УНИВЕРЗИТЕТ У БЕОГРАДУ
МАТЕМАТИЧКИ ФАКУЛТЕТ

Бојана Ј. Лазовић

Примена метода комбинаторне оптимизације за решавање проблема формирања група у настави

Докторска дисертација

Београд, 2018.

UNIVERSITY OF BELGRADE
FACULTY OF MATHEMATICS

Бојана Ј. Лазовић

Combinatorial optimization methods for solving the problems of group formation in classes

Doctoral Dissertation

Belgrade, 2018.

Ментор:

др Мирослав Марић, ванредни професор

Универзитет у Београду, Математички факултет

Чланови комисије:

др Миодраг Матељевић, редовни професор

Универзитет у Београду, Математички факултет

др Милан Божић, ванредни професор

Универзитет у Београду, Математички факултет

др Зорица Станимировић, ванредни професор

Универзитет у Београду, Математички факултет

др Ћемал Долићанин, професор emeritus

Државни Универзитет у Новом Пазару

Датум одбране: ________________________

Овај рад посвећујем мом

највољенијем

брату

Александру Лазовићу,

који би био најсрећнији

и најпоноснији

ујко и брат…

Наслов докторске дисертације: Примена метода комбинаторне оптимизације за решавање проблема формирања група у настави

Резиме

Предмет овог рада је представљање нових математичких модела и алгоритама комбинаторне оптимизације, који се могу применити за решавање проблема формирања група у настави. Наиме, постоје разни проблеми који подразумевају издвајање одређених група индивидуа из коначног скупа, по унапред утврђеним критеријумима груписања. Неки од њих су NP-тешки проблеми комбинаторне оптимизације, разматрани у овој дисертацији: проблем максималне поделе скупа на два дела (енгл. Maximum Set Splitting Problem - MSSP), проблем формирања добро балансиране експерименталне и контролне групе (енгл. Well-Balanced Experimental and Control Group Formation Problem - WBECGFP), проблем формирања *k* добро балансираних група (енгл. Balanced Multi-Weighted Attribute Set Partitioning Problem - BMWASP), проблем формирања група за колаборативно учење (енгл. Collaborative Learning Groups Formation Problem - CLGFP) и проблем минималног репрезентативног скупа (енгл. Minimum Hitting Set Problem - MHSP). Сам процес формирања група представља комплексан и временски веома захтеван задатак, па је из тих разлога неопходна и софтверска подршка како би се он ефикасно и успешно извршио. Неки од проблема на које се може наићи у настави су еквивалентни наведеним NP-тешким проблемима и њиховим специјалним случајевима, посебно када је потребно узети у обзир велики број индивидуа, карактеристика и критеријума за њихово распоређивање у групе.

Циљ истраживања приказаног у овом раду је решавање проблема комбинаторне оптимизације: MSSP, WBECGFP, BMWASP, CLGFP и MHSP. Добијена решења разматраних проблема се могу применити за унапређивање процеса организовања и извођења наставе, процеса дељења и усвајања нових знања, за успешније извођење образовних експерименталних истраживања, за повећање мотивације код студената кроз групни и тимски рад, и све то у циљу што квалитетније наставе математике и рачунарства. С обзиром на разноврсне захтеве који се постављају од стране организатора наставе у погледу броја, величина и састава група које треба формирати, као и критеријума које при томе треба узети у обзир, ова дисертација даје практични допринос методологији што бољег распоређивања индивидуа у групе применом математичких модела и алгоритама комбинаторне оптимизације. Предложени алгоритами су имплементирани у јавно доступне апликације, како би се омогућила што једноставнија примена од стране корисника свих образовних профила.

За разматране проблеме оптимизације, неопходно је најпре дефинисати адекватне математичке моделе. Имајући у виду сложеност проблема, дизајниране су метахеуристичке методе за њихово решавање. Различите варијанте генетског алгоритма (енгл. Genetic Algorithm - GA) приказане су за решавање MSSP и MHSP проблема, док је за решавање проблема WBECGFP, BMWASP и CLGFP приказана метода промењљивих околина (енгл. Variable Neighbourhood Search - VNS), посебно прилагођена сваком од наведених проблема. Иако се представљена истраживања односе на решавање конкретних NP-тешких проблема и на распоређивање студената у специфичне групе, разматрани проблеми имају ширу примену. Предложени модели и методе могу се применити за решавање других проблема, као што су проблеми распоређивања: запослених, задатака, машина, пакета, путовања, часова, лекција, као и других објеката и субјеката.

У циљу потврде адекватности формулисаних математичких модела и ефикасности предложених метахеуристичких метода, спроведен је велики број експеримената. Предложени алгоритми су тестирани на одговарајућим тест примерима из литературе као и на новогенерисаним (јавно доступним) тест примерима. Добијени резултати су поређени са резултатима егзактних или већ постојећих метода, у односу на унапред дефинисане индикаторе поређења. Експериментални резултати су показали да предложени приступи имају боље перформансе у поређењу са другим методама и да могу да обезбеде високо квалитетна решења за различите димензије проблема, у разумном временском року.

Кључне речи: комбинаторна оптимизација, проблеми формирања група у настави, математичко моделирање, метахеуристичке методе, генетски алгоритми, метода промењљивих околина.

Научна област: математика

Ужа научна област: методика наставе математике и рачунарства

Dissertation title: Combinatorial optimization methods for solving the problems of group formation in classes

Abstract

The subject of this thesis is to present new mathematical methods and algorithms of combinatorial optimization, which could be applied for solving the problems of group formation in classes. Namely, there are various problems that require the selection of certain groups forming of individuals from the finite set, based on previously determined grouping criteria. Some of these are NP-hard problems of combinatorial optimization, which are taken into consideration in this thesis: Maximum Set Splitting Problem - MSSP, Well-Balanced Experimental and Control Group Formation Problem - WBECGFP, Balanced Multi-Weighted Attribute Set Partitioning problem – BMWASP, Collaborative Learning Groups Formation Problem - CLGFP and Minimum Hitting Set Problem - MHSP. The process of group formation represents a complex and time-consuming task, thus requiring a necessary software support for the efficient and successful completion of the task. Some of the problems that could come up in the course of teaching are equivalent to the aforementioned NP-hard problems and their special cases, especially when there is a need to take into consideration a large nunber of individuals, charactersitics and criteria for their assigning into groups.

The objective of the research presented in this thesis is solving combinatorial optimization problems: MSSP, WBECGFP, BMWASP, CLGFP и MHSP. The obtained results of the considered problems can be applied for: upgrading the process of the organization and performance of teaching, the process of splitting and the adoption of new knowledge, as well as to achive more successful performance of educational experimental researches, and to increase student’s motivation through group and team work. The objective is to achieve higher quality teaching of mathematics and computing. Taking into consideration various requests put forward by the organizers of teaching, in terms of number, size and group composition needed to be formed, as well as the criteria needed to be taken into account, this thesis provides a practical contribution to the methodology of optimal distribution of individuals into groups by applying mathematical

models and combinatorial optimization algorithms. The proposed algorithms are implemented in publicly available applications, such that users of all educational profiles are able to use them.

The first step in studying the considered optimization problems is to define adequate mathematical models. As the considered problems are very complex, metaheuristic methods are designed in order to find high quality solutions. Different versions of Genetic Algorithm - GA were presented for solving MSSP and MHSP problems, while a Variable Neighbourhood Search - VNS method was presented for solving WBECGFP, BMWASP and CLGFP problems, especially adapted to all of the aforementioned problems. Although the presented researches refer to solving concrete NP-hard problems, as well as to the distribution of students into specific groups, the considered problems have shown to have a wider application. The proposed models and methods can be applied for solving other problems, such as the problems of arrangement of employees, tasks, machines, packets, travel, classes, lectures, as well as organizing other objects and subjects.

In order to confirm the adequacy of the formulated mathematical models and efficiency of the proposed metaheuristic methods, a large number of experiments was conducted. The proposed algorithms have been tested on the relevant instances from literature as well as on newly generated (publically available) instances. The obtained results have been compared with the results of the exact or already existing methods, with reference to the previously defined comparison indicators. Experimental results have shown that the proposed approaches have a better performance compared to other methods and that they can provide high quality solutions for different problem dimensions, within a reasonable execution time.

Key words: Combinatorial Optimization, Problems of Group Formation in Classes, Mathematical Modelling, Metaheuristics, Genetic Algorithms, Variable Neighborhood Search

Scientific field: Mathematics

Scientific subfield: Methodology of teaching of mathematics and computer science

Захвалница

Овом приликом изражавам велику захвалност ментору, проф. др Мирославу Марићу, који је руководио израдом моје докторске дисертације и уваженим члановима комисије: проф. др Миодрагу Матељевићу, проф. др Милану Божићу, проф. др Зорици Станимировић, проф. др Ћемалу Долићанину. Такође, захваљујем се на пруженој подршци и сарадњи колегама: др Јозефу Кратици, др Драгану Ламбићу, др Александру Ђенићу и асистенту Душану Џамићу.

Посебну захвалност дугујем мојим родитељима, Јовану и Славици Лазовић, који су и у најтежим животним околностима имали неограничену количину љубави и храбрости да наставе свој живот због мене. Њихова подршка, бодрење и огромно и несебично ангажовање на чувању и одгајању моје деце, а њихове нове среће и радости, било ми је од највеће помоћи у настављању даљег научног, професионалног и личног усавршавања.

Нову срећу, осмех, радост, љубав и све најлепше што живот може да пружи, дугујем мом супругу Дарку и нашим лепотицама: Анђели и Андријани.

Садржај

1 Увод

1.1 Проблеми комбинаторне оптимизације

Математичка оптимизација подразумева поступак проналажења оптимума, односно екстремних вредности функције на неком скупу при задатим ограничењима. Нека је $f : S \rightarrow R$ реална функција дефинисана на скупу S и нека је $X \subseteq S$ скуп ограничења, тада се проблем оптимизације (минимизације) може формулисати на следећи начин:

$$min\, f(x) \tag{1.1}$$

при ограничењу:

$$x \in X. \tag{1.2}$$

Функција $f(x)$ се назива функција циља. Скуп S представља простор решења, а скуп свих ограничења дефинише простор допустивих решења X. Допустив скуп X је најчешће неки подскуп од R^n, из чега следи да је $x \in X$ заправо вектор $x = (x_1, \dots, x_n)$. Ако је X коначан или бесконачан пребројив скуп, у питању је проблем комбинаторне или дискретне оптимизације. У случају да X припада еуклидском простору R^n, реч је о континуалној или непрекидној оптимизацији. Потребно је пронаћи решење $x \in X$, такво да вредност функције f буде најмања. Може постојати више таквих решења, иако у пракси довољно пронаћи бар једно од њих. Допустиво решење $x^* \in X$ за које важи $(\forall x \in X)\ f(x^*) \leq f(x)$ назива се глобални минимум или оптимално решење. Остала допустива решења (локални минимуми) називају се субоптимална решења.

Проблем максимизације се може свести на проблем минимизације $max\, f(x) = -min\, f(x)$, па је довољно посматрати само један од проблема. Код проблема максимизације циљ је пронаћи глобални максимум, односно допустиво решење $x^* \in X$ за које важи $(\forall x \in X)\ f(x^*) \geq f(x)$.

У зависности од начина формулације, врсте променљивих и стратегија које се користе у процесу решавања проблема оптимизације, постоје два типа оптимизационих модела: модел за програмирање са ограничењима (енгл. Constraint Programming - CP model) и модел математичког програмирања (енгл. Mathematical Programming - MP model). Ова два модела имају исте структуре и исти циљ - проналажење оптималне вредности функције циља систематским одабиром вредности променљивих из скупа допустивих вредности уз поштовање задатих ограничења. Међутим, у односу на начин формулације, CP модел укључује логичка, аритметичка и функционална ограничења, док су у MP моделу заступљена искључиво аритметичка ограничења. Променљиве које фигуришу у аритметичким ограничењима CP модела су дискретне, док су у MP моделима оне дискретне и непрекидне [92]. У односу на стратегије решавања проблема, код CP модела се процесом логичког закључивања одбацују сувишне опције и на тај начин се долази до оптималног решења, док се код MP примењују алгебарске методе које могу бити егзактне или апроксимативне.

У зависности од различитих својстава функције циља и ограничења, постоји неколико врста проблема комбинаторне оптимизације. У случају да су функција циља и ограничења (задата у облику једнакости и/или неједнакости) линеарни, реч је о проблемима линеарног програмирања (енгл. Linear Programming - LP). У оквиру ових проблема често се може наићи на ограничења којим се захтева целобројност неких од координата вектора x. Тада су у питању проблеми мешовитог целобројног линеарног програмирања (енгл. Mixed Integer Linear Programming - MILP). У случају да се захтева целобројност свих координата вектора x (простор допустивих решења је потпростор скупа целих бројева), реч је о проблемима целобројног линеарног програмирања (енгл. Integer Linear Programming – ILP). Ако постоји ограничење којим се захтева да све координате вектора x буду 0 или 1, у питању је 0-1 целобројно линеарно програмирање. Постоје проблеми оптимизације код којих се функција циља не може дефинисати као линеарна. У случају да је f квадратна функција а ограничења линеарна, реч је о проблемима квадратног целобројног програмирања (енгл. Quadratic Integer Programming - QIP). Уколико су ограничења квадратна, проблеми се називају квадратно условљеним (енгл. Quadratically Constrained Programming - QCP).

1.2 Методе комбинаторне оптимизације

Методе комбинаторне оптимизације се деле у две категорије: егзактне и апроксимативне (хеуристичке и симулационе) методе. Теоријски, егзактне методе гарантују оптималност добијеног решења или показују да такво решење не постоји. Међутим, у пракси је њихова примена ограничена само на посебне MP моделе (моделе линеарног програмирања (LP) и моделе целобројног програмирања (IP)). Неке од познатијих егзактних метода које се примењују на LP моделе су: симплекс метода (енгл. Simplex Method) и метода баријере (енгл. Barrier Method) или метода унутрашње тачке (енгл. Interior Point Method), док се на IP моделе примењују: метода одсецања равни (енгл. Cutting Plane Method), метода гранања и ограничавања (енгл. Branch-and-Bound Method), метода гранања и одсецања (енгл. Branch-and-Cut Method) и метода гранања и оцењивања (енгл. Branch-and-Price Method). Више детаља о наведеним методама може се наћи у [42]. Егзактне методе се најчешће користе за добијање оптималних решења проблема на тест примерима мањих димензија.

Уколико не постоји адекватан математички модел и ако је број допустивих решења проблема комбинаторне оптимизације веома велик, примена егзактних метода неће бити од значаја, с обзиром да у највећем броју случајева неће дати резултат. У таквим ситуацијама примењују се апроксимативне методе, које се деле на две подкласе алгоритама: алгоритми апроксимације и хеуристички алгоритми. Иако не гарантују оптималност добијеног решења, предност хеуристичких алгоритама се огледа у проналажењу довољно доброг (високо квалитетног) допустивог решења (најчешће у разумном временском року) за проблеме великих димензија или за NP-тешке проблеме, за чије решавање не постоје алгоритми који се извршавају у полиномском времену. Више детаља о NP-тешким проблемима се може наћи у [73]. У случају да посматрани проблем има више решења, хеуристичком методом се проналазе само нека од тих решења, а не обавезно сва. Поред решења, неке од хеуристичких метода дају и његову прецизност у виду горње или доње границе удаљености од оптималног решења. Према томе, хеуристичке методе настоје успоставити баланс између оптималности и прецизности добијених решења, и времена потребног за њихово проналажење [180].

Хеуристичке методе се најчешће развијају за конкретан проблем или класу проблема, па је њихова примена на друге проблеме ограничена. Међутим, хеуристичке методе које се, уз посебна прилагођавања, могу применити за решавање великог броја оптимизационих проблема називају се метахеуристичке методе.

1.3 Метахеуристичке методе

Метахеуристичке методе су процедуре вишег нивоа (енгл. problem-specific higher-level procedures) специјално дизајниране за ефикасно проналажење довољно доброг решења проблема математичке оптимизације [20, 77]. Састоје се из скупа правила и општих алгоритама оптимизације који подразумевају примену итеративних поступака у процесу поправљања тренутног решења. Метахеуристичке методе не гарантују оптималност добијених решења. Међутим, у пракси се показало да метахеуристичке методе дају веома задовољавајуће резултате за велики низ различитих проблема у разумном временском року, са знатно нижом рачунарском сложеношћу у поређењу са методама исцрпљивања (енгл. exhaustive method) примењеним на исти проблем [138,179].

У зависности од својстава по којима се разврставају, савремене метахеуристичке методе се могу поделити на више начина [20, 77, 180].

- Према полазној мотивацији, издвајају се метахеуристичке методе које су инспирисане природним процесима (енгл. Nature-inspired Metaheuristics) и оне које нису инспирисане природним процесима. Првој класи метахеуристика припадају еволутивни алгоритми (енгл. Evolutionary Algorithms - EA) који су инспирисани процесом еволуције јединки, а у њих спадају: генетски алгоритам (енгл. Genetic Algorithm - GA), генетско програмирање (енгл. Genetic Programming - GP), еволутивно програмирање (енгл. Evolutionary Programming - EP) и други. Методе оптимизације ројевима честица (енгл. Particle Swarm Optimization - PSO), оптимизације мрављим колонијама (енгл. Ant

Colony Optimization - ACO) и оптимизације колонијама пчела (енгл. Bee Colony Optimization - BCO), такође су инспирисане природним појавама. Класи метахеуристика које нису инспирисане природним процесима припадају методе: табу претрага (енгл. Tabu Search - TS), диференцијална еволуција (енгл. Differential Evolution - DE), метода променљивих околина (енгл. Variable Neighbourhood Search - VNS), распршена претрага (енгл. Scatter Search), и друге.

- Према броју решења у оквиру једне итерације, метахеуристичке методе се деле на две класе: метахеуристике засноване на једном решењу (енгл. single point search) и метахеуристике засноване на популацији (енгл. population based). Методе које припадају првој класи користе стратегију засновану на праћењу трајекторије (енгл. trajectory methods) у оквиру које се низом итерација, код којих би требало да важи правило да је свако новодобијено решење боље од тренутног, долази до решења. Неке од ових метода у процесу претраге простора решења користе методу локалне претраге, како би у некој од околина тренутно најбољег решења пронашли ново, потенцијално боље решење. Такве метахеуристике су: метода променљивих околина, табу претрага, симулирано каљење (енгл. Simulated Annealing - SA) и друге. С друге стране, методе које се заснивају на популацији јединки у сваком тренутку располажу са више решења. У оквиру итерација, применом различитих оператора (селектовања, комбиновања, укрштања и измене) формирају се нове популације које би требало да имају квалитетније јединке од претходне популације. Неке од метахеуристика које припадају овој класи су: генетски алгоритми, оптимизација мрављим колонијама, оптимизација колонијама пчела и друге.
- Према приступу коришћења меморије, метахеуристичке методе се разврставају у две класе: метахеуристике које користе меморију (енгл. memory usage methods) и метахеуристике које не користе меморију (енгл. memoryless methods). Метахеуристике које припадају првој класи имају могућност да памте претходна решења и користе информације из меморије током процеса претраге. Један од представника прве класе је

метода табу претраге, док је метода симулираног каљења представник друге класе метода.

Поред наведених подела, метахеуристике је могуће поделити и на: конструктивне, „побољшавајуће“ и хибридне метахеуристике. Конструктивне метахеуристике постепено генеришу само једно допустиво решење. Један од представника ове класе јесте „похлепна“ метода (енгл. Greedy Search - GS). „Побољшавајуће“ метахеуристике имају за циљ да пронађу потенцијално боље решење од полазног произвољног решења, претраживањем његове околине. Процес претраживања се итеративно понавља док год у околини тренутног решења постоји потенцијално боље решење. Примери овох метода су: локална претрага (енгл. Local Search - LS), метода пењања на врх (енгл. Hill Climbing - HC), метода симулираног каљења и друге. У циљу повећања ефикасности метахеуристичких метода, дошло је до дизајнирања хибридних метахеуристика у виду комбинација разним метода. Неке од могућих комбинација су: VNS и TS, VNS и LS, GA и TS, GA и LS, и друге. Метахеуристичка метода која је заснована на GA и LS комбинацији, позната је под називом меметски алгоритам (енгл. Memetic Algorithm - MA).

Проблеми представљени у овој дисертацији решавани су уз помоћ две метахеуристичке методе: генетских алгоритама и методе променљивих околина, тако да су наредна два дела посвећена њиховим описима. Више детаља о осталим методама могу се наћи у [20, 77].

1.3.1 Генетски алгоритми

Генетски алгоритми (GA) су метахеуристичке методе које симулирају одређене спонтане процесе у оквиру природне селекције и репродукције. Идеја симулирања процеса еволуције и њена примена на решавање разноврсних проблема датира још из 40-их година 20 века. Тачније, већ 1948. године Alan Turing, отац модерног рачунарства, предлаже „генетско или еволутивно претраживање” које примењује и на конструисање „машине за учење”. У наредним годинама објављивани су радови у којима се уз помоћ рачунара симулира процес еволуције. Међутим, појам генетских алгоритама по први пут се уводи тек 1975. године са објављивањем књиге „Прилагодљивост у природним и вештачким системима” (енгл. Adaptation in natural and artificial systems) [96] чији је аутор John Holland.

Предложени генетски алгоритми представљени су као рачунарски процеси који симулирају еволутивне процесе популације јединки применом генетских оператора: прилагођености, селекције, укрштања и мутације. У овом процесу свака јединка се на јединствен начин може представити генетским кодом (низом гена). Квалитет јединке одређује се њеном прилагођеношћу окружењу (популацији) на начин да се јединке које су у већој мери прилагођене сматрају „квалитетнијим” у односу на остале јединке, и оне имају већу шансу за репродукцију применом генетских оператора на њихов код. У зависности од критеријума заустављања, овај процес се понавља одређени број пута. Сходно томе, формирају се нове генерације јединки које би требало да буду квалитетније у односу на претходне генерације.

Кодирање и рачунање функције циља

Централну структуру генетског алгоритма чини популација јединки. Почетна популација се најчешће формира на случајан начин (или помоћу неке хеуристичке методе), чиме се обезбеђује разноврсност генетског материјала потребног за проналажење оптималних решења. Популацију најчешће чини између 5 и 500 јединки, при чему јединке представљају решења у претраживачком простору датог проблема. Већи број јединки обезбеђује разноврсност генетског материјала, али може допринети и споријој конвергенцији ка оптималном решењу, с обзиром на број итерација (пролазака кроз све генетске операторе) и времена потребног да се изврши сваки од генетских оператора у оквиру итерације. Свака јединка представља могуће решење у претраживачком простору проблема који се решава. Јединка може бити представљена генетским кодом, при чему је избор функције кодирања један од пресудних фактора успешности GA. Кодирање се врши над одређеном коначном азбуком (најчешће бинарном, целобројном или мешовитом) и зависи од природе самог проблема. Наредни битан фактор успешне примене GA представља дефинисање функције циља којом се одређује вредност сваког решења, односно јединке која га представља. Рачунање вредности функције циља врши се за сваку јединку кроз сваку генерацију и управо овај процес може да изискује највећи временски део извршења GA.

Генетски оператори

Након рачунања вредности функције циља, над популацијом јединки се примењују генетски оператори: функција прилагођености, селекција, укрштање и мутација.

Функција прилагођености

Зависно од специфичности проблема који се решава, постоји већи број начина за одређивање функције прилагођености (енгл. fitness function) јединке у односу на целу популацију. Најчешће се користе разне врсте скалирања вредности функције циља јединке, попут: линеарног скалирања, директног или инверзног скалирања у неки интервал (најчешће у интервал [0,1]), као и сигма одсецање (енгл. sigma truncation scheme). Постоје и методе које се не имплементирају самостално, већ се реализују кроз операторе селекције, а то су: рангирање прилагођености јединке у популацији (енгл. fitness ranking) и имплицитна функција прилагођености (енгл. implicit fitness remapping). Одабрани начин рачунања прилагођености јединки се у великој мери рефлектује на перформансе GA. Више детаља о различитим начинима рачунања функције прилагођености јединки могу се наћи у [77, 118].

Селекција

Након мерења прилагођености јединки, примењује се оператор селекције (енгл. selection operator). Сходно проблему који се решава, бира се тип селекције којом се врши одабир јединки. Најчешће, јединке са ниским вредностима прилагођености имају мање шансе да буду изабране у односу на оне са високим вредностима прилагођености. Јединке које се одаберу оператором селекције даље производе потомство (нове јединке) применом оператора укрштања и мутације. Најпростији облик оператора селекције је проста рулет селекција (енгл. simple roulette selection) у оквиру које је избор јединке директно пропорционалан вредности њене прилагођености. Што је мера прилагођености већа, већа је и шанса да јединка буде изабрана. Међутим, овај тип селекције може довести до брзог губљења доброг генетског материјала који могу имати лошије јединке (јединке са мањом мером прилагођености) и фаворизовања добрих јединки (јединки са већом мером прилагођености), што даље може да доведе до практичне немогућности достизања глобалног оптимума, односно до појаве преурањене конвергенције. Из поменутих разлога, тип селекције који се доста често користи је турнирска селекција (енгл. tournament selection) у оквиру које се одржава онолико турнира колико је потребно да се изабере јединки које ће учествовати у укрштању. Величина турнира N_{tur} је параметар турнирске селекције и он представља укупан број јединки које учествују на турниру. За најчешће унапред задату величину турнира, на случајан начин се генеришу подскупови од по N_{tur} јединки за надметање. У сваком од подскупова симулира се одржавање турнира. Дакле, турнирска селекција се примењује N_{nnel} пута на скупу свих N_{pop} јединки у популацији да би се изабрала N_{nnel} родитеља за укрштање. Иста јединка из тренутне генерације може учествовати у више турнира. „Победник" турнира (јединка са највећом мером прилагођености) пролази селекцију и постаје кандидат за „родитеља" и учествује у стварању нове генерације. Преостале јединке турнира имају шансу да постану родитељи уколико победе на неком од других турнира. Што је већа величина турнира, већа је вероватноћа да боље јединке прођу селекцију, а да се лошије елиминишу. Стандардна турнирска селекција користи целобројну величину турнира, која у појединим случајевима може бити лимитирајући фактор. Избор параметара за турнирску селекцију своди се на две до три вредности, тако да се често дешава да за један параметар конвергенција буде веома спора, а његовим увећање за један, конвергенција буде преурањена па се за решење може узети локални екстремум. Једно од могућих унапређења турнирске селекције представља Фино градирана турнирска селекција (енгл. Fine-Grained Tournament Selection - FGTS), о којој ће бити више речи у одељку 2.4.3. Више детаља о осталим типовима селекције могу се наћи у [77, 118].

Укрштање

Оператор укрштања (енгл. crossover operator) представља оператор генетског алгоритма којим се формирају нове јединке. На случајан начин се упарују јединке које су прошле селекцију, затим се између њих врши размена генетског материјала и тако се добијају нове јединке. Јединке које размењују делове генетског кода представљају „родитеље”, а новонастале јединке „потомке”. Приликом укрштања, очекује се да ће „добре” карактеристике родитеља наследити њихови потомци и да ће новоформиране јединке имати квалитетнији материјал, односно већу прилагођеност окружењу у односу на полазне јединке. Постоји више начина укрштања генетског материјала, а детаљније о њима може се наћи у [77, 118].

Најчешће се користи позиционо укрштање, у оквиру којег се задају тзв. тачке укрштања, односно позиције у кодовима родитеља у односу на које се размењују делови кодова. У зависности од броја тачака укрштања, позициона укршања могу бити: једнопозициона, двопозициона и вишепозициона. Код једнопозиционог укрштања, један потомак добија први део генетског кода од првог родитеља, а други део од другог родитеља. Прецизније, ако се са k означи тачка укрштања, онда први потомак добија генетски код првог родитеља од прве до позиције k, и генетски код другог родитеља од позиције $k+1$ до последње позиције. Генетски код другог потомка формира се од редом преосталих делова кодова оба родитеља. Када су у питању двопозициона укрштања, бирају се редом две тачке укрштања k_1 и k_2 и узајамно се размењују делови кодова оба родитеља од позиције k_1+1 до позиције k_2. Код вишепозициона укрштања, задаје се више тачака укрштања у односу на које се размењују делови кодова.

Поред позиционих укрштања постоји и униформно укрштање, у оквиру којег се (посебно) за сваки пар родитеља на случајан начин генерише одређена бинарна маска дужине генетског кода јединке. Маска се формира тако да вредност 1 на одређеној позицији маске значи да ће први потомак узети посматрани ген од првог родитеља, а други потомак посматрани ген од другог родитеља. Уколико је вредност 0 на одређеној позицији маске, први потомак преузима ген од другог родитеља, а други потомак од првог родитеља.

Карактеристике проблема који се решава утичу на избор оператора укрштања. Уколико је потребно делимично очувати генетску структуру јединки популације, погодна је примена једнопозиционог укрштања. У супротном, када је корисно разбити и измешати блокове у генетским кодовима јединки, примењује се двопозиционо или вишепозиционо укрштање. У случају када су кодови готово независни, најпогоднија је примена униформног укрштања.

Мутација

Оператор мутације (енгл. mutation operator) симулира истоимени еволутивни процес, мењајући генетски код јединке заменом вредности појединих делова кода (гена) са другим вредностима симбола азбуке која се користи. Избор типа оператора мутације може у великој мери да утиче на перформансе GA. Уколико се јединке кодирају бинарно и популација нема некоректних јединки, најчешће се примењује оператор просте мутације (енгл. simple mutation operator). У оквиру овог типа оператора, за сваку јединку се редом процесира сваки ген генетског кода и на случајан начин (са унапред задатом вероватноћом) се одређује да ли ће доћи до измене (мутације) његове вредности или не. Овај процес се може убрзати увођењем тзв. маске (бинарног низа) која се случајно генерише за сваку јединку и по којој се тачно одређује на којој позицији генетског кода јединке долази до промене вредности гена.

С обзиром да се процес мутације извршава са одређеном вероватноћом, пожељно је да она буде веома мала, реда 10^{-3}. Уколико би вероватноћа била знатно већа, GA би се понашао као алгоритам случајне претраге (енгл. random search), док се у случају мање вероватноће претраживачки простор решења не би битније проширио. Изузетак представља појава тзв. „залеђених гена” (енгл. frozen genes), гена који имају исту вредност на одређеној позицији код свих (или великог процента) јединки. Ако се посматра бинарно кодирање и са l означи број залеђених гена, тада простор за претрагу постаје 2^l пута мањи и могућност за преурањену конвергенцију рапидно расте [122]. Из тих разлога, потребно је повећати вероватноћу мутације ових гена за одређени фактор.

Значај оператора мутације је велики јер се управо кроз процес мутације делова кодова јединки добија могућност враћања корисног генетског материјала који се могао изгубити током процеса селекције и укрштања. На овај начин обезбеђује се разноврсност генетског материјала (проширење претраживачког простора решења), спречавајући на тај начин преурањену конвергенцију ка неком од локалних екстремума.

Поред просте мутације користе се и други типови овог оператора, попут: мутације која користи биномну или нормалну расподелу вероватноће, експоненцијална мутација (код које број мутираних гена у коду експоненцијално опада) и друге. Више детаља о различитим типовима оператора мутације може се наћи у [77, 118].

Остале карактеристике GA

Након процеса извршења свих генетских оператора и стварања нових јединки, долази до формирања нове генерације јединки по унапред утврђеној стратегији замене генерација. Стратегија замене генерација јединки представља битан аспект GA, и неки од најчешће примењиваних типова стратегије су:

- Генерацијски GA – у свакој генерацији се мењају све јединке;
- Стационарни GA – један део јединки (најбоље јединке) је повлашћен и директно пролази фазу селекције учествујући у процесу укрштања и мутације, док се преостале јединке генеришу;
- Елитистичка стратегија – набоље јединке (јединке са највећом мером прилагођености) тзв. „елитне” јединке, треба сачувати од измене и елиминације током еволутивног процеса. Из тих разлога, омогућава се њихов директан пролаз кроз све генерације, учествовањем у свакој од фаза генерације заједно са осталим неповлашћеним јединкама. У зависности од броја елитних јединки, њихово претраживање и сортирање од највеће до најмање може успорити генетски алгоритам.

Укупан број генерација (итерација GA) кроз које пролази популација јединки је одређен унапред задатим критеријумом заустављања алгоритма. Могући критеријуми заустављања алгоритма су: достигнут максималан број генерација, достигнуто временско ограничење извршења GA, достигнуто познато оптимално решење, достигнут број узастопних генерација у којима се није поправило најбоље решење, велика сличност јединки у популацији, прекид од стране корисника, и други.

Неки од параметара генетских оператора се могу мењати у току извршења GA. Према начину промене параметара, могу се издвојити две стратегије:

- Статичка промена параметара – код које се унапред задаје линеарно или експоненцијално повећање или смањивање параметара;

- Динамичка промена параметара – код које је промена параметара у директној зависности од тога колико је оператор био до тада успешан и какаве је резулате понудио.

На пример, параметар који се може мењати је вероватноћа мутације. Уколико је у питању њена статичка промена, вероватноћа мутације се мења независно од успеха GA. Код динамичке промене, може се успоставити директна зависност вероватноће мутације од разноврсности јединки у популацији. Што је већа разноврсност јединки, вероватноћа мутација се смањује, и обрнуто, са смањењем разноврсности међу јединкама, повећава се вероватноћа мутације њихових гена.

Основни GA

Основна структура GA приказана је Алгоритмом 1. У првом кораку алгоритма генерисана је иницијална популација P на основну улазних података D и максималног броја јединки N_{pop}. Критеријум заустављања алгоритма дефинише се у наредном кораку. Следи главни део алгоритма који чини петља, приказана у корацима 3-11, која се извршава све док се не испуни критеријум заустављања алгоритма. У оквиру петље, најпре се пролази по свим јединкама i популације и рачуна се њихова функција циља $obj[i]$, а затим се у кораку 7 рачуна и њихова функција прилагођености. У корацима који следе извршавају се преостала три оператора: селекција, укрштање и мутација. Најбоља јединка из последње генерације представља резултат.

Основну варијанту генетског алгоритма представља GA који користи бинарно кодирање, просту рулет селекцију, укрштање у једној тачки и просту мутацију. Иако је методологија простог GA врло једноставна за имплементацију, његова примена се ограничава на мањи број проблема комбинаторне оптимизације. За решавање комплекснијих комбинаторних проблема користе се унапређене варијанте GA.

Детаљанији опис GA методе може се наћи у [77, 118]. Честа примена ове методе за решавање различитих проблема оптимизације показује да GA често производи решења високог квалитета у разумном временском року. Неки од примера успешног коришћења GA методе описани су у [43, 57, 59, 119, 130, 133, 147, 157, 160, 177, 178].

Алгоритам 1: Основна структура GA

Input: D, N_{pop}

Output: solution P_{best}

P ← *GenerateInitialPopulation (*N_{pop}*);*

SelectStopCondition ( ) ;

while** StopCondition ( ) is not satisfied **do

    for $i = 1$ ***to*** N_{pop} ***do***

        $obj[i]$ ← *CalculateObjectiveFunction (*P_i*);*

    endfor

    CalculateFitnessFunction (P*);*

    Selection (P*);*

    Crossover (P*);*

    Mutation (P*);*

endwhile

P_{best}←*Best* (P)*;*

1.3.2 Метода локалне претраге

Локална претрага подразумева процес претраживања околине допустивог решења с циљем проналажења новог, потенцијално бољег решења.

Нека је са x означено једно допустиво решење проблема оптимизације формулисаног моделом (1.1) - (1.2). Под околином $N(x)$ решења x подразумевају се сва решења из допустивог скупа X која се могу добити од x применом неке од елементарних трансформација [159]. Елементи скупа $N(x)$ називају се суседи од x. Уколико се посматра већи број околина једног решења, користе се ознаке N_p $(p = 1, \ldots, p_{max})$ при чему је p_{max} ознака за максималан број околина по којима се врши претраживање. Скуп $N_p(x)$ чине решења која се налазе у p-тој околини тренутног решења, односно она решења која се добијају од тренутног решења x када се на њега примени p пута иста елементарна трансформација.

У складу са наведеном нотацијом, допустиво решење $x \in X$ се дефинише као:

- локални минимум (тренутно решење) уколико важи $(\forall x' \in N(x) \subseteq X)\ f(x) < f(x')$;
- глобални минимум (оптимално решење) уколико важи $(\forall x' \in X)\ f(x) < f(x')$.

Све хеуристичке и метахеуристичке методе су усмерене на проналажење што бољих тренутних решења како би се у што краћем временском року, користећи се специфичним техника побољшања тренутног решења, стигло до оптималног (најбољег) решења. Једна од таквих метода је и метода локалне претраге (Local Search - LS) којом се побољшање тренутног решења тражи у унапред дефинисаној околини решења. Најчешће се за побољшање тренутног решења бира најбоље од свих пронађених побољшања у околини, и та стратегија се назива стратегија најбољег побољшања (енгл. best improvement strategy). Као његова алтернатива користи се и стратегија првог побољшања (енгл. first improvement strategy) којом се тренутно решење ажурира са првим побољшањем нађеним у његовој околини.

Основна структура методе локалне претраге приказана је Алгоритмом 2. У првом кораку алгоритма се рачуна функција циља почетног решења $x_{initial}$, изабраног на случајан начин или неком хеуристиком из простора допустивих решења. У следећем кораку се почетно решење дефинише као тренутно најбоље $x_{best} = x_{initial}$. Иницијална вредност логичке променљиве *improvement* (побољшање) се дефинише у кораку 4. Централни део локалне претраге се састоји од петље која се извршава у редовима 5-15 све док постоји побољшање тренутно најбољег решења. У кораку 8 се за сваког од суседа $x_{temp} \in N_p(\,x_{best})$ рачуна вредност функције циља f_{temp} и ако постоји сусед x_{temp} за чију функцију циља важи $f_{temp} < f_{best}$, он постаје тренутно најбоље решење из околине $x_{best} = x_{temp}$. Када се у корацима 7-14 обиђу сви суседи околине (упоређујући њихове вредности фунције циља са тренутно најбољом вредношћу) најбољи од свих суседа се узима као побољшање иницијалног решења и локална претрага се даље наставља у његовој околини. Претрага околине (енгл. neighborhood exploration - NE), која подразумева испитивање свих суседа тренутно најбољег решења x_{best} представља једну итерацију локалне претраге која се зауставља када у околини последњег пронађеног најбољег решења нема побољшања, односно нема бољег решења од њега.

Један од недостатака методе локалне претраге огледа се у томе што је претрага по свим елементима једне околине временски веома захтеван, а некада и немогућ посао. Унапређене ове методе се огледа у проналажењу начина да се претрага ограничи на суседе који нуде побољшање тренутног решења. Други недостатак се односи на релативно мале шансе у погледу проналажења глобалног минимума, с обзиром да је алгоритам конципиран на начин да се његов рад зауставља када се пронађе први локални оптимум који се не може побољшати, а који се ретко поклапа са глобалним оптимумом.

Алгоритам 2: Структура LS алгоритма

Input: solution $x_{initial}$

Output: solution x_{best}

$f_{initial}$ ← *CalculateObjectiveFunction(*$x_{initial}$*);*

x_{best} ← $x_{initial}$;

f_{best} ← $f_{initial}$;

improvement ← *true;*

while *improvement* ***do***

    improvement ← *false;*

    foreach x_{temp} *in neighbourhood* $N(x_{best})$ ***do***

        f_{temp} ← *CalculateObjectiveFunction(*x_{temp}*);*

        if $f_{temp} < f_{best}$ ***then***

            x_{best} ← x_{temp};

            f_{best} ← f_{temp} ;

            improvement ←*true;*

        endif

    endforeach

endwhile

Инспирисане методом локалне претраге, развијене су многе метахеуристичке методе у оквиру којих су уграђене разне идеје за превазилажење замки локалних оптимума. Неке од таквих метахеуристика су: вишестартно локално претраживање (енгл. Multistart Local Search - MLS), табу претраживање (TS), метода промењивих околина (VNS) и друге.

1.3.3 Метода промењљивих околина

Метода промењљивих околина (Variable Neighbourhood Search - VNS) је метахеуристичка метода коју су предложили Ненад Младеновић и Pierre Hansen 1997. године у [148] за решавање проблема трговачког путника (енгл. Traveling Salesman Problem - TSP). Након промовисања идеје методе, све већи број истраживача почиње да примењује VNS за решавање проблема комбинаторне и глобалне оптимизације [87, 88]. До данашњих дана, VNS се показао као веома успешна метода за решавања многих класа проблема, од којих су неке из области: теорије локација, кластер анализе, планирања, усмеравања возила, дизајна мрежа, вештачке интелигенције, инжењеринга, проблема удруживања, проблема поузданости, биологије, филогеније, геометрије, телекомуникација. Неке од варијанти ове методе посебно развијене за решавања проблема великих димензија или проблема сложене структуре описане су у [54, 55, 56, 87, 128, 141].

Основна идеја VNS методе подразумева употребу унапред дефинисане коначне колекције структура околина у процесу локалне претраге простора допустивих решења, са циљем проналажења глобалног оптимума. Идеја се заснива на три једноставне чињенице [77, 87]:

- Локални минимум у једној околини није обавезно и локални минимум у некој другој околини;
- Глобални минимум је локални минимум у свим околинама;
- За велики број проблема, локални минимуми у разним околинама су релативно близу једни другима.

Наведене чињенице указују да претрагу треба вршити по више околина тренутног решења (локалног минимума), са циљем проналаска глобалног минимума. Како су за велики број проблема локални минимуми различитих околина међусобно релативно блиски, детаљна претрага се врши у најближим околинама локалних минимума, јер се у њима очекује побољшање тренутно најбољег решења. Управо из тих разлога, VNS методом се систематски мењају околине, како у фази претраге за локалним минимумом, тако и у фази изласка из локалног минимума и претраге за глобалним минимумом. Секвенцијалном претрагом околина, добијају се локална оптимална решења, док се систематским мењањем околина претрага усмерава ка делу простора које „обећава”. Наведене чињенице могу се користити на три различита начина: детерминистички, стохастички или комбиновано. У зависности од начина који се примењује, дизајниране су различите варијанте VNS методе. У наредним одељцима представљене су основне варијанте VNS методе, док се више детаља о осталим варијантама може наћи у [77, 87].

Метода промењљивог спуста

Метода промењљивог спуста (енгл. Variable Neighborhood Descent - VND) представља детерминистичку варијанту VNS методе [77, 87]. Најпре се претражује прва околина N_1 тренутно најбољег решења. Уколико се пронађе побољшање тренутног решења, примењује се замена околина и претрага се даље наставља од новог решења. Када се одреди локални минимум у односу на посматрану околину, даље претраживање се наставља у наредној околини N_2. Након одређивања локалног минимума N_2 околине, наредна побољшањима се траже у N_3 околини, и тако даље. Сваки пут када се пронађе побољшање тренутно најбољег решења, претраживање се враћа на прву околину N_1 новог најбољег решења. Претраживање се завршава када се тренутно најбоље решење не може побољшати ни у једној околини или када се достигне унапред задати максималан број околина које се претражују.

Основна структура VND алгоритма дата је Алгоритмом 3. У првом кораку алгоритма се иницијално решење $x_{initial}$, које се најчешће добија на случајан начин или применом неке хеуристике, дефинише као тренутно најбоље решење $x_{best} = x_{initial}$. Величина околине тренутно најбољег решења се у кораку 2 поставља на иницијалну вредност 1. У наредним корацима следи централи део алгоритма који се састоји из *while* петље која се извршава у редовима 3-11, док год критеријум заустављања алгоритма није испуњен. Критеријум заустављања алгоритма је достизање унапред дефинисаног максималног броја околина p_{max}, које се претражују. Централну петљу чине фаза претраживања околине и функција замене околина. У фази претраживања околине, која се извршава у кораку 4, тражи се локални минимум у p-тој околини тренутно најбољег решења x_{best} и он се означава са x_{temp}. У корацима 5-10 се извршава функција замене околина, којом се могуће побољшање x_{temp} упоређује са x_{best} и уколико је боље, оно се дефинише као ново најбоље решење $x_{best} = x_{temp}$ и даља претрага се наставља од његове прве околине $p = 1$. У супротном, уколико x_{temp} није боље од тренутно најбољег x_{best}, претрага се наставља у наредној околини $N_p(x_{best})$, при чему је $p = p + 1$. Након задовољена критеријума заустављања алгоритма, последње добијено најбоље решење x_{best} представља резултат.

