагођен за рад са великим скупом података.

5. 1. Алгоритми засновани на растојању

Методе засноване на растојањима су често пожељне због своје једноставности примене у различитим сценаријима. Алгоритми засновани на растојању могу се, генерално, поделити у два типа:

Партиционо кластеровање (енг. Flat): У овом случају подаци се деле на неколико кластера у једном кораку, обично уз употребу представника партиције. Избор представник партиција и функције растојања је од кључног значаја и регулише понашање основног алгоритма. У свакој итерацији, бирају се најближе тачке представнику партиције, а затим се представник партиције прилагођава тачкама додељеним одређеном кластеру. Неке уобичајене методе за креирање партиција су следеће:

k-Means: Ова метода податке групише у *K* кластера, где се број кластера *K* одређује на различите начине, у зависности од доносиоца одлуке. Тешко је одредити прави број кластера у подацима - зато се алгоритам најчешће примењује више пута, па се на основу мере квалитета кластера или на основу потврде квалитета кластеровања од стране доносиоца одлуке одлучује да ли је резултат задовољавајући. Процес почиње одабиром *K* центроида (удаљености аритметичких средина кластера) и то најчешће на случајан начин. Затим се свакој тачки додељује њему најближи центроид, а затим се у свакој итерацији одређује нови центроид. Овај процес се понавља све док у две узастопне итерације не дође до премештаја једног случаја из једног у други кластер.

Еуклидово растојање се користи да би се израчунала удаљеност. Ова метода се сматра једном од најједноставнијих и најкласичнијих метода за груписање података (Jain, A., 2010) и такође, можда, једна и од највише примењиваних метода због своје једноставности и ефикасне имплементације.

k-Medians: Овом методом, медијана, уместо средње вредности, се користи за одређивање представника партиције. Као и у случају к-means одлуке, представник партиције не мора да буде из оригиналног скупа података. Овај метод је бољи од k-means методе зато што је медијана скупа података мање осетљива на екстремне вредности података.

k-Medoids: Овом методом, представник партиције се бира из оригиналних података. Ова техника се користи код података које чине произвољни објекти где нема смисла говорити о функцијама објеката. На пример, таквав је дискретан скуп тачака где нема смисла говорити о средњој вредности или од медијани. У овим случајевима представници партиције се бирају из скупа података, а итеративне методе се користе у циљу побољшања квалитета ових представника. У свакој итерацији, један од представника се замењује представником из актуелног скупа података како би се проверило да ли је побољшан квалитет груписања. Ова метода захтева много више итерација од претходне две.

Хијерархијско кластеровање (енг. *Hierarchical*): Хијерархијско кластеровање подразумева изградњу једне хијерархијске структуре налик дрвету. Једном формирани кластер се, код хијерархијских метода кластеровања, више не може делити, већ се само може повезивати с другим кластерима. У основи постоје две врсте хијерархијског кластеровања-агломеративне и поделе.

Агломеративне: У првом кораку сви подаци су сопствени кластери. У наредним корацима, два најближа кластера се комбинују у нови, и тако се смањује број кластера за један у сваком наредном кораку. При сваком наредном кораку се или поједини подаци комбинују у нове кластере или се већ постојећи кластери међусобно повезују.

Поделе: Овом методом, користи се приступ од врха ка дну како би се сукцесивно одредиле партиције података у структуру дрвета. Било који алгоритам партиционог кластеровања се користи у сваком кораку. Поделе партиција омогућавају бољу флексибилност како у смислу хијерархијске структуре дрвета тако и нивоа баланса различитих кластера. Није неопходно постићи перфектно избалансирано дрво односа на дубини различитих чворова или дрво чији је степен у свакој грани заправо два. Ово омогућава изградњу структуре стабла која омогућава различите компромисе сваког чвора дубине и тежине (броја података у чвору) (Karypis, G., et al., 1998).

Методе засноване на удаљености су веома популарне у литератури, јер могу да се користе за било који тип података, док је одговарајућа функција растојања погодна за ту врсту података. Дакле, проблем груписања података може да се сведе на проблем проналажења функције растојања за ту врсту података. Из овога произилази да је проналажење функције растојања постала важна област истраживања у обради података (Aggarwal, C., 2003; Wang, F., et al., 2012). Одређене методе су често прилагођене специфичном домену променљивих, као што су категоријалне или временске серије (Das, G., et al., 2000; Gunopulos, D., et al., 2001).

5. 2. Избор мере сличности/растојања

Да би се уопште спровела анализа груписања, неопходно је дефинисати мере блискости два објекта на основу њихових карактеристика. Концепт „сличности" се одређује у зависности од самих података. С обзиром да су подаци у већини случајева вектори стварних вредности, Еуклидска удаљеност између података може послужити као мера те различитости.

За меру $d: \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ кажемо да представља меру различитости (растојања, метрике), објеката $x_i \in \mathbb{R}^n$ и $y_j \in \mathbb{R}^n$ у ознаци $d_{ij} = d(x_i, y_j)$, где је $0 \leq i, j \leq n$, ако задовољава следеће особине за све објекте:

1. $d_{ij} \geq 0$ (услов ненегативности)
2. $d_{ij} = d_{ji}$ (услов симетричности)
3. $d_{ij} \leq d_{ik} + d_{kj}$ (услов неједнакости троугла)

За меру $s_{ij} = s(x_i, y_j)$ кажемо да представља меру сличности објеката x_i и y_j, где је $0 \leq i, j \leq n$, ако задовољава следеће особине:

1. $0 \leq s_{ij} \leq 1$ (услов нормираности)
2. $s_{ij} = 1$, само ако су објекти једнаки
3. $s_{ij} = s_{ji}$ (услов симетричности)

Треба узети у обзир да појединачне променљиве (елементи вектора) могу имати различит значај. Посебно, распон вредности би требало да буде на одговарајући начин градиран како би се добиле разумне вредности удаљености између података. Потешкоће све више долазе до изражаја, нарочито уколико се поред стварних варијабли узму у обзир и целобројне вредности или чак апстрактне класе.

Један од недостатака познатих растојања као што су: Еуклидско растојање, Менхетн растојање, растојање Минковског је што се могу применити само када нам је позната вредност свих параметара тј. особина које посматрамо код објеката. Зато дајемо предлог растојања које може да упоређује објекте код којих из неког разлога нису познати сви параметри.

Такође, поред проблема недостајућих података описаних у делу испод, унешени подаци су шифровани (табела 5. 1.), па се зато дошло на идеју коришћења Хаминговог растојања и предлога новог растојања које користи Хамингово.

Табела 5. 1. Приказ параметра који су испитивани код пацијената и њихово шифровање.

Параметри	**Вредност**
Име	
Пол	1-женски; 2-мушки
Године старости	1. <30; 2. 30-39; 3. 40-49; 4. 50 и више
Маларни раш	0. нема; 1. Има
Дискоидне промене по кожи	0. нема; 1. умерене; 2. изражене
Фотосензитивност	0. нема; 1. има
Ранице у усној дупљи	0. нема; 1. има
Сувоћа усне дупље преко 3м.	0. нема; 1. има
Артралгије	0. нема; 1. има
Артритис	0. нема; 1. умерен; 2. изражен
Сувоћа очију преко 3 месеца	0. нема; 1. има
Проксимална сцлеродерма	0. нема; 1. има

Склеродактилија	0. нема; 1. има
Ранице на врховима прстију	0. нема; 1. има
Раунауд	0. нема; 1. има
Ливедо Ретицуларис	0. нема; 1. има
Отежано гутање	0. нема; 1. има
Телеангиектазија	0. нема; 1. има
Фебрилност	0. нема; 1. има
Губитак на тежини (>5%)	0. нема; 1. има
Малаксалост	0. нема; 1. има
Опадање косе	0. нема; 1. умерено; 2. изражено
Лимфаденопатија	0. нема; 1. има
Епилепсија	0. нема; 1. има
Психијатријске тегобе	0. нема; 1. има
Психолошке тегобе	0. нема; 1. има
ЦВИ	0. нема; 1. има
Побачаји	0. нема; 1. има
Тромбозе	0. нема; 1. артеријске; 2. Венске
Емболије	0. нема; 1. артеријске; 2. Венске
Плеурални излив	0. нема; 1. има
Плућна фиброза	0. нема; 1. има
Калциноза	0. нема; 1. има
ТА	0. нормалан; 1. повишен Пд до 100; 2. повишен Пд >100
СЕ	0. нормалан; 1. <40; 2. 41-60; 3. 61-80; 4. >80
Фибриноген	0. нормалан; 1. повишен; 2. Изражен
Анемија	0. нема; 1. 100<Hb<120< 2. 80<Hb<100< 3. Hb<80
Леукопенија	0. нема; 1. 3000-4000; 2. 2000-3000; 3. <2000

