У Н И В Е Р З И Т Е Т   У   Б Е О Г Р А Д У

МАТЕМАТИЧКИ ФАКУЛТЕТ

Дипломски - мастер рад

РЕШАВАЊЕ ПРОБЛЕМА МУЛТИДИМЕНЗИОНАЛНОГ РАНЦА ПРИМЕНОМ ГЕНЕТСКОГ АЛГОРИТМА

Драгана Јовановић

Београд, јануар 2012.

Ментор: **др Александар Савић**

доцент Математичког факултета у Београду

Чланови комисије:

др Душан Тошић

редовни професор Математичког факултета у Београду

др Владимир Филиповић

доцент Математичког факултета у Београду

Датум одбране:___________________

Садржај

Проблем мултидимензионалног ранца

Резиме

У раду је описана примена генетских алгоритама за решавање проблема вишедимензионалног ранца. Проблем је директно примењив у пракси: при алокацији ресурса са финансијским ограничењима, у комбинаторици, криптографији и примењеној математици.

Репрезентација јединки је целобројна, којој одговара пермутација од *n* предмета. Да ли је предмет изабран или није, одређује се на основу претходно генерисане пермутације. Изабрани су они предмети чији капацитет је мањи или једнак од преосталих капацитета по свакој карактеристици. Свако такво решење је очигледно допустиво, па се не појављује проблем некоректних јединки. На тај начин, омогућена је примена стандардних генетских оператора: фино-градирана турнирска селекција, једнопозиционо укрштање и проста мутација са залеђеним генима.

У циљу побољшавања рада генетског алгоритма, примењене су неке од раније познатих техника. Стационарни генетски алгоритам са елитистичком стратегијом очувава добра решења и значајно доприноси добром квалитету коначног решења. Уклањање вишеструке појаве истих јединки и ограничавање појаве различитих јединки са истом вредношћу функције циља ефективно доприноси избегавању преурањене конвергенције у неком локалном максимуму. Кеширање ГА у доброј мери смањује време извршавања, а не утиче на остале аспекте рада генетског алгоритма.

Генетски алгоритам је тестиран на инстанцама из литературе, максималне величине карактеристика 100, и до 2500 предмета. Добијени су резултати задовољавајућег квалитета у достижном времену извршавања.

Кључне речи: вишедимензионални ранац, генетски алгоритми, комбинаторна оптимизација, метахеуристике.

Abstract

In this paper the genetic algorithm for solving Multidimensional Knapsack Problem (MKP) is described. The problem arises often in resource allocation with financial constraints. A similar problem also appears in combinatory, complexity theory, cryptography and applied mathematics.

The representation of individuals is integer. The permutation representation of individuals is applied. For each element of permutation, it is checked if its characteristics fits total capacity constraints. If answer is positive, the item is added to knapsack. Obviously, any such solution is admissible, so in the solution does not appear incorrect individuals. In this way, it is possible to apply in application standard genetic operators: fine grained tournament selection, one-point crossover, simple mutation with frozen genes.

To improve implementation of algorithm, some of the previously known techniques are applied. Steady state algorithm with elitist strategy preserves good solutions and contributes significantly to the quality of the final solution. Removing multiple instances of the same individuals and limiting the occurrence of different individuals with the same objective function contributes effectively to avoid premature convergence to a local maximum. Cashing technique reduces the execution time, and improves the performance of algorithm.

Genetic algorithm is tested on the publicly-available instances from the literature, maximum size of characteristics 100 and 2500. Proposed genetic algorithm has achieved results of satisfactory quality of the execution time achievable.

Keywords: Multidimensional Knapsack, Genetic Algorithm, Combinatorial Optimization, Metaheuristics

ПРЕДГОВОР

У уводном делу рада дате су опште информације о генетским алгоритмима и метахеуристикама. Други део посвећен је проблему вишедимензионог ранца и општа разматрања о њему: формулација проблема, начинима решавања. У трећем поглављу описан је предложени генетски алгоритам за решавање проблема. Четврто поглавље садржи експерименталне резултате добијене применом генетског алгоритма. У последњем, петом поглављу дат је закључак, као и допринос овог рада.

Желим да се захвалим ментору др Александру Савићу на огромном стрпљењу, мотивацији и корисним саветима приликом израде рада. Такође, желим да се захвалим члановима комисије др Душану Тошићу и др Владимиру Филиповићу на конкретним сугестијама које су ми знатно помогле.

Захваљујем се драгим родитељима и сестрама на стрпљењу и подршци.

Београд, јануар 2012. Кандидат

Драгана Јовановић

1.УВОД

1.1 Проблем вишедимензионалног ранца

Проблем ранца је један од најпознатијих проблема комбинаторне оптимизације. Најосновнија верзија проблема се дефинише на следећи начин: претпоставимо да је дат ранац запремине $b \geq 0$, и скуп од n предмета којима се ранац пуни. Сваки предмет има своју запремину $a_j \geq 0$ и вредност $c_j \geq 0$. Напунити ранац садржајем највеће вредности тако да је укупна вредност која се носи у ранцу максимална:

$$\max \sum_{j=1}^{n} c_j x_j$$

$$\text{при ограничењу:} \qquad \sum_{j=1}^{n} a_j x_j \leq b$$

$$x_j \in \{0,1\}, j = 1, \dots, n$$

где промењљиве x_j имају вредност 1, ако је предмет изабран, иначе 0, тј.

$$x_j = \begin{cases} 1, & \text{предмет је изабран} \\ 0, & \text{иначе} \end{cases}$$