Алгоритам 3: Структура VND алгоритма

Input: solution $x_{initial}, p_{max}$
Output: solution x_{best}
1. $x_{best} \leftarrow x_{initial}$;
2. $p \leftarrow 1$;
3. ***while*** $p \leq p_{max}$ ***do***
4. $\quad x_{temp} \leftarrow$ *ExplorN(*x_{best}, p*);*
5. $\quad$ ***if*** x_{temp} *is better than* x_{best} ***then***
6. $\quad\quad x_{best} \leftarrow x_{temp}$;
7. $\quad\quad p \leftarrow 1$;
8. $\quad$ ***else***
9. $\quad\quad p \leftarrow p + 1$;
10. $\quad$ ***endif***
11. ***endwhile***

Редукована метода промењљивих околина

Rедуковани VNS метод (енгл. Reduced VNS - RVNS) представља стохастичку варијанту VNS методе [77, 87]. RVNS укључује само фазу размрдавања (енгл. shake phase) - фазу изласка из локалног оптимума, без фазе локалне претраге (енгл. local search phase) - фазе проналажења локалног оптимума. RVNS се заснива на систематској промени околина и насумичном избору једног решења у свакој од околина. За потребе алгоритма дефинише се критеријум заустављања, који је често максималан број итерација између два побољшања. Ова метода је погодна за брзу претрагу околина у којима се налазе квалитетна допустива решења, али није погодан за детаљну претрагу околина које садрже решења блиска глобалном оптимуму. RVNS се најчешће примењује за решавање тест примера великих димензија, у случајевима када је потребно избећи недовољно ефикасну и дуготрајну локалну претрагу околина.

Алгоритам 4: Структура RVNS алгоритма

Input: solution $x_{initial}, p_{max}$
Output: solution x_{best}
1. $x_{best} \leftarrow x_{initial}$;
2. $p \leftarrow 1$;
3. *SelectStopCondition();*
4. ***while*** *StopCondition() is not satisfied* ***do***
5. $\quad x_{temp} \leftarrow$ *Shaking(*x_{best}, p*);*
6. $\quad$ ***if*** x_{temp} *is better than* x_{best} ***then***
7. $\qquad x_{best} \leftarrow x_{temp}$;
8. $\qquad p \leftarrow 1$;
9. $\quad$ ***else***
10. $\qquad p \leftarrow p + 1$;
11. $\qquad$ ***if*** $p > p_{max}$ ***then***
12. $\qquad\quad p \leftarrow 1$;
13. $\qquad$ ***endif***
14. $\quad$ ***endif***
15. ***endwhile***

Основна структура RVNS алгоритма дата је Алгоритмом 4. У првом кораку алгоритма се иницијално решење $x_{initial}$ дефинише као тренутно најбоље решење $x_{best} = x_{initial}$. У наредном кораку 2 се величина околине тренутно најбољег решења поставља на иницијалну вредност 1. Критеријум заустављања алгоритма дефинише се у кораку 3, након чега следи централи део алгоритма који се састоји из *while* петље која се извршава у редовима 4-15, док год критеријум заустављања (алгоритма) није испуњен. Петљу чине фаза размрдавања и функција замене околина. У фази размрдавања, која се извршава у кораку 5, тренутно најбоље решење x_{best} се помера ка случајно одабраном решењу x_{temp} које се налази у p-тој околини x_{best}. У корацима 6-14 извршава се функција замене околина, којом се (могуће побољшање) x_{temp} упоређује са x_{best} и уколико је боље, оно се дефинише као ново најбоље решење $x_{best} = x_{temp}$ и даља претрага се наставља од његове прве околине $p = 1$. Уколико x_{temp} није боље од тренутно најбољег x_{best}, претрага се наставља у наредној околини $N_p(x_{best})$, при чему је $p = p + 1$ (или, у случају $p = p_{max}$ поново се креће од $p = 1$). Дакле, у сваком тренутку се разматрају два решења: тренутно најбоље x_{best} и случајно изабрано $x_{temp} \in N_p(x_{best})$. Када се испуни критеријум заустављања алгоритма, последње добијено најбоље решење x_{best} представља резултат.

Основна метода промењљивих околина

Алгоритам 5: Структура BVNS алгоритма

Input: p_{max}

Output: solution x_{best}

InitializeStructures();

$x_{initial}$ ←*RandomSolution();*

$x_{current} \leftarrow x_{initial}$;

p ←*1;*

SelectStopCondition () ;

while** StopCondition () is not satisfied **do

    $x_{temp} \leftarrow$ *Shaking(*$x_{current}, p$*);*

    $x_{temp} \leftarrow$ *LocalSearch(*x_{temp}*);*

    if x_{temp} *is better than* $x_{current}$ ***then***

        $x_{current} \leftarrow x_{temp}$;

        p ←*1;*

    else

        p ← *p +1;*

        if $p > p_{max}$ ***then***

            p ←*1;*

        endif

    endif

endwhile

$x_{best} \leftarrow x_{current}$;

Основна VNS метода (енгл. Basic VNS – BVNS) настала као комбинација претходна два приступа (детерминистичког и стохастичког), користи скуп претходно дефинисаних структура околина у оквиру којих итеративно тражи побољшање тренутног решења проласком кроз две главне фазе: фазе размрдавања и фазе локалне претраге [77, 87]. Број понављања ове две фазе је ограничен унапред дефинисаним критеријумом заустављања алгоритма. Најчешће се за критеријум заустављања узима максималан број итерација или максималан број итерација без побољшања решења или максимално време извршења алгоритма.

Нека променљива $x \in X$ представља произвољно допустиво решење, променљива N_p $(p = 1, \dots, p_{max})$ коначну колекцију унапред дефинисаних структура околина и променљива $N_p(x)$ скуп решења у p-тој околини од x.

Користећи наведену нотацију, структура основне VNS методе може бити представљена следећим корацима:

- Иницијализација: Дефинише се структура околина N_p, проналази се иницијално решење x, поставља се параметар p на иницијалну вредност, дефинише се критеријум заустављања;
- Размрдавање: Насумице се генерише суседно решење $x' \in N_p(x)$;
- Локална претрага: Примењује се метода локалне претраге околине $N_p(x)$, почевши од x' са циљем проналаска локалног оптимума x'';
- Прихватање решења (ажурирање или не): Ако је x'' боље решење од почетног x, поставља се да је $x = x''$ и $p = 1$, па се даља претрага наставља од ажурираног x и његове прве околине. У супротном, претрага се преусмерава на следећу $(p = p + 1)$ околину почетног решења x. У случају да је $p = p_{max}$ даља претрага се наставља од прве $(p = 1)$ околине;
- Провера критеријума заустављања: Уколико критеријум заустављања није задовољен, онда се иде на корак 2, у супротном x се враћа као најбоље пронађено решење.

Структура основног VNS алгоритма приказана је Алгоритмом 5. У првом кораку се дефинише структура околина у простору допустивих решења.

Иницијално решење $x_{initial}$, које се најчешће добија на случајан начин у оквиру другог корака, се у наредном кораку одређује као тренутно решење $x_{current}$ = $x_{initial}$. Вредност променљиве p, која представља редни број околине тренутног решења, се у кораку 4 поставља на 1. Услов заустављања алгоритма дефинише се у кораку 5, и док год он није испуњен извршава се централи део алгоритма који се састоји из петље која се извршава у корацима 6-18. Петљу чине фаза размрдавања, фаза локалне претраге и функција замене околина. У фази размрдавања, која се извршава у кораку 7, тренутно решење $x_{current}$ се помера ка случајно одабраном решењу x_{temp} у p-тој околини тренутног решења. Током фазе локалне претраге, која се извршава у 8 кораку, тражи се побољшање решења $x_{current}$ (побољшање преосталих променљивих) почев од x_{temp}, најчешће у околинама $N_1(x_{current})$ и $N_2(x_{current})$. У наредном кораку се извршава функција замене околина којом се најбоље пронађено решење x_{temp} из околине $x_{current}$ упоређује са тренутним $x_{current}$ и уколико је боље од њега, тренутно решење се ажурира $x_{current} = x_{temp}$ и даља претрага се наставља од његове прве околине $p = 1$. У супротном, уколико најбоље решење из околине x_{temp} није боље од тренутног $x_{current}$, претрага се наставља у наредној околини тренутног решења $N_p(x_{current})$, при чему је $p = p$+1 (p=1, у случају $p = p_{max}$). Дакле, у сваком тренутку се разматрају три решења (тренутно најбоље $x_{current}$, случајно изабрано $x_{temp} \in N_p(x_{current})$ и најбоље пронађено $x_{temp} \in N_p(x_{current})$. Након испуњења критеријума заустављања алгоритма, последње пронађено најбоље решење (локални оптимум) се у кораку 19 враћа као резултат алгоритма, односно као најбоље од свих пронађених решења (глобални оптимум).

Општа метода променљивих околина

Општа метода променљивих околина (енгл. General VNS – GVNS) настала је из BVNS методе заменом локалне претраге VND методом [77, 87]. Структура GVNS алгоритма приказана је Алгоритмом 6. GVNS алгоритам се у односу на BVNS (Алгоритам 5) разликује по улазним подацима и по кораку 8. За улазне податке код GVNS потребно је дефинисати и максималан број околина p'_{max} које се претражују у оквиру VND методе, док се у кораку 8 уместо *LocalSearch* позива *VND* процедура (Алгоритам 3).

Алгоритам 6: Структура GVNS алгоритма

Input: p_{max}, p'_{max}

Output: solution x_{best}

1. *InitializeStructures();*
2. $x_{initial}$ ←*RandomSolution();*
3. $x_{current}$ ← $x_{initial}$;
4. p ←*1;*
5. *SelectStopCondition () ;*
6. ***while*** *StopCondition () is not satisfied* ***do***
7. x_{temp} ← *Shaking(*$x_{current}, p$*);*
8. x_{temp} ← *VND(*x_{temp}, p'_{max}*);*
9.   ***if*** x_{temp} *is better than* $x_{current}$ ***then***
10. $x_{current}$ ← x_{temp};
11. p ←*1;*
12.   ***else***
13. p ← p *+1;*
14.   ***if*** $p > p_{max}$ ***then***
15. p ←*1;*
16.   ***endif***
17.   ***endif***
18. ***endwhile***
19. x_{best} ← $x_{current}$;

1.4 Проблеми формирања група у настави

У ери убрзаног развоја информационих технологија, све чешћи су захтеви и интенције организација за решавањем специфичних проблема кроз групни, а не индивидуални облик рада. Групни облик рада добија на све већем значају, с обзиром да се у таквом окружењу смањује радна оптерећеност појединца и омогућује продуктивније и ефикасније извршавање задатака које поставља организација [6].

Упоредо са захтевима индустије, током последњих тридесет година групни рад је постао и важан аспект образовања [106]. Бројна истраживања су показала да ученици боље памте садржаје уколико су активно укључени у усвајање и обраду информација [11]. Значајан је број радова који говоре у прилог бројних предности групног рада на свим образовним нивоима [167]. Већина академских институција организује курсеве примењујући групни рад као средство помоћу којег студенти размењују своја знања и способности, унапређују вештине потребне за успешно решавања проблема, развијају комуникацијске вештине, и на тај начин се припремају за улазак у индустрију [6]. Посебну пажњу групном облику рада поклањају пословне школе (енгл. business school), које имају за циљ да едукују будуће професионалце који ће бити способни да допринесу економском развоју земље [115]. Битан задатак који пред собом постављају ове школе јесте да код својих студената развију способност за тимским радом, с обзиром да је управо наведена вештина високо вреднована од стране послодаваца [106].

Фундаменталан корак у развоју и перформансама група представља начин на који су групе формиране [6]. Постоји велики број различитих приступа формирању група, међутим нису све групе функционалне [6]. Један од најчешћих разлога нефункционалности групе јесте њено несистематско формирање. Бројна су истраживања у оквиру којих се наводи да неадекватно формирање група утиче на демотивисаност студента и на тај начин ремети процес учења. Према томе, формирање група представља први, важан и комплексан корак у креирању ефикасног окружења које би требало да подстакне успостављање добрих интеракција између чланова групе, и на тај начин да утиче на унапређење процеса учења и интелектуалног раста сваког појединца [39]. Ефикасним груписањем индивидуа, постиже се да оптерећење сваког од чланова групе буде пропорционално распоређено, што води до веће мотивације за радом, до продуктивније размене знања, узајамне помоћи, подршке, развијања комуникацијских и менаџерских вештина, вештина тимског рада, па самим тим и до више различитих решења за решавање постављених проблема [6, 105].

У литератури су постављена бројна питања која се тичу састава групе, типа групе и критеријума које би требало користити при формирању група, како би се креирале оптималне групе у оквиру којих би студенти могли да остваре своје максималне резултате и ефикасност у процесу учења [104]. Сумирањем резултата бројних истраживања дошло се до закључка да у процесу формирања група кључну улогу играју пет фактора [112].

Први фактор се односи на избор врсте и броја карактеристика који ће се користили за груписање индивидуа [21, 150]. Више детаља о карактеристикама индивидуа које су у литератури коришћене у својству одређеног критеријума за груписање, могу се наћи у одељку 3.2.1.

Дефинисање укупног броја група и величине група (одређивањем доње и горње границе бројности чланова групе) представља други фактор [150]. Многе студије тврде да је величина група за колаборативно учење веома важна компонента која утиче на продуктивност учења [104]. Сходно педагошким циљевима, формирање група са једнаким бројем чланова представља чест приступ за формирање група [1, 6, 78, 79, 112, 132, 150, 186, 190, 196].

Трећи фактор укључује дефинисање типа група које се формирају - хетерогене, хомогене, мешовите и балансиране [190]. У оквиру хомогених група (енгл. homogeneous groups), чланови групе имају заједничке карактеристике, односно слични су једни другима у односу на критеријуме груписања. Насупрот хомогеним групама, хетерогене групе (енгл. heterogeneous groups) чине чланови са различитим карактеристикама или различитим вредностима истих карактеристика, односно чланови који се разликују једни од других у односу на критеријуме груписања. Хетерогене групе су суштински еквиваленте максимално разноврсним групама (енгл. maximally diverse groups), у којима се максимизује сума разлика између свих парова индивидуа у групи. Трећу групу чине мешовите групе (енгл. mixed groups), у оквиру којих чланови групе имају одређене карактеристике исте, док се у односу на неке друге карактеристике међусобно разликују; односно ову групу чине чланови који задовољавају хетерогеност за одређене критеријуме а хомогеност за друге критеријуме груписања. У оквиру балансираних група (енгл. balanced groups), чланови групе су на уравнотеженом нивоу у односу на разматране карактеристике. Формирање овог типа група је суштински еквивалентно формирању међу-хомогених група (енгл. inter-homogeneous), где је потребно минимизовати разноликост између група како би се обезбедио сличан састав свих група [169].

Иако је у литератури отворен низ питања која се односе на утицај различитих типова група на перформансе групе и исходе учења, издваја се закључак да групе треба формирати различито у односу на врсту задатка која се решава [163] на начин да њихов састав буде у складу са контекстом учења који укључује: наставни метод који се примењује, наставне садржаје који се обрађују, циљеве које треба остварити и карактеристике студената које се разматрају [104]. Неки истраживачи сматрају да су хетерогене групе применљивије за задатке који захтевају иновативна и креативна решења (дугорочно), с обзиром да у таквом окружењу студенти уче један од другог и тако подстичу стваралачко понашање [15, 169, 204]. Други истраживачи тврде да су хомогене групе функционалније у ситуацијама где је потребно постићи специфичне циљеве (краткорочно и вођено решавање проблема) и где је прогрес по групама потребно обезбедити сличним темпом [15, 204]. Аутори у [1] предлажу формирање мешовитих група које задовољавају хетерогеност за неке од карактеристика а хомогеност за неке друге, омогућавајући флексибилност у односу на број и тип карактеристика које се користе за груписање. У ситуацијама које захтевају уједначено успешно извршавање задатака, истраживачи предлажу формирање балансираних група у односу на неки од специфичних критеријума. Примери наведеног приступа примењени су у [6, 12, 44, 76, 104, 112, 114, 115, 123, 144, 150, 196, 204]. Више детаља о постојећим приступима у формирању различитих типова група могу се наћи у деловима 4.2.1, 4.2.2, 4.2.3 и 5.2.2.

Четврти фактор се односи на пожељне, односно непожељне интеракције између чланова група [112]. Неки истраживачи сугеришу да групе боље функционишу уколико међу њеним члановима владају добри међуљудски односи, односно уколико чланови групе преферирају остале чланове групе или воле да раде са њима из неких разлога [104, 114, 169, 204]. Присутност пријатних интеракција у групи резултира унапређење академских достигнућа и друштвеног понашања [76]. Више детаља о пожељним и непожељним интеракцијама унутар групе може се наћи у одељку 5.2.1.

Пети фактор подразумева одређивање тежине (енгл. weight) сваке од карактеристика и приоритете међу њима [112]. У неколико студија [12, 68, 112, 114, 115, 123, 132, 204] уводи се тежина карактеристика како би се представиле преференције онога ко формира групе, односно релативна важност балансираности група у односу на одређене карактеристике. Па тако, у [204] употреба тежина карактеристика је представљена као једноставан начин да инструктори задовоље различите критеријуме груписања у зависности од различитих образовних контекста. Експерименти су показали да су групе формиране на основу приоритета међу карактеристикама, знатно боље и кохерентније [112].

1.5 Примена метода комбинаторне оптимизације за решавање проблема формирања група у настави

Примена метода комбинаторне оптимизације за решавање проблема формирања специфичних група у настави, односно проблема распоређивања студената у групе (а са циљем пружања помоћи образовним институцијама) има свој континуирани ток у последње три деценије. Проблем груписања студената је означен као један од „најзначајнијих оптимизационих проблема система високог образовања” у [30] и као једна од „најважнијих активности у процесу учења” у [112]. Један од првих примера оваквог проблема представљен је у [93] где је требало распоредити студенте на интервјуе за посао и у пројектне тимове уз максимизовање студентовог избора. Проблем је решен хеуристичким приступом (енгл. Utility-Sequential-Heuristic) којим су студенти најпре распоређени у групе по сопственом избору а затим су из „пренатрпаних” група случајним избором студенти даље распоређивани по најбољем избору. Крајњи циљ решавања овог проблема био је креирање корисничке апликације за распоређивање студената у групе, у оквиру које је имплементиран програм заснован на 0-1 целобројној линеарној формулацији проблема.

Ако се проблем формирања група дефинише као проблем расподеле n студената у m група од по s студената, укупан број свих решења био би $\frac{n!}{m! \cdot (s!)^m}$. Са повећањем вредности променљивих (броја индивидуа и група) и одређених захтева (ограничења) које треба испунити, увећава се и укупан број свих могућих комбинација раасподеле студената у групе, што даље води ка комплекснијем и тежем проблему за решавање [169].

У литератури се могу наћи бројни приступи за решавање овог проблема. Иницијално, они се могу поделити на приступе који не користе моделе и приступе засноване на моделима. Првој групи припадају методе које испитују сва решења, тзв. методе исцрпљивања и методе засноване на постојећим алгоритмима, њиховим модификацијама или новоразвијеним алгоритмима. Другу групу чине оптимизациони приступи који се заснивају на: моделима за програмирање са ограничењима (CP моделима) и моделима математичког програмирања (MP моделима), који се решавају егзактним или апроксимативним методама [169].

Најједноставнији начин за решавање проблема расподеле студената је коришћење методе која испитује сва решења, такозване методе исцрпљивања. Овакав приступ није могуће увек применити, зависно од броја студената, њихових карактеристика, задатих ограничења и група које се формирају [150]. Из тих разлога, чешћа је примена већ постојећих алгоритама, као што су: кластерирање k-средина (енгл. k-means clustering) у [107, 156], кластерирање засновано на матрицама (енгл. matrix-based clustering) у [201], фази кластерирање (енгл. fuzzy clustering) у [149] и хијерархијско кластерирање (енгл. hierarchical clustering) [200]. Међутим, ни примена наведених алгоритама није довела до испуњења свих постављених захтева па су, у циљу добијања бољих резултата, развијене модификоване варијанте већ постојећих алгоритама: модификовани алгоритам за кластерирање k-средина у [78] и модификовани Pareto оптимални скуп (енгл. Pareto optimal set) у [112]. У циљу превазилажења свих недостатака постојећих алгоритама и њихових модификација, развијени су и нови алгоритми, попут: „најуочљивијег“ алгоритма (енгл. squeaky wheel algorithm) у [186] и „манипулативног” алгоритма заснованог на матрицама (енгл. matrix-based manipulation algorithm) у [78]. Ипак, ни један од наведених алгоритама не гарантује оптималност добијених решења, нити је успео да задовољи све захтеве проблема [169].

Како се проблем формирања група дефинише као проблем оптимизације, за његово решавање је потребан добар математички модел. Примена CP модела за решавање проблема формирања група у настави описана је у [156]. Међутим, и поред примене семантичке „веб” технологије (енгл. semantic web technologies), логичког програмирања (енгл. logic programming) и задовољења већег броја задатих ограничења, ни овај модел није испунио све постављене захтеве. Још један пример CP модела за распоређивање студената у групе у односу на њихову доступност представљен је у [167]. С друге стране, примена егзактних и апроксимативних метода заснована на MP моделима је учесталија приликом решавања проблема распоређивања индивидуа у групе. Два мешовита целобројно линеарна модела програмирања за решавање администрационог задатка распоређивања ученика у одељења (примењена на 3 реална примера), представљена су у [76]. Како би се испунили сви захтеви и циљеви постављени од стране администратора академије, формулисан је модел целобројног програмирања за разврставање студената по одељењима у [123]. Модел конвексног квадратног целобројног програмирања (енгл. Convex Quadratic IP model) за решавање расподеле студената на семинаре представљен је у [67]. У [169] проблем распоређивања студената у групе је решаван новим приступом заснованим на бинарном целобројном програмирању кроз формулацију линеарног модела, који је задовољио све постављене захтеве и понудио оптимално решење у разумном временском року. Неки од предложених математичких модела за распоређивање студената у групе могу се наћи у [41, 104, 114, 115, 145].

С обзиром да је проблем проналажења оптималног решења за груписање индивидуа NP-тежак [132], честе су ситуације када због своје комплексности и броја допустивих решења, тест примерима великих димензија проблема не могу бити решавани неким од егзактних решавача. У таквим ситуацијама адекватна математичка формулација проблема постаје веома захтеван задатак па се већина истраживача окреће примени апроксимативних метода које су се показале као добра алтернатива егзактним методама [77, 164]. Неке од метахеуристичких метода које су примењене за решавање проблема распоређивања индивидуа у групе су: оптимизација мрављим колонијама у [79], еволутивни алгоритам детерминистичке грознице (енгл. Deterministic Crowding Evolutionary Algorithm - DCEA) у [196], генетски алгоритам у [1, 6, 78, 105, 112, 138, 150, 179, 190, 204], алгоритам табу претраге у [104], оптимизација ројевима честица у [44, 75, 94, 132], метода промењљивих околина у [22, 23, 49, 55, 56, 87, 88, 89, 148, 187].

Бројна су истраживања у оквиру којих су представљене и решаване различите варијанте проблема распоређивања индивидуа у групе, дефинисане зависно од експлицитно постављених циљева и ограничења. Неке од поменутих варијанти чине следећи проблеми: проблем распоређивања студената у групе које су добро балансиране у односу на одређене карактеристике студената (енгл. well-balanced assignment problem) или проблем балансиране поделе скупа у односу на више карактеристика (енгл. Balanced Multi-Attribute Set Partitioning Problem-BMASP) у [49, 67, 68, 114, 115, 165], проблем максималне разноврсности унутар група (енгл. Maximum Diverse Grouping Problem - MDGP) у [11, 22, 44, 68, 76, 104, 112, 193, 150, 187, 194], као и формирање група разматрањем више критеријума (енгл. Multi-Criteria Group Composition Problem – MCGCP) у [105].

Једно од првих хеуристичких решења за MDGP проблем приказано је у [194]. У [11] представљен је оптимизациони приступ и хеуристика за разврставање студената максимизацијом групне разноврсности. Као перспективнији наводе оптимизациони приступ, у оквиру којег се дефинише математички модел, док се као алтернативни приступ користи хеуристика. Иста врста проблема решавана је и у [49], где су представљене следеће методе: егзактна метода дељења скупа (енгл. set partitioning method), хеуристика заснована на формулацији дељења скупа и две метахеуристичке методе променљивих околина. У [22, 23, 187] представљене су различите варијанте методе променљивих околина за решавање проблема MDGP. Додатне информације о приступима који су коришћени за решавање наведених варијанти проблема могу се наћи у одељцима 4.2.1, 4.2.2 и 5.2.2.

Поред наведених метода комбинаторне оптимизације које су развијене за решавање специфичних проблема формирања група, у литератури се могу наћи и истраживања у оквиру којих су идеје (реализоване кроз математичке моделе) и развијени алгоритми имплементирани у јавно доступне апликације. Нека од таквих истраживања представљена су у [12, 68, 76, 115, 123, 144, 145, 132, 185, 196].

2 Проблем максималне поделе скупа

У овом поглављу описан је нови приступ у решавању проблема максималне поделе скупа на два дела (Maximum Set Splitting Problem - MSSP). MSSP представља проблем у оквиру којег је, за унапред задат коначан скуп елемената и фамилију његових подскупова, потребно пронаћи поделу иницијалног скупа на две непразне партиције, којом би се добио максималан број подељених подскупова фамилије. Проблем MSSP и његова тежинска варијанта су NP-тешки проблеми [73] комбинаторне оптимизације. Доказ да је MSSP проблем NP-тежак приказан је у [136]. Варијанта проблема у оквиру које су сви подскупови фамилије фиксне кардиналности k $(k \geq 2)$ је такође NP-тежак проблем. Штавише, апроксимација проблема MSSP за $k = 3$ и са фактором који је већи од $\frac{11}{12}$, је NP-тежак проблем, што је показано у [84].

2.1 Формулација проблема MSSP

Нека је Ω коначан скуп кардиналности m и нека је дата фамилија његових непразних подскупова $S = \{S_1, \dots, S_n\}$, где је $\bigcup_{j=1}^{n} S_j = \Omega$. Подела скупа Ω на два дела подразумева проналажење уређеног пара (P_1, P_2) непразних партиција скупа Ω, за које важи: $P_1 \cap P_2 = \emptyset$ и $P_1 \cup P_2 = \Omega$. За подскуп S_j $(j = 1, \dots, n)$ може се рећи да је подељен уколико при подели скупа Ω има непразан пресек са обе партиције P_1 и P_2, односно уколико важи: $S_j \cap P_1 \neq \emptyset$ и $S_j \cap P_2 \neq \emptyset$. Ако се са $Obj(P_1, P_2)$ означи укупан број подскупова фамилије S који су подељени уређеним паром партицијама (P_1, P_2), тада се проблем MSSP може формулисати као проблем проналажења максималног броја подељених подскупова (у ознаци $max\ Obj(P_1, P_2)$) у односу на све могуће поделе скупа Ω.

Карактеристике проблема MSSP могу се приказати на два кратка илустративна примера.

Пример 2.1. Нека се скуп Ω_1={1, 2, 3, 4, 5, 6, 7, 8, 9, 10} састоји од десет елемената (m= 10) и четири подскупа (n=4). Подскупови фамилије означени су са: S_1 = {1, 2, 4, 7, 9}; S_2 = {3, 4, 5, 7, 9, 10}; S_3 ={2, 8, 9, 10} и S_4 = {2, 3, 4, 6, 7, 8, 9, 10}. Једно од оптималних решења је уређени пар (P_1, P_2) где је P_1 = {1, 2, 7, 9} а P_2 = {3, 4, 5, 6, 8, 10}. Оптимална вредност функције циља датог решења је 4 с обзиром да су сва четири поскупа подељена: $S_1 \cap P_1$= {1, 2, 7, 9} и $S_1 \cap P_2$= {4} ; $S_2 \cap P_1$ ={7, 9} и $S_2 \cap P_2$ ={3, 4, 5, 10} ; $S_3 \cap P_1$= {2, 9} и $S_3 \cap P_2$ = {8, 10}; $S_4 \cap P_1$ = {2, 7, 9} и $S_4 \cap P_2$ ={3, 4, 6, 8, 10}.

Пример 2.2. Нека се скуп Ω_2={1, 2, 3} састоји од три елемента (m= 3) и три подскупа (n= 3), где су подскупови задати са: S_1 = {1, 2}, S_2 = {1, 3} и S_3 = {2, 3}. Могуће оптимално решење је уређени пар (P_1, P_2) где је P_1 = {1, 2} а P_2 = {3}. Оптимална вредност функције циља је 2 с обзиром да су подељена два скупа: $S_1 \cap P_1$= {1, 2} и $S_1 \cap P_2$=$\emptyset$; $S_2 \cap P_1$= {1} и $S_2 \cap P_2$= {3}; $S_3 \cap P_1$={2} и $S_3 \cap P_2$= {3}.

2.2 Постојећи приступи у решавању проблема MSSP

У литератури се могу наћи бројни приступи у решавању проблема MSSP. У [28, 29] коришћен је приступ заснован на вероватноћи (енгл. probabilistic approach) у циљу креирања детерминистичког тзв. „кернелизационог” (енгл. kernelization) алгоритма за решавање MSSP проблема. Време извршења овог алгоритма ограничено је на $O(2^k)$, где је са k означен укупан број подељених подскупова. Показано је да се овакав алгоритам може ослободити случајног разврставања у групе што даље води ка детерминистички параметризованом алгоритму са временом извршења од $O(4^k)$ за тежинску варијанту проблема MSSP, чиме је добијен први доказ да је проблем лако решив са фиксним параметрима. Техника кернелизације такође се користи у [28, 40, 45, 46, 135, 137]. „ДНК приступ” за решавање проблема MSSP у [27] развијен је као надградња Adleman–Lipton модела додавањем налепница (енгл. sticker-based model) за конструисање простора решења ДНК хеликса и применом ДНК оператора. Хеуристички приступ у решавању MSSP заснован на електромагнетизму (енгл. Electromagnetism-like Method - EM) представљен је у [116]. EM техника заснована на механизму „привлачење-одбијање” комбинује се са техником скалирања и додатно унапређује брзом локалном претрагом. У [141] представљена је метода променљивих околина (VNS) за решавање MSSP проблема.

Неки од примера примене MSSP у разним доменима могу се наћи у [129, 197, 198]. У циљу унапређења перформанси тернарне меморије са адресибилним садржајем (енгл. ternary content addressable memory – TCAM) у [197, 198] користи се алгоритам за дељење скупа (енгл. set splitting algorithm – SSA). Ову технику аутори користе како би решили класификациони проблем са вишеструким спаривањем (енгл. multi-match classification problem). MSSP је нашао своју примену и у стационарној игри за раздвајање (енгл. stationary set splitting game) представљеној у [121], у оквиру које учешће узимају два играча „несигуран" (енгл. unsplit) и „раздвојен" (енгл. split). Несигуран играч стално броји бројне ординале, док раздвојени континуирано покушава да их подели на два стационарна дела.

Први модел квадратног целобројног програмирања (QIP) за решавање проблема MSSP представљен је у [5]. Релаксација ове формулације семидефинитним програмирањем (енгл. Semidefinite Programming - SDP) коришћена је за конструкцију апроксимативног алгоритма са фактором 0.724 за решавање проблема MSSP. Нешто бољи, апроксимативни алгоритам са фактором 0.749 представљен у [202], заснован на на ојачаној SDP релаксацији, унапређен је методом заокруживања (енгл. rounding method) и детаљнијом анализом, додавањем нових неједнакости и унапређењем начина заокруживања. Квадратна целобројна формулација (2.3) - (2.5) коришћена у SDP релаксацији, представљена је моделом који следи.

Користећи се ознакама уведеним у одељку 2.1, параметри и променљиве могу бити дефинисане на следећи начин:

$$y_i = \begin{cases} 1, & i \in P_1 \\ -1, & i \in P_2 \end{cases}, \; i = 1, \dots, m; \tag{2.1}$$

$$z_j = \begin{cases} 1, & S_j \text{ је подељен} \\ 0, & S_j \text{ није подељен} \end{cases}, \; j = 1, \dots, n. \tag{2.2}$$

У складу са уведеном нотацијом, квадратна целобројна формулација проблема MSSP представљена је моделом (2.3)-(2.6):

$$max \sum_{j=1}^{n} z_j \tag{2.3}$$

при ограничењима:

$$\frac{1}{|s_j|-1}\sum_{i_1,i_2\in S_j, i_1\neq i_2}\frac{1-y_{i_1}\cdot y_{i_2}}{2}\geq z_j,\quad j=1,\ldots,n; \tag{2.4}$$

$$z_j\in\{0,1\},\qquad j=1,\ldots,n; \tag{2.5}$$

$$y_j\in\{-1,1\},\qquad i=1,\ldots,m. \tag{2.6}$$

2.3 Формулација целобројног линеарног програмирања за проблем MSSP

У овом одељку представљена је формулација целобројног линеарног програмирања (ILP) за проблем MSSP. Резултати приказани у овом одељку су описани у [130].

Нека су параметар s_{ij} и бинарне променљиве x_i и y_j дефинисане на следећи начин:

$$s_{ij}=\begin{cases}1, & i\in S_j,\\ 0, & i\notin S_j,\end{cases}\qquad i=1,\ldots,m,\quad j=1,\ldots,n; \tag{2.7}$$

$$x_i=\begin{cases}1, & i\in P_1,\\ 0, & i\in P_2,\end{cases}\qquad i=1,\ldots,m; \tag{2.8}$$

$$y_j=\begin{cases}1, & S_j \text{ је подељен},\\ 0, & S_j \text{ није подељен},\end{cases}\qquad j=1,\ldots,n. \tag{2.9}$$

Параметар s_{ij} означава припадност елемента i подскупу S_j, тако да s_{ij} узима вредност 1 када i-ти елемент припада подскупу S_j, док у супротном узима вредност 0. Променљива x_i означава припадност елемента i једној од партиција $\{P_1,P_2\}$. Ако се i-ти елемент налази у партицији P_1, променљива x_i узима вредност 1, у супротном, ако припада партицији P_2, x_i узима вредност 0. Променљива y_j означава подељеност посматраног подскупа S_j, тако да променљива y_j узима вредност 1 уколико је S_j подељен, односно 0 ако није подељен.

Користећи уведену нотацију, формулација целобројног линеарног програмирања проблема MSSP може се представити на следећи начин:

$$max\sum_{j=1}^{n}y_j \tag{2.10}$$

при ограничењима:

$y_j \leq \sum_{i=1}^{m} s_{ij}\, x_i\,, \qquad j = 1, \ldots, n;$ (2.11)

$y_j + \sum_{i=1}^{m} s_{ij}\, x_i \leq |S_j|, \quad j = 1, \ldots, n;$ (2.12)

$y_j \in \{0,1\}, \qquad j = 1, \ldots, n;$ (2.13)

$x_i \in \{0,1\}, \qquad i = 1, \ldots\ , m.$ (2.14)

Функција циља (2.10) је дефинисана тако да сумира број подељених подскупова скупа Ω, а циљ проблема MSSP је њена максимизација. Добра дефинисаност подељеног, односно неподељеног подскупа S_j постиже се ограничењем (2.11), док се ограничењем (2.12) обезбеђује да сваки подскуп S_j има елементе у обе партиције, односно да се сви елементи једног подскупа не могу наћи у тачно једној партицији. Бинарна природа променљивих x_i и y_j одређена је условима (2.13) - (2.14). Може се приметити да изложена формулација (2.10) - (2.14) има укупно $m + n$ бинарних променљивих и $2n$ ограничења.

Нека је уведена ознака: $Obj_{ILP}(x, y) = \sum_{j=1}^{n} y_j$ при ограничењима (2.11) -(2.14). Следеће леме показују да је предложена ILP формулација коректна, тј. да добро описује MSSP проблем.

Лема 1. Нека је $Obj(P_1, P_2)$ подела скупа Ω. Тада постоји решење (x, y) система (2.11) - (2.14) за које важи: $Obj_{ILP}(x, y) \geq Obj(P_1, P_2)$.

Доказ. Уређени пар променљивих (x, y) дефинисан је једнакостима (2.8) и (2.9). Потребно је доказати да ове променљиве задовољавају систем (2.11) - (2.14) и да важи неједнакост: $Obj_{ILP}(x, y) \geq Obj(P_1\,, P_2)$.

Према дефиницији променљиве y_j, важи еквиваленција: S_j је подељен $\Leftrightarrow y_j = 1$. Према томе, број подељених подскупова S_j једнак је $\sum_{j=1}^{n} y_j$, што имплицира неједнакост: $Obj_{ILP}(x, y) \geq Obj(P_1, P_2)$. Ограничења (2.13) - (2.14) су очигледно задовољена дефиницијама (2.8) и (2.9).

Ако подскуп S_j није подељен, онда је $y_j = 0$. У том случају важи неједнакост: $0 = y_j \leq \sum_{i=1}^{m} s_{ij} x_i$, јер су s_{ij} и x_i ненегативни, тако да су ограничења (2.11) задовољена. Такође важи неједнакост: $y_j + \sum_{i=1}^{m} s_{ij} x_i = \sum_{i=1}^{m} s_{ij} x_i \leq \sum_{i=1}^{m} s_{ij} = |S_j|$, што даље повлачи да су и ограничења (2.12) задовољена.

У другом случају, ако је подскуп S_j подељен, тада важи: $y_j = 1$, $S_i \cap P_1 \neq \emptyset$ и $S_i \cap P_2 \neq \emptyset$. Према томе, $(\exists u)(u \in S_j \wedge u \in P_1)$ и $(\exists v)(v \in S_j \wedge v \in P_2)$, што даље имплицира: $s_{uj} = 1, x_u = 1, s_{vj} = 1, x_v = 0$. Следећи корак показује да важи неједнакост: $1 = y_j = s_{uj}\, x_u \leq \sum_{i=1}^{m} s_{ij} x_i$, што повлачи задовољење ограничења (2.11). Као што је претходно имплицирано, важи неједанкост:

$\sum_{i=1}^{m} s_{ij} x_i = x_v s_{vj} + \sum_{i=1,\ i \neq v}^{m} s_{ij} x_i = \sum_{i=1,\ i \neq v}^{m} s_{ij} x_i \leq \sum_{i=1,\ i \neq v}^{m} s_{ij} = |S_j| - 1$,

односно: $y_j + \sum_{i=1}^{m} s_{ij} x_i = 1 + \sum_{i=1}^{m} s_{ij} x_i \leq |S_j|$, тако да су и ограничења (2.12) задовољена.

Лема 2. Ако је (x, y) решење система (2.11) - (2.14), онда постоји подела $Obj(P_1, P_2)$ скупа Ω таква да важи $Obj(P_1, P_2) \geq Obj_{ILP}(x, y)$.

Доказ. Нека је партиција P_1 дефинисана једнакошћу: $P_1 = \{i | x_i = 1 \wedge i \in \Omega\}$, а партиција P_2 као њен комплемент. Потребно је доказати да важи импликација: $y_j = 1 \Rightarrow S_j$ је подељен. Из ограничења (2.11) следе импликације:

$$y_j = 1 \Rightarrow \sum_{i=1}^{m} s_{ij} x_i \geq 1 \Rightarrow (\exists u)\big(s_{uj} = 1 \wedge x_u = 1\big)$$

$$\Rightarrow (\exists u)(u \in S_j \wedge u \in P_1) \Rightarrow S_j \cap P_1 \neq \emptyset .$$

Ограничење (2.12) повлачи импликације:

$$y_j = 1 \Rightarrow y_j + \sum_{i=1}^{m} s_{ij} x_i \leq |S_j| \Rightarrow \sum_{i=1}^{m} s_{ij} x_i \leq |S_j| - 1 \Rightarrow (\exists v)\big(s_{vj} = 1 \wedge x_v = 0\big)$$

$$\Rightarrow (\exists v)(v \in S_j \wedge v \in P_2) \Rightarrow S_j \cap P_2 \neq \emptyset .$$

Из претходно наведеног следи:

$$y_j = 1 \Rightarrow (S_j \cap P_1 \neq \emptyset \ \wedge \ S_j \cap P_2 \neq \emptyset) \Rightarrow S_j \text{ је подељен},$$

што је требало доказати. Према томе, важи да је $\sum_{j=1}^{n} y_j$ већа или једнака броју подељених скупова S_j, што директно имплицира неједнакост:

$$Obj(P_1, P_2) \geq Obj_{ILP}(x, y).$$

Узимајући у обзир наведене леме, дефинише се главна теорема.

Теорема 1. Нека је дата фамилија подскупова $S_1, \ldots, S_n \subseteq \Omega$ и подела коначног скупа Ω на уређени пар дисјунктних непразних партиција (P_1, P_2). Промењљиве x_i, y_j и параметар s_{ij} су дефинисане формулама (2.7) – (2.9). Подела (P_1, P_2) скупа

Ω дели максималан број подскупова S_j $(j = 1, \dots, n)$ ако и само ако постоји оптимално решење (x, y) система (2.10) - (2.14).

Доказ. Смер (⇒) теореме може се лако извести из Леме 1. Такође, други смер (⇐) теореме произилази директно из Леме 2.

Приказ примене CPLEX решавача за решавање MSSP проблема на Примеру 2.1. и Примеру 2.2. следи у даљем тексту.

Пример 2.3. Једно оптимално решење Примера 2.1. је репрезентовано са 1100001010, при чему важи: x_1 = 1, x_2 = 1, x_3 = 0, x_4 = 0, x_5 = 0, x_6 = 0, x_7 = 1, x_8 = 0, x_9 = 1, x_{10} = 0 и y_1 = 1, y_2 = 1, y_3 = 1, y_4 = 1. Вредност фунције циља посматраног решења је 4.

Пример 2.4. Једно оптимално решење Примера 2.2. је репрезентовано са 110, при чему важи: x_1 = 1, x_2 = 1, x_3 = 0 и y_1 = 0, y_2 = 1, y_3 =1. У овом случају, вредност функције циља решења је 2.

На основу модела (2.10)-(2.14) целобројног линеарног програмирања за MSSP, може се формулисати модел целобројног линеарног програмирања за тежинску варијанту MSSP проблема:

$$max \sum_{j=1}^{n} w_j y_j \tag{2.15}$$

при ограничењима (2.11) - (2.14).

Функција циља (2.15) је дефинисана тако да сумира тежине подељених подскупова скупа Ω, а циљ проблема MSSP је њена максимизација. Значења ограничења (2.11) - (2.14) су горе објашњена. Формулација (2.11) - (2.15) има укупно $m + n$ бинарних променљивих и $2n$ ограничења.

2.4 Генетски алгоритам за решавање проблема MSSP

У овој одељку описани су елементи предложеног GA за решавање проблема MSSP. Резултати представљени у овом одељку објављени су у раду [130].

2.4.1 Бинарно кодирање јединки и функција циља

Иницијална популација јединки је генерисана на случајан начин с циљем обезбеђивања максималне разноврсности генетског материјала. За репрезентацију јединки примењено је бинарно кодирање, при чему је број гена (битова) у генетском коду јединке једнак броју елемената посматраног скупа. Према томе, свако од потенцијалних решења (уређени пар (P_1, P_2)) може бити представљено бинарним кодом дужине m. Сваки ген (бит) кода има бинарну вредност 1 или 0, при чему вредност 1 на i-тој позицији кода означава да елемент i припада партицији P_1, док вредност 0 означава да елемент i припада партицији P_2. За решење (поделу) дефинисано на представљени начин, лако се може проверити подељеност сваког од подскупова из фамилије подскупова.