Лимфопенија	0. нема;1. 1000-1500; 2. <1000
Тромбоцитопенија	0. нема; 1. 100-150; 2. 50-99; 3. <50
Фе	0. уредно; 1. смањено; 2. изразито смањено
Еритроцитурија	0. нема; 1. 5-10; 2. >10
Цилиндри	0. нема; 1. има
Протеинурија	0. нема; 1. 0,5-1гр/24ч; 2. 1-3,5гр/24ч; 3. >3,5гр/24ч
Пиурија	0. нема; 1. стерилна; 2. није стерилна
Цоомбсов тест	0. негативан; 1. Позитиван
РФ	0. негативан; 1. Позитиван
ЦРП	0. негативан; 1. позитиван; 2. изражен
АНА	0. негативан; 1. позитиван <1:80; 2. позитиван >1:80
ХЕп-2 АНА	0. негативан; 1. Позитиван
Антицентромерна антитела	0. нема; 1. има
АНЦА	0. негативан; 1. Позитиван
МПО	0. негативан; 1. Позитиван
ПР3	0. негативан; 1. Позитиван
анти См	0. негативан; 1. Позитиван
РнП	0. нема; 1. има
анти дс ДНА	0. негативан; 1. позитиван; 2. изражен
ССА	0. негативан; 1. позитиван; 2. изражен
ССБ	0. негативан; 1. Позитиван
Сцл-70	0. негативан; 1. Позитиван
АцлА ИгГ	0. негативан; 1. позитиван; 2. изражен
АцлА ИгМ	0. негативан; 1. позитиван; 2. изражен
б2ГПИ ИгГ	0. негативан; 1. позитиван; 2. изражен
б2ГПИ ИгМ	0. негативан; 1. позитиван; 2. изражен

ЛА	0. негативан; 1. Позитиван
ВДРЛ	0. негативан; 1. Позитиван
КЦТ	0. негативан; 1. Позитиван
Снижен комплемент	0. негативан; 1. Позитиван
Повишен ИгГ и/или ИгМ	0. негативан; 1. позитиван; 2. изражен
Криоглобулини	0. нема; 1. има
Парапротеин	0. негативан; 1. Позитиван
Кератоцоњуцтивитис сицца	0. нема; 1. има
Промене на очном дну	0. нема; 1. има
Очи остало	0. нема; 1. има
Капилароскопија	0. негативан; 1. Позитиван
Одређивање дифузије плућа	0. негативан; 1. Позитиван
Перикардни излив	0. нема; 1. има
Плућна хипертензија	0. нема; 1. има; 2. Значајно
Сцинтиграфија плућа	0. негативан; 1. Позитиван
Сцинтиграфија пљувачних жлезда	0. негативно; 1. снижен Ак.А Кт; 2. снижен секр. А; 3 снижен А и С
НМР ендокранијума	0. уредан; 1. позитиванваск.; 2. позитиван АФС
РТГ плућа	0. негативан; 1. Позитиван
РТГ шака	0. уредан; 1. дегенеративне промене; 2. Артритис
Поремећена пасажа једњака	0. нема; 1. Има
ЛБТ/ДИФ - биопсија коже	0. уредно; 1. има патологија
Биопсија мале пљувачне жлезде	0. уредно; 1. I, II; 2. III, IV

ПХ биопсије бубрега	0. уредно; 1. мезанцијални Н ИИ; 2. фокално пролиф. ИИИ;
ЕМНГ	3. дифузно пролиф. ИВ; 4.мембрански ГН В; 5. склероза (УЗ)
СБВТ	0. уредан; 1. миопатска; 2. Неуропатска

Да би дефинисали предложено растојање дефинишимо Hamming-ово растојање. Нека је F коначан скуп са q елемената.

Дефиниција 5. 1.: Hamming-ово растојање $d\left(\underline{x},\underline{y}\right)$ између два вектора $\underline{x},\underline{y}\in F^{(n)}$ је број места на којим се ова два вектора разликују.

Предложено растојање користи Hamming-ово растојање и исказне формуле. Коришћење исказних формула у дефиницији растојања је због општости растојања.

Нека су φ и ψ два скупа исказних формула којем припадају формуле које представљају коњункцију литерала. Предложено растојање између ова два скупа исказних формула дефинисано је са:

$$D(\varphi,\psi)=\frac{\max\limits_{A\in\varphi}\min\limits_{B\in\psi}d(A,B)+\max\limits_{B\in\psi}\min\limits_{A\in\varphi}d(A,B)}{2} \qquad (5.\ 1)$$

где је $d(A,B)$ Hamming-ово растојање.

Пре доказа да је предложено растојање метрика, уведимо следеће појмове. Постмарајмо исказне формуле над коначно много слова $p_1,\dots,p_n$. Нека је For скуп свих оваквих формула.

Означимо са $a_1,\dots a_N$, $N=2^n$, све атоме над $p_1,\dots,p_n$ тј. све формуле облика $p_1^{e_1}\wedge\dots\wedge p_n^{e_n}$, за $e\colon\{1,\dots,n\}\to\{0,1\}$ при чему је $p_k^1=p_k$ и $p_k^0=\neg p_k$. Нека је $A=\{a_1,\dots,a_N\}$. Скуп A се може посматрати и као скуп валуација слова $p_1,\dots,p_n$. За било коју формулу α чија су исказна слова међу словима $p_1,\dots,p_n$ и $a\in A$ нека је

$$Val(a\Rightarrow\alpha)=\begin{cases}1, & \vDash a\Rightarrow\alpha\\ 0 & \text{не важи да} \vDash a\Rightarrow\alpha\end{cases}$$

$Val(a\Rightarrow\alpha)$ је заправо истинитосна вредност формуле α при валуацији која одговара атому a. Ако је $Val(a\Rightarrow\alpha)=1$, пишемо и $a\vDash\alpha$.

Међу бинарним низовима дужине n на разне начине се може увести метрика.

На пример, функција $d: AxA \to [0, +\infty]$ дата са

$d(e, f) = |\{i \in \{1, \dots, n\} | e_i \neq f_i\}|$, $e, f: \{1, \dots, n\} \to \{0,1\}$, јесте метрика на A.

1. $d(e, f) = 0 \; akko \; e = f$,
2. $d(e, f) = d(f, e)$,
3. $d(e, f) \leq d(e, g) + d(g, f)$

Неједнакост тругла следи директно из чињенице $e_i \neq f_i \Rightarrow e_i \neq g_i \vee g_i \neq f_i$.

На више начина се задата метрика $d: AxA \to [0, +\infty]$ може са A проширити на For. Нека је $D: For \times For \to [0, +\infty]$ dato sa:

$$D(\alpha, \beta) = \frac{\max_{a \vDash \alpha} \min_{b \vDash \beta} d(a, b) + \max_{b \vDash \beta} \min_{a \vDash \alpha} d(a, b)}{2}$$

***ЛЕМА 5. 1.*:** D је метрика на For.

Доказ: Очигледно је $D(\alpha, \beta) = D(\beta, \alpha)$.

Докажимо да је $D(\alpha, \beta) = 0$ акко $\alpha \leftrightarrow \beta$

$$D(\alpha, \beta) = 0 \Leftrightarrow \max_{a \vDash \alpha} \min_{b \vDash \beta} d(a, b) = 0 \text{ и } \max_{b \vDash \beta} \min_{a \vDash \alpha} d(a, b) = 0$$

$$\Leftrightarrow \text{за свако } a \vDash \alpha, \min_{b \vDash \beta} d(a, b) = 0$$

$$\text{и за свако } b \vDash \beta, \min_{a \vDash \alpha} d(a, b) = 0$$

$$\Leftrightarrow \text{ за свако } a \vDash \alpha \text{ постоји } b \vDash \beta \text{ тако да је } d(a, b) = 0 \text{ тј. } a = b \text{ и}$$

$$\text{за свако } b \vDash \beta \text{ постоји } a \vDash \alpha \text{ тако да је } d(a, b) = 0 \text{ тј. } a = b$$

$$\Leftrightarrow \text{за свако } a \vDash \alpha, a \vDash \beta \text{ и за свако } b \vDash \beta, b \vDash \alpha$$

$$\Leftrightarrow \alpha \vDash \beta \text{ и } \beta \vDash \alpha$$

$$\Leftrightarrow \vDash \alpha \leftrightarrow \beta$$

(Неједнакост троугла) Нека су α, β и γ произвољне формуле. Тада

(1)$d(a, b) \leq d(a, c) + d(c, b)$, sa све $a \vDash \alpha$, $b \vDash \beta$, $c \vDash \gamma$.