Ово је проблем целобројног програмирања. Проблем је димензије један, једино ограничење ранца представља његова запремина. Проблем је у општем случају **NP-**тежак [Kar72], али у специјалном случају када су коефицијенти цели бројеви, може се оптимално решити динамичким програмирањем. Иако тај алгоритам није полиномске сложености, у пракси даје добре резултате. За сад, једина карактеристика ранца је његова запремина. Опис проблема једнодимензионог ранца, као и приближни и тачни алгоритми за његово решавање дали су Martello и Toth [Tot90]. У литератури се среће термин: *0-1 димензиони ранац*, у смислу, јединица је на месту предмета који припада ранцу, односно 0. Ако бисмо додали још неку карактеристику, онда бисмо добили ранац димензије два, у литератури познат као „проблем дводимензионог ранца". Временом су се развијале разне варијанте проблема, са већим бројем димензија и ограничења. Отуда и термин: *мултидимензиони проблем ранца*. Проблем се дефинише овако: дат је ранац коначног капацитета и постоји n предмета, и сваки од њих има m карактеристика. Треба напунити ранац највреднијим предметима, тако да његова унутрашњост буде максимално испуњена. Дате су константе:

d_{ij}, која представља i-ту карактеристику, j-тог предмета,

p_j, као добитак по j-том предмету и

b_i, ограничење карактеристика по свим предметима.

Математичка формулација проблема гласи овако:

$$max \sum_{j=1}^{n} p_j x_j \qquad (1)$$

при ограничењима:

$$\sum_{j=1}^{n} d_{ij} x_j \leq b_i\,, \quad i = 1, \dots, m \quad (2)$$

$$x_j \in \{0,1\}, \qquad j = 1, \dots, n \qquad (3)$$

$$\text{где је } x_j = \begin{cases} 1, & \text{предмет је изабран} \\ 0, & \text{иначе} \end{cases}$$

Једначина (1) представља функцију циља коју треба максимизовати, свако од m ограничења описаних у једначини (2), зове се ограничење ранца, тако да се и проблем мултидимензионог ранца може звати и *проблем m-димензионог ранца.* У ствари, проблем ранца је потпроблем (подскуп) проблема вишедимензионог ранца.

Налажење оптималне вредности проблема ранца наводи нас на основни проблем оптимизације: дат је коначно пребројив скуп S и функција $f: S \to \mathbb{R}$. наћи максимум функције f на скупу S, другим речима одредити

$$max_{x \in S} f(x).$$

Скуп S се назива *допустив скуп,* функција f *функција циља.* За тачку $x \in S$ каже се да је допустиво решење горњег проблема. Задатак је наћи сва допустива решења или бар једно x' такво да је $f(x') = min\, f(x)$ или $f(x') = max\, f(x)$. Таква решења се називају *оптимална решења.* Детаљније о алгоритмима потпуне претраге на коначном скупу S, видети у [Kom96].

1.2 Хеуристике

Проблеми у комбинаторној оптимизацији су најчешће **NP-** тешки, и време изршавања алгоритама за њихово решавање може бити екстремно дугачко. Тачни (егзактни) алгоритми јесу најпожељнији, јер гарантују оптималност добијеног решења, али су за димензије већег проблема готово непримењљиви (нпр: Гаусов метод за решавање система линеарних једначина: за $n > 100$, даје решења која су далеко лошија од оптималног и самим тим неприхватљива.)

Прибегава се хеуристичким методама које најбрже дају решење за проблеме великих димензија; најинтересантнији **NP**-проблеми су управо великих димензија. Опште хеуристике су успешно примењене на скоро све добро познате проблеме комбинаторне оптимизације: проблем трговачког путника, проблем ранца, проблем квадратне асигнације, проблем максималне клике итд. Више о овоме видети у [Čan96]. Решавањем проблема тако великих димензија, добијамо решење које је „довољно добро“. Методе модификоване за специфичан проблем и које пружају управо овакво решење су *метахеуристике.* Најпознатије метахеуристике су: табу претраживање (tabu search), симулирано каљење (simulating annealing), генетски алгоритми (genetic algorithm), метода промењљивих околина (VNS - Variable Neighborhood Search), Лагранжева релаксација (Lagrange relaxation) и остали.

Метахеуристике, самим тим и генетски алгоритми служе за решавање тежих оптимизационих проблема, за које не постоји егзактна математичка метода решавања или су NP-тешки, па се стога не могу решити у прихватљивом времену.

1.3 Генетски алгоритми

Генетски алгоритми (у ознаци ГА) су методе решавања проблема који опонашају процесе еволуције у природи. Компјутерска симулација еволуције први пут се примењује 1954. године на Принстону. Шездесетих година почиње шире да се примењује након низа чланака физичара Фрејзера и Бремермана. Основни елементи које су садржале тадашње симулације, очувале су се до данашњег концепта модерног ГА. Делом „Адаптација у природним и вештачким системима“, Холанд их популаризује '70 –тих година [Hol75]. Холанд, као основни концепт ГА, предлаже опонашање природних процеса еволуције. Свака јединка има диплоидан број хромозома (човек има 46). Делови хромозома су *гени*, и управо су они главни преносиоци особина на следеће генерације. Овај процес обезбеђује *селекција,* која одабира најбоље јединке које ће дати потомство и самим тим пренети добре особине (гене). Укрштање обезбеђује размену и разноврсност генетског материјала. „Паралела“ између природе и ГА приказана је у следећој табели:

Природа	**Генетски алгоритми**
Хромозом, Генотип	**Стринг**
Ген	**Карактер**
Локација	**Позиција карактера у стрингу**
Скуп хромозома	**Популација**

Оператори којима се делује у ГА опонашају неке генетске процесе у биолошким системима. У основној верзији ГА примењују се оператори *селекције, укрштања и мутације.*

Основни концепт простог ГА користи бинарну презентацију јединки, просту селекцију, укрштање у једној тачки и просту мутацију (о којима ће бити речи мало касније). Користи се и термин *канонски генетски алгоритам.* Више о овом концепту може се наћи у светској [Bäc00a, Bäc00b, Bea93a, Bea93b, Gol89, Mic96, Mit99, Müh97], као и у нашој литератури [Čan96, Fil98, Kra00, Kra01b,]. Шематски приказ основних делова генетског алгоритма дат је на Сл. 1.