Алгоритам 7: Рачунање функције циља за проблем MSSP

```
Input: numS, numE
Output: solution, numDiv
1.  numDiv ← 0;
2.  for i=1 to numS do
3.      k ← 0;
4.      j← 0;
5.      while (j<sen[i]) and (k=0 or k=j) do
6.          if y[SE(i,j)] then
7.              k←k+1;
8.          endif
9.          j←j+1;
10.     endwhile
11.     if (k>0 and k<sen[i]) then
12.         numDiv ← numDiv+1;
13.     endif
14. endfor
```

Укупан број подељених подскупова рачуна се помоћу функције циља која је представљена Алгоритмом 7. На почетку се бројачу подељених подскупова *numDiv* додељује вредност 0. У оквиру петље која се извршава у корацима 2-14, пролази се кроз све подскупове i (i=1,.., *numS*) и за сваки од њих, редом се пребројава укупан број елемената k који се налази у једној партицији. Кроз све елементе j (j=1,.., *sen*[i]) једног подскупа i пролази се у петљи која се извршава у корацима 5-10, док год важи услов да је k једнако 0 или j. За сваки елемент j се у кораку 6 проверава припадност једној од партиција, при чему *SE* (i,j) представља редни број j-тог елемента у i –том подскупу, док је y низ (1 ако припада првој партицији, 0 ако припада другој). Уколико је укупан број елемената i-тог подскупа у једној од партиција већи од 0 а мањи од укупног броја елемената посматраног подскупа *sen[i])* (корак11), сматра се да је подскуп подељен, односно да има елемената у обе партиције. У том случају, бројач подељених подскупова увећава се за 1. У супротном, нема увећања бројача већ се прелази на проверу следећег подскупа. Вредност функције циља решења једнака је укупном броју подељених подскупова *numDiv*. Ова вредност се даље максимизује применом генетских оператора.

2.4.2 Функција прилагођености и политика замене генерација

У предложеној GA имплементацији, функција прилагођености f_{ind} јединке ind рачуна се скалирањем вредности функције циља obj_{int} сваке од јединки (укупно N_{pop} јединки) у интервал [0,1]. Како је MSSP проблем максимизације, за скалирање се користи формула:

$$f_{ind} = \frac{obj_{ind} - obj_{ind_{min}}}{obj_{ind_{max}} - obj_{ind_{min}}}, \tag{2.16}$$

где су obj_{ind}, $obj_{ind_{min}}$ и $obj_{ind_{max}}$ редом текућа, минимална и максимална вредност функције циља јединки у популацији. Након скалирања, јединке се распоређују у односу на нерастући низ њихове прилагођености: $f_1 \geq f_2 \geq \cdots \geq f_{N_{pop}}$. Најбоља јединка ind_{max} (са највећом вредности функције циља) добија прилагођеност 1, док најлошија јединка ind_{min} (са најмањом вредности функције циља) добија прилагођеност 0.

Генетски алгоритми обично имају релативно мали број елитних јединки, с обзиром да такве јединке имају двоструку шансу да пређу у следећу генерацију: једном преко оператора селекције и једном као елитне јединке [118]. Међутим, такав приступ се није показао задовољавајућим за решавање проблема MSSP. Како би се добили задовољавајући резултати, било је потребно обезбедити довољан број елитних јединки које би чувале добра решења за „експлоатацију” (максимално истраживање), али и довољан број неелитних јединки потребних за „истраживање”.

Да би се спречила доминација елитних јединки у популацији, њихова прилагођеност се умањује формулом:

$$f_{ind} = \begin{cases} f_{ind} - \bar{f}, & f_{ind} > \bar{f} \\ 0, & f_{ind} \leq \bar{f} \end{cases}; \quad ind = 1, \dots, N_{elite}; \quad \bar{f} = \frac{1}{N_{pop}} \sum_{ind=1}^{N_{pop}} f_{ind}. \quad (2.17)$$

На овај начин, чак и неелитне јединке имају шансу да преживе до следеће генерације. Предложени приступ даје могућност високог степена елитизма без превеликог притиска селекције, а тако и без превише „истраживања” у алгоритму.

Како се елитистичком стратегијом омогућује директан пролаз елитних јединки у следећу генерацију, док се остатак популације (који чине неелитне јединке) заједно са елитним јединкама „бори” за прелазак у следећу генерацију (проласком кроз операторе селекције, укрштања и мутације), наведеним параметрима обезбеђује се замена само једне трећине јединки у свакој генерацији. Вредност функције циља елитних јединки се не мења проласком кроз генерације, што пружа значајну уштеду у времену.

Следећа стратегија која се примењује у предложеној имеплементацији, односи се на вишеструку појаву јединки са истим генетским кодом и на вишеструку појаву јединки са истом вредношћу функције циља [138]. У оба случаја, наведене појаве се сматрају сувишним. Да би се спречио прелазак јединки са истим генетским кодом у следећу генерацију, вредност њихове прилагођености се подешава на нулу. Изузетак представља прво појављивање оваквих јединки у популацији, када оне немају своју копију, па је потребно израчунати њихову прилагођеност. Јединке са истом вредношћу функције циља, али различитим генетским кодом у неким случајевима могу по бројности да доминирају у популацији, што повлачи мању заступљеност осталих јединки доброг генетског материјала. Из тог разлога, корисно је ограничити њиховог број појављивања на максималну дозвољену вредност, константу N_{rv}.

Примена наведене стратегије (која се односи се на вишеструку појаву јединки са истим генетским кодом и на вишеструку појаву јединки са истом вредношћу функције циља) се показала веома ефикасна у погледу очувања разноврсности генетског материјала и спречавања преурањене конвергенције алгоритма. Стратегија се примењују на сваку јединку у популацији, а спроводи се кроз следеће кораке:

Корак 1: Проверава се да ли је генетски код јединке ind идентичан генетском коду неке од претходних јединки, из интервала од 1 до $ind - 1$. Уколико је одговор позитиван, прилагођеност f_{ind} јединке ind се подешава на вредност 0. Иначе се прелази на Корак 2;

Корак 2: Пребројава се укупан број јединки из интервала од 1 до $ind - 1$ које нису добиле прилагођеност 0 у Кораку 1, а које имају исту вредност функције циља као ind. Ако је тај број већи или једнак N_{rv}, прилагођеност f_{ind} јединке ind се подешава на вредност 0.

2.4.3 Селекција

Оператор селекције је механизам којим се бирају јединке које ће произвести потомке у следећој генерацији. У предложеној GA имплементацији користи се унапређени оператор турнирске селекције, познат као Фино градирана турнирска селекција (FGTS) предложена у [66]. Оператор FGTS користи реалан (рационалан) параметар F_{tour} који означава жељену просечну величину турнира. За разлику од стандардне турнирске селекције, у оквиру које су сви турнири исте величине, код FGTS присутне су две величине. Први тип турнира се одржава k_1 пута и његова величина је $\lfloor F_{tour} \rfloor$, док се други тип одржава k_2 пута са учешћем од $\lceil F_{tour} \rceil$ јединки, тако да важи: $F_{tour} \approx \frac{k_1 \cdot \lfloor F_{tour} \rfloor + k_2 \cdot \lceil F_{tour} \rceil}{N_{nnel}}$. Практична примена овог типа селекције и њено поређење са другим операторима описано је у [64, 65, 66].

2.4.4 Укрштање

У оквиру оператора укрштања, јединке које су одабране оператором селекције се на случајан начин групишу у укупно $\lfloor N_{nnel}/2 \rfloor$ парова. Рекомбинацијом генетских материјала сваког од одабраних парова јединки (родитеља) стварају се по две нове јединке (потомка) за следећу генерацију. При решавању проблема MSSP коришћен је оператор једнопозиционог укрштања (детаљније објашњен у одељку 1.3.1) код којег се на случајан начин бира број $k \in \{0, \dots, d-1\}$ који означава тачку укрштања, при чему је d ознака за дужину генетског кода.

2.4.5 Мутација

За оператор мутације у предложеној GA имплементацији одабрана је проста мутација са залеђеним генима (описана у одељку 1.3.1). У свакој генерацији, одређују се позиције на којима ће све или највећи број јединки имати залеђени ген [119, 120, 178], при чему скуп залеђених гена није фиксан већ се може променити током генерација. При решавању задатог проблема, оператор просте мутације је модификован на начин да стопа мутације буде увећана само на залеђеним генима.

2.4.6 Кеширање

Као што је већ наглашено у одељку 2.4.2, рачунање вредности функције циља за сваку јединку и кроз сваку генерацију изискује највећи временски део извршења GA. Са циљем оптимизације GA перформанси, у предложеној имплементацији се користи техника кеширања меморије (енгл. caching technique) која је предложена у [117]. Техника кеширања се примењује на неелитне јединке чија се појава региструје кроз неколико генерација, односно на јединке чија се појава региструје у једној генерацији, па их извесно време (кроз неку од наредних генерација) нема, а затим се опет појаве у некој од наредних генерација. Како се не би губило време на поновно израчунавање њихове функције циља, све неелитне јединке се заједно са својим генетским кодом, вредностима функције циља и индикатором валидности јединке (који је бинаран) учитавају у структуру података која се назива хеш-ред (енгл. hash-queue) табела (у даљем тексту хеш-табела). Хеш-табела је једна од најбржих структура података у програмирању. Њен капацитет се унапред задаје, при чему величина хеш-табеле не утиче на време извршења GA. Процес кеширања обухвата проверу за сваку јединку да ли је вредност њене функције циља већ израчуната и потхрањена у хеш-табели. У случају да је нема у хеш-табели, вредност функције циља јединке се по први пут рачуна и смешта у хеш-табелу по принципу LRU (енгл. last recently used) - уколико је хеш-табела попуњена, брише се вредност функције циља која није најдуже коришћена и потхрањује се нова израчуната вредност. У случају да је вредност функције циља посматране јединке већ уписана у табелу, она се само очитава.

2.4.7 Структура предложеног GA

Алгоритам 8: Структура GA са кеширањем за решавање проблема MSSP

Input: D, N_{elite}, N_{pop}
Output: solution P_{best}
1. $P \leftarrow$ *GenerateInitialPopulation* (N_{pop});
2. *SelectStopCondition ( );*
3. ***while*** *StopCondition ( ) is not satisfied* ***do***
4.   ***for*** $i = N_{elite} + 1$ ***to*** N_{pop} ***do***
5.   ***if*** *ExistInCache*(P_i) ***then***
6. $obj[i] \leftarrow$ *FindInCache*(P_i);
7.   ***else***
8. $obj[i] \leftarrow$ *CalculateObjectiveFunction* (P_i);
9.   *PlaceInCache*($P_i, obj[i]$);
10.   ***if*** *FullCache()* ***then***
11.   *ThrowOutLRUBlockCache();*
12.   ***endif***
13.   ***endif***
14.   ***endfor***
15.   *CalculateFitnessFunction* (P);
16.   *Selection* (P);
17.   *Crossover* (P);
18.   *Mutation* (P);
19. ***endwhile***
20. $P_{best} \leftarrow$ *Best* (P);

Основна структура предложене GA имплементације представљена је Алгоритмом 8. Проширење у односу на основни GA, чија је структура приказана Алгоритмом 1 (у одељку 1.3.1), представља кеширање које се извршава у оквиру *for* петље, у корацима 4 -14. Петљом се пролази кроз све неелитне јединке i и за сваку од њих се у кораку 5 проверава да ли (заједно са израчунатом вредности функције циља) већ постоји у хеш-табели или не. Уколико је посматрана јединка већ уписана у хеш-табели, вредност њене функције циља се очита и у кораку 6 додељује промењљивој $obj[i]$. У супротном, вредност функције циља i-те јединке се рачуна (корак 8) и додељује променљивој $obj[i]$, а затим се оне смештају у хеш-табелу (корак 9). Уколико је хеш-табела попуњена, нова јединка се смешта у табелу по принципу LRU (корак 11).

2.5 Експериментални резултати

У овом одељку представљени су резултати тестирања ILP формулације MSSP проблема, као и резултати потврде коректности и ефикасности предложеног GA алгоритма кроз скуп рачунарских тестирања. Тестирања су изведена на два скупа тест примера преузетих из литературе: минимални репрезентативни скупови (енгл. Minimum Hitting Set – MHS instances) [40] и Steiner троструки системи (енгл. Steiner Triple Systems - STS instances) [71]. Први скуп тест примера (MHS) садржи укупно 10 примера са различитим бројем елемената (m = 50, 100, 250, 500) и различитим бројем подскупова (n = 1000, 10000, 50000), док други скуп тест примера (STS) садржи укупно 7 примера са бројем елемената који се креће у распону од 9 до 243, и бројем подскупова који се крећу у распону од 12 до 9801. Сва тестирања су изведена на PC рачунару са једним процесорским језгром од 2,5 GHz и 1 GB RAM меморије, под Windows оперативним системом. GA алгоритам је имплементиран у програмском језику C. Експериментални резултати приказани у овом одељку презентовани су у раду [130].

Применом ILOG CPLEX 10.1 тестирана је предложена ILP формулација MSSP проблема. CPLEX решавач кроз ILP формулацију MSSP проблема као резултат враћа оптимално решење проблема или горњу границу оптималног решења у случајевима када не успе да пронађе оптимално решење у задатом временском или меморијском ограничењу. Како би се утврдила ефикасност и поузданост предложене GA имплементације, извршено је поређење најбољих резултата добијених овим алгоритмом са резултатима CPLEX решавача.

На основу експеримената са различитим комбинацијама параметара GA алгоритма, одабрани су они који највише одговарају проблему који се решава. На основу резултата тестирања, величина популације од N_{pop} = 150 јединки, са N_{elite}= 100 елитних (које директно прелазе у наредну генерацију) и $N_{nnel} = N_{pop} - N_{elite}$= 50 неелитних јединки, се показала задовољавајућом за представљање доброг компромиса између делова GA претраге који се односе на компоненте експлоатације и компоненте истраживања. За максималан дозвољени број јединки са истом вредношћу функције циља, али различитим генетским кодом, постављена је вредност константе N_{rv}= 40. У [66, 178] извршени су бројни нумерички експерименти над различитим проблемима оптимизације, који су показали да за разматране проблеме оператор FGTS даје најбоље резултате уколико се величина турнира подеси на вредност F_{tour}= 5.4. Уколико би се на 50 неелитних јединки применио оператор FGTS, тада би се спровело k_1=20 и k_2=30 турнира редом величине 5 и 6. Наведене вредност параметра усвојене су као добар избор у предложеној GA имплементацији. Време извршења оператора FGTS је $N_{nnel} \cdot F_{tour}$. У пракси се параметри F_{tour} и N_{nnel} сматрају константама (не зависе од n), што даје константну комплексност времену извршења. Након великог броја експеримента изведених са различитим вредностима вероватноће укрштања, дошло се до закључка да GA достиже најбоље резултате уколико се укрштање изврши са вероватноћом p_{cross} = 0.85. Ова вероватноћа значи да око 85% јединки учествује у стварању потомака разменом свој генетског материјала, док преосталих 15% јединки не учествује у укрштању. За залеђене гене узет је ниво мутације који је 2.5 пута већи на залеђеним генима ($p_{mut} = \frac{1.0}{n}$) у односу на основни ниво мутације гена који нису залеђени ($p_{mut} = \frac{0.4}{n}$). Управо овакав однос нивоа мутација се приликом тестирања показао као добар компромис у циљу враћања изгубљених регија претраживачког простора (генетског материјала) при евентуалној појави залеђених гена. У предложеној GA имплементацији, број кешираних вредности функције циља ускладиштен у хеш-табели је ограничен на N_{cache}= 5000.

Резултати тестирања добијени применом CPLEX решавача и GA алгоритма на MHS тест примерима, приказани су у Табели 2.1. Табела је организована тако да прве две колоне садрже редом податке о укупном броју елемената m и броју подскупова n посматраних тест примера. Трећа колона приказује вредности функције циља f_{opt} оптималних решења добијених CPLEX решавачем у случајевима када је решавач могао да заврши свој рад. Четврта и пета колона садрже податке о резултатима f и времену $t(s)$ извршавања (у секундама) CPLEX решавача. Постојало је временско ограничење од 7200 секунди (2 часа), приближно. Ознака „*опт*” се користи у случајевима где је CPLEX успео да завршио свој рад и да добије оптимално решење, док се ознака „ / ” користи у случају где је CPLEX остао без довољно меморије за рад. Шеста и седма колона се односе на резултате добијене предложеним GA алгоритмом. Резултати су представљени на исти начин као и у случају CPLEX решавача, при чему f_{best} означава вредност функције циља за најбоље добијено решење, док $t(s)$ означава укупно времене извршавања GA алгоритма. Ознака „*опт*” се користи за достигнута оптимална решења.

Табела 2.1: Резултати CPLEX и GA алгоритма за MSSP на MHS тест примерима

m	n	f_{opt}	*CPLEX* f	*CPLEX* $t(s)$	*GA* f_{best}	*GA* $t(s)$
50	1000	1000	*опт*	0.078	*опт*	2.582
50	10000	10000	*опт*	3.265	*опт*	60.039
100	1000	1000	*опт*	0.188	*опт*	4.67
100	10000	10000	*опт*	8.297	*опт*	168.603
100	50000	50000	*опт*	155.203	*опт*	683.147
250	1000	1000	*опт*	0.219	*опт*	8.626
250	10000	10000	*опт*	30.063	*опт*	336.894
500	1000	1000	*опт*	0.500	*опт*	13.325
500	10000	10000	*опт*	106.094	*опт*	437.909
500	50000	50000	/	/	*опт*	2086.517

На основу резултата приказаних у Табели 2.1, може се закључити да CPLEX проналази оптималну вредност функције циља (која је једнака укупном броју понуђених подскупова) за све MHS тест примере, изузев за највећи пример за који му је понестало меморије. Такође, резултати јасно показују да је и GA успео да достигне сва позната оптимална решења, као и оптимално решење за последњи пример, с обзиром да се за њега достиже вредност функције циља од 50000. Због своје робусности, време извршења GA је дуже у поређењу са временом извршења CPLEX решавача.

У циљу провере ефикасности предложених приступа на тежим примерима, извршено је тестирање и на тест примерима покривајућих скупова (енгл. set covering instances) изведених из Steiner троструких система. Резултати ових тестирања приказани су у Табели 2.2. Табела је структуирана на сличан начин као и Табела 2.1, уз додатак колоне са ознаком „*UB*” која садржи вредности горњих граница (енгл. Upper Bound - UB) решења у случајевима где CPLEX није завршио свој рад у временском ограничењу од 7200 секунди. Горња граница се дефинише као вредност која је гарантовано већа или једнака оптималном решењу проблема математичког програмирања. У случају да решење добијено неким алгоритмом достигне горњу границу, такво решење је вероватно оптимално. Ознака „ / ” се користи у другој колони f_{opt} , за случајеве где оптималност решења није доказана.

Табела 2.2: Резултати CPLEX и GA алгоритма за MSSP на STS тест примерима

m	n	f_{opt}	*CPLEX*			*GA*	
			f	UB	$t(s)$	f_{best}	$t(s)$
9	12	10	*опт*		0.031	*опт*	0.193
15	35	28	*опт*		0.343	*опт*	0.233
27	117	/	91	93	7200	91	0.382
45	330	/	253	302	7200	253	0.914
81	1080	/	820	1058	7200	820	2.893
135	3015	/	2278	3001	7200	2278	7.858
243	9801	/	7381	9794	7200	7381	65.409

На основу резултата приказаних у Табели 2.2 може се видети да су се STS тест примери показали изазовнијим за решавање MSSP проблема. CPLEX решавач је успео да оптимално реши само прва два тест примера, док су сви преостали примери остали ван домашаја егзактном решавачу у задатом временском оквиру. С друге стране, GA алгоритам је достигао оба позната оптимална решења при чему је за други пример (m = 15, n = 35) успео да дође до оптималног решења у краћем временском року у поређењу са CPLEX решавачем. За преостале STS примере, GA је добио решења чије су вредности близу вредности горње границе, у кратком временском року извршења.

2.6 Примена MSSP приступа у опште и образовне сврхе

Предложени MSSP приступ може се применити у опште и образовне сврхе, посебно у случајевима када је потребно решити неки од организационих проблема. Извесни организациони проблеми подразумевају расподелу одређеног броја субјеката или објеката (који су распоређени у мање групе по унапред дефинисаним критеријумима) у две дисјунктне подгрупе, тако да се у свакој од новоформираних подгрупа налази бар по један субјекат или објекат у својству представника мање групе којој иницијално припада. Неки од таквих проблема су: расподела задатака на одређени број запослених, као и обратно, расподела запослених на одређени број задатака [141]. У првом случају се за елементе коначног скупа могу узети сви задаци које је потребно да одређени број запослених изврши. Задаци се затим, сходно запосленом који је способан да их изврши, распоређују у мање групе односно подскупове. Наиме, укупан број подскупова једнак је укупном броју запослених. У зависности од циља поделе скупа свих задатака на два дисјунктна подскупа (на пример две локације), овај проблем би се могао свести на решавање MSSP проблема на начин да се максимизује број запослених који би могли да изврше задатке у оквиру два дисјунктна подскупа. У другом случају, за елементе скупа могу се узети запослени које би требало распоредили у подскупове у односу на задатак који одређени број запослених може да изврши. Укупан број подскупова био би једнак укупном броју задатака. Уколико би циљ расподеле био подела скупа запослених на два дисјунктна поскупа (на пример два временска периода, две организационе јединице) тако да се максимизује број задатака који би се могао извршити у оквиру два дисјунктна подскупа, проблем би се могао свести на решавање MSSP.

Примери практичне примене MSSP наведени су у [141]. MSSP може наћи своју употребу у високом образовању приликом распоређивања наставног особља (професора, асистената и стручних сарадника) на летњи и зимски семестар. Најчешће, циљ ове расподеле је максимизовање покривености свих наставних области у оба семестра. За сваку наставну област се формира листа особа (подскуп иницијалног скупа наставног особља) који су квалификовани за одређену област. Да би се уједно и минимизовали трошкови изнајмљивања наставног особља, потребно је сваког од њих распоредити у један од два семестра. Друга практична примена MSSP приступа представљена је на примеру поделе наставних јединица (одређеног броја тематских целина) на два семетра, тако да се максимизује број тематских целина које би се изучавале у оба семестра. MSSP приступ би такође могао наћи своју примену и у сврху састављања тестова за две групе. Сва формулисана питања, која се односе на различите тематске целине (јединице), била би у том случају распоређена у две групе тако да заступљеност тематских целина (јединица) у обе тест групе буде максимизована.

3 Проблем формирања добро балансиране експерименталне и контролне групе

У овом поглављу дефинише се и решава проблем формирања добро балансиране експерименталне и контролне групе (Well-Balanced Experimental and Control Group Formation Problem - WBECGFP). Проблем се може разматрати као полазни задатак едукативних експерименталних истраживања чија је основна сврха тестирање тачности, односно нетачности постављене хипотезе или утврђивање ефикасности предложеног приступа. Уколико се за критеријум разврставања индивидуа у групе узме балансирано распоређивање персоналних карактеристика по групама, тада проблем WBECGFP представља специјални случај NP-тешког проблема MSSP (описаног у другом поглављу). Наиме, решавањем проблема WBECGFP добија се балансирана расподела индивидуа у две партиције, тако што се минимизује разлика у броју индивидуа са истом карактеристиком између експерименталне и контролне групе.

3.1 Формулација проблема WBECGFP

У ситуацијама када едукатор или истраживач треба да утврди тачност постављене хипотезе или ефекте који се могу остварити применом предложеног приступа у процесу организовања и извођења наставе, јавља са потреба за спровођењем експеримента. Један од експеримената који се најчешће користи у те сврхе јесте такозвани контролисани експеримент. Контролисани експеримент се спроводи тако што се посматрана популација дели на две паралелне групе (експерименталну и контролну) које раде под истим, односно непромењеним условима. У оквиру експерименталне групе, чланови се излажу утицају неког фактора (методе, новине) који је предмет истраживања и чији се ефекти мере, док се чланови контролне групе не излажу утицају истог већ се уобичајено понашају. Након спровођења контролисаног експеримента, чланови обеју група се тестирају и добијени резултати се пореде. Да би се обезбедила регуларна основа за валидна поређења резултата, паралелне групе контролисаног експеримента би требало да буду што уједначеније, односно „добро балансиране" у односу на унапред утврђене критеријуме. Појам „добро балансираних" група подразумева да групе садрже приближно исти број индивидуа које испуњавају одређене критеријуме.

Пример примене WBECGFP представља проблем распоређивања студената у уједначену експерименталну и контролну групу. Нека је са A означен укупан број студената који има одређену карактеристику и који су распоређени у експерименталну групу, а са B број студената са истом карактеристиком, распоређених у контролну групу. У односу на посматрану карактеристику, расподела се сматра савршено балансираном, уколико обе групе садрже исти број студената са том карактеристиком, односно ако је A једнако B. Ако њихов број није једнак, потребно је да апсолутна вредност њихове разлике $|A - B|$ буде најмања могућа. Уколико је овај услов задовољен за сваку од карактеристика, расподела се може сматрати добро балансираном, односно оптималном.

У оквиру експеримента који је изведен над групом од 224 студената прве године Београдске пословне школе - Високе школе пословних студија, дат је приказ примене WBECGFP у реалној ситуацији.

3.2 Преглед литературе

У овом одељку је дат преглед персоналних карактеристика које су у својству различитих критеријума за распоређивање индивидуа у групе (појединачно или у комбинацији са другим карактеристикама) коришћене у литератури. Додатно, дат је приказ постојећих приступа у формирању експерименталне и контролне групе, као и мотивациони фактори који су подстакли ово истраживање.

3.2.1 Груписање засновано на карактеристикама

Последњих неколико деценија, истраживачи у области образовања посебну пажњу посвећују разноврсним карактеристикама индивидуа које се разматрају у процесу дефинисања критеријума за њихово распоређивање у групе. У зависности од истраживања које се врши, наставног контекста који се разматра и типа формације, у литератури су предложене различите перспективе груписања.

У [44] наведене су карактеристике ученика које доприносе развоју и динамици група. То су: пол, ниво студија, остварени академски успех, претходни ниво знања, искуство, сфере интересовања, циљеви учења. У [39] карактеристике попут: културно порекло, претходна знања, вештине, стилови учења, тимске улоге ученика и друге, узимају се у обзир приликом формирања група како би се створила добра радна атмосфера у групама и како би се остварили бољи исходи учења.

Различити ставови према групном облику рада, интересовања за поједине предмете, мотивација за рад, степен самопоуздања, степен стидљивости, ниво академског успеха у односу на специфичне курсеве, као и познавање језика на којем се одвија процес наставе, разматрају се као битне карактеристике у процесу формирања група у [79]. Сваку од наведених карактеристика, студенти референцирају са вредношћу у опсегу од 1 до 3, при чему 1 указује на најнижу а 3 на највишу вредност посматране карактеристике.

У [155] разматрају се: профил ученика, ниво знања, когнитивне способности, стилови учења, мотивациона/едукативна сврха и социјалне склоности, док се у [103] анализирају ученички профил, пол, способности, индивидуалне психолошке карактеристике, етничка припадност и обрасци понашања.

Као кључни фактори у формирању група у [21, 140] предлажу се разматрања персоналних и социјалних карактеристика: пола, етичке припадности, мотивације, ставова, интересовања, типова личности (полемичка, екстовертна, интровертна, итд), нивоа успеха. Значај нивоа претходног знања и груписање преферирајућих чланова групе, истиче се у [169], док се у [78] истиче значај персоналних карактеристика, склоности, вештина, знања, понашања.

Студијски програми кроз које је студент прошао, култура земље из које долази, теме интересовања и нивои знања из курса који студент похађа, представљају параметре који су разматрани у процесу формирања група у [163]. У оквиру истраживања, сваки студент је требало да изабере једну од три понуђене опције које су се односиле на ниво знања (почетни, средњи, напредни) и на ниво истраживачког интересовања (нисам заинтересован, заинтересован сам, веома сам заинтересован). У [175] разматрају се карактеристике: интересовање за предмет који се слуша, односи са вршњацима, родне разлике, индивидуалне разлике, културне разлике у оквиру групе, година рођења, карактеристике личности.

Истраживачи у [34, 35, 170] користе карактеристике као што су: расна припадност, пол и способности, док други истичу пресудан утицај психолошких карактеристика, попут самоефикасности [13]. Карактеристике студената попут: пола, етничког статуса, социо-економског статуса и типа личности, анализиране су у [2, 31, 32, 192] и дата је процена њиховог утицаја на исходе учења.

Sternberg тврди да ефикасне групе треба да садрже висок степен извршног, законодавног и судског стила мишљења [182], као и да треба да буду балансиране у односу на ову карактеристику. Исто правило се примењује и у [190], са додатним увођењем листе приоритетних захтева.

Felder-Silverman модел стилова учења коришћен је у радовима [1, 4, 80, 140, 149, 174, 200]. У [4] аутори истражују предности група које чине студенати са различитим стиловима учења, као и њихов утицај на перформансе групе. У студији [140], овај модел се користи за формирање хетерогених група. Начини идентификовања различитих стилова учења у оквиру учења заснованом на игри (енгл. Games-Based Learning) и њихове флуктуације у току процеса учења описани су у [174].

Следећа често разматрана карактеристика јесте улога коју свако од студената може преузети у групи у односу на своје знање, способности и интересовања, у погледу руковођења групом, одржавања међуљудских односа, креативности, доношења одлука и анализирања резултата рада [17]. У ту сврху најчешће примењивана подела улога је заснована на Belbin-овој теорији. Ова теорија примењена је у [12, 144, 156, 196] у својству критеријума за формирање балансираних група за колаборативно учење.

Разматрање три карактеристике (психолошки типови личности, тимске улога и друштвене склоности ученика) уз задовољење критеријума баланса тимских улога и дистрибуције психолошких стилова, предложено је у [12] за формирање група за учење. Формирање група студената на основу различитих психолошких типова личности обезбеђује покривеност свих аспеката неког пројекта, што представља један од услова за добар тимски рад [12]. Различити тестови личности разматрани су и примењени на моделе формирања група у [168, 176]. Аутори у [176] разматрају интересовања ученика према одређеној теми и Myers-Briggs психолошке типове личности, док друга група аутора у [168] креира групе од ученика са различитим карактеристикама личности.

У [154] указује се на значај социјалних вештина студената које утичу на групну интеракцију, повећањем квалитета когнитивних процеса и давањем основа за настанак одређених социо-емоционални варијабли као што су висока мотивација, ниска анксиозност и високо задовољство. Вештина просоцијалне прилагодљивости (енгл. prosocial behaviour/openness skill) је социјална вештина, чији се различити степени развијености код студената користе у [128, 154], у својству критеријума за разврставање студената у ефикасне групе за колаборативно учење.

У [15] разматра се релевантност 14 карактеристика које могу утицати на успешност групног рада и на постизање бољих резултата из математике. Посматране карактеристике су: склоност ка математици, посвећеност, пол, склоност ка енглеском језику, ниво породичног образовања, социо-економски статус породице, етничка припадност, став према групном облику рада, старост, стидљивост (интровертност), религија, интересовање за математику, мотивација за рад, самопоуздање.

У [150] понуђена је могућност разматрања великог броја карактеристика студената, при чему се оне групишу у неколико категорија: биолошке (старост, пол, итд), академске (разред, година студирања, оцене, прелиминарни тестови, самоевалуација, итд), когнитивне (стилови учења, врсте интелигенције, итд), карактеристике личности (вођство, стидљивост, степен мотивисаности, итд) и друге. Како би се проверила валидност методе предложене за формирање група, посебно су разматране три карактеристике: ниво знања, комуникацијске вештине и вештине вођства. Могућност коришћења бројних карактеристика, попут: мотивационе орјентације, постигнућа у учењу, стилови учења и других, представљена је у [105], пре свега у сврху разноврсних потреба едукатора. У циљу формирања оптималних група за учење, дат је конкретан пример у оквиру којег се разматрају сви концепти изабраног курса и поени са прелиминарног теста (ниво знања из области курса), као и додатни услови.

Већина наведених приступа полази од претпоставке да заиста постоје специфичне карактеристике студената које треба разматрати у својству различитих критеријума за формирање ефикасних група, сходно постављеним захтевима. С обзиром да на састав група утичу разни фактори (постављени циљеви, наставне методе, активности и контексти који се разматрају), може се закључити да не постоји најбољи приступ за формирање адекватних и ефикасних група. Заправо, за формирање ефикасних група, потребно је развити приступ који би омогућио флексибилност у односу на број и тип разматраних карактеристика, укључивањем различитих општих и контексно–специфичних критеријума за разврставање индивидуа у групе [204].

3.2.2 Постојећи приступи у формирању експерименталне и контролне групе

Детаљном анализом досадашњих научних истраживања, може се закључити да се у оквиру едукативних експерименталних истраживања још увек није дошло до квалитативног решења проблема WBECGFP. Отежавајући фактори у формирању добро балансираних паралелних група представљају захтеви који налажу разматрање великог броја учесника експеримента као и њихових бројних карактеристика по којима би се они требало равномерно распоредити у две групе. Према томе, сам процес формирања наведених група може представљати прилично комплексан и временски веома захтеван задатак. У том случају једноставне методе не могу наћи своју примену, па је често неопходна софтверска подршка како би се дошло до квалитетног решења у разумном временском року [68].

Генерално гледано, најчешће примењивана метода распоређивања индивидуа у две паралелне групе контролисаног експеримента јесте метода случајне расподеле. Неки од примера тако формираних експерименталних и контролних група дати су у [48, 83, 190] Такође, у великом броју радова уопште није неведено којом методом је извршена расподела студената у експерименталну и контролну групу. Један од таквих примера расподеле описан је у [204]. Такође, истраживачи веома често поделу праве тако што једно или више одељења (група) студената сврставају у експерименталну, а остала одељења (групе) у контролну групу. Пример такве поделе се може наћи у [74, 104]. Бројне су и студије у којима се за експерименталну групу узима група студената (са одређене године студија и/или у оквиру ње са посебног смера) током једне школске године, док у својству контролне групе посматра иста циљна група али током наредне школске године [41, 185]. Ипак, примена случајне методе за формирање експерименталне и контролне групе се може сматрати недовољно поузданим приступом за даља поређења добијених резултата мерења.

3.2.3 Мотивација за истраживање

Имајући у виду значај који контролисани експерименти имају у области експерименталних истраживања, настала је идеја о разматрању проблема формирања добро балансиране експерименталне и контролне групе у односу на бројне карактеристике индивидуа и развијања ефикасног приступа за његово решавање. Тако формиране групе требало би да обезбеде регуларну основу за валидна поређења која представљају битне чиниоце научних истраживања [7].

Мотивациони фактори који су утицали на сагледавање WBECGFP као проблема комбинаторне оптимизације, те у складу с тим и на развијање метахеуристичког приступа за његово решавање, изведени су кроз процес анализе: различитих модела распоређивања индивидуа у групе, различитих комбинација карактеристика у односу на које су вршене расподеле, оптимизационих проблема сличне структуре и понуђених метода за њихово решавање. Најзначајнији међу наведеним факторима описани су у параграфима који следе.

Као што је већ наглашено, проблем WBECGFP се може посматрати као специјални случај NP-тешког проблема MSSP. Други проблем који је послужио као мотивација за прецизније формулисање и решавање WBECGFP, био је проблем добро балансиране расподеле у групе (енгл. well-balanced assignment problem) у односу на разматране карактеристике студената. Више детаља о наведеном проблему и различитим приступима за његово решавање могу се наћи у деловима 4.2.2 и 4.2.3.

Разматрајући значај који свака од карактеристика индивидуа (наведених у одељку 3.2.1) остварује у исходима процеса формирања група, дошло се до идеје груписања најрелевантних међу њима. Додатно узимајући у обзир и њихове различите нивое манифестације, предложене су бројне карактеристике погодне за за балансирано распоређивање индивидуа у две паралелне групе за поређења.

Идеја заступљености свих карактеристика у првој и другој групи, без могућности груписања индивидуа са истом карактеристиком у једној од група, сагласна је са концептом разноврсне, односно хетерогене структуре група, док концепт баланса подразумева уједначен број индивидуа са истом карактеристиком у обе групе. Увођењем приоритета међу карактеристикама, испуњени су додатни захтеви за уравнотеженост група у односу на специфичне карактеристике.

Како је прегледом литературе установљено да проблем формирања добро балансиране експерименталне и контролне групе (заснован на минимизовању разлике у броју индивидуа са истом карактеристиком између група и могућности разматрања приоритета међу карактеристикама) није до сада истражен, у овом поглављу представљен је нови приступ решавању овог проблема. У оквиру предложеног приступа разматрају се следеће карактеристике: пол, година рођења, регион сталног боравка, личне карактеристике (комуникационе вештине, стидљивост, самопоуздање), мотивација за рад, став према групном раду, вештина просоцијалне прилагодљивости, стилови мишљења, стилови учења, психолошки типови личности, улоге у тиму, заинтересованост за математику, математичке способности и ниво знања из математике. Запажено је и да наведена комбинација персоналних карактеристика није раније разматрана у литератури у својству критеријума за балансирано распоређивање индивидуа у групе.

3.3 Предложени приступ у решавању проблема WBECGFP

У овом одељку приказан је математички модел проблема WBECGFP и нови метахеуристички приступ за његово решавање. Представљени приступ се заснива на методи променљивих околина и имплементиран је у апликацију која као резултат враћа балансирану расподелу индивидуа у две групе. Понуђени приступ пружа могућност разматрања великог броја учесника расподеле и великог броја њихових карактеристика на основу којих се могу дефинисати критеријуми за балансирано распоређивање индивидуа у експерименталну и контролну групу.

3.3.1 Нелинеарна математичка формулација проблема WBECGFP

Нека су параметри и променљиве проблема дефинисани на следећи начин:

$S = \{s_1, s_2, s_3, \dots, s_n\}$ - скуп од n индивидуа; (3.1)

$K = \{k_1, k_2, k_3, \dots, k_m\}$ - скуп од m карактеристика индивидуа; (3.2)

$$k_{ij} = \begin{cases} 1, & \text{ако индивидуа } s_i \text{ има карактеристику } k_j \\ 0, & \text{иначе} \end{cases}; \qquad (3.3)$$

$$i = 1, 2, \dots, n, \quad j = 1, 2, \dots, m;$$

$$x_i = \begin{cases} 0, & \text{ако је } s_i \text{ распоређена у групу } A \\ 1, & \text{ако је } s_i \text{ распоређена у групу } B \end{cases}. \qquad (3.4)$$

$$i = 1, 2, \dots, n.$$

Свака индивидуа може имати више карактеристика, и више индивидуа може имати исту карактеристику. Да би се направила веза између индивидуа и њихових карактеристика, уводи се бинарни индикатор k_{ij}. Свакој карактеристици се додељује одговарајућа тежина w_j, чија вредност одређује степен приоритета карактеристике k_j у односу на остале карактеристике. Повећањем тежине неке карактеристике повећава се и приоритет захтева да број индивидуа са том карактеристиком буде што уједначенији у обе групе. Ово је посебно од значаја у случају када се индивидуе не могу распоредити на начин да обе групе имају уједначен број индивидуа у односу на сваку од карактеристика k_j. У супротном, уколико не постоје приоритети међу карактеристикама, подразумевана вредност тежине сваке карактеристике је $w_j = 1$.

С обзиром да је потребно све индивидуе распоредити у две групе A и B, са једнаким бројем чланова, пожељно је да укупан број индивидуа буде паран. У супротном, индивидуа која је „сувишна”, односно нераспоређена, произвољно се распоређује у једну од група.

Нека је са $x_i \in \{0, 1\}$ дефинисана бинарна променљива, таква да је за вредност $x_i = 0$, индивидуа s_i распоређена у групу A, док је у супротном (за вредност $x_i =$

1) индивидуа s_i распоређена у групу B. Ознака A_j користи се за укупан број индивидуа који имају карактеристику k_j у групи A, док се ознака B_j користи за укупан број индивидуа са истом карактеристиком у групи B.

Користећи предложену нотацију, проблем WBECGFP може бити формулисан следећим математичким моделом:

$$min \sum_{j=1}^{m} w_j \left|A_j - B_j\right| \tag{3.5}$$

при ограничењима :

$$A_j = \sum_{i=1}^{n}(1 - x_i)k_{ij}, \qquad j = 1,2, \dots, m; \tag{3.6}$$

$$B_j = \sum_{i=1}^{n} x_i k_{ij}, \qquad j = 1,2, \dots, m; \tag{3.7}$$

$$\sum_{i=1}^{n} x_i = \sum_{i=1}^{n}(1 - x_i); \tag{3.8}$$

$$x_i \in \{0,1\}, \qquad i = 1,2, \dots, n. \tag{3.9}$$

Функција циља (3.5) је дефинисана као сума апсолутних разлика укупног броја индивидуа са истом карактеристиком k_j у групама A и B, узимајући у обзир све карактеристике k_j $(j = 1,2, \dots, m)$ и њихове тежине w_j. Циљ проблема WBECGFP је минимизација функције циља. Ограничења (3.6) и (3.7) одређују редом укупан број индивидуа са карактеристиком k_j у групи A и групи B, док ограничење (3.8) обезбеђује да групе A и B имају једнак број индивидуа. Бинарна природа променљиве x_i дата је условом (3.9).

У предложеном математичком моделу (3.5)-(3.9), функција циља (3.5) је нелинеарна, а све променљиве су целобројне, па предложени модел представља формулацију нелинеарног целобројног програмирања (енгл. Nonlinear Integer Programming - NLIP). Изложени модел нелинеарне математичке формулације проблема WBECGFP има укупно n променљивих и $2m + 1$ ограничења.

3.3.2 Метода променљивих околина за решавање проблема WBECGFP

У овом одељку је описана метода променљивих околина, посебно развијена за решавање проблема расподеле студената у две балансиране групе у односу на карактеристике студената.

Структуре околина и функција циља

Нека променљива N_p $(p = 1, \ldots, p_{max})$ означава низ дефинисаних структура околина, тада променљива $N_p(x)$ представља скуп решења у p-тој околини од x.

Једно решење x разматраног проблема представља једну расподелу студената у групе. Структуре околина дефинисане су на начин да се решење x' налази у p-тој околини решења x, у ознаци $x' \in N_p(x)$, уколико се расподела x' може добити из расподеле x са највише p замена по једног пара студената између различитих група.

Функција циља се дефинише једначином $\sum_{j=1}^{m} w_j |A_j - B_j|$, представљеном у одељку 3.3.1. За рачунање функције циља потребно је за сваку од карактеристика k_j, за $j = (1, \ldots, m)$, израчунати укупан број студената са карактеристиком k_j у групи A (у ознаци A_j) и број студената са истом карактеристиком у групи B (у ознаци B_j). Односно, у свакој од група, за сваког студента проверава се да ли има или нема карактеристику k_j.

Иницијално решење

Иницијално решење WBECGFP проблема добија се RVNS алгоритмом (видети одељак 1.3.3 и Алгоритам 4). Како RVNS представља варијанту VNS методе која укључује само фазу размрдавања, то се његовом применом избегава дуготрајно и сложено извршавање процедуре локалне претраге у оквиру једне VNS итерације. Из наведених разлога, за иницијално решење основног VNS алгоритма (видети одељак 1.3.3 и Алгоритам 5) посебно развијеног за решавање WBECGFP проблема узима се решење добијено RVNS алгоритмом. Иницијално решење RVNS алгоритма добија се случајном расподелом студената у две групе.

Унапређена локална претрага

У фази локалне претраге, током које се детаљно претражују околине тренутног решења, потребно је велики број пута израчунати функцију циља, по једном за свако новодобијено решење у околини.

Нека су са $Char1$ и $Char2$ означена два низа целих бројева дужине m. Низ $Char1[j]$ представља број студената који имају карактеристику k_j у групи A, док $Char2[j]$ представља број студената са истом карактеристиком k_j у (другој) групи B. У том случају, приликом замене студента s_1 из групе A са студентом s_2 из групе B, што даље резултира да s_1 буде у групи B, а s_2 у групи A, за сваку од карактеристика k_j важиће следеће:

$$Char1[j] = Char1[j] - 1, \quad \text{ако } s_1 \text{ има карактеристику } k_j,$$

$$Char2[j] = Char2[j] + 1, \quad \text{ако } s_1 \text{ има карактеристику } k_j,$$

$$Char2[j] = Char2[j] - 1, \quad \text{ако } s_2 \text{ има карактеристику } k_j,$$

$$Char1[j] = Char1[j] + 1, \quad \text{ако } s_2 \text{ има карактеристику } k_j.$$

Осим тога, функција циља не мора да се рачуна од почетка. Довољно је пре промене вредности низова $Char1$ и $Char2$, одузети тренутни допринос карактеристике k_j:

$$obj\_f = obj\_f - t_j|Char1[j] - Char2[j]|,$$

а након промене вредности низова, додати нови допринос карактеристике k_j:

$$obj\_f = obj\_f + t_j|Char1[j] - Char2[j]|.$$

Да би се на овај начин израчунала функција циља, потребна је једна итерација кроз низ карактеристика, временске сложености $O(m)$, што је значајно побољшање у поређењу са временском сложености која у општем случају износи $O(mn)$.