За фиксирано $c \vDash \gamma$, нека је $a_c \vDash \alpha$ такав да је $d(a_c, c) \leq d(a, c)$, за свако $a \vDash \alpha$; дакле $d(a_c, c) = \min_{a \vDash \alpha} d(a, c)$. Како је $d(a_c, b) \leq d(a_c, c) + d(c, b)$, sa било које $b \vDash \beta$, имамо $\min_{a \vDash \alpha} d(a, b) \leq d(a_c, b) \leq d(a_c, c) + d(c, b) = \min_{a \vDash \alpha} d(a, c) + d(c, b)$.

Одавде следи да је

$$\min_{a \vDash \alpha} d(a, b) \leq \min_{a \vDash \alpha} d(a, c) + d(c, b)$$

$$\leq \max_{c\vDash\gamma} \min_{a\vDash\alpha} d(a,c) + d(c,b), \qquad \text{за све } b \vDash \beta, c \vDash \gamma \,.$$

Даље имамо

$$\min_{a\vDash\alpha} d(a,b) \leq \max_{c\vDash\gamma} \min_{a\vDash\alpha} d(a,c) + \min_{c\vDash\gamma} d(c,b)\,, \text{за све } b \vDash \beta.$$

Најзад, ако је b' такав да је $\min_{a\vDash\alpha} d(a,b') \geq \min_{a\vDash\alpha} d(a,b)$, за све $b \vDash \beta$, онда је

$$\max_{b\vDash\beta} \min_{a\vDash\alpha} d(a,b) = \min_{a\vDash\alpha} d(a,b')$$

$$\leq \max_{c\vDash\gamma} \min_{a\vDash\alpha} d(a,c) + \min_{c\vDash\gamma} d(c,b')$$

$$\leq \max_{c\vDash\gamma} \min_{a\vDash\alpha} d(a,c) + \max_{b\vDash\beta} \min_{c\vDash\gamma} d(c,b)$$

Дакле.

$$(2) \max_{b\vDash\beta} \min_{a\vDash\alpha} d(a,b) \leq \max_{c\vDash\gamma} \min_{a\vDash\alpha} d(a,c) + \max_{b\vDash\beta} \min_{c\vDash\gamma} d(c,b)$$

Полазећи од (1), аналогно добијамо

$$(3) \max_{b\vDash\beta} \min_{a\vDash\alpha} d(a,b) \leq \max_{a\vDash\alpha} \min_{c\vDash\gamma} d(a,c) + \max_{c\vDash\gamma} \min_{b\vDash\beta} d(c,b)$$

Из (2) и (3) следи да је $D(\alpha,\beta) \leq D(\alpha,\gamma) + D(\gamma,\beta)$[1] ■

5. 3. Проблем недостајућих података

Овим истраживањем, пошто база података која је коришћена у експерименталном делу има недостајуће податке за поједине атрибуте, дали смо предлог растојања на основу којег можемо попунити недостајуће податке.

Проблем недостајућих података је од великог значаја. Када се говори о механизмима недостајућих података треба утврдити прво разлог недостајања података. Подаци могу недостајати из више разлога. Неки од њих су: подаци нису расположиви, дошло је до грешака у раду са опремом, неконзистентности са другим подацима, па су зато избрисани, нису унешени због неразумевања, нису сматрани битним у тренутку уноса итд. Битна је одлука шта радити са недостајућим подацима?

[1] За доказ ове леме заслужан је проф. др Небојша Икодиновић

Избрисати параметре код којих се јављају недостајући подаци-што није препоручљиво посебно код класификације, нарочито ако недостајуће вредности варијају од параметра до параметра.

Ручно попуњавање недостајућих вредности је заморно и често неизводљиво.

Аутоматско попуњавање: неком општом константом, средњом вредности параметра за све који припадају истој класи.

Највероватнија вредност-закључак који се доноси на основу Бајесове формуле или стаблу одлучивања.

У бази са којом се радило у овом истаживању (база пацијената који болују од аутоимуних болести) до недостајућих података је дошло зато што за неке од пацијената није било потребе да се одраде одређене анализе, јер се пре њих установило о којој болести је реч или није било средстава зато што су поједине анализе скупе, а неки од података недостају због губитка тог податка у тренутку уноса у базу података (у отпусним листама на основу којих су уношени подаци дошло је до превида).

Да би превазишли ово ограничење, а да опет сачувамо компактност налаза тј. да не вршимо брисање налаза код којих постоје недостајући подаци, дали смо предлог горе дефинисаног растојања (5. 1).

Прецизност резултата кластеровања се може смањити уколико се у анализу укључе обележја која нису релевантна за поједине кластере. Одабир релевантних обележја углавном представља предкорак у поступку кластеровања. Пошто постоји и фактор различите значајности сваког од атрибута дали смо предлог рачунања тежинских коефицијената за сваки од њих са циљем издвајања атрибута који описују кластере.

Теорема 5. 5: Нека су $a_i, i = 1, \dots, n$, $a_i \in A$ атрибути који описују објекте $o_j = \left(a_1^j, a_2^j, \dots, a_n^j\right)$, $j = 1, \dots, m$, $o_j \in O$. Уколико су $C_l \subseteq O, l = 1, \dots, k$, кластери, тј. сваки o_j припада тачно једном C_l, $C_l \cap C_r = \emptyset$ за $l \neq r$, $\bigcup_{l=1}^{k} C_l = \boldsymbol{C}$. За свако $a_i, i = 1, \dots, n$ од атрибута $a_i \in A$ важи да је:

$$\lambda_i = max\left\{\frac{P(a_i|C_l)}{P(a_i|\neg C_l)}\right\} \geq 1, \ \forall i \in \{1, .., n\} \text{ и } \forall l \in \{1, \dots, k\} \tag{5. 2}$$

Доказ. Претпоставимо супротно

$$\frac{P(a_i|C_l)}{P(a_i|\neg C_l)} = \frac{\frac{P(a_i \cap C_l)}{P(C_l)}}{\frac{P(a_i) - P(a_i \cap C_l)}{1 - P(C_l)}} = \frac{P(a_i \cap C_l)\big(1 - P(C_l)\big)}{P(C_l)\big(P(a_i) - P(a_i \cap C_l)\big)} < 1 \Leftrightarrow$$

$$P(a_i \cap C_l)\big(1 - P(C_l)\big) < P(C_l)\big(P(a_i) - P(a_i \cap C_l)\big) \Leftrightarrow$$

$$P(a_i \cap C_l) - P(a_i \cap C_l)P(C_l) < P(C_l)P(a_i) - P(C_l)P(a_i \cap C_l) \Leftrightarrow$$

$$P(a_i \cap C_l) < P(C_l)P(a_i) \Leftrightarrow$$

Ако сумирамо по l и леву и десну страну неједнакости добијамо:

$$\sum_{l=1}^{k} P(a_i \cap C_l) < \sum_{l=1}^{k} P(C_l)P(a_i) \Leftrightarrow$$

$$P(a_i) < P(a_i)\sum_{l=1}^{k} P(C_l) \Leftrightarrow$$

$$P(a_i) < P(a_i)$$

Супротно претпоставци. Тиме је теорема доказана. ■

Што је λ_i веће већи је утицај атрибута на припадност кластеру. Из тог разлога λ_i се могу користити као тежине одговарајућих атрибута.

5. 4. Имплементација методе променљивих околина за проблем кластеровања

RVNS и BVNS примењени на проблем кластеровања пацијената имплементирани су на следећи начин. Најпре су израчуната растојања између пацијената применом формуле (5. 1). У фази предпроцесирања врсте матрице растојања (као и одговарајући индекси пацијената) сортиране су у неопадајућем поретку. Ови подаци коришћени су за ефикаснију имплементацију оператора размрдавања. Наиме, како се свако решење карактерише скупом центроида, оператор размрдавања састоји се у замени одговарајућег броја центроида. Прецизније, размрдавање у околини к подразумева да се центроиди замене случајно изабраним објектима који нису центроиди, а удаљени су највише k места од центроида кога мењају. У сваком кораку разматра се замена свих центроида, при чему до замене неће доћи уколико је случајно изабрани објекат најближи том центроиду (уствари то је сам тај центроид). Локално претраживање састоји се у систематској замени једног центроида објектом који није центроид. Оно полази од решења добијеног размрдавањем и извршава се по принципу најбољег побољшања (енг. Best Improvement) док год има побољшања.

RVNS и BVNS методе имплементиране су у C# програмском језику на рачунару HP-15-d055, под оперативним системом Windows 10 Pro (Glišović, N., et al., 2017b). С обзиром на стохастичку природу метода вршено је 100 рестартовања. а као критеријум заустављања задато је максимално време извршавања. Оно је постављено на време рада CPLEX комерцијалног солвера кад год је он налазио оптимална решења или на три минута за примере из литературе.

Све описане методе и алгоритме у овом делу истраживања применили смо на конкретном примеру, бази података пацијената са алергологије и имунологије, Клиничког центра Србије.