Перформансе простог генетског алгоритма се могу побољшати ако се оператори алгоритма и начин кодирања прилагоде природи проблема који се решава.

```
ucitavanje_ulaznih_podataka();
generisanje_pocetne_populacije();
while(!kriterijum_zaustavljanja_ga()) {
for(i = 1; i < N_pop; i++)
obj[i] = Funkcija_Cilja(i);
funkcija_prilagodjenosti();
selekcija();
ukstanje();
mutacija(); }
stampanje_izlaznih_podataka();
```

Сл. 1. Основни концепт генетског алгоритма

Основа ГА је популација. Популација може имати од неколико до пар стотина јединки. Почетна популација обично се бира на случајан начин. Свака јединка популације представља потенцијално решење. Зато је и важан начин кодирања, јер, лоше кодирање утиче на аспекте генетског алгоритма и такав

ГА даће лоше резултате. Још, потребна нам је и оцена јединки, у смислу која је од њих најефикаснија и најприлагођенија околини.

1.3.1 Кодирање и функција прилагођености. Кодирање може бити бинарно или над неком азбуком више кардиналности. Улогу оцене квалитета јединке има *функција прилагођености (fitness function).* Она је, у ствари, кључ процеса селекције јер одређује које ће се јединке елиминисати а које не. Иако то није увек случај, пожељно је да функција прилагођености буде непрекидна и глатка, да би јединке са сличним генетским кôдом имале сличну вредност функције прилагођености. Још, важно је да функција прилагођености, уколико је то могуће, нема много локалних или глобалних екстремума ([Bed93a]). Начин рачунања функције прилагођености у великој мери утиче на перформансе ГА. Начин прелажења од функције циља на функцију прилагођености лежи у самом проблему и некад је потребно комбиновати више начина за њено дефинисање:

- директно преузимање,
- линеарно скалирање,
- сигма одсецање и
- скалирање у јединични интервал.

Оператором селекције бирају се јединке које ће учествовати у следећем кораку: укрштању. Селекцијом и укрштањем се преносе најбоље особине у наредну генерацију. Ако сваки пут преносимо само најбоље особине, неке јединке (која се показала најефикаснијом), може доћи до доминације те јединке у популацији. Ова појава је позната као *преурањена конвергенција.* Процес оптимизације се завршава после неколико корака итерације и губе се делови доброг материјала које имају „лоше“ јединке. Ову негативну појаву спречавамо тако што дупликатима додељујемо нуле фунције прилагођености и самим тим испадају из „игре“.

Проблем који се јавља у каснијој фази извршавања и који је супротан преурањеној конвергенцији је *спора конвергенција*. Разлог је што је разлика између најбоље и осталих јединки врло мала и постоји недовољни градијент у функцији прилагођености и самим тим алгоритам теже долази до оптималног решења.

1.3.2. Селекција. Као што је речено, у природи, јединке које су најприлагођеније средини опстају и репродукују се да би формирале наредну генерацију. Пролазе јединке са најбољим особинама, а потомци имају само добре особине родитеља па и боље. Међутим може се десити да потомци буду и лошији од родитеља, чак и да најбоља јединка не пређе у наредну генерацију. Начин помоћу којег бисмо спречили ову негативну појаву јесте увођење *стационарног генетског алгоритма (steady state)* у комбинацији са елитистичком стратегијом (*elitist strategy*) замене генерација. Овим је обезбеђено да једна или неколико добрих јединки прође у наредни круг без примене оператора селекције, укрштања, мутације.

Подржани су следећи типови селекције:

- *Проста рулет селекција* – најпростији облик оператора селекције. Дефинисана је тако да вероватноћа одабира јединке је директно пропорционална њеном квалитету. Ово и није баш најбоље решење јер је обим узорка на којем вршимо рулет селекцију мали и долази до фаворизовања јединки, њиховој доминацији што повлачи преурањену конвергенцију.
- *Селекција заснована на рангу - (rank based selection)* формира се низ рангова за одговарајуће позиције јединки у популацији, где је вредност ранга сваке јединке функција прилагођености. Предност ове селекције је да шансу у узграђивању потомства имају јединке код којих је вредност разлике у функцији циља велика, као и оне код којих је ова разлика мала.
- *Турнирска селекција* је најпопуларнија селекција. Селекција се врши преко турнира организован између јединки популације. Једини параметар је величина турнира N_{tur}, и он се унапред задаје. Изаберу се подскупови у популацији чија је кардиналност једнака параметру N_{tur}. У сваком подскупу се одржава турнир, одакле имамо „победничку“ јединку. Колико победника желимо да добијемо, толико ћемо и турнира одржати. Овако добијене јединке учествују у следећем процесу – процесу укрштања.

Као велики недостатак сматра се појављивање дупликата јединки у генерацији. Екперименти су показали да чак половина јединки може бити дупликат, чиме се алгоритам успорава. Проблем се решава, нпр, имплицитним брисањем свих вишеструких јединки у популацији. То се постиже додељивањем нуле прилагођеностима таквих јединки чиме је елиминисано њихово појављивање у следећој генерацији.

Други начин је модификација оператора селекције - *фино градирана турнирска селекција,* о којој ће бити мало више речи у одељку 2.