Параметри алгоритма

Да би предложени VNS алгоритам понудио што боље резултате, потребно је одабрати погодне параметре са којима ће алгоритам ефикасно проналазити делове простора решења доброг квалитета. На основу експеримената са различитим комбинацијама параметара, одабрани су они који највише одговарају проблему који се решава. Максималан број околина коришћен у фази размрдавања RVNS алгоритма је $p = 3$, док је максималан број околина коришћених у фази размрдавања VNS алгоритма $p = min\{\frac{n}{6}, 20\}$. Услов заустављања RVNS алгоритма је 20000 узастопних итерација без побољшања решења, док је услов заустављања VNS алгоритма 20 узастопних итерација без побољшања решења.

3.4 Експериментално истраживање

Истраживање је спроведено над групом студената с циљем демонстрације квалитета представљеног WBECGFP модела и потврде ефикасности предложеног VNS алгоритма. У овом подпоглављу је дат детаљан опис: коришћених инструмената у процесу прикупљања података о студентима, начина дефинисања карактеристика студената и осталих улазних параметара расподеле, и основни циљеви истраживања.

3.4.1 Циљеви истраживања

Један од кључних корака у структури научног истраживања јесте прецизно дефинисање истраживачких циљева, односно истраживачких питања. У оквиру истраживања описаном у овом поглављу представљен је нов приступ формирању балансиране експерименталне и контролне групе, и у складу са тим постављена су главна истраживачка питања. Питања се односе на ефикасност примене предложеног VNS алгоритма за решавања проблема WBECGFP и на могућност примене предложеног приступа у настави, у циљу достизања уједначеног академског успеха (завршних оцена) студената двеју група на испиту. На основу питања, формулисане су следеће хипотезе истраживања:

Хипотеза 3.1. Применом предложеног VNS алгоритма у кратком временском року добија се добро балансирана расподела студената у експерименталну и контролну групу, која је знатно равноправнија у односу на расподеле које се иначе користе у настави.

Хипотеза 3.2. Студенти експерименталне и контролне групе, које су формиране предложеним WBECGFP приступом, постижу уједначене резултате и у односу на академски успех студената на испиту.

3.4.2 Учесници истраживања

Студија истраживања је спроведена над групом од 224 студената прве године Београдске пословне школе - Високе школе пословних студија, у оквиру наставе из Математике током првог семестра школске 2014/2015. године. Настава се изводила у оквиру часова предавања и вежби, а сви учесници истраживања су имали једнаке услове за рад. Студенти су унапред информисани о планираном истраживању и о потреби за њиховим континуираним анкетирањем с циљем прикупљања података потребних за регуларну категоризацију сваког студента, у односу на утврђену групу персоналних карактеристика. Такође, студенти су обавештени да ће сви подаци, укључујући и постигнуте резултате на првом колоквијуму као и завршне оцене из предмета Математика, бити коришћени само у истраживачке сврхе.

3.4.3 Прикупљање података

У циљу валидног прикупљања података и дефинисања карактеристика потребних у процесу распоређивања студената у две паралелне групе, коришћени су упитници. За податке који се односе на ниво усвојеног знања из Математике, коришћени су резултати са првог колоквијума. Узимајући у обзир различите вредности испитаних карактеристика студената, дефинисано је укупно 116 карактеристика у односу на које се вршила расподела. С обзиром да је прикупљене квалитативне податке требало претворити у квантитативне податке како би се они могли користити у алгоритму, све карактеристике су представљене преко бинарних вредности, при чему вредност 1 означава да студент има посматрану карактеристику, док вредност 0 значи да је нема.

У првој недељи семестра студентима су подељена два упитника. Први упитник је садржао питања која се односе на пол, годину рођења, регион сталног боравка, личне карактеристике (комуникацијске вештине, стидљивост, самопоуздање), мотивацију за рад, став према групном раду, степен заинтересованости за математику и ниво математичких способности. Питања са вишеструко понуђеним одговором су формулисана за пол, годину рођења и регион сталног боравка, омогућавајући студенту да изабере само један од одговора. Поред тога, за питања која су се односила на личне карактеристике студената (мотивацију за рад, став према групном раду и степен заинтересованости за математику) у својству одговора били су понуђени различити нивои заступљености карактеристике (висок, средњи, низак). За питање о нивоима развијености математичких способности била су понуђена три одговора (одлично, средње, лоше). На основу првог упитника, узимајући у обзир све разматране карактеристике студената и њихове различите вредности, дефинисане су укупно 33 карактеристике.

Да би се утврдио степен развијености вештине просоцијалне прилагодљивости код појединаца, студенти су решавали други упитник који се састојао од самореференцирајућих изјава које су студенти оцењивали на скали од 4 тачке [не слажем се уопште (1) - у потпуности се слажем (4)], а који је преузет из [154]. На основу броја бодова које је сваки од студената освојио решавањем упитника, вршило се распоређивање студената у један од четири интервала развијености вештине просоцијалне прилагодљивости (0-1, 1-2, 2-3, 3-4). У односу на наведене интервале, формиране су 4 нове карактеристике.

У току друге недеље семестра спроведен је трећи упитник посвећен стиловима мишљења. Да би се утврдио степен сагласности, односно несагласности појединца са одређеним стилом, коришћен је упитник „Инвентар стилова мишљења” (енгл. The Thinking Styles Inventory) из [183]. Аналогно истраживању изложеном у [190], за потребе истраживања разматрана је димензија функционалности која обухвата три стила: законодавни, извршни и судски стил. Студентима је понуђено да за сваки од три стила заокруже једну од четири понуђених опција [не слажем се уопште (1)

- потпуно се слажем (4)]. На основу трећег упитника, одређено је нових 12 карактеристика за распоређивање студената у групе.

Како би се на адекватан начин испитали различити стилови учења, студенти су у трећој недељи семестра одговарали на питања у оквиру адаптационе варијанте упитника „Индекс стилова учења” (енгл. The Index of Learning Styles - ILS) који су развили Felder и Soloman 1997. године. Четврти упитник се састојао од 44 ставки преузетих из ILS упитника, као и додатних ставки које су се односиле на димензију Организације. Ова димензија није обухваћена ILS упитником, иако су је Felder и Silverman у [60] представили као посебну димензију још 1988. године. При попуњавању упитника, требало је одабрати по једну од понуђених опција за сваку од ставки која се односила на различите стилове учења. У односу на разматране стилове учења, дефинисано је додатних 10 карактеристика.

У циљу стварања што меродавнијих група за тестирање, уврштена је и категоризација студената сходно Myers-Briggs моделу типова личност. Коришћен је „Myers-Briggs Индикатор” психолошких типова личности (енгл. Myers-Briggs Type Indicator - MBTI**)** из [152, 153], у облику он-лајн упитника доступног на адреси [102]**.** Интроспективни онлајн MBTI упитник конструисале су Katharine Cook Briggs и њена ћерка Isabel Briggs Myers у [152, 153]. Био је то пети по реду упитник који су студенти попуњавали у четвртој недељи семестра. На основу овог упитника, дефинисано је нових 16 карактеристика студената.

Како би се утврдио ниво склоности коју свако од студената има ка одређеној улози у групи (тиму), коришћен је самовреднујући тест Белбинових улога у тиму (енгл. The Belbin Team-Role Self-Perception Inventory - BTRSPI) преузет из [16, 17]. Студенти су овај упитник попуњавали у петој недељи семестра. На основу резултата упитника извршена је класификација студената у складу са нивоом предиспозиције за одређену улогу (који је степенован као низак, просечан, висок или веома висок). Укупно је 36 нових карактеристика формирано на основу овог упитника.

На крају шесте недеље спроведен је први колоквијум из Математике, с циљем утврђивања нивоа усвојеног знања из пређеног градива. Максималан број бодова који се могао освојити на првом колоквијуму, а који ће се у даљем тексту наводити као прелиминарни тест, је 50 поена. Поени са прелиминарног теста су коришћени у циљу класификације студената у оквирима освојених поена (0-10, 11-20, 21-30, 31-40, 41-50), и на тај начин дефинисано је додатних 5 карактеристика.

3.4.4 Параметри распоређивања студената

За потребе истраживања развијена је апликација у коју је имплементиран програм заснован на VNS алгоритму и која нуди могућност решавања представљеног проблема WBECGFP. Апликација је развијена на .NET Framework платформи (варијанта 3.5) у програмском језику C#. Улазни параметри аликације су: листа студената, листа карактеристика и матрица доделе карактеристика студентима, која јасно дефинише да ли i-ти студент има или нема j-ту карактеристику. Осим тога, апликација узима у обзир и тежине карактеристика, смештене у посебан ред матрице. Када програм заврши са радом, апликација као резултат враћа расподелу студената у две групе добро балансиране у односу на број студената са одређеном карактеристиком. Апликација је доступна на адреси [101].

3.4.5 Начини формирања група

У циљу процене ефикасности предложеног VNS алгоритма, спроведена су тестирања на различитим димензија проблема. Генерисани су тест примери различитих димензија засновани на реалним подацима о групама које су чинили: 50, 60, 80, 100, 120, 150, 180, 210 и 224 студената. За сваки од тест примера, студенте је требало распоредити у две што уједначеније (добро балансиране) групе у односу на посматране карактеристике. Алгоритам је тестиран на рачунару са AMD FD-7500 (2.10 GHz) процесором, 8 GB RAM меморије и оперативним системом Windows 8.1.

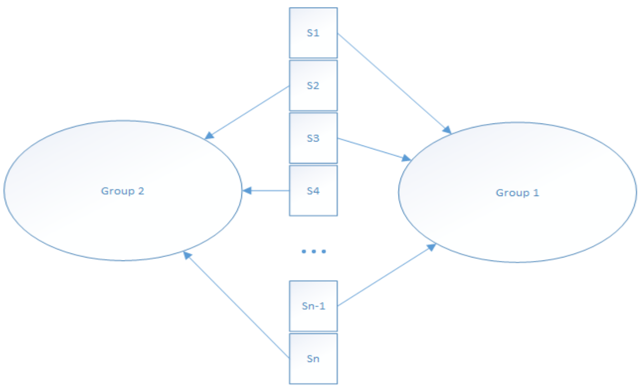
За сваки тест пример студенти су распоређени у две групе на 9 различитих начина:

1. Предложеним VNS алгоритмом,
2. Према презимену,
3. Према броју индекса,
4. Према броју освојених поена на колоквијуму,

5. Случајном расподелом,
6. Према полу студената, а затим по презимену,
7. Према полу студената, а затим по броју индекса,
8. Према полу студената, а затим по броју освојених поена на колоквијуму,
9. Према полу студената, а затим случајном расподелом.

За расподелу према броју освојених поена на колоквијуму, студенти су прво сортирани према успеху на колоквијуму, а затим су распоређивани тако што је први распоређен у прву групу, други у другу, трећи у прву, четврти у другу, итд (Слика 3.1). У погледу расподеле према броју индекса, студенти су прво сортирани према броју индекса, а затим је од прве половине студената формирана једна група, а од друге половине друга група. За поделу према презимену, студенти су прво сортирани по презимену, а затим распоређени у две групе на сличан начин као код расподеле по броју освојених поена на колоквијуму. Код случајне расподеле, студенти су насумично распоређивани у две групе.

Према расподелама наведеним под редним бројем 6, 7, 8 и 9, студенти су прво подељени у две групе - у групу у којој су само мушкарци и у групу у којој су само девојке. Затим су те две групе, свака посебно, подељене аналогно начинима који су наведени под ставкама 2, 3, 4 и 5. На овај начин су добијене по четири групе: G1_мушкарци, G2_мушкарци, G1_жене и G2_жене. Коначне групе расподела под редним бројем 6, 7, 8 и 9, су формиране тако што је прва група добијена спајањем група G1_мушкарци и G1_жене, док је друга група добијена спајањем група G2_мушкарци и G2_жене.



Слика 3.1: Распоређивање студената у односу на резултате са колоквијума

3.4.6 Индикатори поређења

На крају првог семестра одржан је завршни тест (пост-тест) за полагање испита из Математике. Постигнућа студената на пост-тесту су рангирана оценама од 5 до 10. Оцена 5 значи да студент није успешно положио испит; оцена 6 је најслабија пролазна оцена на испиту, док оцена 10 представља најбољу оцену коју студент може да добије на испиту. Завршне оцене студената се не налазе на списку карактеристика у односу на које се студенти распоређују у групе, већ се користе у сврху потврде квалитета расподеле. Прецизније, уколико су студенти равномерно распоређени у две групе у погледу одабраних карактеристикама, очекује се да на крају расподеле у свакој од група буде приближно исти број студената који нису успешно положили испит, приближно исти број студената који су добили оцену 6, оцену 7, итд.

Нека променљива n_{XY} представља укупан број студената из групе X који су на завршном тесту из Математике добили оцену Y. На пример, n_{A7} представља укупан број студената из групе A који су на завршном испиту добили оцену 7.

У циљу потврде ефикасности развијеног VNS алгоритма и процене адекватности предложеног модела, поред рачунања функције циља и времена извршења алгоритма, за свако решење се уводе још две мере квалитета расподеле, које се рачунају у односу на завршне оцене студената. То су:

$$\chi - test = \sqrt{x}\,, \quad \text{где је}$$

$$x = (n_{A10} - n_{B10})^2 + (n_{A9} - n_{B9})^2 + (n_{A8} - n_{B8})^2 + (n_{A7} - n_{B7})^2 + (n_{A6} - n_{B6})^2 + (n_{A5} - n_{B5})^2; \tag{3.10}$$

$$diff = \Delta = |n_{A10} - n_{B10}| + |n_{A9} - n_{B9}| + |n_{A8} - n_{B8}| + |n_{A7} - n_{B7}| + |n_{A6} - n_{B6}| + |n_{A5} - n_{B5}|. \tag{3.11}$$

Мање вредности наведених мера квалитета означавају расподелу већег квалитета.

У сврху поређења сви представљених начина за формирање експерименталне и контролне групе, мерене су вредности четири индикатора:

- f – функција циља расподеле по формули (3.5) ;

- $t(s)$ – време потребно да се добије одговарајућа расподела. Мерење времена има смисла само у случају VNS алгоритма;
- $\chi - test$ – заснован на χ^2 расподели по формули (3.10);
- $diff$ – сума апсолутних разлика по формули (3.11).

3.5 Експериментални резултати

Резултати тестирања свих предложених начина расподеле студената у експерименталну и контролну групу приказани су у Табели 3.1. У првој колони приказани су редни бројеви различитих начина расподеле, које су дефинисане и нумерисане у одељку 3.4.5. Наредне три колоне садрже резултате расподела за различите тест примере које чине редом групе од 50, 60 и 80 студената, затим од 100, 120 и 150 студената, и највеће групе од 180, 210 и 224 студената. За сваки тест пример, мери се ефикасност понуђених начина у односу на четири индикатора: вредност функције циља f, време извршења $t(s)$ у секундама, χ-$test$ и $diff$, представљених у одељку 3.4.6. Вредности индикатора су приказане редом у четири подколоне за сваку од посматраних група студената.

Резултати приказани у Табели 3.1 показују да је функција циља f за сваки тест пример значајно мања у случају расподеле постигнуте VNS алгоритмом него у случају било које друге расподеле. То потврђује да расподеле добијене VNS алгоритмом имају равномерније распоређене карактеристике студената у обе групе и да су самим тим и квалитетније. Такође, може се приметити да је предложени VNS алгоритам и временски веома ефикасан. Расподеле у којима учествује мање од 100 студената су решене за мање од 3 секунде, док је расподела 210 студената најдуже решавана, али и за њу је било потребно 65 секунди. На овај начин је потврђена Хипотеза 3.1.

Добијене вредности индикатора χ-$test$ и $diff$ су значајно ниже у случају расподеле постигнуте VNS алгоритмом него у случају осталих расподела. На основу ових индикатора, може се закључити да су расподелом заснованом на карактеристикама студенти значајно равноправније распоређени и у односу на завршне оцене студената на испиту, чиме се потврђује Хипотеза 3.2.

Табела 3.1: Резултати различитих начина расподеле студената у експерименталну и контролну групу

Расподеле	f	$t(s)$	χ-$test$	$diff$	f	$t(s)$	χ-$test$	$diff$	f	$t(s)$	χ-$test$	$diff$
	Број студената				Број студената				Број студената			
	50				60				80			
1	104	2.04	5.83	12	120	2.48	7.21	16	117	2.97	2.83	6
2	214		7.87	16	278		8.94	20	313		15.62	36
3	212		10.68	20	242		8.72	18	341		12.33	24
4	260		6.48	14	252		9.17	22	301		9.38	18
5	250		8.37	18	326		11.31	22	329		9.38	22
6	226		9.27	18	280		9.59	18	321		12.96	24
7	218		6.48	14	350		11.49	22	327		12.81	28
8	238		8.60	18	222		8.25	16	301		8.72	18
9	262		6.78	14	226		9.80	20	343		11.83	28
	100				120				150			
1	121	7.92	6.48	14	116	14.30	6.93	14	106	13.03	6.16	12
2	351		11.40	24	382		15.87	38	412		14.21	30
3	375		15.81	36	384		15.10	32	434		19.54	42
4	347		11.58	20	356		10.00	22	400		10.49	22
5	377		11.05	24	340		14.97	28	498		17.49	34
6	299		14.63	28	392		16.12	32	472		15.17	32
7	375		16.31	36	400		15.23	32	424		22.32	42
8	353		9.27	22	350		10.00	18	394		9.90	20
9	383		14.07	26	358		12.81	28	404		14.35	32
	180				210				224			
1	100	23.99	3.46	6	92	65.49	9.38	18	93	46.08	7.35	12
2	442		16.49	36	540		19.08	42	535		19.03	42
3	562		22.89	48	608		29.66	62	661		22.58	50
4	388		14.42	34	466		20.30	36	471		10.86	26
5	474		15.36	36	542		26.68	46	505		21.12	36
6	346		16.37	36	490		16.25	36	525		20.45	42
7	582		23.58	48	558		27.20	58	651		22.67	50
8	502		16.25	30	438		11.31	24	523		28.81	48
9	442		16.73	36	510		20.69	42	567		19.95	44

Детаљан приказ највећег примера на којем је тестиран VNS алгоритам, и у оквиру којег се група од 224 студената дели у две групе од по 112 студената, представљен је у Табели 3.2. Прва колона приказује редни број расподеле која се разматра. У наредних 6 колона, приказане су завршне оцене студената из предмета Математика. Свака од колона са завршном оценом, редом кореспондирајући оценама од 10 до 5, је додатно подељена на две подколоне. Прва подколона презентује укупан број студената који су добили одговарајућу завршну оцену и који су распоређени у групу А, док друга подколона представља укупан број студената са истом оценом, распоређених у групу B.

Табела 3.2: Расподела завршних оцена 224 студената добијена VNS алгоритмом

Расподеле	10		9		8		7		6		5	
	A	B	A	B	A	B	A	B	A	B	A	B
1	9	13	8	9	14	14	30	30	30	24	21	22
2	8	14	9	8	9	19	35	25	32	22	19	24
3	14	8	10	7	10	18	38	22	23	31	17	26
4	13	9	7	10	11	17	28	32	29	25	24	19
5	11	11	8	9	11	17	39	21	26	28	17	26
6	10	12	10	7	21	7	24	36	29	25	18	25
7	14	8	10	7	10	18	38	22	22	32	18	25
8	12	10	7	10	14	14	21	39	38	16	20	23
9	13	9	7	10	17	11	24	36	33	21	18	25

Резултати приказани у Табели 3.2 показују да расподела добијена предложеним VNS алгоритмом равномерно распоређује студенте и у односу њихових завршних оцена на испиту. Наиме, број студената који су добили исту завршну оцену а распоређени су у групу А и групу B је доста сличан, а често и идентичан. Највећа разлика је у колони која одговара завршној оцени 6, а коју је у групи А добило 30 студената, док у групи B 24 студената. За остале типове расподеле ове разлике су много веће, што показује да су такве расподеле знатно неравноправније у односу на расподеле добијене VNS алгоритмом.

На основу претходно наведеног, може се закључити да WBECGFP представља квалитетан и ефикасан, нови методички приступ за формирање експерименталне и контролне групе, који пружа могућност коришћења великог броја карактеристика (у конкретном случају 116) у својству критеријума за балансирано распоређивање индивидуа са одређеном карактеристиком у две групе.

4 Проблем формирања *k* добро балансираних група

У овом поглављу дефинише се и решава проблем формирања k добро балансираних група (Balanced Multi-Weighted Attribute Set Partitioning Problem - BMWASP). Проблем BMWASP се састоји у проналажењу поделе (партиције) скупа индивидуа, које имају своје карактеристике, у одређени број (*к*) група тако да новоформиране групе буду што је могуће више балансиране по броју индивидуа са истом карактеристиком. Проблем BMWASP је решаван на начин да пружа могућност разматрања великог броја индивидуа, као и великог броја њихових карактеристика, при чему свака од карактеристика има своју тежину чија вредност одређује степен њеног приоритета у односу на остале карактеристике. Проблем BMWASP представља проширење и надградњу проблема WBECGFP (поглавље 3) и проблема BMASP (енгл. Balanced Multi-Attribute Set Partitioning problem - BMASP). Додатно, BMWASP проблем представља и специјални случај NP-тешког проблема MSSP (поглавље 2).

4.1 Формулација проблема BMWASP

У образовним институцијама често постоји потреба за распоређивањем популације индивидуа у одређени број мањих група чија стуктура репрезентује полазну групу [49]. Особа која настоји да реши овај проблем (најчешће едукатор или организатор наставе) требало би да одреди карактеристике индивидуа које ће користити у њиховој категоризацији, а затим на основу њих да дефинише критеријуме за распоређивање индивидуа у одређени број група. Могућност додатног разматрања релативних приоритета међу дефинисаним критеријумима може допринети већем задовољењу постављених захтева у процесу формирања група [41]. Међутим, проблем распоређивања већег броја индивидуа у групе уз разматрање више персоналних карактеристика и критеријума расподеле (као и релативних приоритета међу карактеристикама и међу критеријумима) представља тежак и временски веома захтеван задатак [163] који због своје сложености захтева употребу неке од рачунарски заснованих метода оптимизације.

У циљу решавања оваквих и сличних захтева, у овом поглављу је описан и решаван проблем BMWASP. Овај проблем може бити посматран као уопштење проблема балансираног партиционисања скупа, познатог у литератури као проблем BMASP. Проблем BMASP је представљен и решаван у [114], и његов циљ је проналажење партиције скупа објеката који имају своје карактеристике, у одређени број група тако да су све групе што је могуће више уједначене по броју објеката са истом карактеристиком. Додавањем тежине свакој од карактеристика, добија се уопштени BMWASP проблем.

Са становишта математичке оптимизације, проблем BMWASP се сврстава у домен проблема са вишеструким циљем одлучивања (енгл. multi-objective decision problems) чија се примена може видети у бројним примерима стварних животним ситуација, од распоређивања студената у студијске групе у [123] до прављења распореда нивоа за JIT траке (енгл. JIT assembly lines) у [181]. Један од примера се може наћи и у образовању, у ситуацијама када истраживачи настоје да процене ефекат који примена нове наставне методе може имати на исходе учења. BMWASP може наћи своју непосредну примену и приликом формирања одељења и студијских група, са унапред задатим карактеристикама и листом приоритета међу њима, а у циљу унапређења процеса организовања и извођења наставе. Неки од осталих примера могу се наћи у у процесу процењивања учинка саветовања, као и у процесу тестирања медицинских третмана.

Поштујући циљеве и ограничења дата од стране администратора Београдске пословне школе - Високе школе струковних студија, BMWASP је директно примењен за распоређивање 229 студената исте студијске групе у 10 балансираних група за ефикасно праћење наставе и усвајање нових садржаја.

4.2 Преглед литературе

У овом одељку дат је преглед литературе који се односи на два општа критеријума за груписање индивидуа, постојеће приступе за решавање проблема добро балансиране расподеле у групе и мотивационе факторе који су иницирали ово истраживање.

4.2.1 Критеријум разноврсности унутар групе

Један од најчешће коришћених критеријума груписања јесте максимизовање разноврсности (енгл. maximizing diversity) унутар група и минимизовање разлика (енгл. minimizing the differences) између група [104]. С једне стране циљ је формирати разноврсне групе, тако да индивидуе имају могућност стицања искуства у раду са особама који се разликују од њих самих, а с друге стране циљ је да тако формиране групе буду што сличније по својој структури. У литератури се наведени начин расподеле индивидуа дефинише и решава као проблем максималног разноврсног груписања (енгл. Maximum Diverse Grouping Problem - MDGP) који се састоји у проналажењу начина да се подели скуп индивидуа у међусобно дисјунктне групе, тако да укупна разноврсност између индивидуа, које припадају истој групи, буде максимизована [61].

У [19] аутори предлажу поједностављен модел у оквиру којег се популација дели на „фамилије” са високим степеном унутар-фамилијарне сличности и међу-фамилијарне различитости. Максимизовање разноликости унутар групе, засноване на различитим нивоима предзнања, и минимизовање разлике између група, засноване на различитим типовима интересовања за учењем, усвојени су као добри принципи расподеле ученика у групе у [44], тако наглашавајући предности формирања сличних мултидисциплинарних група. Формирање интер-хомогених и интра-хетерогених група максимизовањем разноликости унутар групе за неке карактеристике и минимизовањем разноликости између група за друге карактеристике понуђен је као принцип за груписање студената у [150]. Распоређивање студената у групе у [193] одређено је максимизовањем разноврсности чланова групе у погледу бројних карактеристика (година старости, пола, националне припадности, итд) и минимизовањем средње вредности разлика између група. Аутори у [194] формирају максимално разноврсне групе максимизовањем разлика између свих парова студената у оквиру свих групама. У [104] се студенти распоређују у групе које треба да буду максимално разноврсне, уз увођење додатног критеријума - да у групама буде једнако заступљене све релевантне вештине студената, како би се избегли потенцијални недостаци небалансираних група. Максимална разноврсност у групама се постиже максимизовањем суме разлика у вредностима свих карактеристика између сваког пара студента у оквиру сваке од група.

На основу наведеног, може се закључити да су неки истраживачи усмерили своју пажњу на максимизовање различитости унутар групе [104, 190, 194], док су се други фокусирали на минимизовање разлике између група [11, 105, 132]. Међутим, у [11] се наглашава да су ова два приступа заправо „суштински еквивалентна".

4.2.2 Критеријум балансираности међу групама

Бројне су студије које су свој фокус усмериле на распоређивање индивидуа у групе које су међусобно добро балансиране (well-balanced) у односу на одређене карактеристике индивидуа. Критеријум балансиране расподеле, односно успостављања баланса између група подразумева креирање сличних група које ће чинити индивидуе које нису сличне. Суштина овог критеријума се заснива на захтеву да се осигура успешно извршавање задатака од стране свих формираних група [6, 104]. Формирање балансираних група од ученика са добрим, умереним и скромним знањем из програмирања приказано је у [6]. Слично, у [204] вештина програмирања је узета као карактеристика на основу које се формирају балансиране групе. Разматрање шест карактеристика за успостављање баланса међу групама (величина групе, заступљеност полова, академске способности, итд) и сходно њима, подела сваког од разреда у четири групе, уз минималну гаранцију присуства жељеног пријатеља у истој групи, приказано је у [123]. С друге стране, гаранција максималног броја испуњених пријатељских захтева (енгл. friendship requests), балансиране групе сходно карактеристикама студената (полу, потребама за подршком у учењу, нивоима академских способности, „енергетском" понашању) и међусобно једнаке величине група, приказани су у [76]. Истраживачи у [12, 144, 156, 196] користе Belbin-ов модел тимских улога као средство за формирање добро балансираних група за колаборативно учење. Тако формиране групе резултирале су успехом у учењу, унапређењу друштвених односа међу студентима истог тима, увођењу динамике и ефикасности у оквиру група и унапређење перформанси група. Belbin у [16] наглашава да балансиране групе у односу на свих девет тимских улога боље функционишу у односу на небалансиране групе. С тим у вези, Belbin у [16, 17] успоставља услове баланса према којима се за групу може рећи да је балансирана уколико сваку улогу преузима најмање један члан групе. Наиме, савршено балансирана група би била група чији сваки члан има висок или веома висок степен склоности ка једној од девет улога, док би савршено небалансирана група била група чији чланови преузимају исту улогу [144]. У [182, 190] истиче се значај балансирања група у односу на стилове мишљења. Разноврсност Myers-Briggs психолошких стилова у оквиру тима и успостављање баланса између тимова, гарантује да ће сви аспекти пројекта бити успешно решени [153].

4.2.3 Постојећи приступи у решавању проблема добро балансиране расподеле у групе

Генерално, циљ балансиране расподеле (партиционисања) је супротан од стандардног проблема партиционисања, односно класификације субјеката или објеката. У [36] класификација се дефинише као процес у оквиру којег је потребно субјекте или објекте распоредити у групе тако да чланови исте групе буду међусобно слични, а чланови различитих група различити. С друге стране, балансирана расподела подразумева проблем креирања међусобно сличних група које садрже различите чланове. У литератури се могу наћи два основна приступа за решавање проблема добро балансиране расподеле у групе (well-balanced assignment problem), а то су: приступ заснован на ограничењима (енг. the constraint-based approach) и објектни приступ заснован на девијацијама (енгл. the deviation-based objective approach) [67, 68].

Приступ заснован на ограничењима (енг. the constraint-based approach) подразумева одређивање горњих и доњих граница у оквиру ограничења броја студената који испуњавају одређене критеријуме. Уколико није могуће задовољити све услове балансирања, доње и горње границе се проширују ван идеалних мера. Овакав приступ има и своје недостатке. Као прво, потребно је континуирано решавање проблема све док се не достигне изводљиво решење, односно изводљиво распоређивање ученика у групе и као друго, овај приступ може резултирати прихватљивом распоређивању које задовољава услове балансирања, али које не мора уједно да буде и најбоља подела у погледу баланса и квалитета. Аутори у [114, 115, 165] користе наведени приступ за распоређивање студената у добро балансиране пројектне тимове.

Објектни приступ заснован на девијацијама (енгл. the deviation-based objective approach) подразумева минимизирање суме одступања од одговарајућих циљева балансирања. Овакав приступ примењен је у [11] како би максимизовали различитост чланова у оквиру група и минимизирали разлике између група. Предлаже се девет различитих функција циља за мерење сличности између група и тако се добијају нелинеарни оптимизациони модели, док се проблем решава применом оптимизационих метода. Добар пример примене овог приступа у распоређивању студената у балансиране репрезентативне тимове описан је у [49], где аутори користе тежиште (центроид) како би представили сваку од група ентитета, при чему предлажу два различита начина мерења баланса између група, од минималног ка укупном (енгл. min-sum) и од минималног до максималном (енгл. min-max). Модел који је дат са min-sum функцијом циља, минимизује суму пондерисаних раздаљина између тимског тежишта и циљаног вектора карактеристика, док модел са min-max функцијом циља, минимизује максимално пондерисану раздаљину између тимског тежишта и циљаног вектора карактеристика. Аутори примењују свој метод на поделу 120 студената у групе од 5. За решавање ових проблема аутори развијају: егзактну методу засновану на методи дељења скупа (енгл. Set partitioning method), хеуристику засновану на формулацији дељења скупа и две метахеуристичке методе променљивих околина. У [10], аутори представљају студију случаја која се односи на распоређивања 235 студената у осам туторских група, у оквиру које примењују формулацију мешовитог линеарног програмирања са min-sum функцијом циља. Аутори у [67] такође примењују објектни приступ заснован на девијацијама за распоређивање студената на жељене семинаре, успостављајући баланс у погледу пола и броја интернационалних студената у оквиру понуђених семинара, увођењем нове мулти-објективне конвексно квадратне целобројне формулације проблема. За разлику од предходника, не развијају се посебне хеуристике за решавање посматраног проблема, већ се предложени модел директно имплементира у стандардни решавач (енгл. off-the-shelf solver). У [204] предлаже се математички модел који омогућава формирање оптималних група за учење и који задовољава захтеве за груписањем у односу на различите образовне контексте. Баланс се успоставља тако што се за све групе минимизује сума квадрата разлика средњих вредности вештине програмирања за студенте посматране групе и средњих вредности вештине програмирања за све студенте. За решавање представљеног модела предлаже се унапређени генетски алгоритам.

4.2.4 Мотивација за истраживање

Мотивација за истраживачки рад настала је из практичне потребе да се студенти, са исте студијске групе Београдске пословне школе, поделе у одређени број студентских група за наставу, тако да групе буду добро балансиране, односно да свака група има што уједначенији број студената са истом карактеристиком. Додатни мотивациони фактори који су утицали на формулисање и решавање BMWASP као оптимизационог проблема, изведени су из предности и недостатака сродних модела, који се у литератури (одељак 4.2.3) могу наћи као различите варијанте проблема добро балансиране расподеле у групе, а у оквиру којих се разматрају различите захтеви, ограничења и функције циља. Један од таквих модела описан је у [114] где се баланс спроводи кроз јака ограничења (енгл. hard constraints) и проблем BMASP посматра пре у контексту изводљивости (савршено балансиране партиције) него као проблем оптимизације. Аутори анализирају репрезентативну практичну примену методе на примеру формирања група студената у Ротмановој школи за менаџмент, на Универзитету у Торонту. Циљеви анализе у [114] подразумевају одређивање класа проблема балансирања које могу да садрже примере неизводљивости и колико су преовлађујући овакви примери унутар ових класа. Позитивни ефекти принципа разноврсности унутар група и сличности међу групама, уз додатно појачање захтевом за добром балансираношћу међу групама, сублимирани су у својству критеријума за распоређивање индивидуа у BMWASP. Проблем BMWASP се разликује од других сличних проблема у литератури по томе што су подаци бинарни и мултидимензионални са коефицијентима тежине. Насупрот томе, аутори у [14] користе или једнодимензионалне карактеристике или сложене скорове у покушају да интегришу карактеристике који нису лако међусобно мерљиве. Аутори у [146] развијају хеуристички алгоритам и упоређују га са формулацијом мешовитог линеарног програмирања за балансирано партиционисање укључујући само бинарно вредновање карактеристика.

Према томе, BMWASP се може сагледати као нови методички приступ за формирање k добро балансираних група уз могућност разматрања великог броја индивидуа, њихових карактеристика (у конкретном случају преко 100) и листе приоритета међу карактеристикама.

4.3 Предложени приступ у решавању проблема BMWASP

У овом одељку описан је предложени приступ за решавање BMWASP који има за циљ дефинисање одговарајућег критеријума који ће омогућити поређење степена девијације од „савршеног баланса" за различите поделе, при чему се као резултат добија подела које ће минимизовати тај критеријум. Прво се предлаже математичка формулација проблема BMWASP да би се минимизовала укупна удаљеност од идеалног броја сваке карактеристике у свакој групи, укључујући и тежину карактеристика. Наредни корак је представљање мешовите целобројно линеарне реформулације проблема BMWASP и процењивање њене ефикасности кроз скуп рачунарских тестирања. Трећи корак је представљање једноставне и брзе методе за рачунање доње границе коришћењем линеарне релаксације предложене MILP реформулације проблема. У циљу решавања тест примера проблема BMWASP већих димензија на што ефикаснији начин, развијена је и примењена метахеуристичка метода заснована на методи променљивих околина. Резултати представљени у овом одељку, приказани су у [53, 54].

4.3.1 Математичка формулација проблема BMWASP

У овом одељку је представљена математичка формулација BMWASP проблема комбинаторне оптимизације.

За решавања BMWASP, потребно је n индивидуа распоредити у k међусобно дисјунктних група, тако да свака група садржи што уједначенији број индивидуа са одређеном карактеристиком. Једна индивидуа s_i $(i = 1, 2, \dots, n)$ може имати једну или више карактеристика c_j $(j = 1, 2, \dots, m)$. Свака карактеристика c_j има одговарајућу тежину w_j. Повећањем тежине неке од карактеристика повећава се и приоритет захтева да број индивидуа са том карактеристиком буде што уједначенији по групама. Према томе, карактеристике веће тежине имају већи степен приоритета у односу на карактеристике мање тежине. Уколико није другачије наведено, све карактеристике имају исти степен важности и вредност тежине сваке од њих једнака је 1.

Нелинеарна математичка формулација проблема BMWASP

Нека су параметри и променљиве проблема BMWASP дефинисани на следећи начин:

$S = \{s_1, s_2, s_3, \dots, s_n\}$ - скуп n индивидуа; (4.1)

$G = \{g_1, g_2, g_3, \dots, g_k\}$ - скуп k група; (4.2)

$C = \{c_1, c_2, c_3, \dots, c_m\}$ - скуп m карактеристика; (4.3)

w_j - тежине карактеристика c_j, за $j = 1, 2, \dots, m$;

$A = [a_{ij}]$ - матрица карактеристика индивидуа, где је (4.4)

$$a_{ij} = \begin{cases} 1, & \text{ако индивидуа } s_i \text{ има карактеристику } c_j \\ 0, & \text{иначе} \end{cases}; \quad (4.5)$$

$i = 1, 2, \dots, n,\quad j = 1, 2, \dots, m;$

$c_j^{avg} = \frac{1}{k}\sum_{i=1}^{n} a_{ij}$ - идеалан број карактеристике $j = 1, 2, \dots, m$ у свакој групи. (4.6)

Математичка формулација проблема BMWASP користи следеће променљиве:

$$x_{il} = \begin{cases} 1, & \text{ако је индивидуа } s_i \text{ распоређена у групу } g_l \\ 0, & \text{иначе} \end{cases}; \quad (4.7)$$

$i = 1, 2, \dots, n,\quad l = 1, 2, \dots, k.$

Користећи се предложеном нотацијом, проблем BMWASP се може представити формулацијом нелинеарног програмирања (енгл. Nonlinear Programming - NLP):

$$min \ \sum_{j=1}^{m} \sum_{l=1}^{k} w_j \left| y_{jl} - c_j^{avg} \right| \tag{4.8}$$

при ограничењима:

$$y_{jl} = \sum_{i=1}^{n} a_{ij} x_{il}, \quad j = 1,2,\dots,m, \ l = 1,2,\dots,k; \tag{4.9}$$

$$\sum_{l=1}^{k} x_{il} = 1, \qquad i = 1,2,\dots,n; \tag{4.10}$$

$$\sum_{i=1}^{n} x_{il} \leq \left\lceil \frac{n}{k} \right\rceil, \qquad l = 1,2,\dots,k; \tag{4.11}$$

$$\sum_{i=1}^{n} x_{il} \geq \left\lfloor \frac{n}{k} \right\rfloor, \qquad l = 1,2,\dots,k; \tag{4.12}$$

$$x_{il} \in \{0,1\}, \qquad i = 1,2,\dots,n, \ l = 1,2,\dots,k. \tag{4.13}$$

Нека је са BMWASP-O означен NLP модел (4.8)-(4.13). Функција циља (4.8) је дефинисана као сума укупне удаљености сваке од карактеристика од њеног идеалног броја у свакој групи, укључујући и тежине карактеристика. Циљ проблема BMWASP је минимизација вредности функције циља. Ограничењем (4.9) утврђује се укупан број y_{jl} сваке карактеристике у свакој од група. Ограничење (4.10) обезбеђују распоређеност једне индивидуе у тачно једну групу, док се ограничењима (4.11) и (4.12) дефинишу редом максималне, односно минималне величине група. Наведени модел дозвољава да се групе разликују у величини за један уколико n није дељиво са k. У случају да је n дељиво са k, ограничења (4.11) и (4.12) се могу заменити са једним ограничењем:

$$\sum_{i=1}^{n} x_{il} = \frac{n}{k}, \qquad l = 1,2,\dots,k \tag{4.14}$$

Бинарна природа променљиве x_{il} дефинисана је условом (4.13).

Представљена нелинеарна математичка формулација (4.8)-(4.13) проблема BMWASP има укупно $k(n+m)$ променљивих и $mk + n + 2k$ ограничења.

MILP реформулација проблема BMWASP

Изложена формулација BMWASP-O проблема BMWASP је нелинеарна јер у функцији циља фигурише апсолутна вредност. Како апсолутна вредност није непрекидно диференцијабилна функција, то је над њом релативно тешко и захтевно извршавати процедуре стандардне оптимизације. Једноставном модификацијом израза апсолутне вредности [142], ове потешкоће се могу избећи па се и посматрани проблем може реформулисати као проблем мешовитог целобројног линеарног програмирања (MILP). На основу наведеног, у овом одељку је представљена реформулација проблема BMWASP, у ознаци BMWASP-I, која ће омогућити ефикасније решавање тест примера проблема већих димензија и добијање њихових оптималних решења.

Нека је уведена ознака $d_{jl} = y_{jl} - c_j^{avg}$ $(j = 1, 2, \ldots, m,\ l = 1, 2, \ldots, k)$. За свако d_{jl} дефинишу се две нове ненегативне променљиве d_{jl}^+ и d_{jl}^- . Разликом $d_{jl}^+ - d_{jl}^-$ замењује се свако d_{jl} у ограничењима и функцији циља, а додају се и знаци рестрикције $d_{jl}^+ \geq 0$ и $d_{jl}^- \geq 0$. Израз апсолутне вредности $|d_{jl}| = |d_{jl}^+ - d_{jl}^-|$ поједностављује се кад год је $d_{jl}^+ = 0$ или $d_{jl}^- = 0$. Алгебарски, ако је производ променљивих једнак нули $d_{jl}^+ \cdot d_{jl}^- = 0$ онда се наведени израз апсолутне вредности записује као збир две променљиве: $|d_{jl}^+ - d_{jl}^-| = |d_{jl}^+| + |d_{jl}^-| = d_{jl}^+ + d_{jl}^-$.

Увођењем ограничења да је једна од променљивих (d_{jl}^+ или d_{jl}^-) једнака нули , може се представити следећа мешовита целобројно линеарна реформулација проблема BMWASP:

$$min\ \sum_{j=1}^{m} \sum_{l=1}^{k} (w_j\, d_{jl}^+ + w_j\, d_{jl}^-) \tag{4.15}$$

при ограничењима:

$$y_{jl} = \sum_{i=1}^{n} a_{ij} x_{il}\,, \quad j = 1, 2, \ldots, m,\ l = 1, 2, \ldots, k; \tag{4.16}$$

$$\sum_{l=1}^{k} x_{il} = 1, \quad i = 1, 2, \ldots, n; \tag{4.17}$$

$$\sum_{i=1}^{n} x_{il} \leq \left\lceil \frac{n}{k} \right\rceil, \quad l = 1, 2, \ldots, k; \tag{4.18}$$

$$\sum_{i=1}^{n} x_{il} \geq \left\lfloor \frac{n}{k} \right\rfloor, \quad l = 1, 2, \ldots, k; \tag{4.19}$$

$$d_{jl}^+ - d_{jl}^- = y_{jl} - c_j^{avg}, \quad j = 1, 2, \ldots, m,\ l = 1, 2, \ldots, k; \tag{4.20}$$

$$x_{il} \in \{0, 1\}\,, \quad i = 1, 2, \ldots, n\,,\ l = 1, 2, \ldots, k; \tag{4.21}$$

$$d_{jl}^+ \geq 0,\ d_{jl}^- \geq 0, \quad j = 1, 2, \ldots, m,\ l = 1, 2, \ldots, k. \tag{4.22}$$

Ограничење $d_{jl}^{+} \cdot d_{jl}^{-} = 0 \;\; (j = 1, 2, \ldots, m \;\; \wedge \;\; l = 1, 2, \ldots, k)$ није неопходно и може се елиминисати. Нека је дат једноставан пример:

$$min\,(d^{+} + d^{-})$$

при ограничењима:

$$d^{+} + d^{-} = C;$$

$$d^{+} \geq 0 \text{ и } d^{-} \geq 0.$$

Ако је $C \geq 0$ онда је $min(d^{+} + d^{-}) = \; C$, где је $(d^{+},\; d^{-}) = (C, 0)$. Иначе, ако је $C < 0$ онда је $min(d^{+} + d^{-}) = -\,C$, при чему је $(d^{+},\; d^{-}) = (0, -C)$. У оба случаја минимизација ће аутоматски узроковати да је $d^{+} \cdot d^{-} = 0$, па ово нелинеарно ограничење неће бити неопходно.