Пример: Нека су дата два објеката p_1 и p_2 са вредностима три налаза a, b и c тако да имамо следеће податке о објеката:

објекат p_1 има $a \wedge b$ (из неког разлога недостаје податак о вредности атрибута c)

објекат p_2 има $a \wedge \neg b \wedge c$

Ако применимо предложено растојање формула φ која описује први објекат је $a \wedge b \wedge (c \vee \neg c)$ ако претворимо само у конјункције имаћемо две формуле $a \wedge b \wedge c$ и $a \wedge b \wedge \neg c$ (те две формуле припадају скупу формула φ). Док скуп формула другог објеката тј. скупа ψ чини формула $a \wedge \neg b \wedge c$. Ако применимо предложено растојање на ова два објеката добићемо да је растојање између њих:

$$D(\varphi,\psi) = \frac{\max\limits_{A\in\varphi} \min\limits_{B\in\psi} d(A,B) + \max\limits_{B\in\psi} \min\limits_{A\in\varphi} d(A,B)}{2} =$$

$$= \frac{\max\left\{\min\left\{d(a\wedge b\wedge c, a\wedge\neg b\wedge c)\right\}, \min\left\{d(a\wedge b\wedge\neg c, a\wedge\neg b\wedge c)\right\}\right\} + \max\left\{\min\left\{d(a\wedge\neg b\wedge c, a\wedge b\wedge c), d(a\wedge\neg b\wedge c, a\wedge b\wedge\neg c))\right\}\right\}}{2}$$

$$= \frac{\max\left\{\min\left\{1\right\}, \min\left\{2\right\}\right\} + \max\left\{\min\left\{1,2\right\}\right\}}{2} = \frac{2+1}{2} = \frac{3}{2}.$$

Горе описан алгоритам (предложен од стане Chan, EZ, et al., 2004), заједно са предложеним растојањем за недостајуће податке имплементиран је у програмском језику C#. Успешност класификације, овако дефинисаног растојања, показана је кроз поређење резултата са другим методама попуњавања недостајућих података као што су средња вредност и линеарна регресија (Glišović, N., et al., 2016). Резултати поређења су да се најлошији у кластеровању показао метод средње вредности, па линеарна регресија која је била знатно боља од средње вредности и мање боља од предложено растојање. Проценти успешности дати су у табели 5. 2. Резултати су добијени на основу базе која се садржи 45 пацијената са 89 параметара (приказано у табела 5. 1.).

Табела 5. 2. Приказ успешности предложеног алгоритма.

Метод попуњавања недостајућих података	Средња вредност	Линеарна регресија	Наше растојање
Успешност у процентима	80%	93.33%	95.56%

Уколико се укључе тежине тј. примени се теорема 5. 5. успешност се побољша и тада се добије 97.78%.

Такође смо даље у истраживању испититали успешност BVSN и RVNS методе.

С обзиром на стохастичку природу метода вршено је 100 рестартовања, а као критеријум заустављања задато је време извршавања CPLEX комерцијалног солвера (1.09 с). Оптимално решење CPLEX-а је 82.19, што је и постигнуто и код обе варијанте VNS-а, али за знатно краће време (табела 5. 3). Напомињемо да су времена дата у табели 5. 3. изражена у милисекундама.

Имплементиране VNS методе показале су велику стабилност јер је од 100 пуштања у BVNS-у најбоље решење достигнуто 86 пута, док је код RVNS-а то било 83 пута. На основу резултата приказаних у табели 5. 3. може се закључити да је BVNS стабилнија варијанта док је RVNS значајније бржа уз незнатан губитак на стабилности.

Табела 5. 3. Резултати RVNS-а и BVNS-а.

Метода	**Вредност најбољег решења**	**Просечна вредност функције циља**	**Просечно време (мс)**
RVNS	82.19	86.66	0.13
BVNS	82.19	82.24	3.36

Ако се анализира успешност сврставања пацијената у одговарајуће кластере упоређена са оригиналном базом добије се да је код RVNS-а и BVNS-а 91.1% пацијената исправно одређена болест (од 45 пацијената 4 су распоређена у кластере којима не припадају). Медицински гледано, та четири болесника су била из групе лупуса (у изворној бази са том дијагнозом), а распоређени су у сјегрен кластер. Мада, уз консултовање стручњака из медицине ове две болести јесу блиске једна другој и дешава се у пракси да пацијенти са једном дијагнозом „пређу“ временом у другу (из лупуса у сјегрен и обрнуто) (Glišović, N., et al., 2017a). Ситуација се поправља ако се укључе тежине налаза (израчунате преко теореме 5. 5.), тада овај проценат се повећа са 91.1% на 93.33%.

Проблем недостајућих података разматран је и код поштанских података (Glišović, N., et al., 2017c). Методе су тестиране на базама података европске комисије[2] које садрже податке везане за поштански саобраћај. Постоји седам база које за сваку од земаља садрже следеће податке: укупни домаћи и инострани сервис услуга, промет поштанских услуга у домаћем сервису, цене домаћег поштанског сервиса, брзине добијања поште у домаћем и интернационалном сервису, приступне тачке домаћих поштанских сервиса, број запослених у домаћем поштанском сервису, број писама у домаћем и интернационалном сервису. Базе података обухватају податке за период 2004-2011 година за 31 земаљу и то: Аустрија, Белгија, Бугарска, Хрватска, Кипар, Чешка, Данска, Естонија, Финска, Бивша Југословенска Република Македонија, Француска, Немачка, Грчка, Мађарска, Исланд, Ирска, Италија, Латвија, Литванија, Луксембург, Малта, Холандија, Норвешка, Пољска, Португал, Румунија, Словачка, Словенија, Шпанија, Шведска, Велика Британија. Све коришћене базе карактеришу се

[2] http://ec.europa.eu/eurostat/web/postal-services/data/database

недостајућим подацима. Важно је напоменути да постоје земље за које нису наведени подаци ни за једну годину. Те земље искључене су из анализе у одговарајућој бази и подразумевано је да су оне сврстане у један заједнички кластер.

Након искључивања земаља код којих нису постојали подаци ни за једну годину одређен је проценат недостајућих података. Карактеристике сваке од база сумиране су у табели 5. 4.

Табела 5. 4. Опис сваке базе која је анализирана искауана кроз број земаља које се налазе у бази, број Земаља код којих нема података ни за једну годину (број искључених земаља), као и проценат недостајућих података.

Назив базе	**Број земаља**	**Број искључених земаља**	**Проценат недостајућих података**
Укупни домаћи и иностранi сервис услуга	31	8	30%
Промет поштанских услуга у домаћем сервису	30	0	14%
Цене домаћег поштанског сервиса	31	0	7%
Брзине добијања поште у домаћем и интернационалном сервису	31	0	10%
Приступне тачке домаћих поштанских сервиса	31	3	11%
Број запослених у домаћем поштанском сервису	31	0	11%
Број писама у домаћем и интернационалном сервису	31	3	12%

Прво смо уради претпроцесирање, статистичком анализом, које се састојало од одређивање броја група у кластер анализи. Користили смо нормирање података да би избегли велики утицај поједине променљиве која гравитира ка високим апсолутним вредностима код решавања проблема кластеровања.

Од метода нормирања података које се највише користе у обради података су:

- Мин-мах нормирање
- Z -скалирање
- Децимално скалирање

С обзиром на природу података који се користе у овом истраживању (постоје недостајуће вредности, тј. нису нам познате махимална и минимална вредност низа) определили смо се за Z скалирање, као методу адекватну у овом кораку обраде података.

$$\mathrm{y}' = \frac{y - y_s}{\sigma_y},$$

где је y' нова (нормирана вредност), y изворна вредност атрибута, y_s средња вредност, а σ_y стандардна девијација познатих (постојећих) атрибута. Средња вредност и стандардна девијација се рачунају на уобичајени начин:

$$y_s = \frac{1}{n}\sum_{i=1}^{n} y_i$$

$$\sigma_y = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - y_s)^2}$$

Статистичком анализом како података тако и њихове природе формиране су групе сличних земаља са сличним вредностима атрибута (одређени су интервали референтних вредности у свакој групи). Затим смо применили BVNS алгоритам за кластеровање у оквиру сваке базе података на онолико кластера колико је одређено у претпроцесирању података. Алгоритам је тестиран на рачунару истих карактеристика као и код базе пацијената. Резултати тестирања дати су у табели 5. 5.

Табела 5. 5. Резултати рада BVNS-а за сваку базу података дата по оптималним решењима, успешности и времену доласка до оптималних решења.