1.3.3. Укрштање. Дефинише се као поступак у којем се на случајан начин размењује генетски материјал - делови кодова (хромозома) два решења из родитеља и тако добијају два нова решења (деца). Нова генерација би требало да одржи добре особине претходне генерације. Оператор укрштања делујући на сваки пар родитеља са собом повлачи *вероватноћу укрштања* p_u на свакој позицији на хромозому. Ако је вероватноћа p_u једнака 0.5, онда хромозоми на том месту имају подједнаку шансу да се укрсте. Добијају се деца која имају особине родитељског пара са вероватноћом 1- p_u. Постоје:

- укрштање у једној тачки,
- укрштање у две тачке и

- униформно укрштање.

Могуће је дефинисати укрштање у *N* тачака, али се најчешће среће укрштање је у једној и две тачке. Сваки пар родитеља генерише „маску“, бинарни низ дужине као генетски кôд родитеља.

1.3.3.1. Униформно укрштање. На случајан начин се генерише *маска дужине N* за сваки пар јединки. Сваки бит маске може имати вредност 0 или 1. Бит маске са вредношћу 1 значи да се на тој позицији врши замена битова (хромозома) родитеља.

Алгоритам униформног укрштања може се описати следећим псеудокодом:

```
mi=0, za svako i=1,2,...,N;
        za svako i=1,2,...,N:
Odrediti_slucajnu_vrednost ξ iz [0,1];
        Ako je ξ≤pu, tada je mi=1;
Rezultat je vektor maske m'.
```

Пример униформног укрштања:

родитељ 1 родитељ 2	XXXXXXXXXX YYYYYYYYYY
маска	1 0 1 0 0 1 0 0 0 1
потомак 1 потомак 2	YXYXXYXXXY XYXYYXYYYX

1.3.3.2. Укрштање у једној тачки. На случајан начин бира се позиција једног бита. Након изабраног бита подстринг се размењује између хромозома.

родитељ 1 родитељ 2	XXXXXXXXXX YYYYYYYYYY
маска	0 0 0 0 0 0 1 1 1 1
потомак 1 потомак 2	XXXXXXYYYY YYYYYYXXXX

1.3.3.3. Укрштање у две тачке. На случајан начин се бира позиција два бита. Врши се размена подстрингова између два изабрана бита.

родитељ 1 родитељ 2	XXXXXXXXXX YYYYYYYYYY
Маска	0 0 1 1 1 1 0 0 0 0
потомак 1 потомак 2	XXYYYYXXXX YYXXXXYYYY

1.3.4. Мутација. Приликом селекције и укрштања јединки које ће дати нову генерацију, бивају одбачене јединке које имају мало „лошије“ особине, али ипак и добре. Мутацијом обезбеђујемо повраћај бар једног дела изгубљених гена. Мутација враћа дивергенцију у популацију, у смислу да спречава да популација конвергира и заврши у локалном минимуму. Врши се са одређеном вероватноћом која се назива *степен мутације* p_m. Код ГА, мутација се обично врши са малом вероватноћом: од 0.001 до 0.01. Примењују се два алгоритма мутације:

- случајна мутација
- уређена мутација

1.3.4.1. Случајна мутација. Позиција бита који мутира бира се на случајан начин и врши се операција негације гена.

0 1 **0** 1 0 1 0 1 0 **1**

0 1 **1** 1 0 1 0 1 0 **0**

1.3.4.2. Уређена мутација. Уређена мутација ради по следећем принципу: на случајан начин одаберу се позиције два бита. Мутација, односно негација врши се само на оне битове које се налазе између одабрана два.

0 0 1 **0** 1 **1** 0 1 0 1

0 0 1 **1** 0 **0** 0 1 0 1

1.3.5. Критеријум заустављања. Генетски алгоритми су стохастичке методе које врше претрагу простора допустивих решења. Пошто претрага може бити неограничена, тешко је формално дефинисати услов заустављања. Зато се он обично унапред задаје. Зауставни критеријум може бити:

- достигнут је унапред задати број генерација,
- сличност јединки у популацији,,
- понављање најбоље јединке одређени (максимални) број пута,
- достигнуто оптимално решење (ако је оно унапред познато),
- ограничено време извршавања ГА,
- прекид од стране корисника и др.

Пошто сваки од њих има добре и лоше стране, најбоље је комбиновати више њих.

1.3.6. Остали аспекти генетског алгоритма. Један од битних аспеката ГА је *политика замене генерација.* Примењују се следеће стратегије:

- *генерацијска* – у свакој генерацији мења све јединке у популацији,
- *стационарна* – где се део генерације замењује новим, а остале су непромењене из претходне генерације,
- *елитистичка* – политика у којој се део добре генерације штите од измене или елиминације током еволутивном процеса. Овакав механизам се назива и *елитизам.* Коришћењем елитизма добија се на уштеди времена изршавања, а може се и лакше проценити квалитет јединке.

Једна од могућности побољшања ГА је комбиновање са другим различитим хеуристикама. Комбиновање се може применити на самом почетку, у току извршавања генетског алгоритма у свакој генерацији, на неком њеном делу или само на једној јединки.

Друга могућност побољшања реализације ГА, његових резултата и времену извршавања се заснива на техникама паралелизације и кеширања (први пут је примењено у [Kra99a]), детаљније се може наћи у [Kra00].

Генералне особине ГА су: мала брзина извршавања, лако се паралелизују, на рачунару су једноставни за имплементацију, представљају добру хеуристику за комбинаторне проблеме. У посебно добре особине спадају: чување и коминовање информација од добрих родитеља (crossover) и могућност мењања перформанси у зависности од улазних параметара. Некада је спорији од других метода, али се може прекинути у било ком тренутку.

1.3.7. Примене ГА. Свеукупно гледајући врлине и примедбе генетских алгоритама, они су се успешно показали у решавању проблема комбинаторне оптимизације широког спектра. Примењују се у решавању **NP-** тешких проблема, вештачкој интелигенцији и унапређивању једноставних програма, али и уметности и музици. Од првих примена симулације еволуције у игрању једноставних игара, преко решавања комплексних инжењерских проблема помоћу стратегије еволуције. Налазе примену и у контроли робота, разним проблемима учења и оптимизације.