Неке од одлучујућих променљивих реформулације BMWASP-I су цели бројеви, док је осталим променљивим дозвољено да не буду цели бројеви. Према томе, предложени BMWASP-I модел (4.15-4.22) представља мешовито целобројно линеарну реформулацију проблема BMWASP. Предложена реформулација има $k(3m + n)$ променљивих и $2mk + n + 2k + nk$ ограничења.

Линеарна релаксација проблема BMWASP

Доња граница (енгл. Lower Bound - LB) се дефинише као вредност која је гарантовано мања или једнака оптималном решењу. У случају да решење добијено неким алгоритмом достигне доњу границу, вероватно је реч о оптималном решењу. За брзо и једноставно рачунање доње границе најчешће се користи линеарна релаксација проблема математичког програмирања (скраћено LP релаксација проблема од енгл. linear programming relaxation) при чему се као доња граница узима вредност решења LP релаксације проблема.

У овом одељку представљена је LP релаксација MILP реформулације проблема BMWASP. LP релаксација проблема формулише се раскидањем веза између индивидуа и карактеристика. У том случају расподела индивидуа у групе постаје ирелевантна, па се проблем BMWASP може једноставно решити као проблем распоређивања расположивих карактеристика у групе. Рачунањем вредности решења за „идеалну“ (добро) балансирану расподелу карактеристика по групама, добија се вредност доње границе.

Нека је са $I = (n, m, k, w_j, A)$ уведена ознака за тест примере BMWASP проблема са k група и n индивидуа, где је свака индивидуа окарактерисана вектором од m карактеристика кроз матрицу карактеристика A, задатом са (4.4)-(4.5). Тада се вредност LB може израчунати по формули:

$$LB = \sum_{j=1}^{m} w_j \left(\theta_j \cdot \lambda_j + \left(k - \theta_j\right) \cdot \left(1 - \lambda_j\right)\right) \tag{4.23}$$

где је:

$$\theta_j = \left(\sum_{i=1}^{n} a_{ij}\right) mod\ k, \qquad j = 1, 2, \ldots, m; \tag{4.24}$$

$$\Delta_j = \sum_{i=1}^{n} \frac{a_{ij}}{k} - \left\lfloor \sum_{i=1}^{n} \frac{a_{ij}}{k} \right\rfloor, \qquad j = 1, 2, \ldots, m; \tag{4.25}$$

$$\lambda_j = \begin{cases} max\left(\Delta_j, 1 - \Delta_j\right), \Delta_j \leq \frac{1}{2} \\ min\left(\Delta_j, 1 - \Delta_j\right), \Delta_j \geq \frac{1}{2} \end{cases}, \quad j = 1, 2, \ldots, m. \tag{4.26}$$

Предложени математички модел (4.23)-(4.26) представља LP релаксацију проблема BMWASP. Уколико постоји „идеално“ балансирана расподела карактеристика у групе, одговарајућа вредност функције циља (4.23) једнака је доњој граници решења проблема BMWASP.

Карактеристике LP релаксације проблема BMWASP могу се приказати на једном кратком илустративном примеру.

Пример 4.1. Потребно је 5 студената распоредити у 3 групе тако да групе буду међусобно добро балансиране у односу на 4 карактеристике студената. Веза између студената и карактеристика, као и параметри LP релаксације проблема приказани су у Табели 4.1. У првој колони табеле налазе се ознаке за студенте и параметре релаксације. У наредне четири колоне приказани су подаци који се односе на карактеристике студената у ознакама {A1, A2, A3, A4}, док последња колона даје приказ укупног броја група. Веза између студента и карактеристике приказана је бинарним вредностима (1 ако студент има одређену карактеристику и 0 ако је нема).

Као што се може видети из Табеле 4.1, у „идеалној“, односно добро балансираној расподели, у свакој од група ће се наћи по једна од карактеристика

{A2, A3, A4}, док ће карактеристике A1 бити у две група по 2, а у једној од група само једна A1 карактеристика. Идеалан број карактеристике A1 у све три групе био би 1.666667, што значи да ће одступање њеног реалног броја од идеалне вредности бити или |2-1.666667|=0.333333 или |1-1.666667|=0.666667. Вредност промењљиве θ za карактеристику A1 iznosi 2. Ова промењљива показује колико пута ће одступање бити 0.333333 (што је одговарајућа вредност промењљиве λ), док ће у $(k-\theta)$ пута одступање имати вредност (1-0.333333), односно 0.666667. Ова „идеална" расподела некад може да се направи, а некад не. За постигнуту „идеалну" расподелу, LB је вредност њене функције циља.

Табела 4.1: Карактеристике LP релаксације проблема BMWASP

	A1	A2	A3	A4	Broj grupa
Студент 1	1	0	1	1	k =3
Студент 2	1	1	1	0	
Студент 3	1	0	0	1	
Студент 4	1	1	1	0	
Студент 5	1	1	0	1	
Сума	5	3	3	3	
Идеална вредност (Сума/k)	1.666667	1	1	1	
θ (Сума mod k)	2	0	0	0	
λ	0.333333	0	0	0	

У оригиналном проблему, студент има више карактеристика, тако да би се у случају размештаја на пр. Студента 1 у прву групу због карактеристике A4, аутоматски у истој групи нашле и карактеристике A1 и A3, односно оне би биле индуковане самом расподелом. У LP релаксацији се уклања веза студент-карактеристика, па се врши обична расподела карактеристика у групе.

4.3.2 Метода промењљивих околина за решавање BMWASP

Успешна примена VNS методе за решавање проблема из бројних области послужила је као подстицај за имплементацију нове варијанте ове методе у циљу решавања BMWASP проблема. Предност предложеног VNS алгоритма огледа се у имплементацији ефикасне и брзе локалне претраге засноване на стратегији замене индивидуа (енгл. swap-based local search). У даљем тексту следи детаљније објашњење свих аспеката предложене VNS методе, која је заједно са експерименталним резултатима описана у [53, 54, 139].

Простор решења проблема BMWASP

Простор претраге решења BMWASP проблема обухвата све могуће поделе скупа од n индивидуа у k група, при којима свака од формираних група g_l садржати најмање $\left\lfloor \frac{n}{k} \right\rfloor$ а највише $\left\lceil \frac{n}{k} \right\rceil$ индивидуа.

Нека је са $F_{n,n_1,n_2,\dots,n_k}$ означен број начина на који се n индивидуа може распоредити у k група, при чему група g_l садржи n_l индивидуа ($n = \sum_{l=1}^{k} n_l$). Секвенцијалном процедуром из [184] одређује се број $F_{n,n_1,n_2,\dots,n_k}$:

1. Нека се прво n_1 индивидуа распореди у g_1 групу. Укупан број свих могућих начина да се од n индивидуа изабере n_1, једнак је броју свих комбинација од n елемената класе n_1. Према томе, број могућих начина да се формира група g_1 једнак је:

$$\binom{n}{n_1} = \frac{n!}{n_1!(n-n_1)!} \ . \tag{4.27}$$

2. У наредном кораку врши се распоређивање n_2 индивидуа у другу групу g_2. Од укупно n индивидуа, n_1 је већ распоређено у прву групу g_1, тако да остаје $n - n_1$ индивидуа које треба распоредити у другу групу. Број могућих начина да се од преосталих $n - n_1$ индивидуа изабере тачно n_2, једнак је броју комбинација од $n - n_1$ елемената класе n_2. Према томе, број могућих начина за формирање група g_2 је:

$$\binom{n-n_1}{n_2} = \frac{(n-n_1)!}{n_2!(n-n_1-n_2)!} \ . \tag{4.28}$$

На основу претходног, укупан број могућих начина за формирање прве две групе g_1, g_2 је:

$$\binom{n}{n_1} \cdot \binom{n-n_1}{n_2} = \frac{n!}{n_1!(n-n_1)!} \cdot \frac{(n-n_1)!}{n_2!(n-n_1-n_2)!} = \frac{n!}{n_1!n_2!(n-n_1-n_2)!} \ . \tag{4.29}$$

3. У корацима који следе распоређују се остале индивидуе у преостали број група, све док не остане $n_к$ индивидуа и последња група g_k. Постоји само један начин да се формира последња група, што се може записати као:
$$\binom{n-n_1-n_2-\cdots-n_{k-1}}{n_k} = \frac{(n-n_1-n_2-\cdots-n_{k-1})!}{n_k!(n-n_1-n_2-n_2-\cdots-n_k)!}\,. \tag{4.30}$$
Према томе, множењем свих биномних коефицијената добијамо:

$$\begin{aligned}
&\binom{n}{n_1}\cdot\binom{n-n_1}{n_2}\cdot\ \ldots\ \cdot\binom{n-n_1-n_2-\cdots-n_{k-1}}{n_k}\\
&= \frac{n!}{n_1!\,n_2!\cdot\ldots\cdot\ n_{k-1}!\,(n-n_1-n_2-\cdots-n_{k-1})!}\cdot\frac{(n-n_1-n_2-\cdots-n_{k-1})!}{n_k!\,(n-n_1-n_2-\cdots-n_k)!}\\
&= \frac{n!}{n_1!\,n_2!\cdot\ldots\cdot\ n_k!\,(n-n_1-n_2-\cdots-n_k)!}\\
&= \frac{n!}{n_1!\,n_2!\cdot\ldots\cdot\ n_k!\,0!}\\
&= \frac{n!}{n_1!n_2!\cdot\ldots\cdot n_k!}.
\end{aligned} \tag{4.31}$$

У BMWASP моделу свака група садржи најмање $\left\lfloor\frac{n}{k}\right\rfloor$ а највише $\left\lceil\frac{n}{k}\right\rceil$ индивидуа. (Ако неке од група садрже исти број индивидуа, у том случају обележавање група неће бити од значаја. Тада се група 1 може обележити као група 2, а група 2 као група 3 и тако даље.) Нека се са $\hat{k}$ означи број група које садрже $\left\lceil\frac{n}{k}\right\rceil$ индивидуа, тада $k-\hat{k}$ група садржи $\left\lfloor\frac{n}{k}\right\rfloor$ индивидуа. Ако је n дељиво са k онда важи $\hat{k}=k$, иначе $\hat{k}=n\,mod\,k$. Дакле, постоји укупно $\hat{k}!\cdot(\,k-\hat{k}\,)!$ начина да се обележе групе. Према томе, коначан број начина на који се n индивидуа може распоредити у k група, при чему важи:
$$n=\left(\left\lceil\frac{n}{k}\right\rceil-\left\lfloor\frac{n}{k}\right\rfloor\right)\cdot\left(\hat{k}\cdot\left\lceil\frac{n}{k}\right\rceil+\left(k-\hat{k}\right)\cdot\left\lfloor\frac{n}{k}\right\rfloor\right)+\left(1-\left\lceil\frac{n}{k}\right\rceil+\left\lfloor\frac{n}{k}\right\rfloor\right)\left(\hat{k}\cdot\frac{n}{k}\right), \tag{4.32}$$

једнак је:

$$F_{n,n_1,n_2,\ldots,n_k}=C_{n,k}=\frac{n!}{\left(\left\lceil\frac{n}{k}\right\rceil!\right)^{\hat{k}}\cdot\left(\left\lfloor\frac{n}{k}\right\rfloor!\right)^{(k-\hat{k})}\cdot\hat{k}!\cdot(\,k-\hat{k}\,)!}\,. \tag{4.33}$$

На пример, укупан број начина да се 20 индивидуа распореди у 4 групе од по 5 индивидуа је $C_{20,4}$ = 488 864 376.

Основне карактеристике методе промењљивих околина за BMWASP

У циљу решавања проблема BMWASP, предложени VNS алгоритам користи шест улазних параметара: n - број индивидуа, m - број могућих карактеристика за сваку индивидуу, k - број група за поделу, p_{max} - максимална величина околине, w_j - тежина карактеристике и $[a_{ij}]$ - матрица која повезује карактеристике са индивидуама. Основна структура предложеног VNS алгоритма за решавања проблема BMWASP представљена је Алгоритмом 9.

На почетку се дефинишу структуре околина. Једна замена околине (енгл. swap neighborhood) подразумева замену између једног пара индивидуа који припадају различитим групама. Према томе, околина је дефинисана на начин да се решење x_{temp} налази у p–тој околини решења $x_{current}$, у ознаци $x_{temp} \in N_p(x_{current})$, ако се подела x_{temp} може добити из поделе $x_{current}$ са највише p замена (по једног пара) индивидуа између различитих група.

Иницијално решење $x_{initial}$ се добија у другом кораку алгоритма на случајан начин. Затим се на њега примењује RVNS алгоритам (видети одељак 1.3.3 и Алгоритам 4) како би се на брз и ефикасан начин добило добро иницијално решење $x_{current}$ (корак 3) за основни VNS алгоритам. Критеријум заустављања RVNS алгоритма је достизање максималног броја понављања без побољшања тренутног решења. Максималан број понављања (итерација) је 30000. Иницијална вредност промењљиве p (величине околине) дефинисана је кораком 4. У наредном кораку дефинише се критеријум заустављања VNS алгоритма. Централни део VNS алгоритма се састоји из петље која се извршава у оквиру корака 6-18. Петља се понавља све док се критеријум заустављања не испуни, што подразумева 30 понављања без побољшања решења. У кораку 7, спроводи се фаза размрдавања на решењу $x_{current}$. У оквиру фазе размрдавања тренутно најбоље решење $x_{current}$ насумично се помера ка новом решењу у његовој p-околини (која садржи решења добијена са p замена једног пара индивидуа који припадају различитим групама).

Након размрдавања, у реду 8 над добијеним решењем x_{temp} примењује се унапређена локална претрага. Затим се у корацима 9-11 извршава функција замене околина којом се пореде тренутно најбоље решење $x_{current}$ и најбоље пронађено решење x_{temp} у његовој p околини. Уколико је x_{temp} боље од $x_{current}$, тренутно најбоље решење се ажурира и параметра p добија вредност 1. У супротном, ако x_{temp} није боље решење од тренутно најбољег, величина околине p се увећава за један, односно претрага се наставља у следећој околини $x_{current}$ решења. Вредност параметра p се поново поставља на 1 у корацима 14-16 уколико она премаши максималну вредност параметра p_{max}. Коначно, када се главна петља заврши, најбоље решење које је пронашао VNS се у кораку 19 именује са x_{best} и враћа као најбољи резултат.

Алгоритам 9: Структура VNS алгоритма за решавање проблема BMWASP

Input: n, m, k, p_{max}, w_j, a_{ij}

Output: solution x_{best}

1. *InitializeStructures();*
2. $x_{initial} \leftarrow$ *RandomSolution();*
3. $x_{current} \leftarrow$ *RVNS(*$x_{initial}$*);*
4. $p \leftarrow 1$;
5. *SelectStopCondition();*
6. ***while*** *StopCondition() is not satisfied* ***do***
7. $x_{temp} \leftarrow$ *Shaking(*$x_{current}, p$*);*
8. $x_{temp} \leftarrow$ *LocalSearch(*x_{temp}*);*
9.   ***if*** x_{temp} *is better than* $x_{current}$ ***then***
10. $x_{current} \leftarrow x_{temp}$;
11. $p \leftarrow 1$;
12.   ***else***
13. $p \leftarrow p + 1$;
14.   ***if*** $p > p_{max}$ ***then***
15. $p \leftarrow 1$;
16.   ***endif***
17.   ***endif***
18. ***endwhile***
19. $x_{best} \leftarrow x_{current}$;

У фази унапређене локалне претраге користе се две стратегије: стратегија брзе замене (енгл. swap-based strategy) по једног пара индивидуа које припадају различитим групама, како би се на брз и ефикасан начин добила побољшања тренутног решења и стратегија најбољег побољшања (енгл. best improvement strategy) којом се од свих побољшања пронађених у околини тренутног решења бира најбоље.

Структура предложене унапређене локалне претраге дата је Алгоритмом 10. У првом кораку алгоритма рачуна се вредност функције циља тренутног решења $x_{current}$. Овај корак није неопходан уколико се вредност функције циља ажурира, на сличан начин као што је описано у даљем тексту. У другом кораку се тренутно решење проглашава најбољим (x_{best}). Иницијална вредност логичке променљиве $improvement$ (побољшање) одређује се у кораку 3. Централни део локалне претраге састоји се из петље која се извршава у редовима 5-36, док год је могуће побољшати тренутно најбоље решење (заменом пара индивидуа). За сваки појединачни пар индивидуа који припадају различитим групама, у корацима 10-13 добија се ново решење x_{temp}, тако што се прво изврши замена индивидуа у групама а затим ажурира вредност функције циља.

Нека се у оквиру тренутног решења $x_{current}$ елемент s_{i_1} налази у групи g_{l_1}, а елемент s_{i_2} у групи g_{l_2}. Решење које се добија након премештања елемента s_{i_1} у групу g_{l_2} и елемента s_{i_2} у групу g_{l_1}, обележава се са x_{temp}. Пошто се елемент s_{i_1} измести из групе g_{l_1} и убаци у групу g_{l_2}, нови број карактеристика у групама g_{l_1} и g_{l_2} утиче на вредност функције циља новог решења. Међутим, како остале групе g_l (l =1, 2,...,$k \wedge l \neq l_1, l \neq l_2$) остају непромењене, број карактеристика у овим групама нема утицаја на вредност функције циља новог решења. Да би се искористила поменута чињеница и убрзала локална претрага, формира се помоћна матрица Y= (y_{jl}) у оквиру које елемент y_{jl} представља број индивидуа у групи g_l које имају карактеристику c_j. Такође, иницијализује се низ c_j^{avg} (на пример, после уноса података) којим се за сваку карактеристику c_j дефинише њен идеалан број у свакој од група. Користећи претходно наведено, у једној итерацији алгоритма проласком кроз све карактеристике у корацима 15-27 може се израчунати вредност функције циља новог суседног решења.

Алгоритам 10: Унапређена локална претрага за проблем BMWASP

Input: solution $x_{current}$

Output: solution x_{best}

$f_{current} \leftarrow$ *CalculateObjectiveFunction(*$x_{current}$ *);*

$x_{best} \leftarrow x_{current}$;

$f_{best} \leftarrow f_{current}$;

improvement $\leftarrow$ *true;*

while *improvement* ***do***

improvement $\leftarrow$ *false;*

for $l_1 = 1$ *to* k ***do***

foreach *student* s_{i_1} *in group* g_{l_1} ***do***

for $l_2 = l_1 + 1$ ***to*** k ***do***

foreach *student* s_{i_2} *in group* g_{l_2} ***do***

$x_{temp} \leftarrow x_{current}$;

$f_{temp} \leftarrow f_{current}$;

$x_{temp}^{g_{l_1}} \leftarrow (x_{temp}^{g_{l_1}} \setminus \{s_{i_1}\}) \cup \{s_{i_2}\}$;

$x_{temp}^{g_{l_2}} \leftarrow (x_{temp}^{g_{l_2}} \setminus \{s_{i_2}\}) \cup \{s_{i_1}\}$;

foreach *attribute* c_j ***do***

if $a_{i_1 j} \neq a_{i_2 j}$ ***then***

$f_{temp} \leftarrow f_{temp} - w_j \cdot \left| y_{jl_1} - c_j^{avg} \right| - w_j \cdot \left| y_{jl_2} - c_j^{avg} \right|$;

if $(a_{i_1 j} = 1)$ *and* $(a_{i_2 j} = 0)$ ***then***

$y_{jl_1} \leftarrow y_{jl_1} - 1$;

$y_{jl_2} \leftarrow y_{jl_2} + 1$;

else

$y_{jl_1} \leftarrow y_{jl_1} + 1$;

$y_{jl_2} \leftarrow y_{jl_2} - 1$;

endif

$f_{temp} \leftarrow f_{temp} + w_j \cdot \left| y_{jl_1} - c_j^{avg} \right| + w_j \cdot \left| y_{jl_2} - c_j^{avg} \right|$;

endif

endforeach

if $f_{temp} < f_{best}$ ***then***

$x_{best} \leftarrow x_{temp}$;

$f_{best} \leftarrow f_{temp}$;

improvement $\leftarrow$ *true;*

endif

endforeach

endfor

endforeach

endfor

endwhile

Условом задатим у линији 16 прескачу се карактеристике који не утичу на вредност функције циља (карактеристике које обе индивидуе или имају или немају, односно за које важи $a_{i_1 j} \neq a_{i_2 j}$). Пре него што се ажурира број карактеристике c_j у групама које су укључене у замену индивидуа, у кораку 17 вредност функције циља се умањује за тренутни допринос посматране карактеристике c_j. Након што се помоћна матрица $Y = (y_{jl})$ ажурира у оквиру корака 18-24, вредност функције циља се у кораку 25 увећава за нов допринос карактеристике c_j у тим групама. На овај начин се вредност функције циља приликом локалне претраге не израчунава од почетка, већ се ажурира у само једном проласку кроз све карактеристике. У корацима 28-31, најбоље пронађено решење се ажурира ако је x_{temp} боље од тренутно најбољег x_{best} и вредност променљиве $improvement$ (побољшање) се подешава на вредност $true$. Очигледно је да се промена вредности функције циља за свако ново решење које је добијено заменом околине извршава са сложеношћу $O(m)$.

4.4 Експериментални резултати

У овом одељку представљени су резултати тестирања математичке формулације BMWASP-O и њене MILP реформулације BMWASP-I проблема BMWASP и резултати потврде ефикасности предложеног VNS алгоритма, кроз скуп рачунарских тестирања. Сва рачунарска тестирања су изведена на Intel Core i5-2400 процесору од 3.10GHz са 4GB RAM меморије под Ubuntu оперативним системом. Експериментални резултати приказани у овом одељку представљени су у [53, 54].

У циљу тестирања BMWASP-O модела изведени су рачунарски тестови применом CPLEX CP 12.5 егзактног решавача. Овај решавач се користи за решавање модела са великим скупом аритметичко-логичких ограничењима. С друге стране, за тестирање BMWASP-I модела коришћен је егзактни CPLEX MIP 12.5 решавач који се примењује за решавање проблема мешовитог целобројног линеарног програмирања. Решавач проблема програмирања са ограничењима доказује оптималност решења тако што показује да се боље решење од тренутног не може наћи, док решавач проблема мешовитог целобројног линеарног програмирања доказује оптималност решења користећи се доњом границом. За све тест примере дефинисано је временско ограничење од 14400с (4 часа).

Предложени VNS алгоритам је имплементиран на .NET Framework платформи коришћењем програмског језика C#. Као што је раније напоменуто, VNS алгоритам се извршава све док се не достигне услов заустављања од 30 понављања (итерација) без побољшања решења. Вредност параметра p_{max} се поставља на $min\,\{\left\lceil \frac{n}{6} \right\rceil, 20\}$.

Тестирања су прво изведена на тест примерима који су генерисани насумице мењањем броја индивидуа у распону од 10 до 250, обухватајући 12 нивоа вредности карактеристика и 7 нивоа поделе група. Затим је тестирање спроведено на практичном примеру са 229 студената и 116 карактеристика, који је изведен из реалне ситуације у Београдској пословној школи.

Прегледом литературе нису пронађени јавно доступни тест примери за BMWASP, па је за тестирање предложених модела и VNS алгоритма генерисано 300 тест примера проблема са различитим вредностима параметара. Параметри коришћени за генерисање ових тест примера су:

- број индивидуа: 10, 25, 50, 100, 250;
- број карактеристика за сваку индивидуу: 5, 10, 15, 20, 30, 40, 50, 60, 70, 80, 90, 100;
- број група: 2, 5, 10, 15, 20, 25, 30;
- карактеристике за сваку индивидуу: дискретна униформна расподела на интервалу [0, 1];
- тежина сваке карактеристике: дискретна униформна расподела на интервалу [1, 3].

Оваква дистрибуција параметара омогућава генерисање разноврсних тест примера проблема, па су предложени модели и алгоритам тестирани и процењени у различитим условима. Сви тест примери су тестирани уз временско ограничење од 14400s (4 сата). За тест примере уведена је ознака попут, на пример, и50к20г5 где се разматра: 50 индивидуа, 20 карактеристика и 5 група. Генерисани BMWASP тест примери су доступни на адреси [98].

Комплетни резултати тестирања добијени CPLEX решавачима и VNS алгоритмом на тест примерима (и10к5г2 - и25к100г5) мањих димензија приказани су у Табелама 4.2 и 4.3.

Табела 4.2: Поређење резултата CPLEX CP решавача, CPLEX MIP решавача и VNS алгоритма на тест примерима (и10к5г2 – и10к100г5)

Тест примери	CP решавач (BMWASP-O)		MIP решавач (BMWASP-I)		VNS алгоритам			LB
	f	t	f	t	f_{best}	t_{avg}	σ	
и10k5г2	3.00	0.26	3.00	0.08	**3.00**	0.07	0.00	3.00
и10k10г2	11.00	0.56	11.00	0.03	**11.00**	0.14	0.00	11.00
и10k15г2	26.00	0.48	26.00	0.08	**26.00**	0.20	0.00	22.00
и10k20г2	26.00	0.55	26.00	0.09	**26.00**	0.24	0.00	18.00
и10k30г2	48.00	0.57	48.00	0.06	**48.00**	0.36	0.00	20.00
и10k40г2	52.00	0.61	52.00	0.08	**52.00**	0.47	0.00	26.00
и10k50г2	94.00	0.75	94.00	0.11	**94.00**	0.59	0.00	50.00
и10k60г2	104.00	0.94	104.00	0.09	**104.00**	0.69	0.00	72.00
и10k70г2	134.00	0.98	134.00	0.12	**134.00**	0.80	0.00	68.00
и10k80г2	144.00	1.06	144.00	0.13	**144.00**	0.90	0.00	84.00
и10к90г2	158.00	1.16	158.00	0.17	**158.00**	1.01	0.00	88.00
и10k100г2	195.00	1.20	195.00	0.15	**195.00**	1.15	0.00	89.00
Просечно време:		*0.76*		*0.10*		*0.60*		
и10k5г5	13.60	6.45	13.60	0.05	**13.60**	0.08	0.00	13.60
и10k10г5	33.20	48.57	33.20	0.10	**33.20**	0.13	0.00	31.20
и10k15г5	56.40	71.27	56.40	0.16	**56.40**	0.20	0.00	40.80
и10k20г5	59.20	78.42	59.20	0.15	**59.20**	0.26	0.00	37.60
и10k30г5	134.00	180.10	134.00	0.55	**134.00**	0.37	0.00	87.20
и10k40г5	172.00	293.40	172.00	0.77	**172.00**	0.49	0.00	118.40
и10k50г5	210.80	335.90	210.80	0.50	**210.80**	0.59	0.00	144.80
и10k60г5	261.60	493.00	261.60	1.01	**261.60**	0.70	0.00	173.60
и10k70г5	306.40	516.66	306.40	1.70	**306.40**	0.83	0.00	195.20
и10k80г5	340.40	590.85	340.40	0.85	**340.40**	0.93	0.00	258.40
и10k90г5	419.20	811.50	419.20	2.18	**419.20**	1.05	0.00	282.40
и10k100г5	452.40	901.74	452.40	2.39	**452.40**	1.20	0.00	286.40
Просечно време:		*360.65*		*0.87*		*0.60*		

Табеле 4.2 и 4.3. су формиране тако да прва колона садржи информацију о ознаци тест примера. Наредне четири колоне приказују редом вредности функције циља f добијених оптималних решења и време извршења t (изражено у секундама) алгоритама CPLEX CP и MIP решавача. Шеста колона приказује вредност функције циља f_{best} за најбоља решења добијена VNS алгоритмом у 20 независних серија извршења. Седма колона, у ознаци t_{avg}, се односи на просечно укупно време извршења VNS алгоритма, до задовољења критеријума заустављања (достигнути број итерација без побољшања). Осма колона, у ознаци σ, садржи податке о стандардној девијацији у односу на вредност функције циља. Стандардна

девијација се рачуна по формули $\sigma = \sqrt{\frac{\sum_{i=1}^{N}(x_i-\bar{x})^2}{N-1}}$, где су x_i вредности функције циља, $\bar{x}$ њена просечна вредност, тј. $\bar{x} = \frac{\sum_{i=1}^{N} x_i}{N}$, а N број извршавања VNS алгоритма на свакој инстанци. У овој експерименталној анализи N узима вредност N=20. Последња колона у ознаци LB приказује вредност доње границе добијене методом LP релаксације. Оптималне вредности функције циља добијене VNS алгоритмом су означене тамним фонтом.

Табела 4.3: Поређење резултата CPLEX CP решавача, CPLEX MIP решавача и VNS алгоритма на тест примерима (и25к5г2 - и25к100г5)

Тест примери	CP решавач (BMWASP-O)		MIP решавач (BMWASP-I)		VNS алгоритам			LB
	f	t	f	t	f_{best}	t_{avg}	σ	
и25k5г2	1.00	5.60	1.00	0.03	**1.00**	0.11	0.00	1.00
и25k10г2	14.00	180.92	14.00	0.06	**14.00**	0.19	0.00	14.00
и25k15г2	15.00	208.28	15.00	0.04	**15.00**	0.34	0.89	15.00
и25k20г2	27.00	670.43	27.00	0.08	**27.00**	0.41	1.73	25.00
и25k30г2	48.00	2620.94	48.00	0.11	**48.00**	0.64	3.69	34.00
и25k40г2	66.00	3647.57	66.00	0.22	**66.00**	0.85	3.51	34.00
и25k50г2	94.00	5565.21	94.00	0.36	**94.00**	1.01	3.86	48.00
и25k60г2	-	-	130.00	0.52	**130.00**	1.18	4.26	72.00
и25k70г2	-	-	146.00	1.15	**146.00**	1.40	2.89	64.00
и25k80г2	-	-	194.00	1.70	**194.00**	1.57	3.28	72.00
и25k90г2	-	-	209.00	1.52	**209.00**	1.94	8.57	85.00
и25k100г2	-	-	252.00	2.50	**252.00**	1.90	5.71	110.0
Просечно време:		-		*0.70*		*1.00*		
и25k5г5	-	-	20.80	0.16	**20.80**	0.12	0.00	20.80
и25k10г5	-	-	23.20	0.22	**23.20**	0.27	0.93	23.20
и25k15г5	-	-	61.20	14.46	**61.20**	0.42	0.96	59.20
и25k20г5	-	-	88.00	74.08	**88.00**	0.55	1.75	79.20
и25k30г5	-	-	130.00	713.30	**130.00**	0.82	1.51	100.00
и25k40г5	-	-	176.80	2121.98	**176.80**	1.10	2.19	116.80
и25k50г5	-	-	-	-	220.00	1.27	5.38	139.20
и25k60г5	-	-	-	-	345.60	1.61	3.32	187.20
и25k70г5	-	-	-	-	359.60	1.78	5.88	206.40
и25k80г5	-	-	-	-	433.60	1.86	3.43	242.40
и25k90г5	-	-	-	-	528.00	2.36	4.47	284.80
и25k100г5	-	-	-	-	586.80	2.84	6.66	312.80
Просечно време:		-		-		*1.30*		

На основу резултата тестирања 48 тест примера мањих димензија који су приказани у Табели 4.2 и 4.3, може се видети да је CP решавач успео да реши 31 тест пример (и10к5г2 – и25к50г2) док је MIP решавач решио 42 (и10к5г2 – и25к40г5) тест примера. VNS алгоритам је пронашао сва позната оптимална решења (42/42) веома брзо и имао је слично време извршења попут CPLEX решавача на мањим тест примерима, али и значајно краће време на већим тест примерима, са просеком од 1.3s.

Резултати тестирања и поређења алгоритама на преостала 252 тест примера (и25к5г10 - и250к100г25) већих димензија, показали су супериорност BMWASP-I формулације у односу на BMWASP-O формулацију. CPLEX CP решавач није могао оптимално да реши ниједан од наведених тест пример (0/252) у прецизно задатом временском року, док се CPLEX MIP решавач показао подобним за оптимално решавање 57 од 252 тест примера у задатом временском року. У осталим примерима, веома чест проблем на који се наилазило у раду са MIP решавачем била је величина дрвета гранања и одсецања, које би у појединим случајевима постајало толико велико да за њега не би било довољно меморијског простора.

Предложени VNS алгоритам је са значајно краћим просечним временом извршења пронашао 54 од 57 познатих оптималних решења. Додатно, у још три тест примера (и250к15г10, и250к15г15 и и250к15г20), за која нису нађена оптимална решења након коришћења MIP решавача, решења која је понидио VNS алгоритам су достигла LB. Како је вредност функције циља за ова три тест примера једнака вредности LB, реч је о оптималним решењима. За преостале тест примере VNS алгоритам је понудио високо квалитетна решења у кратком временском року. Комплетни резултати тестирања VNS алгоритма на тест примерима већих димензија (и25к5г10 - и250к100г25) приказани су редом у Табелама 4.4, 4.5 и 4.6. Све табеле су структуиране тако да поред колоне која се односи на назив тест примера, имају и колоне које редом приказују: вредност функције циља f_{best} за најбоља решења добијена VNS алгоритмом у 20 независних серија извршења, просечно време t_{avg} извршења VNS алгоритма и стандардну девијацију σ у односу на вредност функције циља. Колона са ознаком LB приказује вредност доње границе добијене методом LP релаксације. Све добијене оптималне вредности функције циља означене су тамним фонтом.

Табела 4.4: Резултати тестирања VNS алгоритма на тест примерима (и25к5г10 - и50к100г25)

Тест примери	f_{best}	t_{avg}	σ	LB	Тест примери	f_{best}	t_{avg}	σ	LB
и25к5г10	**24.40**	0.10	0.00	24.40	и50к5г2	**5.00**	0.20	0.00	5.00
и25к10г10	**76.60**	0.30	0.43	76.60	и50к10г2	**4.00**	0.50	0.00	4.00
и25к15г10	111.00	0.50	1.32	103.80	и50к15г2	**8.00**	0.90	1.01	8.00
и25к20г10	184.20	0.60	1.47	154.60	и50к20г2	**15.00**	1.30	1.75	15.00
и25к30г10	262.00	0.90	1.93	199.20	и50к30г2	34.00	1.80	2.75	26.00
и25к40г10	403.40	1.20	1.93	300.80	и50к40г2	61.00	2.70	4.68	39.00
и25к50г10	521.20	1.30	1.52	371.80	и50к50г2	93.00	3.00	5.45	53.00
и25к60г10	615.20	1.70	2.83	416.00	и50к60г2	133.00	3.70	7.27	77.00
и25к70г10	767.20	2.00	3.15	445.60	и50к70г2	144.00	4.60	7.38	60.00
и25к80г10	744.60	2.30	3.47	466.60	и50к80г2	193.00	5.20	8.78	79.00
и25к90г10	991.20	2.20	1.13	628.60	и50к90г2	233.00	5.80	10.90	99.00
и25к100г10	1073.60	2.80	4.69	684.40	и50к100г2	250.00	5.90	13.02	108.00
и50к5г5	**19.20**	0.30	0.00	19.20	и50к5г10	**31.00**	0.30	0.00	31.00
и50к10г5	**37.60**	0.70	0.00	37.60	и50к10г10	**62.60**	0.80	0.00	62.60
и50к15г5	**44.80**	1.80	1.65	44.80	и50к15г10	**108.00**	2.30	1.66	108.00
и50к20г5	**75.20**	2.30	1.63	75.20	и50к20г10	155.80	2.80	3.11	146.00
и50к30г5	147.20	3.30	3.95	111.20	и50к30г10	267.80	4.20	5.24	196.60
и50к40г5	190.00	4.40	5.13	138.40	и50к40г10	397.40	5.70	8.03	293.20
и50к50г5	260.00	5.80	7.66	153.60	и50к50г10	548.00	7.20	6.32	379.20
и50к60г5	369.20	6.60	7.43	200.80	и50к60г10	622.00	7.70	6.74	393.20
и50к70г5	458.40	7.90	8.43	240.80	и50к70г10	816.60	8.90	8.21	471.40
и50к80г5	503.20	8.60	11.04	252.80	и50к80г10	961.80	11.20	7.30	589.20
и50к90г5	572.80	11.50	12.48	272.80	и50к90г10	1115.20	12.40	9.57	651.40
и50к100г5	644.00	11.50	13.14	321.60	и50к100г10	1222.80	13.00	9.33	704.40
и50к5г15	**72.27**	0.30	0.00	72.27	и50к5г20	**81.90**	0.30	0.00	81.90
и50к10г15	**70.40**	1.30	1.34	70.40	и50к10г20	**141.30**	0.70	0.00	141.30
и50к15г15	175.47	2.50	3.12	169.60	и50к15г20	246.20	2.20	1.24	227.20
и50к20г15	203.33	3.00	3.11	174.67	и50к20г20	294.10	3.00	2.87	262.00
и50к30г15	360.00	4.10	8.03	286.13	и50к30г20	536.40	4.00	3.87	410.60
и50к40г15	654.67	6.00	10.17	485.07	и50к40г20	682.40	6.70	3.74	493.30
и50к50г15	802.67	8.80	7.83	570.67	и50к50г20	914.20	7.60	3.95	594.50
и50к60г15	905.47	8.60	7.32	614.40	и50к60г20	1237.90	9.90	7.96	879.50
и50к70г15	1161.07	9.40	9.61	792.53	и50к70г20	1293.60	8.80	4.80	895.60
и50к80г15	1285.20	11.80	9.91	830.67	и50к80г20	1809.20	9.80	7.93	1152.00
и50к90г15	1540.67	12.40	10.87	978.13	и50к90г20	1878.90	13.50	7.06	1184.90
и50к100г15	1765.73	14.10	7.31	1101.33	и50к100г20	2341.10	13.40	10.66	1499.60
и50к5г25	**57.12**	0.40	0.00	49.44					
и50к10г25	97.36	1.20	0.82	70.24					
и50к15г25	195.76	2.20	1.60	98.56					
и50к20г25	312.00	2.80	1.49	162.24					
и50к30г25	578.24	3.50	3.64	226.88					
и50к40г25	677.12	4.70	3.06	319.68					
и50к50г25	1016.24	6.00	2.57	545.76					
и50к60г25	1252.24	6.80	4.37	600.64					
и50к70г25	1303.28	8.80	3.74	488.80					
и50к80г25	1605.84	10.50	2.52	706.88					
и50к90г25	2028.08	11.00	3.70	900.32					
и50к100г25	2286.08	11.70	4.23	1016.00					

Табела 4.5: Резултати тестирања VNS алгоритма на тест примерима (и100к5г2 – и100к100г30)

Тест примери	f_{best}	t_{avg}	σ	LB	Тест примери	f_{best}	t_{avg}	σ	LB
и100к5г2	**6.00**	0.70	0.00	6.00	и100к5г5	**16.80**	1.20	0.00	16.80
и100к10г2	**10.00**	1.50	0.00	10.00	и100к10г5	**32.80**	3.30	0.00	32.80
и100к15г2	**4.00**	3.30	0.00	4.00	и100к15г5	**50.40**	7.70	0.39	50.40
и100к20г2	**31.00**	4.10	0.00	31.00	и100к20г5	54.80	13.80	1.83	45.60
и100к30г2	34.00	9.70	3.02	32.00	и100к30г5	108.40	24.00	3.70	77.60
и100к40г2	56.00	13.20	5.24	38.00	и100к40г5	207.60	30.90	6.81	144.00
и100к50г2	82.00	18.50	4.32	52.00	и100к50г5	271.20	43.70	7.12	160.80
и100к60г2	124.00	21.90	6.52	66.00	и100к60г5	340.40	49.40	9.09	149.60
и100к70г2	156.00	23.60	7.18	76.00	и100к70г5	493.20	52.30	15.45	220.80
и100к80г2	184.00	34.30	9.42	64.00	и100к80г5	597.60	78.40	16.73	281.60
и100к90г2	227.00	37.80	10.18	101.00	и100к90г5	647.20	0.40	17.91	305.60
и100к100г2	273.00	41.90	15.47	111.00	и100к100г5	714.40	86.90	16.34	276.80
и100к5г10	**17.00**	1.60	0.00	17.00	и100к5г15	**66.13**	1.50	0.00	66.13
и100к10г10	**57.20**	5.10	0.00	57.20	и100к10г15	**82.67**	7.10	0.80	82.67
и100к15г10	**107.00**	12.10	1.25	107.00	и100к15г15	145.07	16.40	1.94	141.87
и100к20г10	140.40	17.70	4.88	122.80	и100к20г15	195.73	24.90	5.03	176.27
и100к30г10	255.20	28.80	6.31	192.20	и100к30г15	386.53	35.10	6.50	303.47
и100к40г10	397.60	43.70	8.71	252.20	и100к40г15	628.40	51.60	14.61	427.20
и100к50г10	519.80	51.90	16.10	324.40	и100к50г15	795.33	60.60	18.09	516.53
и100к60г10	783.80	55.70	16.62	428.40	и100к60г15	1040.67	79.20	19.79	624.00
и100к70г10	937.40	77.00	12.56	511.00	и100к70г15	1208.67	78.40	18.39	720.53
и100к80г10	1077.20	00.10	2.55	507.60	и100к80г15	1280.13	111.10	18.20	670.67
и100к90г10	1239.80	86.20	18.95	626.20	и100к90г15	1668.00	116.00	29.06	830.40
и100к100г10	1395.00	111.60	17.58	687.80	и100к100г15	2096.67	126.90	25.68	1085.87
и100к5г20	**82.60**	1.50	0.00	82.60	и100к5г25	**45.12**	1.70	0.00	45.12
и100к10г20	**152.20**	4.70	0.00	152.20	и100к10г25	**133.28**	7.80	0.85	133.28
и100к15г20	**275.20**	12.60	0.51	275.20	и100к15г25	198.72	22.40	4.56	177.92
и100к20г20	348.60	25.40	3.77	337.00	и100к20г25	343.04	28.10	9.49	276.48
и100к30г20	492.60	38.90	7.73	415.50	и100к30г25	555.68	42.70	13.00	362.40
и100к40г20	802.40	47.70	11.37	621.70	и100к40г25	824.80	54.90	11.71	468.64
и100к50г20	901.80	62.40	9.44	646.30	и100к50г25	989.68	66.70	9.19	543.68
и100к60г20	1303.70	76.70	17.36	908.30	и100к60г25	1502.96	78.60	13.07	692.16
и100к70г20	1583.30	96.70	10.80	1063.00	и100к70г25	1756.72	107.50	19.05	886.56
и100к80г20	1837.50	98.40	25.65	1229.10	и100к80г25	2002.72	128.60	16.30	875.68
и100к90г20	2431.40	91.20	14.24	1568.10	и100к90г25	2196.96	117.70	26.05	1053.60
и100к100г20	2488.50	124.40	20.60	1524.60	и100к100г25	2598.32	135.70	25.52	1259.52
и100к5г30	**102.73**	1.60	0.00	102.73					
и100к10г30	**267.67**	4.90	0.00	267.67					
и100к15г30	364.20	17.70	2.41	354.13					
и100к20г30	439.60	32.20	2.75	419.53					
и100к30г30	756.80	40.30	8.48	645.60					
и100к40г30	1265.47	45.30	9.22	1018.00					
и100к50г30	1510.60	72.90	12.30	1184.93					
и100к60г30	1901.20	89.70	15.54	1395.87					
и100к70г30	2365.07	88.10	13.36	1723.20					
и100к80г30	2695.40	104.60	16.40	1899.20					
и100к90г30	3075.93	111.00	13.49	2141.67					
и100к100г30	3714.67	129.00	21.48	2582.47					

Табела 4.6: Резултати тестирања VNS алгоритма на тест примерима (и250к5г2 – и250к100г30)

Тест примери	f_{best}	t_{avg}	σ	LB	Тест примери	f_{best}	t_{avg}	σ	LB
и250к5г2	**6.00**	4.00	0.00	6.00	и250к5г5	13.60	126.80	0.00	13.60
и250к10г2	**7.00**	9.00	0.00	7.00	и250к10г5	29.60	17.60	0.00	29.60
и250к15г2	**14.00**	16.90	0.00	14.00	и250к15г5	35.20	46.80	0.00	35.20
и250к20г2	**19.00**	28.80	0.00	19.00	и250к20г5	46.80	89.00	1.80	44.00
и250к30г2	28.00	68.70	2.18	24.00	и250к30г5	126.40	162.90	3.81	105.60
и250к40г2	67.00	91.00	3.01	59.00	и250к40г5	181.60	210.50	5.51	128.00
и250к50г2	76.00	130.00	3.91	52.00	и250к50г5	301.20	259.90	8.78	199.20
и250к60г2	112.00	159.70	6.97	62.00	и250к60г5	372.40	364.60	14.05	211.20
и250к70г2	142.00	177.90	9.05	78.00	и250к70г5	452.80	417.60	15.83	207.20
и250к80г2	176.00	236.30	10.68	82.00	и250к80г5	556.40	512.80	15.59	212.00
и250к90г2	227.00	236.40	12.12	105.00	и250к90г5	664.80	651.20	19.56	268.80
и250к100г2	248.00	267.90	14.09	94.00	и250к100г5	812.80	612.00	23.35	292.00
и250к5г10	36.20	8.50	0.00	36.20	и250к5г15	**37.87**	11.00	0.00	37.87
и250к10г10	64.80	25.00	0.00	64.80	и250к10г15	**83.47**	28.30	0.00	83.47
и250к15г10	81.80	93.30	0.54	81.80	и250к15г15	**130.40**	93.10	1.67	130.40
и250к20г10	144.60	131.80	2.48	137.40	и250к20г15	239.33	160.70	5.29	223.73
и250к30г10	292.80	238.40	8.19	231.00	и250к30г15	412.80	249.50	9.06	333.07
и250к40г10	473.40	304.30	9.85	325.40	и250к40г15	562.00	348.30	11.29	347.73
и250к50г10	629.80	333.80	15.13	359.60	и250к50г15	1013.33	401.60	9.06	588.80
и250к60г10	779.00	488.80	19.98	400.40	и250к60г15	1074.93	644.00	24.33	537.60
и250к70г10	973.80	578.10	25.49	468.40	и250к70г15	1525.87	720.90	31.88	814.93
и250к80г10	1093.20	638.40	23.79	506.60	и250к80г15	1641.47	761.90	36.33	778.40
и250к90г10	1402.80	801.10	30.41	629.80	и250к90г15	2056.53	834.60	40.35	929.60
и250к100г10	1622.80	958.10	36.56	687.60	и250к100г15	2433.47	897.70	38.77	1114.67
и250к5г20	65.00	11.60	0.00	65.00	и250к5г25	**79.68**	12.10	0.00	79.68
и250к10г20	120.40	30.60	0.00	120.40	и250к10г25	**203.20**	28.80	0.00	203.20
и250к15г20	219.90	95.50	0.80	219.90	и250к15г25	190.40	141.50	3.00	185.76
и250к20г20	261.90	161.90	4.47	239.60	и250к20г25	337.44	184.80	6.31	305.60
и250к30г20	462.90	283.60	12.35	332.00	и250к30г25	664.48	282.10	9.46	541.60
и250к40г20	795.80	335.20	16.74	523.40	и250к40г25	1028.08	368.30	13.91	652.64
и250к50г20	1003.70	373.30	17.91	540.80	и250к50г25	1378.80	455.00	30.78	842.72
и250к60г20	1478.60	542.20	30.94	779.00	и250к60г25	1636.88	607.50	20.47	810.56
и250к70г20	1798.10	612.80	33.71	885.10	и250к70г25	2081.36	631.40	38.35	952.48
и250к80г20	2367.30	825.40	48.02	1142.60	и250к80г25	2784.80	802.80	33.71	1300.80
и250к90г20	2420.20	883.20	45.36	1124.00	и250к90г25	3064.48	992.40	39.99	1346.40
и250к100г20	2996.10	1050.30	36.95	1344.00	и250к100г25	3684.72	990.60	41.91	1557.44
и250к5г30	**86.53**	13.30	0.00	86.53					
и250к10г30	**215.73**	27.90	0.00	215.73					
и250к15г30	232.60	143.10	2.99	228.33					
и250к20г30	465.13	207.40	7.05	433.60					
и250к30г30	879.00	321.70	16.60	709.80					
и250к40г30	1186.67	358.90	22.56	751.33					
и250к50г30	1401.73	547.90	31.35	716.73					
и250к60г30	1805.40	697.60	25.97	877.60					
и250к70г30	2611.07	621.10	32.01	1255.67					
и250к80г30	2822.53	923.70	39.25	1361.53					
и250к90г30	3431.80	1043.60	52.68	1655.80					
и250к100г30	4153.60	1166.00	46.11	1784.53					

Да би се извршило објективно поређење времена извршења алгоритама, просечно време VNS алгоритма упоређено је са просечним временом CP решавача, али само на тест примерима у којима су оба алгоритма достигла исти резултат. На исти начин је извршено поређење просечног времена извршења VNS алгоритма и MIP решавача. Резиме поређења времена извршења алгоритама приказан је у Табели 4.7. Табела се састоји из четири колоне. У првој колони налазе се називи алгоритама који се пореде. Друга колона даје приказ укупног броја тест примера у којим су алгоритми (који се пореде) добили истим резултат. У трећој и четвртој колони приказане су редом вредности просечног времена t_{avg} извршења VNS алгоритма и CPLEX (CP/MIP) решавача. На основу приказаних резултата може се закључити да предложени VNS алгоритам има краће просечно време извршења у поређењу са CP и MIP решавачима, на подскупу тест примера у оквиру којих су добијени исти резултати.