BVNS примењен на базама	**Најбоље решење**	**Број достигнутих најбољиих решења (успешност)**	**Време доласка ПРОСЕЧНО до оптималног решења (секундама)**
Укупни домаћи и иностранi сервис услуга	7954014.83	88	0.16
Промет поштанских услуга у домаћем сервису	20188.09	100	0.3
Цене домаћег поштанског сервиса	2.40	99	0.45
Брзине добијања поште у домаћем и интернационалном сервису	371.46	92	0.64
Приступне тачке домаћих поштанских сервиса	1163584.85	92	0.65
Број запослених у домаћем поштанском сервису	303556.29	100	0.14
Број писама у домаћем и интернационалном сервису	14360498.72	95	0.62

За сваку од ових база дајемо преглед кластера који су добијени где су различитим бојама представљени различити кластери (табела 5. 6.). Постигнута је прецизност кластеровања од 90%-96.96%.

Табела 5. 6. Кластери добијени применом методе BVNS. Различите боје приказују различите кластере. За сваку базу сврстане су земље у три кластера.

Укупни домаћи и инострани сервис услуга	**Промет поштанских услуга у домаћем сервису**	**Цене домаћег поштанског сервиса**	**Брзине добијања поште у домаћем и интернационалном сервису**	**Приступне тачке домаћих поштанских сервиса**	**Број запослених у домаћем поштанском сервису**	**Број писама у домаћем и интернационалном сервису**
Bulgaria	Belgium	Belgium	Belgium	Belgium	Belgium	Bulgaria
Czech Republic	Italy	Ireland	Czech Republic	Bulgaria	Spain	Czech Republic
Estonia	Denmark	Greece	Denmark	Czech Republic	Czech Republic	Denmark
Greece	Sweden	Denmark	Germany	Denmark	Denmark	Estonia

Croatia	Netherlands	Germany	Estonia	Estonia	Hungary	Ireland
Cyprus	Austria	Luxembourg	Ireland	Ireland	Netherlands	Greece
Latvia	Poland	Austria	Greece	Greece	Austria	Croatia
Lithuania	Bulgaria	Poland	Spain	Croatia	Poland	Cyprus
Luxembourg	Czech Republic	Portugal	Croatia	Cyprus	Finland	Latvia
Hungary	Estonia	Slovakia	Cyprus	Latvia	Sweden	Lithuania
Austria	Ireland	Finland	Latvia	Luxembourg	Romania	Luxembourg
Romania	Greece	Sweden	Lithuania	Hungary	Bulgaria	Hungary
Slovenia	Croatia	United Kingdom	Luxembourg	Malta	Estonia	Malta
Slovakia	Cyprus	Iceland	Hungary	Netherlands	Ireland	Austria

Sweden	Latvia	Bulgaria	Malta	Austria	Greece	Poland
Iceland	Lithuania	Czech Republic	Netherlands	Poland	Croatia	Portugal
Norway	Luxembourg	Estonia	Austria	Portugal	Italy	Romania
Former Yugoslav Republic of Macedonia	Hungary	Spain	Poland	Romania	Cyprus	Slovenia
Denmark	Malta	Croatia	Portugal	Slovenia	Latvia	Slovakia
Spain	Portugal	Cyprus	Slovenia	Slovakia	Lithuania	Finland
Poland	Romania	Latvia	Slovakia	Finland	Luxembourg	Sweden

Finland	Slovenia	Lithuania	Finland	Sweden	Malta	Iceland
Italy	Slovakia	Hungary	Sweden	United Kingdom	Portugal	Norway
	Finland	Malta	United Kingdom	Iceland	Slovenia	Former Yugoslav Republic of Macedonia
	Iceland	Netherlands	Iceland	Norway	Slovakia	Germany
	Norway	Romania	France	Former Yugoslav Republic of Macedonia	Iceland	Spain
	United Kingdom	Slovenia	Italy	Spain	Norway	Netherlands
	Germany	Former Yugoslav Republic of Macedonia	Norway	Germany	Former Yugoslav Republic of Macedonia, the	Italy

	Spain	France	Romania		Germany	
	France	Italy	Bulgaria		France	
		Norway	Former Yugoslav Republic of Macedonia, the		United Kingdom	

Овако гурписани подаци могу се даље користити у циљу што боље анализе земаља чији су се поштански сервиси и услуге разматрани. Предложени приступ може омогућити примену бенчмаркинг и других алата у циљу анализе поштанских сервиса када су расположиви подаци некомплетни.

У циљу упоређивања предложеног растојања као и методе VNS за кластеровање тестирање смо одрадили и на базама података доступним на интернету (енг. UCI Repository of machine learning databases for classification)[3] и упоредили резултате са постојећима из новије литературе (Yuan-Ting Yan et al., 2016.). Опис база дат је у табели 5. 7.

Табела 5. 7. Опис база колико имају инстанци, колико атрибута и класа.

Назив базе	**Број истанци**	**Број атрибута**	**Број класа**	**Број недостајућих података**
Anneal	898	38	6	37
Bands	540	39	2	66
B.cancer	699	10	2	2
Credit	690	15	2	8
CVR	435	16	2	73

[3] UCI Repository of machine learning databases for classification, http://archive.ics.uci.edu/ml/datasets.html

Dermatology	366	34	6	2
Heart-h	294	13	2	13
Heart-c	303	13	5	3
Hepatitis	155	19	2	21
H.colic	368	27	2	177
L.cancer	32	56	2	3
MM	961	5	2	10

Неке од база података су имале категоријалне порменљиве заступљене у већој мери, а пошто се нисмо бавили проблемима шифровања таквих променљивих као ни тражења мера сличности из тог разлога те базе нисмо тестирали. Применили смо VNS методу са растојањем за недостајуће податке које смо предложили и добили резултате који су показали да је наша предложена метода боља од метода које су предложили и тестирали Yuan-Ting Yan и сарадници у свом раду који се бави истим проблемима. Резултати ових тестирања дати су у табели 5. 8.

Табела 5. 8. Прецизност поређених метода. N/A вредност представља базе на којима нисмо тестирали VNS из разлога што су садржале категоријалне променљиве.

Назив методе / Назив базе података	**SNNE**	**MGNE**	**NNNE**	**Bag 1**	**Bag 2**	**NN**	**VNS**	**Стабилност (ВНС)**
Anneal	0.836	N/A	0.836	**0.901**	N/A	N/A	N/A	N/A
Bands	**0.773**	0.756	0.746	0.764	0.727	0.555	N/A	N/A

B.cancer	0.935	0.938	0.936	0.938	0.939	0.65	**0.956**	100
Credit	**0.860**	0.855	0.848	0.858	0.857	0.499	N/A	N/A
CVR	0.942	0.945	0.942	**0.965**	0.964	0.513	0.876	100
Dermatology	0.886	0.879	0.861	0.849	0.844	0.277	**0.891**	100
Heart-h	**0.816**	0.806	0.806	0.812	0.76	0.656	0.725	100
Heart-c	0.526	0.519	0.516	0.52	0.524	0.437	**0.680**	40
Hepatitis	0.676	0.676	0.682	0.663	0.681	0.551	**0.897**	100
H.colic	0.735	0.723	0.701	0.778	0.633	0.583	**0.922**	100
L.cancer	0.524	0.522	0.498	0.503	0.517	0.347	**0.719**	100
MM	0.834	**0.836**	0.801	0.829	0.84	0.504	0.649	100
Средња вредност	0.779	0.769	0.764	0.782	0.753	0.507	**0.813**	

Од девет база у осам је успешност од 100%, а просечна вредност функције циља је за око 4% боља него код најбоље методе из литературе.

ШЕСТО ПОГЛАВЉЕ: ЗАКЉУЧАК

Код проблема оптимизације односа корист/трошак моделирана неизвесност применом троугаоних фази бројева је омогућила да се побољша модел линеарног програмирања из литературе у смислу да се неизвесност, која се појављује како у функцији циља тако и у ограничењима, третира за сваки параметар ограничења и/или за сваки коефицијент функције циља и/или за сваки коефицијент ограничења. Добијени резултати су показали да су предложене методе врло сличних перформанси, а да су супериорне у односу на постојеће методе из литературе.

Главни недостатак алгоритама је потреба да се дефинишу почетна стања која задовољавају ограничења, за сваку промењљиву. Још једна мана је избор одговарајућег почетне вредности температуре (T) (када је у питању алгоритам симулираног каљења), односно, одговарајућег броја генерација (када је у питању генетски алгоритам), јер „погрешан одабир“ ових вредности може довести до дуже претраге. Ова природа конвергенције је типичан аспект недетерминистичких метода претраге, као што су симулирано каљење и генетски алгоритам.

Анализа односа корист/трошак је изведена и у условима задовољења ограничења и времена постизања резултата насупрот вредности функције циља. Треба уочити да, уколико доносилац одлуке намерава да достигне бољи ниво задовољења ограничења (тј. и/или параметре) то ће довести до горих резултата у односу на вредност функције циља и обрнуто.