Неке од новијих примена су:

- дискретни локацијски проблеми [Alp03, Dre03, Fil00, Kra98, Kra99b, Kra00, Kra01, Sta07a],
- хаб локацијски проблеми [Fil06, Kra05, Kra06, Kra07],
- минимална довезана допуна графа [Lju00a, Lju00b, Lju01, Lju03, Lju04, Trm00],

- метричка димензија графа [Kra08a, Kra08b, Kra08c],
- уопштени проблеми придруживања [Chu97, Raid04, Sav97, Sav08a, Sav08b, Sav09],
- дизајн рачунарских мрежа [Kra02, Kra08d],
- проблем упаривања и распореда посаде авиона [Sua09],
- задовољивост формула [Ognj01, Ognj04b],
- вишедимензионално проблем ранца [Puc06, Rai05],
- минимална међузависност бинарних низова [Kov08],
- максимално балансирана повезана партиција графа [Đur08],
- избор индекса података [Kra03],
- проблем путујућег трговца [Gol09],
- рутирање и одабир преносиоца [Kra08c],
- проблем дискретно урађене медијане [Sta07a],
- покривање чворова графа [Mil08],
- проблем мултипроцесорске отворене продавнице [Mat09],
- проблем примања наруџбина [Rom09],
- бојење чворова графа [Juh06].

2. Проблем мултидимензионалног ранца

2.1 Дефиниција проблема

Као што је дефинисано формулама (1)-(3), проблем ранца се своди на налажење максималне вредности функције циља, што је конкретно, максимизовање његовог унутрашњег простора.

Ако $m = 1$, проблем се своди на „обичан“ проблем ранца, (описан у првом делу) који се решава динамичким програмирањем. За $m = 2$, проблем се назива *„проблем дводимензионог ранца“.*

У даљем тексту, сваки пут кад се помене „проблем ранца“, мисли се на „проблем мултидимензионалног ранца“.

У табели испод, дат је пример инстанце димензије 8x2. Врста „редни број предмета“ постоји да би се знало који је предмет укључен, а који није. Узимајући предмете редом, а поштујући ограничења, добијамо за вредност функције циља 17. Предмети који улазе у ранац су под редним бројевима 1,2 и 3. Предмет 4 или неки други се не може наћи у ранцу јер би прекорачио ограничење ранца. Решење се може представити векторски (1,1,1,0,0,0,0,0), где је јединица на месту предмета који је изабран.

Величина Инстанце	8	2							ограничења b_i
редни бр.предмета	1	2	3	4	5	6	7	8	
добици d_i	**8**	**6**	**3**	**5**	**9**	**3**	**14**	**7**	
карактеристике предмета	3	2	4	3	2	1	5	1	≤9
	1	1	2	4	2	1	3	3	≤5

2.2 Постојећи начини решавања

Проблем ранца се први пут појављује у оквиру капиталног буџета у раду [Lor55], [Man57]. Тачним методама, заснованим на идеји динамичког програмирања, проблем су решавали Гилмор и Гомори (Gilmore and Gomory, 1966), Вајгартнер и Нес (Weingartner and Ness 1967), као и Ших (Shih, 1979) методом гранања и ограничавања. Затим, Чу и Бизли (Chu&Beasley, 1998) хибридном еволуционом теоријом (EA). Више о овоме видети у радовима [Lor55], [Man57], [Gil66], [Wei67], [Shi79], [Chu98], редом.

Проблем је **NP-** тежак јер је уопштење проблема ранца за који је доказано у [Kar72] да је **NP-** тежак.

3. Предложени генетски алгоритам

Како су сви општи појмови генетских алгоритмама описани су у претходном одељку, у овој секцији ће бити описане само оне караткеристике које су примењене на дати проблем.

3.1 Кодирање и функција циља

Примењено је целобројно кодирање јединки, где се прво генерише пермутација од n предмета. Генетски кôд се састоји од $n-1$ гена, од којих сваки одређује један елемент пермутације. Последњи елемент пермутације је познат на основу n - 1 претходних елемената, па због тога није кодиран у генетском кôду. На основу вредности i -тог гена ($i = 1,\ldots,n$) , рачуна се вредност i -тог елемента g_i пермутације на следећи начин:

- Постојећи елементи пермутације (1 до $i-1$) се нотирају као искоришћени;
- На основу вредности i -тог гена g_i се добија i -ти елемент пермутације p_i, као g_i -та неискоришћена вредност;
- Пошто би се могло догодити да таква вредност не постоји ($g_i > n-i-1$), претходно се врши нормализација $g_i = g_i \bmod (n-i-1)$. На тај начин, вредност p_i увек постоји.

На пример, гену 3011 одговара пермутација $\begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 4 & 1 & 3 & 5 & 2 \end{pmatrix}$. Број 3 означава да први елемент у низу треба бити четврти неискоришћени од 1 2 3 4 5, а то је 4. Вратимо се на ген, следећи број је 0, а то је први неискоришћен из низа 1 2 3 5. И то је 1, као други елемент пермутације. Трећи елемент пермутације је други неискоришћен из низа 2 3 5, што је тројка. На исти начин ће четврти елемент пермутације бити 5, и последњи неискоришћен (одређен помоћу претходних четири) је број 2.