Табела 4.7: Поређење просечног времена извршења на тест примерима за које алгоритми достижу иста решења

Поређења	Број тест примера са истим резултатима	VNS (t_{avg})	CP/MIP решавач (t_{avg})
VNS са CP	31 (31+0)	0.50	556.00
VNS са MIP	96 (42+54)	5.98	50.81

4.5 Примена BMWASP приступа у Београдској пословној школи

У Београдској пословној школи – Високој школи струковних студија постојала је реална потреба да се у школској 2016/2017. години расподели 229 студената са исте године у 10 група тако да свака група буде репрезентативна у односу на посматрану популацију студената. Школска управа бира релевантне карактеристике студената (укупно 116) и одређује њихов релативни значај и приоритет. Неке од разматраних карактеристика су: пол, година старости, ниво претходног знања, земља порекла, итд. Након што се одреде карактеристике и њихове тежине, задатак је распоредити студенте у одређени број група и одредити меру балансираности такве расподеле. Примена предложеног BMWASP приступа и резултати тестирања алгоритама на наведеном примеру приказани су у [53, 54].

Креирана је апликација за BMWASP у коју је имплементиран програм заснован на предложеном VNS алгоритму. Апликација је развијена на .NET Framework платформи (варијанта 3.5) у програмском језику C#. Улазни подаци се преузимају из фајла у који је смештена листа студената, листа карактеристика и бинарна матрица доделе карактеристика студентима. У посебан ред матрице смештају се тежине карактеристика тако да се и овај параметар узима у обзир приликом расподеле студената. Када програм заврши са радом, апликација као резултат враћа фајл са расподелом студената у k балансираних група. Ова апликација је доступна на адреси [99].

Табела 4.8: Поређење резултата тестирања алгоритама - Београдска пословна школа: 229 студената, 10 група и 16 карактеристика

Алгоритми	Карактеристике	Вредности
CPLEX CP решавач	вредност функције циља након 8 сати рада	**1151.20**
	број побољшаних решења	75
	применарадне меморије	149.20 MB
	број истражених грана	7 326 914
CPLEX MIP решавач	доња граница	390.40
	горња граница	**709.80**
	број побољшаних решења	19
	применарадне меморије	658.20 MB
	број итерација	118 828 066
VNS	најбоља вредност функције циља	**653.00**
	просечна вредност функције циља	671.84
	просечно време извршења	567.60 s
	применарадне меморије	36.50 MB
	стандардна девијација	11.81
	релативна стандардна девијација	1.8
LP релаксација	доња граница	390.40

У Табели 4.8 представљена су поређења резултата добијених CPLEX CP решавачем, CPLEX MIP решавачем и VNS алгоритмом на реалном примеру који броји 229 индивидуа, 116 карактеристика и 10 група. Табела је формирана из три колоне. Колоне редом приказују називе различитих приступа у решавању реалног проблема, карактеристике сваког од њих и вредности посматраних карактеристика. CPLEX CP решавач је заустављен након 28800 секунди (8 сати) рада, док је CPLEX MIP решавач заустављен након 86400 секунди (24 сата) рада.

Резултати приказани у Табели 4.8 показују да просечна вредност функције циља добијена VNS алгоритмом за мање од 10 минута је много боља од вредности функције циља добијене паралелним извршавањем CPLEX CP решавача на 4 рачунара након 8 сати рада. Резултати такође показују да CPLEX MIP решавач није успео да пронађе бољу горњу и доњу границу у односу на предложени VNS алгоритам и методу LP релаксације којом је добијена вредност доње границе.

5 Проблем формирања група за колаборативно учење

У овом поглављу разматра се и решава проблем формирања група за колаборативно учење (Collaborative Learning Groups Formation Problem – CLGFP), Опсежна примена какву ове групе имају у методичком и општем значају, иницирала је представљање новог приступа за решавање проблема CLGFP. За разлику од ранијих истраживања, предложени приступ узима у обзир ниво претходног академског знања, међуљудске односе и вештину просоцијалне прилагодљивости студената. С обзиром на комплексност и временску захтевност решавања, CLGFP проблем представља NP-тежак проблем комбинаторне оптимизације [132, 196].

5.1 Формулација проблема CLGFP

Колаборативно учење представља ефикасну стратегију за учење [204] која стимулативно делује на студенте да кроз рад у малим групама, стремећи остваривању заједничких циљева, унапређују све перформансе процеса учења [50]. Из когнитивне перспективе, колаборативним учењем долази се до виших нивоа учења, развоја критичког начина размишљања, заједничког схватања, развијања вештине решавања проблема и дугорочног памћења научених материјала. Из социјалне перспективе, кроз овакав облик учења студенти унапређују своје комуникацијске вештине, међуљудске односе, развијају позитивне ставове према члановима групе, што стимулативно делује на процес дељења и усвајања нових знања кроз интеракцију са осталим члановима групе [204]. Постоје многи примери позитивних исхода међуљудске сарадње и колаборативне размене знања у процесу учења [108], као и бројне студије које говоре у прилог унапређења самог процеса учења управо кроз овакав облик рада.

Полазна претпоставка за успешно спровођење колаборативног учења је формирање група по јасно дефинисаним критеријумима. Истраживачи у области колаборативног учења тврде да многи неуспешни исходи групног рада потичу из самог процеса формирања групе [110, 172]. Ради спречавања потенцијалних проблема, дошло је до развоја бројних метода које се примењују у процесу формирања група и које играју кључну улогу у унапређењу успеха који се остварује колаборативним учењем [109, 173] и на тај начин повећавају могућност напретка у учењу код студената [79].

Подела студената у мање групе представља веома тежак задатак с обзиром на велики број могућности њиховог распоређивања, а када се на то дода и захтев за укључивањем више променљивих, тада решавање овог проблема налаже веома захтевна израчунавања [190]. У том случају софтверска подршка може бити од велике помоћи едукаторима у процесу формирања група према специфичним критеријумима [154].

За разлику од ранијих истраживања, проблем CLGF се дефинише као проблем расподеле студената у четворочлане групе приближно истог квалитета које треба да задовоље три критеријума:

- групе треба да буду хетерогене у погледу нивоа претходног знања студената из области која се учи;
- групе треба да чине студенти са неутралним међуљудским односима (који немају негативних осећања једни према другима, нити су блиски пријатељи);
- групе треба да буду хомогене у односу на вештину просоцијалне прилагодљивости студената.

Карактеристике студената које су узете у обзир приликом формулисања критеријума за расподелу студената у групе, одабране су не само на основу њиховог утицаја на процес колаборативног учења већ и на основу могућности избора јасне стратегије за расподелу студената, која ће покрити све могућности и омогућити ефикасну имплементацију предложеног приступа.

Проблем CLGF може имати широку практичну примену, посебно у ситуацијама када је потребно распоредити велики број индивидуа у мање групе (тимове) узимајући у обзир више карактеристика. Коришћење CLGFP у едукативне сврхе, може бити од велике користи како за студенте тако и за наставнике, а све са крајњим циљем унапређења процеса организовања и извођења наставе, размене и усвајања нових знања и вештина између студената, и њиховим бољим исходима учења. Један од позитивних примера примене CLGFP приказан је приликом распоређивања 108 студената са исте студијске групе, Београдске пословне школе – Високе школе струковних студија, у балансиране четворочлане групе за колаборативно учење.

5.2 Преглед литературе

У овом одељку је дат преглед специфичних персоналних карактеристика које су изабране као критеријуми за распоређивање индивидуа у четворочлане групе за колаборативно учење, постојећи приступи у формирању група за колаборативно учење, као и мотивациони фактори који су инспирисали ово истраживање.

5.2.1 Груписање засновано на специфичним карактеристикама

Примена колаборативног учења као наставне методе, захтева од наставника да донесе две важне одлуке. Прва одлука подразумева одабир карактеристика студената које могу битно утицати на исходе колаборативног учења. Неке од карактеристика предложених у претходним истраживањима [13, 34, 35, 170, 182] су: раса, пол, способности, самоефикасност и стилови учења Поред наведених карактеристика, и социјалне вештине студената могу значајно утицати на исходе колаборативног учења, с обзиром да интеракције између чланова групе играју битну улогу у групном раду [154].

Друга важна одлука се односи на дефинисање типа група које се формирају - хетерогене или хомогене (видети одељак 1.4). Према општим критеријумима типови група се могу комбиновати, па тако група може бити хомогена у односу на једну карактеристику, а хетерогена у односу на неке друге карактеристике [78]. Поједине студије, у оквиру којих су разматране композиције група, указују на предност хомогених група у односу на хетерогене, под условом да је променљива у односу на коју се формирају групе позитивно повезана са учењем или колаборативним процесом. Такав пример представља висок степен кохезивности или висок степен социјалних способности у [33, 82, 95, 97]. С друге стране, постоје студије [79, 104, 111, 132, 150, 190] у оквиру којих су разматране процедуре за формирање група и у којима су хетерогене групе преферирајуће у односу на хомогене, у погледу академског успеха, пола, националности, оријентације, способности.

Способности студената

Хетерогене групе формиране у односу на способности студената имају позитиван утицај на студенте, без обзира на њихове способности [169]. Међутим, у тако формираним групама постоји могућност да студенти са вишим нивоом способности (енгл. high-ability students) буду незадовољни, јер подучавање слабијих студената захтева додатни труд; док с друге стране, студенти нижих способности (енгл. low-ability students) могу осетити непријатност из разлога што често имају потребу да затраже помоћ од осталих чланова групе [2]. Када је реч о хомогеним групама формираним у односу на споменуту карактеристику, такве групе имају позитиван ефекат на студенте са високим нивоом способности, али слабије студенте могу оставити без довољно могућности за напредак [190]. Чланови хомогених група које су састављене само од студената са високим нивоом способностима знатно чешће траже тачан одговор или саму суштину проблема, док чланови група које укључују само студенте са ниским способностима више оклевају у тражењу помоћи [192]. Дакле, могло би се закључити да процес колаборативног учења најбоље функционише у оквиру хетерогених група које се састоје из чланова са високим и са ниским нивом способности, иако постоји могућност да велике разлике између чланова групе умање ефекте сарадње [192].

Међуљудски односи између студената

Истраживања показују да већина метода заснованих на употреби рачунара не узима у обзир међуљудске односе између индивидуа, као параметар за формирање група. Међутим, лоши међуљудски односи између чланова групе могу довести до нефункционисања таквих група и до негативног искуства у учењу и исходима [4] и до потребе за расформирањем група које су формиране на чисто академским принципима. У истраживању [150] две од девет група формираних без разматрања међуљудских односа између чланова група, нису биле у могућности да функционишу управо због личних несугласица које су постојале између појединих студената.

У оквиру неколико метода у којима су разматрани међуљудски односи [104, 169, 186], то је урађено само са циљем да се избегну негативни односи између чланова групе. Студентима је дата могућност да бирају чланове своје групе, под условом да такав избор не ремети параметре постављене од стране особе која формира групе.

Добри међуљудски односи и осећај заједнице наводе се у [125] као предуслов бољег извршења задатака. Нижи ниво социјалних интеракција може имати негативан утицај на ефикасност колаборативног учења [124]. Ако се групе за колаборативно учење формирају од међусобно пожељних чланова за сарадњу, оне ће функционисати боље [126]. У [204] дефинисана је хипотеза по којој задовољавање жеље студената у одабиру чланова своје групе може бити од помоћи у успостављању боље сарадње. Међутим, групе формиране по студентовом слободном избору су врло често засноване на пријатељским односима, што може довести до слабијих резултата у учењу услед недостатка више различитих перспектива [2].

Постоје студије у оквиру којих се показало да добри пријатељи нису увек најбољи избор за партнере у групама за колаборативно учење, с обзиром да су неки од преферирајућих колега били добри само у социјалним оквирима [169, 185]. Из наведених разлога, могло би се закључити да би виши ниво добрих међуљудских односа између студената требало да позитивно утиче на виши ниво међусобне сарадње, под условом да неформална конверзација не буде приоритетна у поређењу са дискусијама које су орјентисане ка решавању задатака. Према томе, било би пожељно да групе за колаборативно учење не укључују студенте који су блиски пријатељи нити студенте који не могу сарађивати [185], све у циљу стварања добре радне и професионалне атмосфере.

У оквиру приступа описаног у [204], међуљудски односи студената разматрани су у својству критеријума за формирање група за колаборативно учење. С друге стране, у [169, 104, 186] представљене су методе у оквиру којих се разматра појам компатибилности (који се заснива на рејтингу појединца који се формира од стране осталих студената), рефлектујући на тај начин у којој мери поједини студенти желе или не желе да сарађују са осталим студентима. Методе су дизајниране тако да, поред задовољења критеријума које задаје наставник, имају за циљ да задовоље и жеље студената. Овакав приступ представља комбинацију два приступа, студентовог слободног избора и наставниковог избора, у процесу креирања група за колаборативно учење.

Вештина просоцијалне прилагодљивости

Просоцијално понашање подразумева „широку категорију конкретних понашања која су дефинисана једним значајним сегментом друштва и/или друштвене групе као генерално корисно за друге људе” [158]. Отвореност позитивно утиче на особу да вољно прихвата нове идеје и да буде спремна да искуси и позитивне и негативне емоције [37]. Вештина просоцијалне прилагодљивости је конструкција заснована на просоцијалном понашању и отворености. Аутори у [154] описују наведену вештину као „отвореност ка мишљењима других људи, способност да се ствари сагледају из другачијих перспектива и способост да се помогне другима”.

Хетерогеност у односу на вештину просоцијалне прилагодљивости може бити главни узрок незадовољства перформансама групе, недостатка ефикасности сарадње и поделе одговорности [154]. Повећана хетерогеност може проузроковати ситуацију у којој се задаци чешће деле међу члановима групе [113]. Појединачни нивои развијености социјалних вештина сваког од чланова групе су од мањег значаја и утицаја на ефективност процеса колаборативног учења у односу на конфигурацију социјалних вештина целе групе [154].

Из тих разлога, хомогене групе формиране у односу на вештину просоцијалне прилагодљивости требало би да буду ефикасније од хетерогених група без обзира на просечан ниво развијености ове вештине [154]. Истраживање представљено у [154] је показало да студенти испољавају висок ниво задовољства оствареним резултатима и ефикасношћу колаборативног учења у хомогеним групама које су формиране у односу на поменуту друштвену вештину.

5.2.2 Постојећи приступи у формирању група за колаборативно учење

Генерално гледано, групе за колаборативно учење се најчешће формирају коришћењем метода које се заснивају на: студентовом слободном избору (енгл. student selection), случајном избору (енгл. random selection) или наставниковом избору (енгл. teacher selection) [190]. Групе формиране по студентовом слободном избору се врло често заснивају на пријатељским односима или на заједничким интересовањима у погледу одређених тема, што најчешће може резултирати формирању хомогених група [2]. Пријатељски односи између чланова групе могу олакшати процес колаборативног учења, али могу довести и до слабијих резултата у учењу услед недостатка различитих начина сагледавања ствари и појава. Према тврђењу изложеном у [169], добри пријатељи не морају нужно бити и добри партнери у учењу. Исто тако, приликом формирања хомогених група постоји могућност да њени чланови избегну да укључе појединце са слабије развијеним социјалним вештинама [190].

С друге стране, применом методе случајног избора за формирање група, смањује се могућност да се појединци осећају одбаченим или издвојеним. Међутим, групе формиране случајним избором могу бити структуиране тако да се састоје само од слабијих појединаца, односно од студената са ниским способностима. Осим тога, постоји могућност да припадници случајно одабраних група не могу да раде заједно, па се такве групе морају преуредити [150]. Метода заснована на наставниковом избору чланова групе подразумева да наставници користе специфичне карактеристике појединаца, обично њихове способности или претходна постигнућа, у циљу формирања група [190]. Наставници који довољно добро не познају своје студенте не могу у потпуности искористити предности ове стратегије, с обзиром да не располажу са довољно поузданим информација за формирање група.

У оквиру неколико студија [44, 79, 104, 105, 132, 150, 169, 186, 190, 204], развијене су различите варијанте рачунарских метода које могу бити од велике помоћи у процесу формирања група за колаборативно учење. У [79] представљена је метода заснована на оптимизацији мравље колоније, која користи опште перформансе студената и академске способности студената за формирање хетерогених група. Метода представљена у [105] користи се поступком унапређеног генетског алгоритма како би се формирале „интер-хомогене" групе, узимајући у обзир освојене поене са прелиминарног теста и нивое савладаног градива одређеног курса. Варијанта методе оптимизације роја честица коришћена је у [132] за формиране „интер-хомогених" група, узимајући у обзир интересе студената и нивое разумевања градива. У [44] се формирају „интер-хомогене" и „интра-хетерогене" групе за учење у односу на две карактеристике студената (претходни ниво знања и интересовања), такође користећи методу оптимизације роја честица.

У [15] је представљен модел векторског простора у оквиру којег је разматрано неколико персоналних карактеристика (пол, став према групном раду, интересовање за математику, мотивација за успех, самопоуздање, стидљивост, нивои знања енглеског језика и математике). Нови приступ за формирање група представљен је и у [190]. Овај приступ уврштује три карактеристике које се односе на одређене стилове размишљања студената заједно са методом заснованом на генетском алгоритму. Приступ који користи генетски алгоритам за груписање студената на основу више њихових карактеристика (нивоа знања, комуникативних и лидерских вештина) са циљем задаовољења хомогености међу групама (inter-homogeneous grouping) развијен је и у [150]. Још један пример примене унапређеног генетског алгоритма за формирање група за колаборативно учење описан је у [204]. Предложени приступ задовољава неколико општих и контекстно-специфичних критеријума за груписање студената.

5.3 Предложени приступ у решавању проблема CLGFP

У овом одељку је представљен нови приступ за формирање група за колаборативно учење, у оквиру којег се узимају у обзир различити нивои (претходног) академског постигнућа сваког од посматраних студената, њихови међуљудски односи и развијеност вештине просоцијалне прилагодљивости код сваког од њих. Процес формирања група (који укључује неколико параметара) представљен је као проблем математичког програмирања. Предложена је математичка формулација проблема и развијена метода променљивих околина за његово решавање. Резултати описани у овом одељку, приказани су у раду [128].

5.3.1 Критеријуми за формирање група

Кроз процес колаборативног учења спроведена је студија истраживања са студентима прве године Београдске пословне школе – Високе школе струковних студија у оквиру наставе из предмета Статистика. Предложено је да се формирају четворочлане групе које су хетерогене у односу на ниво претходног знања студената из области која се учи, које чине студенти са неутралним међуљудским односима (који немају негативан став о осталим члановима групе, нити су међусобно блиски пријатељи) и које су хомогене у односу на вештину просоцијалне прилагодљивости.

Да би се установио ниво претходног знања студента из предмета који се разматра, коришћени су резултати са првог колоквијума, који ће се у даљем тексту звати прелиминарни тест. Прелиминарни тест се састојао од четири математичка проблема, које је требало решити у временском периоду од три сата. Поред решења задатих проблема, било је потребно приложити и детаљан опис свих корака у поступку решавања проблема. Укупан број поена које је студент могао да освоји на колоквијуму био је скалиран на интервалу вредности [0,100], тако да је ниво претходног знања за сваког студента представљен са једним бројем из датог интервала. Унутрашња конзистентност (поузданост) прелиминарног теста одређена је према Cronbach α коефицијенту [38] и добијена је прихватљива поузданост теста од 0.78. На основу резултата са прелиминарног теста, студенти су распоређени у „разумно хетерогене” групе приближно једнаког квалитета у односу на број освојених поена. Под „разумно хетерогеном” групом се подразумева група састављена од студената са ниским, средњим и високим успехом [79]. Иницијална препорука да групе за колаборативно учење треба да чине студенти са „измешаним способностима” (енгл. mixed-ability) изложена је у [172]. Такве групе би се састојале од по једног члана са високим, два са просечним и једног са ниским способностима.

Да би се утврдили међуљудски односи између студената, студентима је подељен упитник у оквиру којег је требало да наведу имена колега са којима би желели да сарађују, са којима не би желели да сарађују и са којима би могли да сарађује али не и да заједно уче. Мање пожељни односи су представљани већим бројем, док су бољи односи означавани са мањим бројем. На основу добијених података, формирана је матрица међуљудских односа студената. Горње ограничење броја који је додељиван за дефинисане односа није постојало, тако да је наставницима омогућено да сами бирају скалу позитивних бројева којом су означавали различите односе.

Ниво развијености вештине просоцијалне прилагодљивости код појединаца, мерена је тако што су студенти добили задатак да одговоре на упитник из [154], састављен од самореференцирајућих изјава које је требало оценити помоћу скале од четири броја (не слажем се уопште [1] - потпуно се слажем [4]). На основу резултата упитника, за сваког студента израчуната је просечна вредност свих одговора. Добијени број из интервала [1,4] узет је за ниво развијености ове вештине. Сразмерно нивоу развијености вештине просоцијалне прилагодљивости, студенти су распоређени на ранг листи.

5.3.2 Математичка формулација проблема CLGFP

Како је CLGFP проблем комбинаторне оптимизације, у овом одељку је описан математички модел проблема за оптимално распоређивања студената у четворочлане групе приближно истог квалитета, по утврђеним критеријумима. Сви резултати презентовани у овом одељку, представљени су у раду [128].

Да би се формирале разумно хетерогене групе у односу на ниво претходног академског успеха, креирана је ранг листа студената (од најслабијег до најуспешнијег) на основу освојених поена са прелиминарног теста. У [79] се дефинише „мера добре хетерогености” A_i (енгл. measure of goodness of heterogeneity) помоћу формуле:

$$A_i = \frac{S_1 - S_4}{1 + \left|\frac{S_1 - S_4}{2} - S_2\right| + \left|\frac{S_1 - S_4}{2} - S_3\right|}; \tag{5.1}$$

у којој S_1 представља максималан број поена, S_4 минималан број поена, а S_2 и S_3 поене преосталих студената формиране i-те „мале" групе. На основу представљене мере хетерогености, подела студената у групе би по овом критеријуму требало да буде спроведена на следећи начин.

Прва четворочлана група G_1 би се формирала од студента са максималним бројем освојених поена и студента са минималним бројем поена, и од два студента чији су освојени поени најприближнији аритметичкој средини поена најбољег и најслабијег студента. У циљу формирања следеће четворочлане групе, одабрани студенти би се уклонили са ранг листе и процес би се понављао.

Међутим, због разматрања и других критеријума у процесу формирања група, коначна подела студената у групе за колаборативно учење се разликује од поделе која би се добила претходно наведеним поступком. Недостатак претходног начина мерења хетерогености групе се огледа у томе што мале разлике у броју освојених поена S_j могу довести до великих варијација мере хетерогености A_i, што може довести до диспропорције са мерама осталих карактеристика које се користе у методи. Из тих разлога, уводи се прилагођена варијанта претходне формуле:

$$A_i = \left|\frac{S_1+S_4}{2} - S_2\right| + \left|\frac{S_1+S_4}{2} - S_3\right| - S_1 + S_4 + Max; \tag{5.2}$$

у којој Max представља максималан број поена са прелиминарног теста, а $1 \le i \le [\frac{N}{4}]$, где је N укупан број студената. Што је вредност A_i мања, то је хетерогеност групе G_i већа. Због пропорције са мерама осталих карактеристика, препоручено је да се резултати са прелиминарног теста скалирају до 100 поена.

Нека је са F_{xy} означена матрица која представља међуљудске односе између студената, где је $1 \le x, y \le N$ [104]. F_{xy} не мора бити једнако са F_{yx}. У студији [104] дефинисана је само једна опција у случају када студент x не жели да буде у истој групи са студентом y. У оквиру методе која се представља у овом поглављу, предлажу се две опције за дефинисање непожељних чланова групе како би, у случају да се све жеље студената не могу испунити, ипак постојала разлика између мање и више непожељних партнера за сарадњу. Из тих разлога уводи се следећа формула којом се дефинишу међуљудски односи:

$$F_{xy} = \begin{cases} I_0 \text{ , ако } x \text{ у потпуности жели да избегне } y \\ I_1 \text{, ако x сматра да } y \text{ није најбољи избор за сарадника у групи} \\ I_2 \text{ , ако } x \text{ жели да сарађује са у} \\ 0 \text{ , ако } x \text{ има неутралан став према } y \end{cases} \quad ;(5.3)$$

Што су вредности I_i веће, за $0 \leq i \leq 2$, то је распоређеност студената x и y у исту групу мање пожељно. У спроведеном истраживању коришћене су следеће вредности: I_0=10000, I_1=1000, I_2=10.

Према томе, мера међуљудских односа у групи дата је формулом:

$$B_i = \sum_{x \in G_i} \sum_{y \in G_i} F_{xy}; \quad (5.4)$$

Што је вредност B_i мања, то су међуљудски односи у групи G_i бољи.

У односу на вештину просоцијалне прилагодљивости, групе би требало да буду хомогене. Потребно је направити ранг листу студената у односу на нивое развијености вештине просоцијалне прилагодљивости код сваког од њих. Уколико два студента имају исти ниво ове вештине, њихов ранг треба да буде исти. Разлике између рангова чланова четворочлане групе би требало да буду што мање, како би група била хомогенија у односу на наведену карактеристику.

Према томе, мера хомогености мале групе G_i је дата формулом:

$$C_i = \sum_{j=1}^{3} \sum_{k=j+1}^{4} \left| R_j - R_k \right|; \quad (5.5)$$

у којој су $\{R_j, R_k\}$ рангови студената у односу на вештину просоцијалне прилагодљивости. Што је вредност C_i мања, то је хомогеност групе G_i већа.

Како би се у односу на претходно наведене карактеристике обезбедило оптимално распоређивање студената у четворочлане групе, предложена је математичка формулација проблема CLGFP:

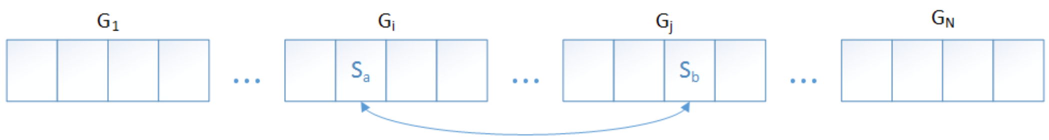
$$min\big(max(w_0 A_i + w_1 B_i + w_2 C_i)\big) \, . \quad (5.6)$$

Функција циља (5.6) описаног математичког модела представља број нерешених захтева четворочлане групе, односно меру квалитета једне групе. Што је мера квалитета мања, група је боље креирана. За рачунање мере квалитета групе узимају се у обзир приоритети међу карактеристикама. Приоритети су представљени преко вредности тежина карактеристика које се означавају константама: w_0, w_1, w_2. (У истраживању је w_0=10, w_1=1, w_2=1.) Максимизовање функције циља подразумева проналажење групе која у посматраној подели има највећи број нерешених захтева, односно чија је мера квалитета највећа. Према томе, циљ проблема CLGFP је минимизовање максималног броја нерешених захтева четворочланих група, односно, проналажење поделе у којој се налази група са најмањим максималним бројем нерешених захтева.

5.3.3 Метода променљивих околина за решавање проблема CLGFP

У овом одељку су описане основне карактеристике методе променљивих околина (VNS) посебно развијене за решавање проблема CLGFP. Резултати представљени у овом одељку приказани су у раду [128].

Нека променљива x представља једно решење предложеног проблема, а променљива N_k $(k = 1,\ldots, k_{max})$ низ унапред одабраних структура околина, тада се са $N_k(x)$ означава скуп решења у k-тој околини од x. Једно решење x проблема представља једну расподелу студената у четворочлане групе за колаборативно учење. Структуре околина су дефинисане на начин да се решење x' налази у k-тој околини x, ако се расподела студената x' може добити из расподеле x са k замена по једног пара студената између различитих група. Једна замена пара студената између група $(Gi, Sa) \leftrightarrow (Gj, Sb)$ се дефинише заменом једног студента Sa из групе Gi са другим студентом Sb из групе Gj, што резултира да је након замене студент Sa измештен у групу Gj, а студент Sb у групу Gi (Слика 5.1).



Слика 5.1: Једна замена студената

Функција циља једног решења дефинисана је формулом $max(w_0A_i + w_1B_i + w_2C_i)$, представљеном у одељку 5.3.2. Да би се израчунала функција циља једне расподеле, потребно је за сваку групу израчунати број нерешених захтева $w_0A_i + w_1B_i + w_2C_i$, и након тога издвојити највећу такву вредност (максималан број нерешених захтева).

Уколико је позната функција циља текућег решења, приликом рачунања функције циља новодобијеног решења из његове околине, није неопходно поново рачунати број нерешених захтева за сваку групу. Потребно је израчунати нерешене захтеве само за две групе које су учествовале у замени студената и на основу тога ажурирати функцију циља нове расподеле.

Иницијално решење алгоритма се формира случајном расподелом студената у групе. Затим се на њега примени RVNS алгоритам (приказан Алгоритмом 4) и добијено решење представља иницијално решење за основни VNS алгоритам (приказан Алгоритмом 5). У фази локале претраге примењује се стратегија најбољег побољшања, а претрага се врши само у првој околини.

Максималан број околина у RVNS алгоритму је $k_{max} = 3$, док је у основном VNS алгоритму $k_{max} = min\{\frac{N}{3}, 20\}$. Параметар *iteration_number* је улазни параметар апликације који одређује максималан број итерација (корака) алгоритма пре испуњења услова заустављања. Услов заустављања RVNS алгоритма је 1000*$iteration\_number$ узастопних итерација алгоритма без побољшања функције циља, док је услов заустављања VNS алгоритма $iteration\_number$ узастопних итерација алгоритма без побољшања функције циља.

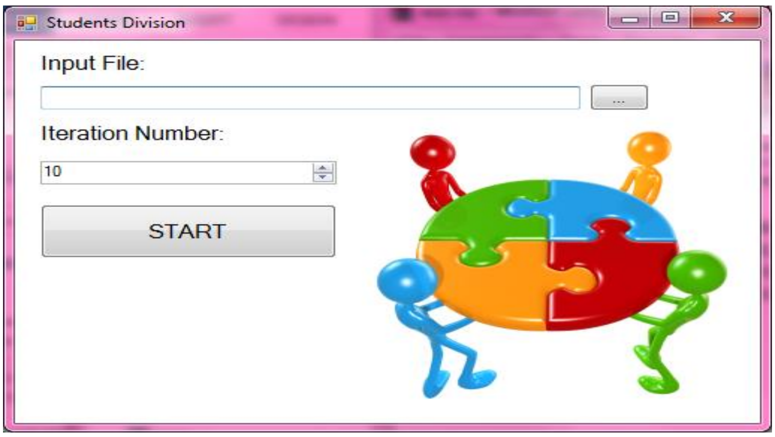
5.3.4 Апликација за CLGFP

За потребе истраживања развијена је апликација која решава предложени проблем комбинаторне оптимизације. Апликација је развијена на .NET Framework платформи (варијанта 3.5) и програмском језику C#. За решавање проблема CLGFP у апликацији је имплементиран програм заснован на предложеном VNS алгоритму. За улазне податке, апликација захтева фајл са подацима о студентима. За сваког студента потребно је да се у фајлу налазе следећи подаци:

- ИД;

- Идентификатор студента;
- Број освојених поена на прелиминарни тесту;
- Вештина просоцијалне прилагодљивости;
- Списак студената које у потпуности жели да избегне;
- Списак студената које сматра да нису најбољи избор за сараднике;
- Списак студената са којима жели да сарађује.

ИД представља јединствени идентификациони број студента. Идентификатор студента је текстуални опис студента: његово име, презиме, број индекса ... Број освојених поена на прелиминарном тесту представља број поена које је студент добио на колоквијуму. Број поена треба да буде скалиран у опсегу вредности [0,100]. Вештина просоцијалне прилагодљивости показује позицију студента на ранг листи формираној на основу посматране карактеристике. Уколико два или више студента имају исти ниво вештине просоцијалне прилагодљивости, додељује им се исти број (ранг) на листи. Управо из тих разлога, следећи или више следећих бројева биће прескочено. Списак студената који се у потпуности желе избећи, оних који нису најбољи избор за сарадњу и оних са којима се жели сарађивати, је представљен одговарајућим идентификационим бројевима који су раздвојени зарезима. Након што програм учита податке из фајла, покреће се алгоритам који распоређује студенте у четворочлане групе. У програму постоји параметар који означава укупан број итерација VNS алгоритма. Већа вредност параметра повећава шансе да алгоритам понуди боље решење, али уједно повећава и време извршења програма.



Слика 5.2: Приказ апликације за проблем CLGFP

Када програм заврши са радом, као резултат кориснику се враћа нови фајл, у којем се налази добијена расподела студената у групе. Свака од четворочланих група садржи имена одабраних студената записаних у четири реда, заједно са вредностима карактеристика групе (A_i, B_i, C_i), које су за сваку групу посебно израчунате. Предложена апликација је доступна на адреси [100].

5.4 Експериментално истраживање

Са циљем потврде ефикасности развијеног VNS алгоритма, извршено је тестирање алгоритма на различитим димензијама проблема и добијени резултати су упоређени са резултатима добијеним методом случајног избора (Монте Карло методом). С друге стране, у циљу потврде адекватности предложеног CLGFP модела, спроведен је експеримент над групом студената Београдске пословне школе. Резултати представљени у овом одељку, приказани су у раду [128].

5.4.1 Циљеви истраживања

У оквиру истраживања описаног у овом поглављу представљен је нов приступ формирању четворочланих група за колаборативно учење, и у складу са тим постављена су главна истраживачка питања. Питања су се односила на ефикасност примене предложеног VNS алгоритма за решавања проблема CLGFP и на могућност примене предложеног приступа у настави са циљем унапређења исхода колаборативног учења студената. На основу питања, формулисане су следеће хипотезе истраживања:

Хипотеза 5.1. Применом предложеног VNS алгоритма добија се расподела студената у четворочлане групе за колаборативно учење, која у знатно већој мери и у кратком временском року испуњава све задате критеријуме у односу на расподелу добијену Монте Карло методом случајног избора.

Хипотеза 5.2. Студенти група формираних предложеним приступом постижу бољи академски успех на тесту провере знања у поређењу са студентима група које су формиране методом случајног избора или методом студентовог избора.

5.4.2 Експеримент у настави

У својству наставне методе, колаборативно учење је примењено на часовима предавања и вежби предмета Статистика над групом од 108 студената прве године Београдске пословне школе, у летњем семетру школске 2014/2015. години**.** На часовима је изучавана наставна област - Регресиона и корелациона анализа. Сви студенти су били почетници у наведеној области, и никада раније нису били укључени у активности колаборативног учења.

Формирање група за тестирање

У циљу процене адекватности предложеног CLGFP модела, истраживање је спроведено иницијалним распоређивањем студената у две групе за тестирање, експерименталну и контролну групу. Распоређивање студената је извршено коришћењем процедуре усклађивања (енгл. matching procedure) имајући у виду затхев да групе буду што уједначеније у односу на пол студената и у односу на њихове резултате са прелиминарног теста.

Експерименталну групу је чинило 56 студената (43 особа женског пола, 13 особа мушког пола; просечне старости 20 година), док је контролну групу чинило 52 студента (40 особа женског пола, 12 особа мушког пола; просечне старости 20 година). Резултати са прелиминарног теста за студенте експерименталне и контролне групе су приказани у Табели 5.1. Табела је организована на начин да прва колона садржи информацију о називима група, док је у другој колони дат приказ укупног броја студената по групама. Наредне две колоне приказују вредности аритметичких средина броја бодова студената и вредности стандардних девијација (просечног одступања свих података од њихове аритметичке средине) редом за експерименталну и контролу групу.

Да би се извршило поређење резултата са прелиминарног теста за два независна узорка, коришћен је Студентов t-тест са нивоом значајности од 5% (α =0.05). Студентов t-тест укључује тестирање нулте хипотезе против алтернативне хипотезе. Нулта хипотеза је поставка којом се тврди да не постоји значајна разлика између аритметичких средина броја бодова два независна узорка, док се алтернативном хипотезом тврди супротно, да постоји значајна разлика између аритметичких средина два независна узорка.

Табела 5.1: Резултати са прелиминарног теста

Групе	Укупан број студената	Аритметичка средина	Стандардна девијација
Експериментална	56	22.96	8.70
Контролна	52	23.00	9.60

Применом Студентовог t-теста на резултате приказане у Табели 5.1 добијена је вредност параметра $p = 0.984$. Ова вредност параметра p се пореди са нивом значајности α =0.05. Како је вредност параметра p већа од вредности параметра α, може се са сигурношћу од 95 % тврдити да није одбачена нулта хипотеза, односно да није постојала значајна разлика између експерименталне и контролне групе у погледу резултата које су студенти постигли на прелиминарни тесту. На овај начин је показано да су студенти обе групе били у сличним условима пре упоређивања исхода колаборативног учења.

Просечан ранг вештине просоцијалне прилагодљивости за експерименталну групу је био 3.20 а за контролну групу 3.22. У погледу међуљудских односа, студенти из експерименталне групе навели су у просеку 1.73 имена студената које су желели у потпуности да избегну, затим 0.86 имена студената које нису сматрали најбољим избором за сарадњу и 3.43 имена студената са којима су желели да сарађују. Студенти из контроле групе навели су у просеку 1.69 имена студената које су желели у потпуности да избегну, затим 0.88 имена студената које нису сматрали најбољим избором за сарадњу и 3.36 имена студената са којима су желели да сарађују. Ови подаци показују да нису постојале значајне разлике између експерименталне и контролне групе ни у односу на наведене карактеристике.

Коришћењем предложеног приступа заснованог на VNS алгоритму, студенти распоређени у експерименталну групу су даље подељени у четрнаест четворочланих група, док су студенти из контролне групе распоређени у девет четворочланих група методом студентовог слободног избора и у четири четворочлане групе методом случајног избора. Метода случајног избора је примењена у процесу распоређивања студената који нису успели самостално да се распореде у неку од група. Студенти из експерименталне и контролне групе били су физички одвојени током експеримента.

Просечна мера хетерогености, заснована на резултатима са прелиминарног теста, за четворочлане групе формиране у оквиру експерименталне групе била је 75.57 (минимум 66, максимум 83), док је за четворочлане групе формиране у оквиру контролне групе била 86 (минимум 47, максимум 100). Ова разлика у мерама показује да је хетерогеност група формираних у експерименталној групи, у односу на претходни ниво знања студената, била већа. Просечна мера хомогености четворочланих група, у односу на вештину просоцијалне прилагодљивости, у експерименталној групи била је 67.36 (минимум 0, максимум 152), а у контролној групи 88.7 (минимум 27, максимум 151) што потврђује да је експериментална група била боље подељена на хомогене групе. У погледу међуљудских односа, експериментална група се састојала од десет, а контролна од три четворочлане групе у којима студенти нису ни пријатељи ни непријатељи. Преостале четири четворочлане групе из експерименталне групе су биле састављене од студената који су желели да буду заједно у групи.