За будуће истраживање, код симулираног каљења, било би интересантно проширити ово изучавање кроз поређење ових добијених резултата са резултатима који се добијају ако узмемо друге вредности за контролне параметре алгоритма, углавном почетне вредности температуре, а исто тако и покушати да се прилагоди температура у складу са природом проблема. Код генетског алгоритма треба анализирати шта се дешава за ако се мењају вредности генетских оператора. Још један интересантан аспект за будући рад би био оцењивање (процена) понашања алгортима у решавању нелинеарних проблема.

За проблем кластеровања када су подаци непотпуни постигнути теоријски доприноси су потврђени кроз праксу. Експериментални резултати показали су висок степен успешности како на конкретним проблемима тако и на проблемима којима су се бавили други научници. У поређењу са резултатима из литературе, предложена метода променљивих околина постигла је значајно боље резултате.

Проблем који збуњује истраживаче кластер анализе је одређивање коначног броја образованих кластера (познато као правило заустављања или критеријум заустављања). Нажалост, не постоји објективна стандардна процедура и не постоји интерни статистички критеријум за решавање овог проблема. Највећи недостатак је то што истраживачи морају да укључе ad hock методе које су иначе релативно комплексне. Једна врста критеријума заустављања је релативно просто истраживање мера сличности или растојања између кластера у сваком узастопном кораку, са дефинисаним кластер решењима када је мера сличности једна одређена вредност.

Постоји одређени број специфичних процедура које су предложене али се ни једна није показала као најбоља у свим ситуацијама. Такође, истраживачи морају дати чврсте процене, са концептом теоријских односа који може предложити природан број кластера. Може се покренути процес у којем одређени критеријуми, на основу практичних испитивања, показују да резултати морају бити прегледни и разумљиви за употребу када се одреди природан број кластера, тј. од 3–6, и тада најбоље решење за овај број кластера представља избор најбољег резултата међу свим добијеним за одабране бројеве кластера. У коначној анализи је вероватно најбоље да се узме један број кластер решења (нпр. 2, 3, 4) и да се донесе одлука са свим решењима, користећи априори критеријуме и практичну оцену, здрав разум или теоријске оцене. Кластер решења ће бити побољшана када се нађу решења за концептуалне аспекте проблема.

Основни проблем јесте где повући црту, тако да остане оптимални број кластера. Треба рећи да овај проблем нема задовољавајуће решење. Итеративне методе захтевају од корисника да унапред одреди број кластера. У статистичком смислу нулта-хипотеза о непостојању структуре унутар неког скупа објеката није сасвим јасна, па ни смислена.

У друштвеним наукама доминирају два приступа одређивању броја кластера: хеуристички приступ и формални тестови. Први приступ је најчешћи, а односи се на субјективно постављање границе на дендрограму добијеном хијерахијском кластеризацијом. Основни критеријум јесте смисленост или интерпретабилност добијеног решења. Други начин, подједнако субјективан јесте анализа коефицијената (коефицијенти фузије) који показују сличности међу кластерима при сукцесивном спајању кластера. Нагло опадање (или повећање вредности код мера удаљености) указује на мању повезаност међу кластерима који се спајају. Нагли скок указује на спајање два релативно различита кластера.

На крају, важно је закључити да кластер анализа даје истраживачима емпиријску и објективну методу за извођење једног од најбитнијих задатака као што је груписање. Да ли за сврху упрошћавања, истраживања или потврде, анализа груписања је један врло моћан аналитички апарат који има врло широку примену. Ипак, ова техника повлачи одговорност истраживача, па је нужна одређена доза опреза приликом њеног коришћења. Уколико се правилно користи, ова анализа има потенцијал да открије податке који до тада нису откривени помоћу других метода. Такође, правилно руковање захтева велико знање, како се због лоше употребе не би јавили погрешни резултати и закључци.

У овом делу истраживања и, у вези са проблемом постављања дијагнозе пацијентима,

су два циља:

- Први циљ је био да се пронађе што бољи начин превазилажења недостајућих података, ради прецизније класификације и развоја метода за класификацију пацијената која би омогућила лакшу и бољу анализу података и развила систем за одлучивање.
- Други циљ је био да се да предлог кластеровања са недостајућим подацима, тј. да се овај NP тежак проблем имплементира кроз метахерустичку методу промењљивих околина (VNS) и покаже њена успешност у решавању овог проблема на конкретном проблему кластеровања пацијената који болују од аутоимуних болести. Резултати приказани у табели 5. 3. указују да су методе засноване на VNS метахеуристици достигле оптимално решење (добијено помоћу CPLEX солвера) за значајно краће време.

Литература

[1] Aarts, E., and Lenstra, J., (1997), *Local Search in Combinatorial Optimization*, John Wiley & Sons, Inc New York, ISBN: 0471948225.

[2] Aggarwal, C., (2003), Towards systematic design of distance functions for data mining applications, *KDD Conference*.

[3] Allison, PD., (2001), *Missing data*, Sage university papers series on quantitative applications in the social sciences. Sage, Thousand Oaks.

[4] Al-Sultan, K.S., (1995), A tabu search approach to the clustering problem, *Pattern recognition*, 28 (9), 1443–1451.

[5] Bandyopadhyay, S., and Maulik, U., (2002), An evolutionary technique based on Kmeans algorithm for optimal clustering in R^N, *Information Science*, 146, 221–237.

[6] Banfield, J., and Raftery, A., (1993), Model-Based Gaussian and Non-Gaussian Clustering, *Biometrics*, Vol. 49, No. 3, pp 803–821.

[7] Barber, T. J., and Boardman, J. T., (1988), Knowledge –Based Project Control Employing Heuristic Optimisation, *IEE Proceedings*, 135(8): 529 – 538.

[8] Bellman, R.E., and Zadeh, L.A., (1970), Decision making in a fuzzy Environment, *Management Science*, 17 (4), pp 141-164.

[9] Bezdek, JC, (1980), A converegence theorem for the fuzzy ISODATA clustering algorithms, *IEEE Trans Pattern Anal Mach Intell*.;2:1-8.

[10] Blazewics, J., Ecker, K., Pesch , E., Schmidt , G. and Weglarz , J., (1996), *Scheduling Computer and Manufacturing Processes*, Springer-Verlag.

[11] Bose, S., Das, C., Dutta, S., and Chattopadhyay, S., (2012), A novel interpolation based missing value estimation method to predict missing values in microarray gene expression data, *In: Proceedings of ernational conference on communications, devices and intelligent systems (CODIS)*, pp 318–321.

[12] Chan, EZ, Ching. WK, Ng, MK, and Huang, JZ., (2004), An optimization algorithm for clustering using weighted dissimilarity measures, *Pattern Recognit.*; 37: 943-952.

[13] Chen, S., and Tsai, M., (2011), Time-cost trade-off analysis of project networks in fuzzy environments, *European Journal of Operational Research*, 212, pp 386-397.

[14] Chen, SM, and Chen, HH, (2000), Estimating null values in the distributed relational databases environments, *Cybern Syst.*, 31 (8):851–871.

[15] Chiu, Y.S. P., and Chiu, S. W., (2005), Incorporating expedited time and cost of the end product into the product structure diagram, *International Journal of Machine Tools & Manufacture*, Vol: 45, pp. 987 – 991.

[16] Das, G., and Mannila, H., (2000), Context-based similarity measures for categorical databases. *PKDD Conference*, pages 201–210.

[17] Davidović, T., (2006), Raspoređivanje zadataka na višeprocesorske sisteme primenom metaheuristika [докторска дисертација]. [Београд]: Математички факултет.

[18] De Jong K. A., (1975), An analysis of the behaviour of a class of genetic adaptive systems, Doctoral dissertation, University of Michigan.

[19] De, P., Dunne, E. J., Ghosh, J. B., and Wells C. E. (1995), The discrete time-cost trade-off problem revisited, *European Journal of Operational Research*, 81: 225– 238.

[20] Demeulemeester, E. L., Herroelen, W. S., and Elmaghraby, S. E., (1996), Optimal procedures for the discrete time cost trade off problem in project networks, *European Journal of Operational Research*, 88: 50 – 68.

[21] Donders, AR, van der Heijden, GJ, Stijnen, T., Moons, KG, (2006), Review: a gentle introduction to imputation of missing values, *J Clin Epidemiol*, 59(10):1087–1091.

[22] Dorigo, M., Di Caro, G., Gambardella, L.M., (1999), Ant algorithms for discrete optimization, *Artif. Life*, 5 (2), pp. 137–172.

[23] Dorigo, M., Maniezzo, V., Colorni, A., (1996), Ant system: optimization by a colony of cooperating agents,IEEE Trans. Syst. Man Cybern., 26, pp. 29–41.

[24] Dugošija, Đ., (2011), *Linearno programiranje*, Zavod za udžbenike Beograd, ISBN: 978-86-17-17051-4.