На основу пермутације, а имајући у виду ограничења ранца по сваком предмету, одређујемо који су предмети изабрани, на следећи начин:

- У сваком кораку ажурирамо преостале капацитете за сваку карактеристику. На почетку се преостали капацитет поставља на укупни капацитет за сваку карактеристику. За сваки изабрани предмет се преостали капацитети смањују за одговарајућу вредност у матрици D за сваку карактеристику;

- У кораку i се проверава да ли се предмет pi може убацити у ранац, односно да ли су преостали капацитети већи или једнаки од вредности у матрици D за предмет pi;
- Ако је одговор потврдан, предмет убацујемо у ранац, и ажурирамо преостале капацитете;
- Ако је одговор негативан, предмет не улази у садржај ранца, преостали капацитети остају непромењени.

С обзиром на то да је свако ограничење испоштовано, сва решења су допустива, па се не појављује проблем некоректних јединки.

На примеру из претходне секције, рачунамо функцију циља генерисану пермутацијом на случајан начин. Тренутна максимална вредност је 25, што је много боље од претходног рекорда. Изабрани предмети су под редним бројем 7, 1 и 6, тј: (1,0,0,0,0,1,1,0).

величина инстанце	8	2							ограничења b_i
редни бр.предмета	7	4	1	8	2	5	3	6	
добици d_i	**14**	**5**	**8**	**7**	**6**	**9**	**3**	**3**	
карактеристике предмета	5	3	3	1	2	2	4	1	≤9
	3	4	1	3	1	2	2	1	≤5

Пермутацијом добијамо нови низ предмета:

величина инстанце	8	2							ограничења b_i
редни бр.предмета	5	2	8	6	4	7	1	3	
добици d_i	**9**	**6**	**7**	**3**	**5**	**14**	**8**	**3**	
карактеристике предмета	2	2	1	1	3	5	3	4	≤9
	2	1	3	1	4	3	1	2	≤5

Оваква јединка за најбољу вредност ранца даје 26, што је и максимална функција циља. Изабрани предмети су: (1,1,0,0,1,1,0,0).

Проблем је прилично мале димензије, па се и рачун могао извести у пар корака. Са димензијом проблема расте и комплексност проблема, као и његово временско извршавање.

3.2 Генетски оператори

Пошто предложено кодирање и одговарајућа функција циља увек представља само допустива решења, омогућена је примена стандардних генетских оператора. Постоји велико искуство у решавању разних проблема помоћу ГА [Đur08, Fil08, Fil00, Fil06, Kra98, Kra99a, Kra99b, Kra00, Kra01, Kra02, Kra03, Kra05, Kra06, Kra07, Kra08a, Kra08b, Kra08d, Kra11, Ognj01, Ognj04b, Sav08a, Sav08b, Sav09, Sav10, Sta07a].

На основу природе проблема и претходног искуства изабрани су: фино градирана турнирска селекција, једнопозиционо укрштање и проста мутација са залеђеним генима.

За оператор **селекције** одабрана је фино-градирана турнирска селекција, описана у [Fil98], [Fil03],[Fil06]. Као што је описано, селекција се врши преко турнира организованих између јединки популације. Једини параметар је величина турнира N_{tur}, и он се унапред задаје. Изаберу се подскупови у популацији чија је кардиналност једнака параметру N_{tur}. У сваком подскупу се одржава турнир, одакле имамо „победничку“ јединку. Колико победника желимо да добијемо, толико ћемо и турнира одржати. Овако добијене јединке учествују у следећем процесу – процесу укрштања.

Модификован оператор селекције - *фино градирана турнирска селекција,* уместо целобројног параметра турнирске селекције, овде се користи жељена *просечна* величина турнира N_{ftur}. Нпр, ако је $N_{ftur} = 5.5$ и уколико је укупан број турнира 20, онда можемо бирати десет турнира величине 5, и десет турнира величине 6.

Код оператора **укрштања** користи се једнопозиционо укрштање. На случајан начин бира се позиција једног бита неелитне јединке. Након изабраног бита подстринг се размењује између хромозома. Укрштање се врши са вероватноћом $p_u=0.85$.

Код оператора **мутације** користи се случајна мутација, која спречава губитак генетског материјала. Мутација се врши тако што се на случајан начин мењају одређени гени. Дешава се да одређен број јединки има исти ген на истој позицији. Ова врста гена је позната као „замрзнути гени“. Оператори селекције и укрштања не могу деловати на замрзнуте гене. Зато је укључена модификација која их обрађује. Лоша особина замрзнутих гена је што сужавају простор претраге за једну целу димензију, повећавају вероватноћу мутације и негативну појаву – превремена конвергенција. ГА се може побољшати повећањем учесталости мутације само за замрзнуте гене. Ови гени се одређују у свакој генерацији.

3.3 Остали аспекти ГА

У оквиру генетског алгоритма, такође су примењене неке од раније познатих техника за побољшавање решења и убрзавање рада. Стационарни генетски алгоритам са елитистичком стратегијом очувава добра решења и значајно доприноси добром квалитету коначног решења. Уклањање вишеструке појаве истих јединки и ограничавање појаве различитих јединки са истом вредношћу функције циља ефективно доприноси избегавању преурањене конвергенције у неком локалном максимуму.

Популација се састоји од 150 јединки (почетна популација се генерише на случајан начин). Примењен је стационарни ГА са елитистичком стратегијом, где се у свакој генерацији мења само једна трећина популације (50 јединки), док две трећине најбољих јединки („елитне јединке") директно прелази у следећу генерацију без примене генетских оператора. На тај начин се чувају добра решења од могуће „несрећне" примене генетских оператора. Пошто елитне јединке остају непромењене током генерација, вредност функције циља се директно преноси из претходне генерације, па нема потребе за поновним рачунањем. Детаљни опис ове технике може се видети у [Kra00], [Kra01].