Процес колаборативног учења

Експеримент је спроведен у временског периода од месец дана. Током прве недеље, студенти су добили задатак да проуче материјал из области Регресивне и корелационе анализе. Следеће недеље студенти су учили нову лекцију, уз помоћ професора, асистената и колега у групама за колаборативно учење, кроз интерактивни процес учења, усвајањем нових знања и решавањем различитих проблема. На крају недеље свака од формираних четворочланих група, у оквиру експерименталне и контролне групе, добила је задатак да уради пројекат. Пројекат се састојао из низа питања која су представљала кораке за решавање задатих проблема. Сваки од чланова групе је требало најпре да самостално реши проблеме, а затим би сви чланови групе дискутовали и заједно дефинисали коначна решења задатих проблема. Групе су биле у обавези да доставе професору сва индивидуална решења, заједно са групним решењима. У трећој недељи, студенти су добили задатак да у оквиру формираних група за колаборативно учење, током предавања и вежби, заједно решавају проблеме, у трајању од 3 сата дневно. На крају месеца, студенти су награђени за успешно завршен пројекат са максималних 10 поена од укупно 100 поена које су могли освојити на целом испиту.

Након периода колаборативног учења, студенти су тестирани. Пост-тест је спроведен на исти начин као прелиминарни тест. Студенти су индивидуално решавали пост-тест, као у случају прелиминарног теста. У циљу процене поузданости пост-теста израчунат је Cronbach α коефицијент. Добијени коефицијент (Cronbach α) износи 0,81 што представља добру унутрашњу конзистенцију (поузданост) пост-теста. Максималан број поена на прелиминарном тесту и пост-тесту био је исти, по 30 поена.

5.5 Експериментални резултати

У овом одељку приказани су резултати провере ефикасности и поузданости предложеног VNS алгоритма кроз скуп рачунарских тестирања и резултати исхода предложеног приступа у односу на академски успех студената. Експериментални резултати приказани у овом поглављу описани су у раду [128]

5.5.1 Резултати тестирања VNS алгоритма

У оквиру експеримента за тестирање предложеног VNS алгоритма, коришћени су тест примери засновани на реалним подацима студената прве године Београдске пословне школе. Тест примери су генерисани на основу група које су чинили редом 16, 24, 32, 48, 64, 96 и 100 студената. VNS алгоритам је тестиран на рачунару са AMD FD-7500 (2.10 GHz) процесором и 8GB RAM меморије, на 64-битној варијанти Windows 8.1 оперативног система.

У циљу процене добијених резултата, за сваки тест пример је генерисана и случајна расподела студената у четворочлане групе и мерен је квалитет тако добијених расподела. Да би се обезбедило регуларно поређење, за методу случајне расподеле узета је Монте Карло метода. Монте Карло алгоритам је извршаван у неколико итерација и на крају је бирана најбоља добијена расподела. Како се са повећањем броја студената повећава и број могућих комбинација расподеле студената у групе, одлучено је да се у експерименту за методу случајне расподеле не користи константан број итерација, већ број од 2000*N итерација, где је N укупан број студената. С друге стране, за услов заустављања рада VNS алгоритма узето је добијање истог квалитета расподеле у 10 узастопних итерација алгоритма. Једна итерација VNS алгоритма обухвата један пролазак кроз фазу размрдавања и фазу локалне претраге.

Резултати тестирања Монте Карло и VNS алгоритма приказани су у Табели 5.2. У првој колони табеле дат је приказ укупног броја студената које је требало распоредити у групе. Друга и трећа колона приказују перформансе Монте Карло алгоритма, при чему се у другој колони налази добијена вредност функције циља f (рачуната по формули 5.6) која уједно приказује и квалитет добијене расподеле, док се у трећој колони налази време t извршења Монте Карло алгоритма, дато у секундама. Четврта и пета колона приказују перформансе предложеног VNS алгоритма, при чему је вредност функције циља f приказана у четвртој колони, а време t извршења VNS алгоритма у петој колони.

Резултати приказани у Табели 5.2 показују да Монте Карло и VNS алгоритам дају исте резултате само у случају најмањег тест примера, када је укупан број студената 16. У том случају VNS алгоритам проналази најбољу расподелу у краћем временском периоду. При решавању преосталих тест примера, VNS алгоритам даје боље резултате у поређењу са Монте Карло алгоритмом (углавном у краћем временском периоду), чиме се потврђује Хипотеза 5.1. Монте Карло алгоритам би могао постићи бољу брзину ако би користио мањи број итерација. Међутим, у том случају достизање доброг решења са Монте Карло методом би било много мање вероватно.

Ако би се представљени проблем пробао решити неком од метода исцрпљивања, у случају када n студената треба поделити на $m = n/4$ група, могући број различитих расподела би био $\frac{\binom{n}{4}\cdot\binom{n-4}{4}\cdot\binom{n-8}{4}\cdots\binom{8}{4}\cdot\binom{4}{4}}{(m)!} = \frac{n!}{(m)!\cdot 24^m}$. На пример, ако би 24 студента требало распоредити у 6 четворочланих група, укупан број свих могућих комбинација би био приближно једнак 2^{42}. Узимајући у обзир да је за сваку комбинацију потребно да се прође кроз списак студената у једној итерацији како би се израчунала функција циља добијеног решења, време које би било потребно да се израчуна вредност функције циља за све комбинације (на процесору на којем се вршило тестирање) би захтевало више од 913 дана.

Табела 5.2: Перформансе Монте Карло и VNS алгоритма

Укупан број студената	Монте Карло алгоритам		VNS алгоритам	
	f	$t(s)$	f	$t(s)$
16	1006	0.41	1006	0.28
24	957.3	0.84	920	0.47
32	959	1.45	909	0.81
48	1000	3.28	848	3.36
64	1070	6.23	848	5.07
96	1207.3	14.33	1010	8.91
100	1221.7	15.57	955.4	11.33

Поред задовољавајућих резултата који се остварују помоћу предложеног VNS алгоритма, може се констатовати да је алгоритам и временски веома ефикасан у односу на димензије проблема које се решавају. Просечно време потребно за решавање тест примера је 4.32 секунде, док је за решавање највећег тест примера у оквиру којег се 100 студената распоређује у 25 група, потребно 11.33 секунде што је веома кратко време, узимајући у обзир укупан број могућих комбинација такве расподеле.

5.5.2 Исходи предложеног приступа у погледу академског успеха студената

Како би се утврдило да ли након колаборативне активности постоји значајна разлика у исходима учења између студената распоређених у експерименталној и контролној групи, коришћен је Студентов t-тест са нивоом значајности од α = 0,05 (аналоган тесту представљеном у одељку 5.4.2) за поређење резултата са пост-теста. Вредности аритметичких средина и стандардних девијација, израчунатих на основу резултата са пост-теста за студенте експерименталне и контролне групе, приказане су у Табели 5.3. Табела је структурирана попут Табеле 5.1. У првој и другој колони дати су редом називи група и њихова бројност, док се у трећој и четвртој колони налазе редом подаци о аритметичким срединама освојених бодова и стандардним девијацијама (просечним одступањима од аритметичке средине) за разматране групе.

Применом Студентовог t-теста на резултате приказане у Табели 5.3 добија се вредност параметра $p = 0.042$. Упоређујући резултате група формираних предложеним VNS приступом (експериментална група) са резултатима група формираних традиционалним начинима - методом случајног избора или методом студентовог слободног избора (контролна група), може се запазити да је добијена вредност p нижа од задате вредности $\alpha = 0.05$ што говори у прилог томе да се са 95% сигурношћу може тврдити да се нулта хипотеза одбацује, односно да се алтернативна хипотеза не одбацује (видети одељак 5.4.2). Другим речима, аритметичке средине броја бодова са пост-теста двеју група се значајно разликују (за 4.88 поена) у корист предложеног приступа. У односу на стандардну девијацију узорака, резултати указују на то да се VNS методом може добити ниже стандардно одступање од аритметичке средине броја бодова, показујући на тај начин да је VNS метода стабилнија у поређењу са традиционалним методама.

На основу статистичке анализе, може се закључити да се применом предложеног приступа може остварити значајно унапређење академског постигнућа код студената из експерименталне групе, с обзиром да су студенати наведене групе били успешнији у савладавању предвиђених наставних садржаја у поређењу са студентима из контролне групе, што уједно потврђује Хипотезу 5.2. С обзиром на то да статистичка анализа показују и стабилност VNS методе, ова метода се показала као адекватна за формирање група које би подстакле колаборативно учење.

Добијени резултати су у складу са претходним истраживањима описаним у [44, 79, 105, 109, 132, 150, 173, 190], потврђујући да методе које се користе у процесу формирања група имају позитиван утицај на исходе колаборативног учења.

Бољи успех студената експерименталне групе може се дефинисати као директна последица примене приступа заснованог на предложеном VNS алгоритму. За разлику од контролне групе, у експерименталној групи је узето у обзир да групе за колаборативно учење треба да буду „разумно хетерогене” у погледу способности студената [79, 172]. А према [2] хетерогене групе засноване на способности унапређују процес учења код студената свих нивоа способности. Претходне студије [150, 190] су такође показале да хетерогене групе засноване на способности индивидуа карактеришу боље перформансе него групе које су формиране методом случајног избора.

Табела 5.3. Резултати са пост-теста

Групе	Укупан број студената	Аритметичка средина	Стандардна девијација
Експериментална	56	23.96	10.41
Контролна	52	19.08	14.10

Постоји мишљење да групе за колаборативно учење треба да буду формиране од студената који немају ни позитиван ни негативан став једни према другима [169, 185]. У експерименталној групи, десет четворочланих група је испунило овај захтев, док је преосталих четири група формирано од студената који су у пријатељским односима. Алгоритмом је било предвиђено да се избегну преферирајући студенти, међутим тежина тог захтева је за овај експеримент била релативно мала у односу на остале критеријуме, с обзиром да претходна истраживања показују да хетерогеност група која је заснована на способности [2, 150, 190] и избегавање студената који су у лошим међуљудским односима [104, 124, 125, 186] треба да има већи приоритет од овог захтева. Са друге стране, у контролној групи постоје три четворочлане групе које су формиране од студената који немају ни позитиван ни негативан став једни према другима, и десет група формираних на основу студентовог слободног избора. Иако је у експерименталној групи у већој мери испуњен захтев да четворочлане групе не буду формиране од студената који су у пријатељским односима, постоји потреба за додатним истраживањем, како би се утврдило да ли овај захтев значајно утиче на успех колаборативног учења.

Како би се осигурало да добијена расподела студената на најбољи могући начин утиче на процес колаборативног учења, групе су конфигурисане и у односу на вештину просоцијалне прилагодљивости студената. Према [154] формирање хомогених група у односу на ову вештину, смањује могућност за незадовољством перформансама група, недостатком ефикасности сарадње и поделом задатака.

Формирање група за колаборативно учење у односу на наведене карактеристике је имало значајног утицаја на већи академски успех који су остварили студенти експерименталне групе у односу на студенте контролне групе.

6 Проблем минималног репрезентативног скупа

Проблем минималног репрезентативног скупа (Minimum Hitting Set Problem - MHSP) је фундаменталан проблем комбинаторне оптимизације [72] и један од најпознатијих NP-тешких проблема [73]. MHSP представља проблем у оквиру којег је за унапред задату колекцију подскупова датог коначног скупа потребно пронаћи скуп са минималним бројем елемената, такав да сваки од подскупова из колекције садржи бар један елемент нађеног скупа и да не постоји ниједан други подскуп добијеног скупа са истим својством. Проблем MHSP је еквивалентан класичном проблему минималног покривајућег скупа (енгл. Minimum Set Cover Problem - MSCP), па се позитивни и негативни апроксимативни резултати за MHSP могу извести директно из резултата класичног MSCP проблема [8, 25, 90]. MSCP представља проблем код којег је за дати „универзални“ коначни скуп U и његову колекцију T подскупова чија је унија једнака U, потребно пронаћи најмању подколекцију $T' \subseteq T$ чија је унија скупова једнака U.

6.1 Формулација проблема MHSP

Нека је дат коначан скуп $P = \{x_1, \dots, x_m\}$ и колекција C његових непразних подскупова $C = \{P_{1,\dots,}P_n\}$, где је $P_i \subseteq P$ и i = 1,…,n. Тада се минимални репрезентативни скуп колекције C може дефинисати као скуп $H \subseteq P$ за који важи: $\forall P_i \in C,\ P_i \cap H \neq \emptyset \wedge \neg\exists H' \subset H : P_i \cap H' \neq \emptyset$ за $i = 1, \dots, n$. Наиме, скуп H има непразан пресек са сваким од подскупова P_i и не постоји ниједан подскуп од H који би задовољио наведене услове. За такав скуп H се може рећи да репрезентује све подскупове P_i из колекције C и да мањи од њега (са истим својствима) не постоји. У том случају сваки од подскупа P_i садржи бар по један елемент из скупа H.

Уколико постоји више минималних репрезентативних скупова у C, они формирају колекцију минималних репрезентативних скупова $\Omega = \{H_1, \dots, H_k, \dots, H_{|\Omega|}\}$. Проналажење елемената ове колекције је познато као NP-тежак проблем [73]. Величина репрезентативног скупа једнака је броју елемената у H, односно његовој кардиналности $|H|$. Према томе, проблем MHSP се своди на проблем проналажења репрезентативног скупа минималне кардиналности.

У даљем тексту следе два илустративна примера која приказују карактеристике проблема MHSP.

Пример 6.1. Нека се први скуп P'= $\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$ састоји од десет елемената (m = 10) и четири подскупа (n = 4). Подскупови су задати са: P_1 = $\{1, 3, 4, 9\}$; P_2= $\{2, 3, 4, 6, 8\}$; P_3= $\{1, 2, 3, 5, 6, 7\}$; P_4 = $\{1, 3, 5, 9, 10\}$. Једно од оптималних решења је скуп H= $\{3\}$. Оптимална вредност функције циља је 1, с обзиром да важи: $P_1 \cap P_2 \cap P_3 \cap P_4 = \{3\}$. Према томе, у наведеном случају само један елемент је довољан да репрезентује све подскупове из колекције.

Пример 6.2. Нека је дат други скуп P''=$\{1, 2, 3\}$ који садржи три елемента ($m = 3$) и три подскупа (n = 3). Подскупови су формирани тако да : P_1 = $\{1, 2\}$; P_2 = $\{1, 3\}$; P_3 = $\{2, 3\}$. Једно од оптималних решења је скуп H = $\{1, 2\}$. Оптимална вредност функције циља је 2, с обзиром да важи: $P_1 \cap P_2$=$\{1\}$; $P_1 \cap P_3$= $\{2\}$; $P_2 \cap P_3$= $\{3\}$. Према томе, у овом случају оптимално је узети два елемента за представнике свих подскупова у колекцији.

6.2 Постојећи приступи у решавању проблема MHSP

Бројни оптимизациони и хеуристички приступи, као и више других исцрпљујућих алгоритама за решавање проблема MHSP предложени су у литератури. Као што се може видети у [18], MHSP проблем представља један од кључних проблема у комбинаторици коначних скупова. Познат као NP-тежак проблем, MHSP захтева такозвану „фокусирајућу” хеуристику како би се повећала ефикасност претраге и/или ограничила величина повратног скупа. Такви приступи имају потенцијал да редукују проблем MHSP на комплексност полиномског времена по цену целовитости. Идентификовање минималних репрезентативних скупова дате колекције скупова је проблем који се појављује у многим доменима. Један од таквих је и дијагноза на основу модела (енгл. model-based diagnosis – MBD). Након што је у [166] показано да се дијагнозе могу сагледати као решења MHSP проблема, представљени су бројни покушаји да се овај проблем реши у контексту дијагнозе на основу модела. У [3] предложен је „малог утрошка” апроксимативни MHSP алгоритам назван STACCATO. Овај алгоритам користи хеуристичку функцију преузету из приступа заснованог на статистичком софтверу за локализацију грешака. STACCATO је примењен у контексту дијагностичког проблема заснованог на моделу. У [47, 81, 166, 195] MHSP проблем је решен коришћењем такозваног „дрвета репрезентативних скупова” (енгл. hit-set tree) за чију је имплементацију потребна меморија експоненцијалне величине. У [203] представљен је метод који користи дрво за пребројавање скупова (енгл. set-enumeration trees) како би се добили сви скупови минималног утрошка у контексту дијагнозе на основу модела. Аутори закључују да овај метод има комплексност експоненцијалног времена по питању броја елемената у скуповима.

У [62] представљена је прва целобројно линеарна формулација проблема MHSP, која је коришћена у овом истраживању. У [62, 63] проблем MHSP је представљен као 0/1 проблем целобројног програмирања, при чему се показало да је могуће пронаћи MHS скупови минималне кардиналности са алгоритмом који захтева меморију линеарне величине (док се истовремено може очекивати експоненцијално време за потребна рачунања). Алгоритам Quine-McCluskey представљен у [143, 162], који се користи у оквиру логичке оптимизације (енгл. logic optimization), примењен је у циљу извођења главних импликаната монотоне булеанске функције (дуални проблем MHSP проблему). Међутим, наведени алгоритам има ограничену употребу због своје експоненцијалне комплексности. Алгоритам за имплицитни MHSP проблем представљен је у [26], док је алгоритам гранања и редуковања (енгл. branch-and-reduce) дизајниран за решавање MHSP проблема описан у [171].

Бројни хеуристички приступи предложени су како би се MHSP рачунање учинило приступачније великим системима. У [133] описан је акпроксимативни метод за решавање MHSP проблема коришћењем генетских алгоритама. Предложена функција прилагођености има за циљ проналажење решења минималне кардиналности. Иако се не приказује анализа комплексности времена, претпоставља се да је утрошак лошији од GA који је предложен у овом поглављу. Стохастични алгоритми, анализирани у оквиру задовољења ограничења у [70] и пропозиционог задовољења у [160], примери су приступа који су независни од домена, а користе се за решавање MHSP проблема. Наведени алгоритми су се показали знатно ефикаснијим у поређењу са исцрпљујућим метода.

Као један од основних комбинаторних проблема, MHSP је нашао своју примену и у доменима: Булове алгебре, рачунске биологије и претраге података (енгл. data mining) [72]. Већина алгоритамске литературе је усмерена на проблем проналажења колекције MHS скупова, али постоји и обимна литература у којој су приказани алгоритми за генерисање колекције MHS скупова, било експлицитно било у својству неког сродног проблема, под другим именом. Неки од домена у оквиру којих је генерисање колекције MHS скупова нашло своју примену су: комбинаторика у [58], Булова алгебра у [69, 86], дијагнозе грешака у [3, 24, 134, 161, 166, 195], претраге података у [9, 52] и рачунарска биологија у [85, 188, 189]. Приступи за генерисање колекције MHS скупова се могу поделити у неколико категорија: приступи који се заснивају на итерацији скупа (енгл. set iteration approaches), у којима се почиње од мале подфамилије за коју се налазе сви MHS скупови а затим се итеративно подфамилији додају скупови и уједно врши ажурирање колекције MHS скупова [9, 18, 52, 81, 166, 195]; приступи „завади па владај" (енгл. divide and conquer) у оквиру којих се врши подела улазне фамилије на дисјунктне подфамилије, проналазе њихови MHS скупови одвојено, па се затим они комбинују [3, 24, 69, 134]; MHS изградиви приступи (енгл. MHS buildup approaches) у којима се формирају скупови елемената за које се очекују или се може гарантовати да ће бити подскупови колекције MHS скупова, а затим се итеративно додају елементи све док и они не постану MHS скупови [91, 151]; у потпуности репрезентативни приступи (енгл. full cover) који унапређују приступ „завади па владај" уз помоћ хиперграфске леме која омогућава ефикаснију рекомбинацију [161].

6.3 Математичка формулација проблема MHSP

Формулација целобројног линеарног програмирања (ILP) за решавање MHSP проблема први пут је изложена у [62].

Нека су параметри и променљиве дефинисани на следећи начин:

$$p_{ij} = \begin{cases} 1, \; j \in P_i \\ 0, \; j \notin P_i \end{cases}, \quad j = 1, \ldots, m, \;\; i = 1, \ldots, n; \tag{6.1}$$

$$y_j = \begin{cases} 1, \; j \;\; \text{је} \;\; \text{представник} \\ 0, \; j \;\; \text{није представник} \end{cases}, \;\; j = 1, \ldots, m. \tag{6.2}$$

Параметар p_{ij} узима вредност 1 ако j-ти елемент припада подскупу P_i, у супротном узима вредност 0. Бинарна променљива y_j узима вредност 1 ако је j-ти елемент изабрани представник у минималном гађајућем скупу, у супротном узима вредност 0.

На основу наведене нотације, формулација целобројног линеарног програмирања за проблем MHSP представљена је на следећи начин:

$$min \sum_{j=1}^{m} y_j \tag{6.3}$$

при ограничењима :

$$\sum_{j=1}^{m} p_{ij} y_j \geq 1, \qquad i = 1, \ldots, n; \tag{6.4}$$

$$y_j \;\; \in \{0, 1\}, \qquad j = 1, \ldots, m. \tag{6.5}$$

Функција циља (6.3) минимизује укупни број елемената који представљају све подскупове P_j колекције C. Ограничење (6.4) обезбеђује да сваки подскуп P_j има бар по једног представника у минималном репрезентативнем скупу. Бинарна природа променљиве y_j одређена је условом (6.5).

6.4 Генетски алгоритам за решавање проблема MHSP

С обзиром да MHSP представља NP-тежак проблем комбинаторне оптимизације, где величина простора решења експоненцијално расте са димензијом проблема, уобичајени начин да се ова врста проблема реши је примена неке од метахеуристичких метода. Генетски алгоритми представљају механизме претраге који могу брзо да лоцирају области високих перформанси веома великих и комплексних простора за претрагу. С друге стране, генетски алгоритми се могу конципирати тако да омогуће и ефикасну локалну претрагу. Из наведених разлога, развијен је генетски алгоритам у оквиру којег је имплементирана хеуристика локалне претраге. Хеуристика је имплементирана са циљем побољшања квалитета решења проблема. Резултати представљени у овом одељку су презентовани у раду [43].

Основна структура GA са LS представљена је Алгоритмом 11. Хеуристика локалне претраге се примењује на сваку јединку i популације P (у свакој од генерација), након рачунања функције циља $obj[i]$, а пре примене генетских оператора.

Алгоритам 11: Структура GA са LS за решавање проблема MHSP

Input: D, N_{pop}

Output: solution P_{best}

P ← *GenerateInitialPopulation* (N_{pop});
SelectStopCondition ( );
while *StopCondition ( ) is not satisfied* ***do***
    for i*=1* ***to*** N_{pop} ***do***
        $obj[i]$ ← *CalculateObjectiveFunction* (P_i);
        HeuristicLocalSearch (P_i);
    endfor
    CalculateFitnessFunction (P);
    Selection (P);
    Crossover (P);
    Mutation (P);
endwhile
P_{best} ← $Best(P)$;

6.4.1 Кодирање јединки

Кодирање јединки је бинарно, при чему се свако од потенцијалних решења (кандидат за минимални репрезентативни скуп H) представља бинарним кодом дужине m. Број битова у генетском коду јединке једнак је броју елемената скупа који се посматра. Ген чија је вредност 1 на j-тој позицији кода означава да елемент j припада репрезентативном скупу, док вредност 0 показује супротно. Дефинисано на овај начин, лако се може проверити за сваки подскуп да ли се у њему налазе елементи који га представљају, или не.

Пример 6.3. Нека је једно од решења репрезентовано са: 10001001. У складу са дефинисаним начином кодирања решења, важило би: $y_1 = y_5 = y_8 = 1$ и $y_2 = y_3 = y_4 = y_6 = y_7 = 0$. Према томе, елементи $\{y_1, y_5, y_8\}$ би били представници иницијалног скупа и кандидати за елементе MHS.

6.4.2 Функција циља и хеуристика локалне претраге

Вредност функције циља једног решења једнака је укупном броју елемената-представника, односно кардиналности репрезентативнег скупа. Потребно да сваки од подскупова има свог представника у репрезентативнем скупу. У складу са формулацијом проблема MHSP, потребно је минимизовати укупан број представника. У циљу побољшања решења, у оквиру којег неки од подскупова могу бити без представника или их могу имати више него што је потребно, примењује се хеуристика локалне претраге, којом се врши убацивање „потребног", односно брисање „непотребног" представника.

Функција циља и хеуристика локалне претраге представљен је Алгоритмом 12. У првом кораку алгоритма, укупан број представника *numR* се поставља на вредност 0. Затим се у петљи која се извршава у корацима 2-4, а у оквиру које се рачуна укупан број представника, пролази кроз све елементе i (i=1,..,*ne*) иницијалног скупа и за сваки од њих проверава да ли јесте или није представник. Ознака $y[i]$ се користи за низ свих елемента, где вредност 1 указује да је i-ти елемент представник, док 0 показује супротно.

Локална претрага се извршава у корацима 5-45. У првом делу алгоритма, који се извршава у оквиру *for* петље (кораци 5-15), убацује се по један представник у подскупове без представника. Пролази се редом кроз све елементе *j* (*j=0,..,sen*[*k*]-1) сваког од подскупова *k* (*k=1,..,ns*) и проверава се да ли има представника. Уколико се пронађе бар један представник *y*[*SE*(*k*,*j*)], прелази се на испитивање следећег подскупа. У случају да се претрагом дошло до последњег елемента *j*=*sen*[*k*] подскупа и у њему се није пронашао ниједан представник (*not y*[*SE*(*k*,*j*)]), у корацима 11-12 се на случајан начин бира један његов елемент и поставља за представника. У наредном кораку следи увећање бројача представника.

У другом делу хеуристике локалне претраге, која се извршава у оквиру *while* петље (кораци 17-45), врши се избацивање по једног вишак представника из подскупова који поред посматраног *i*-тог представника имају бар још једног представника, односно имају укупно бар два представника. У кораку 16, вредност логичке променљиве *improvement* (побољшање) се поставља на вредност *true* (тачно) и док год има побољшања, извршава се *while* петља. (Претпоставка је да постоји представник који је вишак, и да ће се његовим избацивањем извршити побољшање решења.) У оквиру *while* петље пролази се по свим елементима *i* иницијалног скупа и за сваки од њих се проверава да ли јесте представник $y[i]$. Уколико је *i* представник, алгоритам се наставља извршавати у корацима 22-42, на начин да се даље проверава да ли подскупови који садрже посматрани представник имају бар два представника. У супротном, уколико *i* није представник, прелази се на корак 43, односно на проверу следећег елемента.

Нова логичка променљива *twoR* којом се одређује да ли постоје два представника, уводи се у кораку 22. Затим се са петљом *for*, која се извршава у корацима 23-36, пролази по свим подскуповима *j* (j=1,..,$esn[i]$) који садрже *i*-тог представника. (Ако би се пролазило кроз све подскупове, успорио би се процес извршења GA.) У новој *while* петљи, која се извршава у корацима 27-32, пролази по свим елементима *k* (k=0,..,$sen[q]$-1) поскупа *q* ($q = ES(i,j)$ док год је број његових представника $r<2$. Испитивање да ли је *k*–ти елемент представник скупа *q*, врши се у кораку 28.

У случају да се проласком кроз све елементе скупа *q* не пронађу два представника, посматрани *i*-ти представник се не може брисати и петља *for* (којом се пролази кроз све скупове који имају *i*-тог представника) се завршава. У супротном, уколико сви подскупови *q* (који садрже посматраног представника) имају бар по два представника, односно уколико важи услов из корака 37, у наредном кораку се врши брисање посматраног представника из репрезентативнег скупа, тачније *i*-ти елемент више није представник ($y[i]$=0). Сходно претходном, бројач представника се смањује за један у кораку 39. Једном када се промена изврши, LS наставља да тражи следећег представника чијом би се променом поправило решење.

Алгоритам 12: Рачунање функције циља и хеуристика LS за проблем MSSP

```
Output: solution NumR
/* функција циља
numR ←0;
for i =1 to ne do
    numR ← numR+y[i] ;
endfor

/* локална претрага
for k=1 to ns do
    j← 0;
    while (j<sen[k]) and (not y[ SE(k,j)]) do
        j←j+1;
    endwhile
    if (j=sen[k]) then
        i ← GetRandom(sen[k]);
        y[i] ← 1;
        numR ← numR+1;
    endif
endfor
improvement ← true;
while (improvement) do
    improvement ← false;
    i← 0;
    while (not improvement) and (i<ne) do
      if (y[i]) then
        twoR ← true;
        for j=1 to esn[i] do
            r ← 0;
            q = ES(i,j);
            k ← 0;
            while (k<sen[q]) and (r<2) do
                if (y[ SE(q,k) ]) then
                        r ← r+1;
                endif
                k ← k+1;
            endwhile
            if (r<2) then
                twoR ← false;
            endif
        endfor
        if (twoR) then
            y[i] ← 0;
            numR ← numR-1;
            improvement ← true;
        endif
      endif
      i←i+1;
    endwhile
endwhile
```

6.4.3 Генетски оператори и остале карактеристике GA

У предложеној GA имплементацији, прва генерација јединки је генерисана на случајан начин. У свакој генерацији, популација броји N_{pop}= 150 јединки. Функција прилагођености (фитнес) f_{ind} јединке ind ($ind = 1,..,N_{pop}$) рачуна се скалирањем вредности функције циља obj_{ind} у интервал [0,1]. Како је MHSP проблем минимизације, вредности јединки се инверзно скалирају у јединични интервал, па се за скалирање користи формула:

$$f_{ind} = \frac{obj_{ind_{max}} - obj_{ind}}{obj_{ind_{max}} - obj_{ind_{min}}}, \tag{6.6}$$

где су obj_{ind}, $obj_{ind_{min}}$ и $obj_{ind_{max}}$ редом текућа, минимална и максимална вредност функције циља јединки у популацији. Након одређивања прилагођености сваке јединке, јединке се распоређују у односу на нерастући низ њихове прилагођености: $f_1 \geq f_2 \geq \ldots \geq f_{N_{pop}}$. Најбоља јединка ind_{min} (са најмањом вредности функције циља) добија прилагођеност 1, док најлошија јединка ind_{max} (са највећом вредности функције циља) добија прилагођеност 0.

Да би се добили задовољавајући резултати, у предложеној GA имплементацији примењују се стратегије описане у одељку 2.4.2. Елитистичком стратегијом обезбеђује се 100 елитних и 50 неелитних јединки. У циљу спречавања доминације елитних јединки у популацији, њихова прилагођеност се умањује применом формуле (2.17).

Могућност вишеструког појављивања јединки са истим генетским кодом се отклања подешавањем вредности њихове прилагођености на нулу, изузев у случају првог појављивања. Укупан број јединки са истом вредношћу функције циља а различитим генетским кодом, у свакој генерацији се ограничава константом N_{rv}= 40.

У циљу оптимизовања времена извршења GA, користи се техника кеширања описана у одељку 2.4.6. Максималан број јединки које се заједно са вредностима функције циља могу сместити у хеш-табелу ограничен је на N_{cache}= 5000.

Оператор селекције, који се користи у предложеној имплементацији, је Фино градирана турнирска селекција FGTS, описана у одељку 2.4.3 и 2.5. Реалном параметру kоји означава просечну величину турнира F_{tour} додељује се вредност

5.4. За N_{nnel} = 50 неелитних јединки, одговарајуће константе k_1 и k_2 узимају редом вредности 20 и 30.

За укрштање насумице одабраних $\left\lfloor \frac{N_{nnel}}{2} \right\rfloor$ парова неелитних јединки, користи се оператор укрштања у једној тачки (описан у одељку 1.3.1) са вероватноћом укрштања p_{cross}= 0.85 (одељак 2.4.4 и 2.5). Наведена вероватноћа подразумева размену генетског материјала код оквирно 85% јединки, док се код преосталих 15% јединки не очекује укрштање већ стварање потомака идентичних родитељима.

У циљу спречавања превремене конвергенције и губљења подобласти простора претраге, оператор мутације се модификује на начин да се стопа мутације увећава код сваког залеђеног гена (одељак 1.3.1 и 2.4.5). У складу са наведеним, узето је да је стопа мутације (која зависи од димензије n) буде 2.5 пута већа ($p_{mut} = \frac{1.0}{n}$) на залеђеним у односу на ниво мутације ($p_{mut} = \frac{0.4}{n}$) код незалеђених гена. Након сваке мутације, врши се провера броја јединица у сваком од подскупова, а затим се, на местима где је то потребно, врше принудне мутације, с циљем формирања валидних јединки.

6.5 Експериментални резултати

У овом одељку приказани су резултати потврде коректности и ефикасности предложеног GA алгоритма кроз скуп рачунарских тестирања. Тестирања су спроведена на тест примерима минималног репрезентативнег скупа (Minimum Hitting Set – MHS instances) преузетих из [40] и детаљније приказаних у одељку 2.5. Сва тестирања су извршена на PC рачунару са једним процесорским језгром од 2.5 GHz и 1 GB RAM меморије под Windows оперативним системом. GA алгоритам је кодиран у програмском језику C. Експериментални резултати приказани у овом одељку су представљени у раду [43].

Како би се утврдила ефикасност и поузданост предложене GA имплементације, извршено је поређење најбољих резултата добијених овим алгоритмом са најбољим познатим резултатима и са резултатима добијеним егзактним ILOG CPLEX 10.1 решавачем. Тестирања изведена са CPLEX решавачем спроведена су у циљу добијања оптималних решења за проблем MHSP на MHS тест примерима.

Резултати тестирања добијени применом CPLEX решавача и GA алгоритма на MHS тест примерима приказани су у Табели 6.1. Табела је структуирана тако да се у првој и другој колони налазе подаци о броју елемената m и броју подскупова n за сваки од десет тест примера. Најбоља позната решења за све тест примере (преузета из [25]) су приказана у трећој колони табеле. У наредне четири колоне налазе се подаци који приказују резултате добијене CPLEX решавачем. Четврта колона даје приказ добијених оптималних вредности функције циља f_{opt}, уз које стоји ознака „*опт*". Вредности функције циља најбољих добијених решења f_{best} за преосталих седам тест примера приказане су у петој колони, при чему се ознака „*н.п.*" користи у случајевима где је CPLEX достигао најбоље познато решење. Шеста колона под називом „LB" се односи на доњу границу решења и њене вредности су приказане у случајевима где CPLEX није успео да добије оптимална решења. За сваки од тест примера, време извршења CPLEX-а (изражено у секундама) приказано је у седмој колони. Последње три колоне Табеле 6.1 садрже информације о резултатима који су постигнути применом имплементираног GA. У оквиру осме колоне приказане су најбоље вредности функције циља добијене GA, при чему се ознаке „*опт*" и „*н.п.*" користе у већ претходно наведену сврху. Просечно време (изражено у секундама) потребно да GA достигне своје најбоље решење је приказано у деветој колони, док се последња колона односи на укупан број генерација N_{gen} кроз које је GA мор да прође да би добио одговарајуће решење.

Табела 6.1: Поређења CPLEX и GA резултата на MHS тест примерима

m	n	$f_{bestknown}$	CPLEX				G A		
			f_{opt}	f_{best}	LB	$t\ (s)$	f_{best}	$t\ (s)$	N_{gen}
50	1000	*ново* 6	6 *опт*			33.64	*опт*	3.239	2304
50	10000	10	9 *опт*			25783.36	*н.п.*	77.146	2608
100	1000	6	6 *опт*			866.281	*опт*	5.778	2767
100	10000	9		*н.п.*	5	59140	*н.п.*	210.767	2830
100	50000	39		40	27	78836.44	40	694.656	3619
250	1000	10		*н.п.*	6	62294.42	11	10.571	3679
250	10000	12		*н.п.*	5	61562.39	13	426.919	3492
500	1000	10		*н.п.*	5	44749.72	11	17.31	3832
500	10000	16		17	4	44322.61	17	550.356	3914
500	50000	21		*н.п.*	5	12953.05	*н.п.*	2657.596	3796

На основу резултата приказаних у Табели 6.1 може се закључити да је CPLEX решавач, тестиран на ILP формулацији проблема MHSP, погодан за проналажење оптималних вредности функције циља за решења добијена на MHS тест примерима мањих димензија, с обзиром да за прва три примера добија оптималне вредности функције циља. За преосталих седам примера, CPLEX решавач проналази решења чија оптималност није доказана у задатом временском ограничењу. За примере: (m, n) = {(100, 10000); (250, 1000); (250, 10000); (500, 1000); (500, 50000)} CPLEX достиже најбоље познате вредности функције циља, док за два примера (m, n) = {(100, 50000); (500,10000)} добијене вредности функције циља се разликују за 1 од најбољих познатих вредности и веће су од добијених доњих граница. С друге стране, резултати приказани у Табели 6.1 показују да предложени GA алгоритам достиже у веома кратком временском року оптимална решења за примере: (m, n) = {(50, 1000); (100, 1000)}, а најбоља позната решења за примере (m, n) = {(50, 10000); (100, 10000); (500, 50000). У преосталим случајевима најбоље добијене вредности функције циља су веома близу најбољим познатим (разликују се за 1) и достигнуте су у веома кратком временском року. За први пример (m, n) = (50, 1000) GA успева да у року од 3,239 секунде пронађе ново најбоље познато и оптимално решење, са функцијом циља чија је вредност 6, уместо дотадашње најбоље познате вредности 7.

Поређењем експерименталних резултата приказаних у Табели 6.1, лако се може уочити подесност примене алгоритамских компоненти за решавање проблема MHSP. Сагледавајући добијена GA решења, димензије тест примера и време извршења које је прилично кратко, знатно краће у поређењу са CPLEX решавачем, може се закључити да се предложеним GA приступом достижу веома добри резултати. Са циљем валидне процене поузданости GA перформанси, алгоритам је извршаван 20 пута за сваки тест пример (изузев за највећи). Пример највеће димензије тестиран је 10 пута због релативно дуготрајног процеса израчунавања функције циља. За критеријуме заустављања GA узети су: максималан број генерација N_{gen}= 5000 и понављање непромењене најбоље вредности функције циља кроз N_{rep}= 2000 узастопних генерација.

Примери оптималних решења и њихове вредности функције циља за проблем MHSP, које је CPLEX добио за прва три MHS тест примера: (m, n) = {(50, 1000);

(50, 10000); (100, 1000)}, приказани су у Табели 6.2. Прве две колоне табеле дају приказ броја елемената m и броја подскупова n, док су наредне две колоне резервисане за приказ оптималних вредности функције циља f_{opt} и примере оптималних решења проблема MHSP.

Табела 6.2: CPLEX оптимална решења за MHSP на MHS тест примерима

m	n	CPLEX	
		f_{opt}	оптимална решења
50	1000	6	6 9 23 34 37 40
50	10000	9	1 3 4 10 21 24 29 33 35
100	1000	6	28 57 59 65 67 76

6.6 Примена MHSP приступа у опште и образовне сврхе

Репрезентативне групе налазе своју примену у образовним институцијама, али и у другим друштвеним сегментима, попут процеса испитивања јавног мњења. Само формирање репрезентативног скупа подразумева селектовање представника сваке од циљних група које се дефинишу сходно унапред задатим критеријумима, са крајњим циљем издвајања најмањег броја индивидуа које се истовремено могу третирати као представници што већег броја посматраних група. У образовним институцијама, конкретна примена MHSP приступа би се могла наћи у спровођењу експерименталних истраживања чији би циљ био унапређење извођења наставе, у процесу састављања тестова, али и за формирање различитих репрезентативних група студената, међу којима је и избор чланова студентског парламента.

7 Закључак

У овом раду су представљене метахеуристичке методе за решавање NP-тешких проблема комбинаторне оптимизације: MSSP, WBECGFP, BMWASP, CLGFP и MHSP. За решавање проблема MSSP и MHSP развијене су варијанте генетских алгоритама, док је за решавање проблема WBECGFP, BMWASP и CLGFP коришћена метода променљивих околина посебно прилагођена сваком од проблема.

За проблем MSSP развијен је модел мешовитог целобројног линеарног програмирања (број променљивих и ограничења је релативно мали у поређењу са димензијом проблема) и дат је детаљан доказ коректности формулације. Модел је примењен за проналажење оптималних решења тест примера мањих димензија применом егзактног решавача ILOG CPLEX 10.1. Предложене су и математичке формулације проблема WBECGFP, BMWASP и CLGFP. За проблеме WBECGFP и BMWASP представљене су њихове нелинеарне математичке формулације, док је за проблем BMWASP додатно предложена и његова мешовита целобројно линеарна реформулација BMWASP-I, која је развијена у циљу оптималног решавања тест примера проблема већих димензија. Резултати експерименталне студије, у оквиру које је тестирано 300 примера, показали су да је BMWASP-I формулација супериорнија у односу на BMWASP формулацију. У циљу представљања једноставне и брзе методе за рачунање доњих граница развијена је и линеарна релаксација предложене BMWASP-I формулације. Адекватност предложених математичких формулација проблема WBECGFP, BMWASP и CLGFP, потврђена је њиховом практичном применом у реалним ситуацијама кроз експерименте који су изведени над различитим групама студената Београдске пословне школе, у оквиру наставе математике и статистике.

Варијанта генетског алгоритма имплементирана за решавање проблема MSSP користи: бинарно кодирање јединки, ограничени број различитих јединки са истом вредношћу функције циља, унапређени оператор фино градиране турнирске селекције, укрштање у једној тачки, унапређену мутацију са концептом залеђеним гена и технику кеширања која знатно убрзава процес рачунања функције циља. За проблем MHSP развијена је нова варијанта генетског алгоритма, која је додатно унапређена имплементацијом хеуристике локалне претраге. Метода локалне претраге примењује се након рачунања функције циља како би се убацио „потребан“, односно избрисао „непотребан“ представник у посматраном решењу, у циљу добијања квалитетнијих решења пре примене генетских оператора. Ефикасност предложених генетских алгоритама тестирана је на тест примерима преузетим из литературе.

У оквиру методе променљивих околина посебно развијене за проблеме WBECGFP, BMWASP и CLGFP, користи се бинарно кодирање. За сваки проблем дефинишу се њему прилагођене специфичне структуре околина. За иницијално решење основног VNS алгоритма узима се решење добијено RVNS алгоритмом (како би се на брз и ефикасан начин добило добро и изводљиво решење), док се за иницијално решење RVNS узима случајна расподела студената у проблемски дефинисан број група.

За сваки од проблема WBECGFP, BMWASP и CLGFP развијена је унапређена и ефикасна метода локалне претраге, којом свеукупна претрага добија на квалитету а самим тим се остварује и унапређење методе VNS, констрисане за њихово решавање. У фази унапређене локалне претраге користи се стратегија брзе замене по једног пара индивидуа које припадају различитим групама и тако се на брз и ефикасан начин добијају побољшања тренутног решења. Стратегија најбољег побољшања којом се од свих побољшања пронађених у околини тренутног решења бира најбоље решење, је стратегија на којој се заснива процедура локалне претраге.

За проблеме WBECGFP и BMWASP, унапређење локалне претраге се огледа у побољшању временске сложености рачунања функције циља у односу на општи случај. Наиме, за свако новодобијено решење у оквиру функције замене индивидуа, функција циља не мора да се рачуна од почетка. Довољно је за сваку од карактеристика (прескачу се карактеристике који не утичу на вредност функције циља, односно кактеристике које обе индивидуе или имају или немају) најпре одузети њен тренутни допринос од вредности функције циља тренутног решења, а затим (након изршене замене индивидуа) на вредност функције циља додати нови допринос посматране карактеристике.

Временска сложеност рачунања функције циља код проблема CLGFP је такође побољшана кроз функцију замене индивидуа. Приликом рачунања функције циља новодобијеног решења, није потребно за сваку од формираних група рачунати број нерешених захтева, већ је довољно израчунати нерешене захтеве само за две групе које су учествовале у замени индивидуа и на основу тога ажурирати вредност функције циља новог решења (наћи нови максимални број нерешених захтева).

Поузданост и ефикасност предложених варијанти VNS алгоритама је тестирана на различитим димензијама тест примера проблема узимајући у обзир реалне податке. У том погледу, експериментални резултати су показали да VNS алгоритам развијен за проблем WBECGFP достиже квалитетније и балансираније расподеле студената у две групе у односу на расподеле добијене осталим разматраним метода, и то у кратком временском року. За потребе тестирања VNS алгоритма развијеног за проблем BMWASP генерисано је 300 јавно доступних тест примера (од малих до великих димензија). Резултати тестирања су показали да VNS алгоритам за BMWASP достиже сва оптимална решења на тест примерима мањих димензија која су добијена егзактним решавачима ILOG CPLEX 12.5, уједно пружајући решења високог квалитета за тест примере великих димензија у кратком временском року извршења. У циљу процене ефикасности VNS алгоритма предложеног за проблем CLGFP, спроведена су тестирања на различитим димензијама тест примера генерисаних на основу реалних података. Поређење је извршено у односу на Монте Карло методу и анализом перформанси алгоритама показано је да VNS даје боље резултате у кратком временском року извршења. За велике тест примере, поред фактора уштеде времена, добијена решења имала су мали број неиспуњених захтева.