[25] Elbeltagi, E., Hegazy, T., and Grierson, D., (2005), Comparison among five evolutionary-based optimization algorithms, *Advanced Engineering Informatics*, 19: 43 – 53.

[26] Eshtehardian, E., Afshar, A., and Abbasnia, R., (2008), Time-cost optimization: using GA and fussy sets theory for uncertainties in cost, *Construction Management and Economics*, 26: 679 – 691.

[27] Feng, C. W., Liu, L., and Burns, S. A., (1997), Using genetic algorithms to solve construction time-cost trade-off problems, *Journal of Computing in Civil Engineering*, 11(3): 184 – 189.

[28] Fern, X. and Brodley, C., (2003), Random projection for high dimensional data clustering: a cluster ensemble approach, *In Proceedings of ICML 2003*, pages 186-193.

[29] Fern, X. and Lin, W., (2008), Cluster Ensemble Selection, *Statistical Analysis and Data Mining*, 1(3):128-141.

[30] Fernández Pierna, J.A., Massart, D.L., (2000), Improved algorithm for clustering tendency, *Anal. Chim. Acta* 408 13–20.

[31] Filho, J., Treleaven, P., and Alippi, C., (1994), Genetic-algorithm programming environments, *IEEE Computer Society*, pp. 28-43.

[32] Fondahl, J. M., (1961), A non-computer approach to the critical path method for the construction industry, *Technical Report, No. 9, Construction Institude*, Department of Civil Engineering, Stanford University.

[33] Fredizzi, M., Kacprzyk, J., and Verdegay, J. L. (1991), A survey of fuzzy optimization and mathematical programming, In M. Fedrizzi & Kacprzyk (Eds.), *Interactive Fuzzy Optimization*, Springer-Verlag, pp 15-28.

[34] Garey, M., Johnson, R., David, S., (1979), *Computers and intractability: A guide to the theory of NP-completeness*, W. H. Freeman, ISBN 0-7167-1045-5.

[35] Glišović, N., and Rašković, M., (2016), The system for decision support in the diagnosis of systemic autoimmune diseases", *4rd Int. Conf. "Contemporary Problems of Mathematics, Mechanics and Informatics",* The State University of Novi Pazar, 19, 20 and 21. June 2016, Novi Pazar, Serbia.

[36] Glišović, N., (2013), Time-cost trade-off analysis of project using fuzzy-genetic approach, *Economics and social science*, vol. 13 br. , pp. 121-126.

[37] Glišović, N. (2014), Comparison of a fuzzy genetic and simulated annealing algorithm approach for project time-cost tradeoff, *Journal of Applied Mathematics*, 07/2014; (Article ID 817921):12.

[38] Glišović, N., Davidović, T., Bojović, N., Knežević, N.,(2017), Statističke i matematičke metode za rešavanje problema klasterovanja poštanskih podataka kada su oni nepotpuni, *XXXV Simpozijum o novim tehnologijama u poštanskom i telekomunikacionom saobraćaju, PosTel 2017,* pp 23-31, 5-6 decembra.

[39] Glišović, N., Davidović, T., Rašković, M., (2017), Klasterovanje kada podaci nedostaju korišćenjem metode promenljivih okolina, *XLIV Simpozijum o operacionim istraživanjima SYMOPIS*, pp. 158-165, Zlatibor, 25-28 septembra, ISBN 978-86-7488-135-4.

[40] Glišović, N., Rašković, M., (2017), Optimization for classifying the patients using the logic measures for missing data“, *Scientific publications of the State University of Novi Pazar*, vol. 9 no. 1, ISSN 2217-5539 (Print), ISSN 2466-3778.

[41] Glišović, N., Ribeiro, R. A., Milenković, M., Bojović, N., Petrović, V., (2012), A SA-based solution procedure for fuzzy time-cost tradeoff, *International Scientific Conference From Global Crisis to Economic Growth Which Way to Take?*, Faculty of Economics, Belgrade, 20-22.09.

[42] Goyal, S. K., (1975), A note on a simple CPM time-cost tradeoff algorithm, *Management Science,* Vol. 21 (6), pp. 718 – 722.

[43] Goyal, S. K., (1996), A simple time-cost tradeoff algorithm, *Production Planning & Control*, 7(1): 104 – 106.

[44] Gunopulos, D., and Das, G., (2001), Time-series similarity measures, and time series indexing, *ACM SIGMOD Conference*.

[45] Hadjitodorov, S., Kuncheva, L. I., and Todorova, L. P., (2006), Moderate Diversity for Better Cluster Ensembles, *Information Fusion Journal,* 7(3): 264-275.

[46] Hansen, P., Mladenović, N., (1997), Variable neighborhood search for the *P*-median, *Locat Sci*, 5: 207–226.

[47] Hansen, P., Mladenović, N., (2001), Variable neighborhood search: principles and applications for the *p*-median, *Eur J Oper Res*, 130:449–467.

[48] Hanss, M., (2005), *Applied Fuzzy Arithmetic: An Introduction with Engineering Applications*, Springer.

[49] Jain, A., (2010), Data clustering: 50 years beyond *k*-means, *Pattern Recognition Letters*, 31(8): 651– 666.

[50] Jiang, J.-H., Wang, J.H., Chu, X., Yu, R.-Q., (1997), Clustering Data using a Modified Integer Genetic Algorithm (IGA), *Anal. Chim. Acta* 354 (1997) 263–274.

[51] John C. Nash, (2000), The (dantzing) simplex method for linear programming, University of Ottawa.

[52] Kao, Y.-T., Zahara, E., and Kao, I.-W. (2008). A hybridized approach to data clustering, *Expert Systems with Applications*, 34(3), 1754–1762.

[53] Karypis, G. and Kumar, V., (1998), Multilevel k-way partitioning scheme for irregular graphs, *Journal of Parallel and Distributed Computing*, 48(1):96–129.

[54] Kelly, J.E. (1961), Critical path planning and scheduling: mathematical basis, *Operations Research*, 9, pp 167–179.

[55] Kirkpatrick, S., Gelatt, CD., and Vecchi, M.P., (1983), Optimization by simulated annealing, *Science* 220 (1983), 4598, 671-680.

[56] Klement, E. P., Mesiar, R., and Pap, E., (2000), *Triangular Norms*, Kluwer Academic Publishers, Dordrecht.

[57] Kratica, J., (2000), Paralelizacija genetskih algoritama za rešavanje nekih NP-kompletnih problema, Doktorska disertacija, Beograd.

[58] Krishna, K., and Murty (1999), Genetic K-means algorithm, *IEEE Transactions on Systems Man and Cybernetics B Cybernetics*, 29, 433–439.

[59] Lai, Y., and Hwang, C., (1992), *Fuzzy mathematical programming*, Methods and Applications, Springer-Verlag.

[60] Lai, Y.J., and Hwang, C.L., (1994), *Fuzzy multiple objective decision making*, Methods and Applications, Springer-Verlag.

[61] Li, H., and Love, P., (1997), Using improved genetic algorithms to facilitate time-cost optimization, *Journal of Construction Engineering and Management*, 123(3): 233 – 237.

[62] Li, H., Cao, J. N., and Love, P. E. D., (1999), Using machine learning and GA to solve time-cost trade-off problems, *Journal of Construction Engineering and Management*, 125(5): 347 – 353.

[63] Little, RJA, Rubin, DB, (2002), *Statistical analysis with missing data*, 2nd edn. Wiley, New York.

[64] Luke, S., (2009), *Essentials of metaheuristics*, (Vol. 113). Raleigh: Lulu.

[65] Martini, R., and Reinelt, G., (2011), The linear ordering problem exact and heuristic methods in combinatorial optimization, *Applied Mathematical Sciences*, Springer-Verlag Berlin Heidelb erg, vol. 17.

[66] Milenković, M., Bojović, N., Ribeiro, R.A., and Glisović, N., (2012), A fuzzy simulated annealing approach for project time-cost tradeoff, *Journal of Intelligent & Fuzzy Systems,* 23, pp. 1-13.

[67] Mladenović, N. and Hansen, P., (1997), Variable neighborhood search, *Computers and Operations Research*, 24(11); pp 1097–1100.

[68] Mladenović, N., (1995), A variable neighbodhood algorithm – a new metaheuristic for combinatorial optimization, *Abstracts of papers presented at Optimization Days*, Montreal, 112.

[69] Mladenović, N., Hansen, P., (2005), Variable neighborhood search. in: Burke E. K. Kendall G. Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques: Boston, MA: Springer US; pp 211-238.

[70] Mladenović, N., Hansen, P., (2007), Variable neighborhood search: Principles and applications, *European Journal of Operational Research*., 2007;130: pp 449-467.

[71] Moselhi, O., and Lorterapong, P., (1993), Least impact algorithm for resource allocation, *Canadian journal of civil engineering*, Vol: 20(2), pp. 180 – 188.