Као што је речено, приликом извршавања генетског алгоритма, може доћи до негативне појаве преурањена конвергенција (као доминација јединке). Још, може се десити да јединке са истом функцијом циља, а различитим генетским кодовима доминирају у популацији. У сваком случају, генетски алгоритам може завршити у локалном оптимуму. Први проблем решава се тако што се јединкама додељују нуле прилагођености функције, а други тако што ће се ограничити број појављивања оваквих јединки на неку константу. У овој имплементацији број оваквих јединки је ограничен на 40.

Пре свега, треба нам оцена квалитета јединке, тј. **функција прилагођености**. У предложеном алгоритму, вредности јединки се скалирају у интервал [0,1] тако да најбоља јединка има прилагођеност вредности 1, а најлошија 0. У случају проблема максимизације користи се следећа формула:

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

Најбоља јединка је она са највећом вредношћу x_{max}, јединка са најмањом вредношћу x_{min} и x текућа вредност јединке у популацији. Овако задата функција прилагођености није дефинисана ако све јединке имају исту вредност. Ово ће се десити само ако су све јединке исте, а то знати да алгоритам конвергира у тој вредности, па и није потребно даље извршавање алгоритма. Рад ГА се прекида и штампа се извештај.

У циљу спречавања рачунања вредности функције циља за јединке које имају исти кôд у различитим генерацијама, уводи се техника **кеширања,** [Kra99],[Kra01]. Јединке за које је вредност функције већ израчуната смештају се у кеш меморију. Другим речима, за време извршавања ГА, проверава се да ли је за текућу јединку раније израчуната вредност функције прилагођености. Ако јесте, онда се та вредност узима из кеш меморије. У супротном, рачуна се вредност функције циља и смешта се у кеш меморију.

4. Екпериментални резултати

Сва тестирања и израчунавања вршена су на једном процесору рачунара Intel Quad Core 2.5 GHz, са 1GB радне меморије. Алгоритам је кодиран у програмском језику **C**. Тестиране су неке инстанце из литературе, димензије до 2500 предмета и до 100 карактеристика. Могу се наћи на интернет адресама:
http://hces.bus.olemiss.edu/tools.html,
http://people.brunel.ac.uk/~mastjjb/jeb/orlib/files/

Прва колона садржи назив инстанци; у другој колони је димензија проблема, и то у форми: број предмета и број карактеристика; у трећој колони су уписани или најбољи познати резултати или оптимално решење, у зависности од литературе; четврта, пета и шеста колона, редом, описују резултате генетског алготима, за које време је добијено најбоље решење и укупно време извршавања алгоритма; колона **rel** означава процентуалну грешку у добијеним ГА резултатима и **sigma** означава средње квадратно одступање, изражено у процентима. Критеријум заустављања био је када број генерација достигне број 5000 генерација са највише 1000 понављања најбоље јединке.

Табела 1. Експериментални резултати за инстанце димензије до *(n,m)=(2500,100),* преузетих са http://hces.bus.olemiss.edu/tools.html

назив инстанце	димензија		најбоље познато	ГА	t (s)	ttot (s)	rel (%)	sigma (%)
	пред.	кар						
mk_gk01	100	15	3766	3729	2,831	5,367	0,809	0,511
mk_gk02	100	25	3958	3912	2,815	5,395	0,54	0,399
mk_gk03	150	25	5650	5577	6,292	11,218	0,409	0,334
mk_gk04	150	50	5764	5696	6,488	12,008	0,531	0,285
mk_gk05	200	25	7557	7440	10,091	17,268	0,354	0,191
mk_gk06	200	50	7672	7568	9,728	18,413	0,328	0,262
mk_gk07	500	25	19215	18848	60,024	91,648	0,414	0,261
mk_gk08	500	50	18801	18498	64,251	97,137	0,329	0,138
mk_gk09	1500	25	58085	56727	756,956	897,036	0,185	0,118
mk_gk10	1500	50	57292	56169	728,184	898,514	0,132	0,073
mk_gk11	2500	100	95231	93703	2299,114	2795,112	0,134	0,068

Тестиране инстанце су максималне димензије 2500 и 100. Генетски алгоритам је добио резултате доброг квалитета. Највећа разлика у решењу се јавља код инстанци изразито већих димензија. Као „екстремни случајеви“ издвајају се инстанце mk_gk09, mk_gk010 и mk_gk11, где је процентуална разлика у решењу 2.36%, 1.96% и 1.6%, редом. Узимајући у обзир квалитет решења, као и добро време извршавања, ГА се сматра успешним.

Табела 2. Експериментални резултати за инстанце мање димензије преузетих са http://shah.freeshell.org/gamultiknapsack/

назив инстанце	димензија		Оптимално	ГА	t (s)	ttot (s)	rel (%)	sigma (%)
	пред.	кар.						
hp1	28	4	3418	*opt*	0,112	0,647	0,06	0,147
hp2	35	4	3186	*opt*	0,312	1,019	0,348	0,55
pb1	27	4	3090	*opt*	0,091	0,612	0,165	0,286
pb2	34	4	3186	*opt*	0,124	0,828	0,159	0,319
pb4	29	2	95168	*opt*	0,006	0,528	0	0
pb5	20	10	2139	*opt*	0,009	0,399	0	0
pb6	40	30	776	*opt*	0,215	1,152	0,664	0,843
pb7	37	30	1035	*opt*	0,186	1,078	0,633	0,98
sent01	60	30	7772	*opt*	1,059	2,487	0,95	1,522
sent02	60	30	8722	*opt*	1,004	2,579	0,522	0,548
weing1	28	2	141278	*opt*	0,101	0,611	0,147	0,351
weing2	28	2	130883	*opt*	0,063	0,577	0,312	0,57
weing3	28	2	95677	*opt*	0,106	0,619	0,883	1,17
weing4	28	2	119337	*opt*	0,071	0,596	0,079	0,353
weing5	28	2	98796	*opt*	0,255	0,744	2,148	1,673
weing6	28	2	130623	*opt*	0,124	0,629	0,511	0,554
weing7	105	2	1095445	1095206	1,956	4,539	0,265	0,219
weing8	105	2	624319	618901	1,948	4,569	3,43	3,583

Као што се види из **Табеле 2.**, генетски алгоритам је добио најбољи резултат у 18 од 20 случајева. Време извршавања није веће од 5 секунди, и релативна грешка је мања од 4, па се сматра да је ГА био успешан.