Са методичког становишта, предложени MSSP, WBECGFP, BMWASP, CLGFP и MHSP приступи могу наћи своју непосредну примену на свим образовним инстанцама, широког су дијапазона перспективе и могу бити од помоћи едукатору на плану педагошке имплементације. Постоји флексибилност у избору и броју карактеристика, избору приоритета међу карактеристика и у броју индивидуа које могу бити обухваћене процесом распоређивања, а све у циљу задовољења педагошких и истраживачких потреба. WBECGFP и BMWASP приступи су посебно примењиви у случајевима где је потребно узети у обзир велики број индивидуа, њихових карактеристика и могућност коришћења приоритета међу карактеристикама, и где треба ефикасно распоредити индивидуе у две или више балансираних група. Могућност коришћења CLGFP приступа у процесу формирања четворочланих група за колаборативно учење, може бити од велике користи за студенте и наставнике у ситуацијама када је потребно распоредити велики број студената у мање групе на основу више карактеристика студената. Предложени MHSP приступ се може користити у сврху формирања различитих репрезентативних група студената, састављања разних тестова, избора чланова студентског парламента. С друге стране, MSSP приступ може се применити у случајевима када је потребно извршити расподелу већег броја индивидуа у две дисјунктне групе, на начин да се у свакој од те две групе налази бар по једна индивидуа као представник мање групе којој иницијално припада (за расподелу предавача на два семестра, поделу наставних јединица на два семестра, коришћење различитих наставних метода у оба семестра, формулисање тестова за две групе на основу питања која покривају све обрађене тематске целине).

7.1 Научни допринос рада

Научни допринос ове дисертације представљају следећи најважнији резултати:

- Нова формулација мешовитог целобројног линеарног програмирања за проблем MSSP.
- Нелинеарна математичка формулација проблема WBECGFP.
- Нелинеарна математичка формулација проблема BMWASP.
- Мешовита целобројно линеарна реформулација проблема BMWASP.
- Линеарна релаксација проблема BMWASP.
- Математичка формулација проблема CLGFP.
- Развој генетских алгоритама за проблеме MSSP и MHSP.
- Имплементација хеуристике локалне претраге у генетски алгоритам, с циљем побољшања квалитета решења проблема MHSP.
- Примена метахеуристичке методе промењљивих околина за решавање проблема WBECGFP, BMWASP и CLGFP, посебно прилагођене сваком од наведених проблема.

- Генерисање 300 јавно доступних тест примера за проблем BMWASP.
- Нови методички приступ формирању експерименталне и контролне групе, уз могућност коришћења великог броја карактеристика (у конкретном случају преко 100) у својству критеријума за балансирано разврставање индивидуа у групе.
- Нови методички приступ формирању *к* добро балансираних групa, који омогућава разматрање бројних индивидуа, њихових карактеристика (у конкретном случају преко 100) и листу приоритета међу карактеристикама.
- Нови методички приступ распоређивању индивидуа у четворочлане групе за колаборативно учење, који узима у обзир ниво претходног академског знања, међуљудске односе и вештину социјалне прилагодљивости студента.
- Креирање јавно доступних апликација намењених за решавање проблема WBECGFP, BMWASP и CLGFP, које као резултат дају поделе на одговарајуће групе.
- Педагошки допринос процесу организовања и извођења наставе, и процесу учења.

Из свега претходно наведеног, може се закључити да се развијени приступи могу успешно применити за решавање описаних и њима сличних проблема. Истраживање приказано у овом раду даје допринос областима комбинаторне оптимизације и посебно метахеуристичким методама, као и педагошки допринос целокупном процесу организовања и извођења наставе, и процесу дељења и усвајања нових знања кроз групни облик рада.

Литература

[1] S. Abnar, F. Orooji, F. Taghiyareh, *An evolutionary algorithm for forming mixed groups of learners in web based collaborative learning environments*, Proceedings of the 2012 IEEE international conference on technology enhanced education Kerala, India: IEEE. F, (2012), 1–6.

[2] P.C. Abrami, B. Chambers, C. Poulsen, C. De Simone, S. d'Apollonia, J. Howden, *Classroom connections: understanding and using cooperative learning*, Toronto: Harcourt Brace & Company, 1995.

[3] R. Abreu, A. J. C. Van Gemund, *A Low-Cost Approximate Minimal Hitting Set Algorithm and its Application to Model-Based Diagnosis*, Proceedings of the 8th Symposium on Abstraction, Reformulation and Approximation (SARA09), Lake Arrowhead, CA, USA, 2009.

[4] E. Alfonseca, R. M. Carro, E. Martin, A. Ortigosa, P. Paredes, *The impact of learning styles on student grouping for collaborative learning: a case study*, User Modeling and User-Adapted Interaction, 16 (3-4) (2006), 377-401.

[5] G. Andersson, L. Engebretsen, *Better approximation algorithms for set splitting and not-allequal sat*, Information Processing Letters, 65 (1998), 305–311.

[6] Z. C. Ani, A Yasin, M. Z. Husin, Z. A. Hamid, *A method for group formation using genetic algorithm*, International Journal on Computer Science and Engineering, 2(9) (2010), 3060–3064.

[7] D. Ary, L. Jacobs, C. Sorensen, D. Walker, *Introduction to Research in Education*, 1972.

[8] G. Ausiello, A. Datri, M. Protasi, *Structure preserving reductions among convex optimization problems*, Journal of Computer and System Sciences, 21(1) (1980), 136-153.

[9] J. Bailey, T. Manoukian, K. Ramamohanarao, *A fast algorithm for computing hypergraph transversals and its application in mining emerging patterns*, Proceedings of the Third IEEE International Conference on Data Mining, IEEE, (2003) p. 485.

[10] B. Baker, C. Benn, *Assigning pupils to tutor groups in a comprehensive school*, Journal of the Operational Research Society, 52(6) (2001), 623–629.

[11] K.R. Baker, S.G. Powell, *Methods for assigning students to groups: a study of alternative objective functions*, Journal of the Operational Research Society, 53(4) (2002), 397–404.

[12] J. M. Balmaceda, S. Schiaffino, J. A. Diaz-Pace, *Using constraint satisfaction to aid group formation in CSCL*, Inteligencia Artificial, 17(53) (2014), 35-45.

[13] A. Bandura, *Self-efficacy: the exercise of control*, New York: W.H. Freeman and Company, 1997.

[14] M. Beheshtian-Ardekani, M. A. Mahmood, *Education development and validation of a tool for assigning students to groups for class projects*, Decision Sciences, 17(1) (1986), 92–113.

[15] R. Bekele, *Computer-assisted learner group formation based on personality traits*, Ph.D. Dissertation, University of Hamburg, 2005.

[16] R. M. Belbin, *Management teams: Way they succeed or fail*, Oxford: Butterworth-Heineman, 1981.

[17] R. M. Belbin, *Team roles at work*, Oxford: Butterwoth-Heinemann, 1993.

[18] C. Berge, *Hypergraphs: Combinatorics of finite sets*, Amsterdam: Elsevier-North Holland, The Netherlands, 1989.

[19] J. Bhadury, E. J. Mighty, H. Damar, *Maximizing workforce diversity in project teams: A network flow approach*, Omega, 28(2) (2000), 143–153.

[20] C. Blum, A. Roli, *Metaheuristics in combinatorial optimization: Overview and conceptual comparison*, ACM Computing Surveys, 35(3) (2003), 268-308.

[21] J. H Bradley, F. J. Herbert, *The effect of personality type on team performance, Journal of Management Development*, 16 (1997), 337-353.

[22] J. Brimberg, S. Janićijević, N. Mladenović, D. Urošević, *Solving the clique partitioning problem as a maximally diverse grouping problem*, Optimization Letters, 11(6) (2017), 1123–1135.

[23] J. Brimberg, N. Mladenović, D. Urošević, *Solving the maximally diverse grouping problem by skewed general variable neighborhood search*, Information Sciences, 295 (2015), 650–675.

[24] N. Cardoso, R. Abreu, *Mhs2: A map-reduce heuristic-driven minimal hitting set search algorithm*, in: J. A. M. Lourenãgo, E. Farchi (Eds.), Multicore Software Engineering, Performance and Tools, Lecture Notes in Computer Science, 8063 (2013), 25–36.

[25] T. M. Chan, E. Grant, *Exact algorithms and APX-hardness results for geometric packing and covering problems*, Computational Geometry, 47(2) (2014), 112-124.

[26] K. Chandrasekaran, R. Karp, E. Moreno-Centeno, S. Vempala, *Algorithms for implicit hitting set problems*, Proceedings of the twenty-second annual ACM-SIAM symposium on Discrete Algorithms, 2011, 614-629.

[27] W. L. Chang, M. Guo, M. Ho. *Towards solution of the set-splitting problem on gel-based DNA computing*, Future Generation Computer System, 20(5) (2004), 875-885.

[28] J. Chen, S. Lu, *Improved algorithm for weighted and unweighted set splitting problems*, Lecture Notes in Computer Science, 4598 (2007), 573–547.

[29] H. Chen, S. Lu, *Improved parameterized set splitting algorithms: A probabilistic approach*, Algorithmica 54 (2009), 472–489.

[30] T. C. E. Cheng, *Operations research and higher education administration*, Journal of Educational Administration, 31(1) (1993), 77–90.

[31] E. G. Cohen, *Designing groupwork: Strategies for the heterogeneous classroom*, New York: Teachers College Press, 1986.

[32] E. G. Cohen, *Restructuring the classroom: conditions for productive small* group, Review of Education Research, 64(1) (1994a), 1–35.

[33] S. G. Cohen, D. E. Bailey, *What makes teams work: Group effectiveness research from the shop floor to the executive suite*, Journal of Management, 23 (1997), 239–290.

[34] E. G. Cohen, R. A. Lotan, *Working for equity in heterogeneous classrooms: sociological theory in practice*, New York: Teachers College Press, 1997.

[35] R. Cordero, N. Di Tomaso, G. F. Farris, *Gender and race/ethnic composition of technical workgroup: relationship to creative productivity and morale*, Journal of Engineering and Technology Management, 13 (1996), 205–221.

[36] R. M. Cormack, *A review of classification*, Journal of the Royal Statistical Society, A (General) (1971), 321–367.

[37] P. T. Jr Costa, R. R. McCrae, *Revised NEO personality inventory and NEO five-factor inventory professional manual*, Psychological Assessment Resources, Odessa, 1992.

[38] LJ. Cronbach, *Coefficient alpha and the internal structure of tests*, Psychometrika, 16(3) (1951), 297–334.

[39] W. Cruz, S. Isotani, *Group formation algorithms in collaborative learning contexts: A systematic mapping of the literature*, Collaboration and Technology, (2014), 199-214.

[40] V. Cutello, G. Nicosia, *A clonal selection algorithm for coloring, hitting set and satisfiability problems*, Lecture Notes in Computer Science, 3931 (2006), 324-337.

[41] R. Cutshall , S. Gavirneni , K. Schultz, *Indiana University's Kelley School of Business uses integer programming to form equitable, cohesive student teams*, Interfaces, 37(3) (2007), 265–276.

[42] M. Čangalović, Đ. Dugošija, V. Kovačević-Vujčić, S. Simić, J. Vuleta, *Kombinatorna optimizacija: matematička teorija i algoritmi*, Društvo operacionih istraživača Jugoslavije, 1996.

[43] B. L. Ćendić, *A genetic algorithm for the minimum hitting set*, Scientific Publications of the State University of Novi Pazar, 6(2) (2014), 107-117.

[44] M.-I. Dascalu, C.-N. Bodea, M. Lytras, P. O. De Pablos, A. Burlacu, *Improving e-learning communities through optimal composition of multidisciplinary learning groups*, Computers in Human Behavior, 30 (2014), 362–371.

[45] F. Dehne, M. Fellows, F. Rosamond, *An FPT algorithm for set splitting*, Lecture Notes in Computer Science, 2880 (2003), 180–191.

[46] F. Dehne, M. Fellows, F. Rosamond, P. Shaw, *Greedy localization, iterative compression, modeled crown reductions: New FPT techniques, and improved algorithm for set splitting, and a novel 2k kernelization of vertex cover*, Lecture Notes in Computer Science, 3162 (2004), 127–137.

[47] J. DeKleer, *Diagnosing intermittent faults*, Proceedings of 18th International Workshop on Principles of Diagnosis, 2007.

[48] V. Demirer, I. Sahin, *Effect of blended learning environment on transfer of learning: an experimental study*, Journal of Computer Assisted Learning, 29 (2013), 518–529.

[49] J. Desrosiers, N. Mladenovi´c, D. Villeneuve, *Design of balanced MBA student teams*, Journal of the Operational Research Society, 56 (1) (2005), 60–66.

[50] P. Dillenbourg, *What do you mean by collaborative learning*? Collaborative Learning-Cognitive and Computational Approaches, (1999), 1–19.

[51] C. Domingo, N. Mishra, L. Pitt, *Efficient read-restricted monotone CNF/DNF dualization by learning with membership queries*, Machine learning, 37 (1) (1999), 89–110.

[52] G. Dong, J. Li, *Mining border descriptions of emerging patterns from dataset pairs*, Knowledge and Information Systems, 8(2) (2005), 178–202.

[53] D. Džamić, B. Ćendić, M. Marić, A. Đenić, *Solving balanced multi-weighted attribute set partitioning problem with variable neighborhood search*, Filomat, (2018) (accepted for publication).

[54] D. Džamić, M. Marić, A. Đenić, B. Lazović, *A variable neighborhood search for creating student groups with similar characteristics*, Zbornik radova - XLIV Simpozijum o operacionim istraživanjima, (2017), 152 – 157.

[55] A. Đenić, M. Marić, Z. Stanimirović, P. Stanojević, *A variable neighbourhood search method for solving the long-term care facility location problem*, IMA Journal of Management Mathematics, (2016), p. 8.

[56] A. Đenić, N. Radojičić, M. Marić, M. Mladenović, *Parallel VNS for bus terminal location problem*, Applied Soft Computing, 42 (2016), 448–458.

[57] B. Đurić, J. Kratica, D. Tošić, V. Filipović, *Solving the maximally balanced connected partition problem in graphs by using genetic algorithm*, Computing and Informatics, 27(3), (2008), 341–354.

[58] T. Eiter, K. Makino, G. Gottlob, *Computational aspects of monotone dualization: A brief survey*, Discrete Applied Mathematics, 156(11) (2008), 2035–2049.

[59] E. Falkenauer, *Evolutionary algorithms: Applying genetic algorithms to real-world problems*, Springer, New York, (1999), 65-88.

[60] R. M. Felder, L. K. Silverman, *Learning and teaching styles in engineering education*, Engineering Education, 78(7) (1988), 674-681.

[61] T. A. Feo, M. Khellaf, *A class of bounded approximation algorithms for graph partitioning*, *Networks*, 20(2) (1990), 181–195.

[62] A. Fijany, F. Vatan, *New approaches for efficient solution of hitting set problem*, Proceedings of the winter international symposium on Information and communication technologies, Cancun, Mexico: Trinity College Dublin, 2004.

[63] A. Fijany, F. Vatan, *New high performance algorithmic solution for diagnosis problem*, Proceedings of the IEEE Aerospace Conference, 2005.

[64] V. Filipović, *Fine-grained Tournament Selection Operator in Genetic Algorithms*, Computing and Informatics, 22 (2003), 143-161.

[65] V. Filipović, D. Tošić, *Experimental Results in Applying of Fine Grained Tournament Selection*, Proceedings of the 10th Congress of Yugoslav Mathematicians, Belgrade, 21-24.01. (2001), 331-336.

[66] V. Filipović, J. Kratica, D. Tošić, I. Ljubić, *Fine grained tournament selection for the simple plant location problem*, Proceedings of the 5th Online World Conference on Soft Computing Methods in Industrial Applications, (2000), 152-158.

[67] R. Forrester, K. Hutson, T. To, *Improving the quality of the assignment of students to first-year seminars*, OR Insight, 26(2) (2013), 120-139.

[68] R. Forrester, K. Hutson, *Balancing faculty and students preferences in the assignment of students to groups*, Decision Sciences Journal of Innovative Education, 12(2) (2014), 131-147.

[69] M. L. Fredman, L. Khachiyan, *On the complexity of dualization of monotone disjunctive normal forms*, Journal of Algorithms, 21 (3) (1996), 618–628.

[70] E. C. Freuder, R. Dechter, M. L. Ginsberg, B. Selman, E. P. K. Tsang, *Systematic versus stochastic constraint satisfaction*, Proceedings of the International Joint Conference on Artificial Intelligence, (1995), 2027-2032.

[71] D. R. Fulkerson, G. L. Nemhauser, L. E. Trotter, *Two computationally difficult set covering problems that arise in computing the l-width of incidence matrices of Steiner triple systems*, Mathematics Programme of Study, 2 (1974), 72–81.

[72] A. Gainer-Dewar, P. Vera-Licona, *The Minimal Hitting Set Generation Problem: Algorithms and Computation*, SIAM Journal on Discrete Mathematics, 31(1) (2017), 63-100.

[73] M. Garey, D. Johnson, *Computers and intractability: A guide to the theory of NP completeness*, San Francisco: Freeman, 1979.

[74] A. A. Genlott, Å. Grönlund, *Improving literacy skills through learning reading by writing: The iWTR method presented and tested*, Computers & Education, 67 (2013), 98–104.

[75] F. Ghorbani, G. A. Montazer, *Learners grouping improvement in elearning environment using fuzzy inspired PSO method*, Proceedings of the 3rd international conference on E-learning and E-teaching, Tehran, Iran, (2012), 65–70.

[76] A. W. Gill, *The school class allocation problem-maximising friendship requests subject to diversity requirements*, Journal of the Operational Research Society, 66(7) (2014), 1-10.

[77] F. Glover, G.A. Kochenberger, *Handbook of metaheuristics*, International Series in Operations Research & Management Science, 57 (2003).

[78] A. Gogoulou, E. Gouli, G. Boas, E. Liakou, M. Grigoriadou, *Forming Homogeneous, Heterogeneous and Mixed Groups of Learners*, Proceedings of the 11th International Conference on User Modeling: Workshop on Personalisation in E-Learning Environments at Individual and Group Level, Corfu, Greece, (2007), 33-40.

[79] S. Graf, R. Bekele, *Forming heterogeneous groups for intelligent collaborative learning systems with ant colony optimization*, In M. Ikeda, K. Ashley, T.-W. Chan (Eds.), Intelligent Tutoring Systems, Springer Berlin Heidelberg, Lecture Notes in Computer Science, 4053 (2006), 217-226.

[80] S. Graf, S. R. Viola, T. Leo, Kinshuk, *In-depth analysis of the Felder-Silverman learning style dimensions,* Journal of Research on Technology in Education, 40(1) (2007), 79-93.

[81] R. Greiner, B. A. Smith, R. W. Wilkerson, *A correction to the algorithm in Reiters theory of diagnosis,* Artificial Intelligence, 41 (1), (1989), 79-88.

[82] S. M. Gully, D. S. Devine, D. J. Whitney, *A meta-analysis of cohesion and performance: Effects of level of analysis and task interdependence*, Small Group Research, 26 (1995), 497–520.

[83] R. Gürbüz, O. Birgin, *The effect of computer-assisted teaching on remedying misconceptions: The case of the subject "probability"*, Computers & Education, 58 (2012), 931–941.

[84] V. Guruswami, *Inapproximability results for set splitting and satisfiability problems with no mixed clauses,* Algorithmica, 38 (2004), 451–469.

[85] O. Hädicke, S. Klamt, *Computing complex metabolic intervention strategies using constrained minimal cut sets*, Metabolic engineering, 13(2) (2011), 204–213.

[86] M. Hagen, *Algorithmic and computational complexity issues of MONET*, PhD, Friedrich-Schiller Universität Jena, 2008.

[87] P. Hansen, N. Mladenović, *Variable neighbourhood search*. In F. Glover, G. Kochenagen, editors, Handbook of Metaheuristics, Dordrecht, (2003), 145-184.

[88] P. Hansen, N. Mladenovic, J. Brimberg, J. A. M. Pérez, *Variable neighborhood search*. In M. Gendreau, J.-Y. Potvin (Eds.), Handbook of Metaheuristics, International Series in Operations Research & Management Science, 146 (2010), 61-86.

[89] P. Hansen, N. Mladenovic, J. A. M. Pérez, *Variable neighbourhood search: methods and applications,* Annals of Operations Research, 175(1) (2010), 367–407.

[90] M. Hasan, S. M. S. Hossain, MD. M. Rahman, M. S. Rahman, *Solving minimum hitting set problem and generalized exact cover problem with light based devices*, International Journal of Unconventional Computing, 7(1-2) (2011), 125-140.

[91] C. Hébert, A. Bretto, B. Crémilleux, *A data mining formalization to improve hypergraph minimal transversal computation*, Fundamenta Informaticae, 80(4) (2007), 415–434.

[92] S. Heipcke, *Comparing constraint programming and mathematical programming approaches to discrete optimisation- the change problem*, Journal of the Operational Research Society, 50(6) (1999), 581–595.

[93] A.V. Hill, J. D. Naumann, N. L. Chervany, *SCAT and SPAT: Large-scale computer-based optimization systems for the personnel assignment problem*, Decision Sciences, 14(2) (1983), 207–220.

[94] T. F. Ho, S. J. Shyu, F. H.Wang, C. T. Li, *Composing high-heterogeneous and high-interaction groups in collaborative learning with particle swarm optimization*, World Congress on Computer Science and Information Engineering, (2009), 607-611.

[95] A. Holen, *The PBL group: Self-reflections and feedback for improved learning and* growth, Medical Teacher, 22 (2000), 485–488.

[96] J. H. Holland, *Adaptation in natural and artificial systems*, University of Michigan Press, 1975.

[97] J.-L. Hsu, H.-W. Chou, W.-Y. Hwang, S.-B. Chou, *A two-dimensional process in explaining learners' collaborative behaviors in CSCL*, Educational Technology and Society, 11 (2008), 66–80.

[98] http://geogebra.matf.bg.ac.rs/software/bmwasp/instances.zip

[99] http://geogebra.matf.bg.ac.rs/software/bmwasp/app.zip

[100] http://geogebra.matf.bg.ac.rs/software/forming_groups/app.zip

[101] http://geogebra.matf.bg.ac.rs/software/two_groups/app.rar

[102] http://www.personalitypathways.com/type_inventory.html

[103] Y.-M. Huang, T.-T. Wu, *A systematic approach for learner group composition utilizing U-learning portfolio*, Educational Technology & Society, 14(3) (2011), 102–117.

[104] R. Hubscher, *Assigning students to groups using general and context-specific criteria*, IEEE Transactions on Learning Technologies, 3(3) (2010), 178–189.

[105] G. J. Hwang, P.Y. Yin, C. W. Hwang, C. C. Tsai, *An enhanced genetic approach to composing cooperative learning groups for multiple grouping criteria*, Journal of Educational Technology & Society, 11(1) (2008), 148–167.

[106] D. Jackson, *An international profile of industry-relevant competencies and skill gaps in modern graduates*, The International Journal of Management Education, 8(3) (2010), 29–58.

[107] D. Jin, Z. Qinghua, D. Jiao, G. Zhiyong, *A method for learner grouping based on personality clustering*, Proceedings of the 10th international conference on computer supported cooperative work in design, Nanjing, China: IEEE, (2006), 1–6.

[108] F. Johansson, *The Medici effect: breakthrough, insights at the intersection of ideas, concepts and cultures*, Boston, MA: Harvard Business School Press, 2004.

[109] D.W. Johnson, R.T. Johnson, *Cooperative Classrooms*, In M. Brubacher, R. Payne, K. Rickett (Eds.), Perspectives on Small Group Learning: Theory And Practice, Ontario: Rubicon Publishing Inc, (1990), 119-131.

[110] D.W. Johnson, R.T. Johnson, *The internal dynamics of cooperative learning groups*. In R. Slavin, et al. (Eds.), Learning to Cooperate, Cooperating to Learn, New York: Plenum, (1985), 103-124.

[111] D.W. Johnson, R.T. Johnson, E. J. Holubec, *Cooperation in the classroom*, Edina, MN: Interaction Book Company, 1998.

[112] M. M. B. Jozan, F. Taghiyareh, *An evolutionary algorithm for homogeneous grouping to enhance web-based collaborative learning*, International Journal of Computer Science Research and Application, 3(1) (2013), 74-85.

[113] S. J. Karau, K. D. Williams, S*ocial loafing: A meta-analytic review and theoretical Integration*, Journal of Personality and Social Psychology, 65 (1993), 681–706.

[114] D. Krass, A. Ovchinnikov, *Constrained group balancing: Why does it work?*, European Journal of Operational Research, 206(1) (2010), 144–154.

[115] D. Krass, A. Ovchinnikov, *The University of Toronto's Rotman School of Management uses management science to create MBA study groups*, Interfaces, 36(2) (2006), 126–137.

[116] J. Kratica, *An Electromagnetism-like method for the maximum set splitting problem*, Yugoslav Journal of Operations Research, 23(1) (2013).

[117] J. Kratica, *Improving performances of the genetic algorithm by caching*, Computing & Artificial Intelligence, 18 (1999), 271–283.

[118] J. Kratica, *Paralelizacija genetskih algoritama za rešavanje nekih NP- kompletnih problema*, Doktorska disertacija, Matematički fakultet, Beograd, 2000.

[119] J. Kratica, M. Čangalović, V. Kovačević-Vujčić, *Computing minimal doubly resolving sets of graphs*, Computers and Operations Research, 36(7) (2009), 2149-2159.

[120] J. Kratica, V. Kovačević-Vujčić, M. Čangalović, *Computing strong metric dimension of some special classes of graphs by genetic algorithms*, Yugoslav Journal of Operations Research, 18(2) (2008), 143-151.

[121] J. Kratica, V. Kovačević-Vujčić, M. Čangalović, *Computing the metric dimension of graphs by genetic algorithms*, Computational Optimization and Applications, 44(2) (2009), 343-361.

[122] J. Kratica, Z. Stanimirović, D. Tošić, V. Filipović, *Two genetic algorithms for solving the uncapacitated single allocation p-hub median problem*, European Journal of Operational Research, 182(1) (2007), 15-28.

[123] B. Krauss, J. Lee, D. Newman, *Optimizing the assignment of students to classes in an elementary school*, INFORMS Transactions on Education, 14(1) (2013), 39–44.

[124] K. Kreijns, P. A Kirschner, W. Jochems, *Identifying the pitfalls for social interaction in computer-supported collaborative learning environments: a review of the research*, Computers in Human Behavior, 19 (2003), 335-353.

[125] K. Kreijns, P. A. Kirschner, W. Jochems, *The Sociability of Computer-Supported Collaborative Learning Environments*, Educational Technology & Society, 5(1) (2002), 8–22.

[126] J. A. Kulikand , C. L. C. Kulik, *Effects of ability grouping on student achievement*, Equity & Excellence in Education, 23(1-2) (1987), 22-30 .

[127] R. H. Kwon, G. V. Dalakouras, C. Wang, *On a posterior evaluation of a simple greedy method for set packing*, Optimization Letters, (2008), 587-597.

[128] D. Lambić, B. Lazović, M. Marić, A. Đenić, *A novel metaheuristic approach for collaborative learning group formation*, Journal of Computer Assisted Learning, (2018), 1-10. https://doi.org/10.1111/jcal.12299

[129] P. B. Larson, S. Shelah, *The stationary set splitting game*, Mathematical Logic Quarterly, 54(2) (2008), 187-193.

[130] B. Lazović, M. Marić, V. Filipović, A. Savić, *An integer linear programming formulation and genetic algorithm for the maximum set splitting problem*, Publications de l'Institut Mathématique, 92(106) (2012), 25–34.

[131] L. Liberti, N. Maculan, Y. Zhang, *Optimal configuration of gamma ray machine radiosurgery units: the sphere covering subproblem*, Optimization Letters, 3 (2009), 109–121.

[132] Y. T. Lin, Y. H. Huang, S. C. Cheng, *An automatic group composition system for composing collaborative learning groups using enhanced particle swarm optimization*, Computers & Education, 55(4) (2010), 1483–1493.

[133] L. Lin, Y. Jiang, *Computing minimal hitting sets with genetic algorithms*, Proceedings of the Internation Workshop on Principles of Diagnosis, Semmering, 2002.

[134] L. Lin, Y. Jiang, *The computation of hitting sets: Review and new algorithms*, Information Processing Letters, 86 (4) (2003), 177–184.

[135] D. Lokshtanov, S. Saurabh, *Even faster algorithm for set splitting!*, Proceedings of the International Workshop on Parameterized and Exact Computation, (2009), 288-299.

[136] L. Lovász, *Coverings and colorings of hypergraphs*, Proceedings of the 4th Southeastern Conference on Combinatorics, Utilitas Mathematics, Springer Berlin Heidelberg, (1973), 3-12.

[137] S. Lu, *Randomized and deterministic parameterized algorithms and their applications in bioinformatics*, PhD, Texas A&M University, 2009.

[138] M. Marić, *An efficient genetic algorithm for solving the multi-level uncapacitated facility location problem*, Computing and Informatics, 29(2) (2010), 183-201.

[139] M. Marić, D. Džamić, B. Lazović, *Rešavanje problema balansiranog particionisanja skupa primenom metode promenljivih okolina*, Osmi simpozijum „Matematika i primene", Matematički fakultet, Univerzitet u Beogradu, Beograd, Srbija, 17- 18. Nov, 2017.

[140] E. Martín, P. Paredes, *Using learning styles for dynamic group formation in adaptive collaborative hypermedia systems*, Proceedings of the First International Workshop on Adaptive Hypermedia and Collaborative Web-based Systems, (2004), 188-198.

[141] D. Matić, *Рјешавање неких проблема у настави примјеном метода комбинаторне оптимизације*, PhD, Matematički fakultet, Beograd, 2013.

[142] B. A. McCarl, T. H. Spreen, *Applied mathematical programming using algebraic systems*, Cambridge, 1997.

[143] E. J. McCluskey, *Minimization of boolean functions*, The Bell System Technical Journal, 35(5) (1956), 1417-1444.

[144] N. Meslec, P. L. Curşeu, *Are balanced groups better? Belbin roles in collaborative learning groups*, Learning and Individual Differences, 39 (2015), 81–88.

[145] D. Meyer, *Opt assign – a web-based tool for assigning students to groups*, Computers & Education, 53(4) (2009), 1104–1119.

[146] J. Mingers, F. A. O'Brien, *Creating student groups with similar characteristics: a heuristic approach,* Omega, 23(3) (1995), 313–321.

[147] M. Mitchell, *An Introduction to Genetic Algorithms*, Cambridge, Massachusetts: MIT Press, 1999.

[148] N. Mladenović, P. Hansen, *Variable neighborhood search*, Computers & Operations Research, 24(11) (1997), 1097–1100.

[149] G. A. Montazer, M. S. Rezaei, M. S. *A new approach in e-learners grouping using Hybrid Clustering Method*, Proceedings of the international conference on Education and e-Learning Innovations, Sousse, Tunisia: IEEE, (2012), 1–5.

[150] J. Moreno, D. A. Ovalle, R. M. Vicari, *A genetic algorithm approach for group formation in collaborative learning considering multiple student characteristics*, Computers & Education, 58(1) (2012), 560–569.

[151] K. Murakami, T. Uno, *Efficient algorithms for dualizing large-scale hypergraphs*, Discrete Applied Mathematics, 170 (2014), 83–94.

[152] I. Myers, *Manual: The Myers-Briggs type indicator*, Princeton, N.J.: Educational Testing Service, 1962.

[153] I. Myers Briggs, P. B. Myers. *Gifts differing: understanding personality type*, Davies-Black Publishin, 1980.

[154] M. Notari, A. Baumgartner, W. Herzog, *Social skills as predictors of communication, performance and quality of collaboration in project-based learning*, Journal of Computer Assisted Learning, 30 (2014), 132–147.

[155] X. Ou, D. You, *A digital collaborative design model based on learner characteristics in art design*, American Journal of Engineering and Technology Research, 11(9) (2011), 3056–3061.

[156] A. Ounnas, H. C. Davis, D. E. Millard, *A framework for semantic group formation in education*, Educational Technology & Society, 12(4) (2009), 43–55.

[157] A. Paszynska, M. Paszynski, *Application of a hierarchical chromosome based genetic algorithm to the problem of finding optimal initial meshes for the self-adaptive hp-FEM*, Computing and Informatics, 28(2) (2009), 209–223.

[158] L. A. Penner, J. F. Dovidio, J. A. Piliavin, D. A. Schroeder, *Prosocial behavior: Multilevel perspectives*, Annual Review of Psychology, 56(2005), 365–392.

[159] M. Pirlot, *General local search methods*, European Journal of Operational Research, 92 (1996), 493-511.

[160] M. Qasem, A.prugel-bennett, *Complexity of max-sat using stochastic algorithms*, Proceedings of the 10th annual conference on Genetic and evolutionary computation, (2008), 615-616.

[161] T. Quaritsch, I. Pill, *PyMBD: A library of MBD algorithms and a light-weight evaluation platform*, Proceedings of International Workshop on Principles of Diagnosis, (2014), 1-5.

[162] W. Quine, *A way to simplify truth functions*, American Math Monthly, 62 (1955), 627- 631.

[163] L. Razmerita, A. Brun, *Collaborative learning in heterogeneous classes*, The 3rd International Conference on Computer Supported Education, Netherlands, (2011), 189–194.

[164] C. Reeves, *Modern heuristic techniques for combinatorial* problems, New York: John Wiley & Sons, 1993.

[165] G. Reeves, E. Hickman, *Assigning MBA students to field study project teams: A multi-criteria approach*, Interfaces, 22(5) (1992), 52–58.

[166] R. Reiter, *A theory of diagnosis from first principles*, Artificial Intelligence, 32(1) (1987), 57-95.

[167] T. M. Roeder, R. M. Saltzman, *Schedule-based group assignment using constraint programming*, INFORMS Transactions on Education, 14(2) (2014), 63–72.

[168] R. H. Rutherfoord, *Using personality inventories to help form teams for software engineering class projects*, Proceedings of the 6th Annual Conference on Innovation and Technology in Computer Science Education, (2001), 73-76.

[169] H. Sadeghi, A. Kardan, *A novel justice-based linear model for optimal learner group formation in computer-supported collaborative learning environments*, Computers in Human Behavior, 48 (2015), 436-447.

[170] V. Savicki, M. Kelley, D. Lingenfelter, *Gender and group composition in small task groups using computer-mediates communication*, Computers in Human Behavior, 12(2) (1996), 209–224.

[171] L. Shi, X. Cai, *An exact fast algorithm for minimum hitting set*, Computational Science and Optimization, 1 (2010), 64–67.

[172] R. E. Slavin, *Developmental and motivational perspectives on cooperative learning: a* reconciliation, Child Development, 58(5) (1987), 1161–1167.

[173] R. E. Slavin, *When does cooperative learning increase achievement?* Psychological Bulletin, 94 (1983), 429–445.

[174] M. Soflano, T. M. Connolly, T. Hainey, *Learning style analysis in adaptive GBL application to teach SQL*, Computers & Education, 86 (2015), 105-119.

[175] P. Souren, P. Seetharaman, I. Samarah, P. Mykytyn, *Impact of heterogeneity and collaborative conflict management style on the performance of synchronous global virtual teams*, Information and Management, (2003), 303–321.

[176] B. W. Speck, *Fostering collaboration among students in problem-based learning*, New Directions for Teaching and Learning, 95 (2003), 59-66.

[177] Z. Stanimirović, *A Genetic Algorithm Approach for the Capacitated Single Allocation PHub Median Problem*, Computing and Informatics, 29(1) (2010), 117–132.

[178] Z. Stanimirović, J. Kratica, Đ. Dugošija, *Genetic algorithms for solving the discrete ordered median problem*, European Journal of Operational Research, 182(3) (2007), 983–100.

[179] Z. Stanimirović, M. Marić, S. Božović, P. Stanojević, *An efficient evolutionary algorithm for locating long-term care facilities*, Information Technology and Control, 41(1) (2012), 7789.

[180] P. Stanojević, *Egzaktne i metaheuristicke metode za rešavanje NP-teških lokacijskih problema*, Doktorska disertacija, Matematički fakultet, Beograd, 2016.

[181] G. Steiner, J. S. Yeomans, *Optimal level schedules in mixed-model, multi-level JIT assembly systems with pegging*, European Journal of Operational Research, 95(1) (1996), 38–52.

[182] R. J. Sternberg, *Thinking styles*, New York: Cambridge University Press, 1998.

[183] R. J. Sternberg, R. K.Wagner, *MSG thinking styles inventory manual*, Unpublished manuscript, 1991.

[184] M. Taboga. *Lectures on probability theory and mathematical statistics*, Createspace Independent Pub, 2012.

[185] Đ. Takači, G. Stankov, I. Milanović, *Efficiency of learning environment using GeoGebra when calculus contents are learned in collaborative groups*, Computers & Education, 82 (2015), 421-431.

[186] S. L. Tanimoto, *The squeaky wheel algorithm: automatic grouping of students for collaborative projects*. Proceedings of the Workshop Personalisation in Learning Environments at Individual and Group Level in Conjunction with 11th International Conference User Modeling, (2007), 79-80.

[187] D. Urošević, *Variable neighborhood search for maximum diverse grouping problem*, Yugoslav Journal of Operations Research, 24(1) (2014), 21-33

[188] A. Vazquez, *Optimal drug combinations and minimal hitting sets*, BMC systems biology, 3(1) (2009), 81.

[189] P. Vera-Licona, E. Bonnet, E. Barillot, A. Zinovyev, *OCSANA: optimal combinations of interventions from network analysis*, Bioinformatics, 29(12) (2013), 1571–1573.

[190] D. Y. Wang, S. S. J. Lin, C. T. Sun, *DIANA: A computer supported heterogeneous grouping system for teachers to conduct successful small learning groups*, Computers in Human Behavior, 23(4) (2007), 1997–2010.

[191] C. Wang, M. T. Thai, Y. Li, F. Wang, W. Wu, *Optimization scheme for sensor coverage scheduling with bandwidth constraints*, Optimization Letters, 3 (2009), 63–75.

[192] N. Webb, *Peer interaction and learning in small groups*, International Journal of Educational Research, 13 (1989), 21–39.

[193] R. R. Weitz, M. T. Jelassi, *Assigning students to groups: A multi-criteria decision support system approach*, Decision Sciences, 23 (1992), 746–757.

[194] R. Weitz, S. Lakshminarayanan, *An empirical comparison of heuristic methods for creating maximally diverse groups*, Journal of the Operational Research Society, 49 (1998), 635–646.

[195] F. Wotawa, *A variant of Reiters hitting-set algorithm*, Information Processing Letters, 79 (2001), 45-51.

[196] V. Yannibelli, A. Amandi, *A deterministic crowding evolutionary algorithm to form learning teams in a collaborative learning context*, Expert Systems with Applications, 39(10) (2012), 8584–8592.

[197] F. Yu, R. Katz, and T. Lakshman, *Efficient multi-match packet classification for network security applications*, IEEE Journal on Selected Areas in Communications, 24(2) (2006).

[198] F. Yu, T. V. Lakshman, M. A. Motoyama, R. H. Katz, *SSA: A Power and memory efficient scheme to multi-match packet classification*. EECS Department, University of California, Berkeley, 2005.

[199] B. Yuan, M. Orlowska, S. Sadiq, *Extending a class of continuous estimation of distribution algorithms to dynamic problems*, Optimization Letters, 2 (2008), 433–443.

[200] D. Zakrzewska, *Cluster analysis in personalized e-learning systems*. In N. T. Nguyen & E. Szczerbicki (Eds.), Intelligent systems for knowledge management, studies in computational intelligence, Heidelberg, Germany: Springer, 252 (2009), 229–250.

[201] K. Zhang, L. Cui, H. Wang, Q. Sui, *An improvement of matrix-based clustering method for grouping learners in e-learning*, Proceedings of the 11th international conference on computer supported cooperative work in design, Melbourne, Australia: IEEE, (2007), 1010–1015.

[202] J. Zhang, Y. Ye, Q. Han, *Improved approximations for max set splitting and max NAE SAT*, Discrete Applied Mathematics, 142 (2004), 133–149.

[203] X. Zhao, D. Ouyang, *Improved algorithms for deriving all minimal conflict sets in modelbased diagnosis*, Proceedings of the 3rd International Conference on Intelligent Computing (ICIC07), Lecture Notes in Computer Science, 4681 (2007), 157-166.

[204] Y. Zheng, C. Li, S. Liu, W. Lu, *An improved genetic approach for composing optimal collaborative learning groups*, Knowledge-Based Systems, 139 (2018), 214–225.

Биографија аутора

Бојана Лазовић (удато Ћендић) рођена је 29. марта 1979. године у Краљеву. Основну школу „Четврти краљевачки батаљон” завршила је као носилац дипломе „Вук Караџић”. Гимназију у Краљеву завршила је са одличним успехом. Током школовања учествовала је и награђивана на такмичењима из математике. Образовање наставља на Математичком факултету у Београду, где је дипломирала са просечном оценом 8.22 на смеру Професор математике и рачунарства. Докторске академске студије студијског програма Методика наставе математике и рачунарства уписала је фебруара 2009. године. Све испите предвиђене планом и програмом докторских студија положила је закључно са јануаром 2011. године, са просечном оценом 9.83.

У периоду од септембра 2005. до децембра 2006. године радила је као професор математике, и рачунарства и информатике у Гимназији у Краљеву. Од децембра 2006. до маја 2014. године радила је у Шумарској школи у Краљеву, као професор математике, и рачунарства и информатике. Од маја 2014. године до јула 2015. године била је запослена као стручни сарадник у настави, а потом је 2015. године изабрана у звање асистента на Катедри за математику и статистику у Београдској пословној школи – Високој школи струковних студија.

Од 2015. до 2016. године била је ангажована на пројекту Министарства туризма, трговине и телекомуникација и Друштва математичара Србије под називом „Пријемни испит – равноправно за све“.

Прилог 1.

Изјава о ауторству

Име и презиме аутора __Бојана Лазовић__________________________

Број индекса ____2028/2008____

Изјављујем

да је докторска дисертација под насловом

Примена метода комбинаторне оптимизације за решавање проблема формирања групa у настави

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

Потпис аутора

У Београду, ____________________

____

Прилог 2.

Изјава о истоветности штампане и електронске варијанте докторског рада

Име и презиме аутора ___Бојана Лазовић___________________________________

Број индекса___2028/2008____

Студијски програм ___Методика наставе математике и рачунарства___________

Наслов рада __Примена метода комбинаторне оптимизације за решавање проблема формирања група у настави______________________________________

Ментор __др Мирослав Марић____

Изјављујем да је штампана варијанта мог докторског рада истоветна електронској варијанти коју сам предао/ла ради похрањена у **Дигиталном репозиторијуму Универзитета у Београду.**

Дозвољавам да се објаве моји лични подаци везани за добијање академског назива 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 аутора

У Београду, ________________________

____

Прилог 3.

Изјава о коришћењу

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

Примена метода комбинаторне оптимизације за решавање проблема

формирања група у настави

која је моје ауторско дело.

Дисертацију са свим прилозима предао/ла сам у електронском формату погодном за трајно архивирање.

Моју докторску дисертацију похрањену у Дигиталном репозиторијуму Универзитета у Београду и доступну у отвореном приступу могу да користе сви који поштују одредбе садржане у одабраном типу лиценце Креативне заједнице (Creative Commons) за коју сам се одлучио/ла.

1. Ауторство (CC BY)

2. Ауторство – некомерцијално (CC BY-NC)

3. Ауторство – некомерцијално – без прерада (CC BY-NC-ND)

4. Ауторство – некомерцијално – делити под истим условима (CC BY-NC-SA)

5. Ауторство – без прерада (CC BY-ND)

6. Ауторство – делити под истим условима (CC BY-SA)

(Молимо да заокружите само једну од шест понуђених лиценци.
Кратак опис лиценци је саставни део ове изјаве).

Потпис аутора

У Београду, __________________________

____

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 прерада**. Дозвољавате умножавање, дистрибуцију и јавно саопштавање дела, без промена, преобликовања или примене 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 прерада**. Дозвољавате умножавање, дистрибуцију и јавно саопштавање дела, без промена, преобликовања или примене 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.