[72] Moussourakis, J., and Haksever, C., (2004), Flexible model for time/cost tradeoff problem, *Journal of Construction Engineering and Management*, 130, pp 307-314.

[73] Mualik, U., and Bandyopadhyay, S., (2002), Genetic algorithm based clustering technique, *Pattern Recognition*, 33, 1455–1465.

[74] Murthy, C. A., and Chowdhury, N., (1996), In search of optimal clusters using genetic algorithms, *Pattern Recognition Letters*, 17, 825–832.

[75] Pap, E., (1999), *Fazi mere i njihova primena*, Univerzitet u Novom Sadu, Prirodno-matematički fakultet, Novi Sad.

[76] Pires, F. M., Pires, J. M., Ribeiro, R. A., (1996), Solving fuzzy optimization problems: flexible approachesusing simulated annealing, P*roceedings of the World Automation Congress (WAC'96),* Montpellier, France, May.

[77] Pires, F., Ribeiro, R., (1998), A new risk function for fuzzy linear programming, *Proceedings of the World Automation Congress (WAC'98)*, Alaska, May, 138, pp. 1-10.

[78] Reeves, A., (1995), *Modern heuristic techniques for combinatorial problems*, McGraw-Hill.

[79] Ribeiro, R. A., Pires, F. M., (1999), Fuzzy linear programming via simulated annealing, *Kybernetica*, 35 (1), pp. 57-67.

[80] Ribeiro, R.A., and Varela, L.R., (2003), Fuzzy optimization using Simulated Annealing: An example Set, *In Fuzzy Sets Based Heuristics for Optimization*, Studies in Fuzziness and Soft Computing Series, Springer, 126, pp 159-180.

[81] Rothlauf, F., (2011), *Design of modern heuristics: principles and application*, Springer Science & Business Media.

[82] Schneider, J. and Kirkpatrick, S., (2007), *Stochastic optimization*, Springer Science & Business Media.

[83] Selim, S. Z., and Al-Sultan, K., (1991), A simulated annealing algorithm for the clustering problem, *Pattern Recognition*, 24(10), 1003–1008.

[84] Selim, S. Z., and Ismail, M. A., (1984), K-means type algorithms: A generalized convergence theorem and characterization of local optimality, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6, 81–87.

[85] Shelokar, P. S., Jayaraman, V. K., and Kulkarni, B. D. (2004), An ant colony approach for clustering, *Analytica Chimica Acta*, 509, 187–195.

[86] Siemens, N., (1971), A Simple CPM time-cost trade off algorithm, *Management Science*, 17(6): B-354 – B-363.

[87] Sorensen, K., (2012), Metaheuristics – the metaphor exposed, *International Transportations in Operational Research* doi:10.1111/itor.12001.

[88] Suhanic, G., (2001), *Computer-aided project management*, first addition, Oxford University Press, Inc. New York.

[89] Sun, L.-X., Xie, Y.-L., Song, X.-H., Wang, J.-H., Yu, R.-Q., (1994), Cluster analysis by simulated annealing, *Comput. Chem.*, 18 103–108.

[90] Sung, C. S., and Jin, H. W., (2000), A tabu-search-based heuristic for clustering, *Pattern Recognition*, 33, 849–858.

[91] Szczubialka, K., Verdú-Andrés, J., Massart, D.L., (1998), A new method of detecting clustering in the data, *Chemometr. Intell. Lab. Syst.* 41 145–160.

[92] Talbi, E.G., (2009), *Metaheuristics: from design to implementation*, (Vol. 74), John Wiley & Sons.

[93] Teodorović, D., Lucic, P., et al. (2006), Bee colony optimization: Principles and applications, *In Neural network applications in electrical engineering*, NEUREL 2006, pp. 151–156.

[94] Tran, T.N., Wehrens, R., Buydens, L.M.C., (2003), Knn density-based clustering for high dimensional multispectral image, In: Proc, *2nd GRSS/ISPRS Joint Workshop on Remote Sensing and Data Fusion over Urban Areas Workshop*, pp. 147–151.

[95] Vanhoucke, M., (2005), New computational results for the discrete time/cost tradeoff problem with time-switch constraints, *European Journal of Operational Research*, 165: 359 – 374.

[96] Vanhoucke, M., and Debels, D., (2007), The discrete time/cost trade off problem: extensions and heuristic procedures, *J Sched,* 10: 311 – 326.

[97] Varela, L.R., Ribeiro, R.A., and Pires, F.M., (2002), Simulated annealing & fuzzy optimization, *Proceedings of the 10th Mediterranean Conference on Control and Automation - MED2002*, Lisbon, Portugal, pp 1-9.

[98] Wang, F. and Sun, J., (2012), Distance metric learning in data mining, *SDM Conference (Tutorial)*.

[99] Welch, J.W., (1983), *Journal of Statistic and Computer Simulation*, 15, 17–25.

[100] Xing, B., and Gao, W.J., (2014), *Innovative computational intelligence: a rough guide to 134 clever algorithms*, Switzerland: Springer International Publishing.

[101] Yang, I.T., (2007), Performing complex project crashing analysis with aid of particle swarm optimization algorithm, *International Journal of Project Management*, 25, pp 637-646.

[102] Yang, I.T., (2007), Using elitist particle swarm optimization to facilitate bicriterion time-cost tradeoff analysis, *Journal of Construction Engineering and Management, ASCE*, 133, Pp 498-505.

[103] Yuan-Ting, Y., Yan-Ping, Z., Yi-Wen, Z., Xiu-Quan, D., (2016), A selective neural network ensemble classification for incomplete data, *Int. J. Mach. Learn. & Cyber., Springer*, doi 10.1007/s13042-016-0524-0.

[104] Zadeh, L., (1965), Fuzzy Sets, *Information and Control*, pp. 338-353.

[105] Zheng, D. X. M., Ng, S. T., and Kumaraswamy, M. M., (2004), Applying a genetic algorithm-based multiobjective approach for time-cost optimization, *Journal of Construction Engineering and Management*, 130 (2): 168 – 176.

[106] Zheng, D. X. M., Ng, S. T., and Kumaraswamy, M. M., (2005), Applying pareto ranking and niche formation to genetic algorithm-based multiobjective time-cost optimization, *Journal of Construction Engineering and Management*, 131(1): 81 – 91.

[107] Zimmermann, H., (1991), *Fuzzy Set Theory and its Applications*, Kluwer Academic Publishers.

Биографија аутора

Наташа Глишовић је рођена 17. октобра 1984. у Рашки. Основну школу *Иво Лола Рибар (садашњи назив Рашка)* и Гимназију у Рашки завршила је са одличним успехом, као носилац Вукове дипломе. На Математички факултет у Београду, смер Рачунарство и информатика, уписала се 2003. године. Дипломирала је 2009. године када је и стекла Звање дипломирани математичар.

Од септембра 2009. до децембра 2010. била је запослена у X Београдској Гимназији Михаило Пупин у Београду. Од јануара 2011. до септембра 2013. била је запослена на Математичком институту САНУ као истраживач-приправник. Од септембра 2013. до данас запослена је на Државном Универзитету у Новом Пазару (до фебруара 2016-е као истраживач-приправник, а од фебруара 2016-е као асистент) на Департману за Математичке науке. Од 2011. године до данас ангажована је на пројекту Министарства просвете, науке и технолошког развоја под називом „*Развој нових информаципно-комуникационих технологија, коришћењем напредних математичких метода, са применама у медицини, енергетици, е-управи, телекомуникацијама и заштити националне баштине“,* број 044006. Учесник је ERASMUS+ програма 2016.

Прилог 1.

Изјава о ауторству

Потписани-а НАТАША ГЛИШОВИЋ

број уписа 2035/2009

Изјављујем

да је докторска дисертација под насловом

ОПТИМИЗАЦИЈА ПРОБЛЕМА УПРАВЉАЊА ОДНОСИМА КОРИСТИ И ТРОШКОВА ПРИ РАСПОРЕЂИВАЊУ ПРОЈЕКАТА ПРИМЕНОМ МЕТАХЕУРИСТИЧКИХ АЛГОРИТАМА

- 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, 29.12.2017.

Прилог 2.

Изјава о истоветности штампане и електронске верзије докторског рада

Име и презиме аутора Наташа Глишовић

Број уписа 2035/2009

Студијски програм Математика

Наслов рада Оптимизација проблема управљања односима користи и трошкова при распоређивању пројеката применом метахеуристичких алгоритама

Ментор др Миодраг Рашковић

Потписани Наташа Глишовић

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, 29.12.2017.

Прилог 3.

Изјава о коришћењу

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

Оптимизација проблема управљања односима користи и трошкова при распоређивању пројеката применом метахеуристичких алгоритама

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, 29.12.2017.

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.