Табела 3.1. Експериментални резултати за инстанце мање димензије преузетих са http://shah.freeshell.org/gamultiknapsack/

назив инстанце	димензија		оптимално	ГА	t (s)	ttot (s)	rel (%)	sigma (%)
	пред.	кар.						
weish01	30	5	4554	*opt*	0,100	0,645	0,661	0,769
weish02	30	5	4536	*opt*	0,096	0,643	0,122	0,213
weish03	30	5	4115	*opt*	0,092	0,645	0,162	0,415
weish04	30	5	4561	*opt*	0,051	0,599	0,294	0,37
weish05	30	5	4514	*opt*	0,085	0,637	0,06	0,267
weish06	40	5	5557	*opt*	0,198	1,009	0,116	0,165
weish07	40	5	5567	*opt*	0,328	1,140	0,469	0,876
weish08	40	5	5605	*opt*	0,168	0,980	0,044	0,090
weish09	40	5	5246	*opt*	0,229	1,028	0,193	0,421
weish10	50	5	6339	*opt*	0,342	1,371	0,759	1,052
weish11	50	5	5643	*opt*	0,337	1,362	2,056	3,014
weish12	50	5	6339	*opt*	0,408	1,420	1,656	1,399
weish13	50	5	6159	*opt*	0,398	1,425	1,766	2,317
weish14	60	5	6954	*opt*	0,669	1,911	1,312	0,757
weish15	60	5	7486	*opt*	0,599	1,820	1,789	1,596
weish16	60	5	7289	7287	0,398	1,640	1,16	1,282
weish17	60	5	8633	*opt*	0,746	2,005	0,402	0,376
weish18	70	5	9580	*opt*	1,115	2,738	0,837	0,777
weish19	70	5	7698	*opt*	1,282	2,843	2,175	1,572
weish20	70	5	9450	*opt*	1,143	2,721	1,100	0,92

Табела 3.2. Експериментални резултати за инстанце мање димензије, преузете са интернет адресе: http://shah.freeshell.org/gamultiknapsack/

назив инстанце	димензија		Оптимално	ГА	t (s)	ttot (s)	rel (%)	sigma (%)
	пред.	кар.						
weish21	70	5	9074	9033	1,024	2,575	2,599	1,533
weish22	80	5	8947	8929	1,022	2,930	2,607	2,098
weish23	80	5	8344	*Opt*	1,839	3,631	3,056	2,557
weish24	80	5	10220	10215	1,838	3,640	1,493	1,042
weish25	80	5	9939	9928	1,817	3,509	0,99	0,948
weish26	90	5	9584	9563	1,937	3,918	2,33	1,718
weish27	90	5	9819	9721	2,546	4,539	2,592	1,659
weish28	90	5	9492	*opt*	1,339	3,477	3,459	2,67
weish29	90	5	9410	*opt*	2,145	4,060	4,39	3,411
weish30	90	5	11191	11182	1,975	4,047	1,262	0,982
mknap1_1	6	10	3800	*opt*	0,000	0,186	0	0
mknap1_2	10	10	8706,1	*opt*	0,000	0,256	0	0
mknap1_3	15	10	4015	*opt*	0,008	0,293	0	0
mknap1_4	20	10	6120	*opt*	0,042	0,434	0,041	0,09
mknap1_5	28	10	12400	*opt*	0,114	0,687	0,44	0,775
mknap1_6	39	5	10618	*opt*	0,146	0,965	0,162	0,161
mknap1_7	50	5	16537	*opt*	0,644	1,696	0,151	0,206

У **Табели 3.1.** и **Табели 3.2.** тестиране су инстанце максималне инстанце 50x5. Најбољи резултат није остварен у само 8 од 37 случајева, што је мало више од 20% укупног броја инстанци, а највећа разлика у резултату је изражена у инстанцама weish21 и weish27, са вредношћу 41 и 98. Процентуална грешка није већа од 1%, тако да се може сматрати да је ГА и у овом случају био успешан.

Табела 4. Експериментални резултати за инстанце димензије *(n,m) = (100,5)*

назив инстанце	димензија		најбоље познато	ГА	t (s)	ttot (s)	rel (%)	sigma (%)
	пред.	кар.						
mknapcb1_01	100	5	24381	23737	2,172	4,490	2,307	1,378
mknapcb1_02	100	5	24274	23556	2,332	4,688	2,261	1,997
mknapcb1_03	100	5	23551	23121	2,109	4,588	3,153	1,526
mknapcb1_04	100	5	23534	23123	1,725	4,104	1,453	1,024
mknapcb1_05	100	5	23991	23513	2,299	4,549	2,583	1,411
mknapcb1_06	100	5	24613	24315	2,074	4,463	2,203	1,065
mknapcb1_07	100	5	25591	24860	1,482	3,952	2,464	1,470
mknapcb1_08	100	5	23410	22771	2,917	5,028	2,402	1,372
mknapcb1_09	100	5	24216	23658	2,311	4,840	2,844	1,587
mknapcb1_10	100	5	24411	23918	2,276	4,627	2,178	1,440
mknapcb1_11	100	5	42757	42484	2,668	4,992	1,506	1,188
mknapcb1_12	100	5	42545	42078	3,080	5,227	1,707	0,938
mknapcb1_13	100	5	41968	41514	2,759	5,053	1,184	0,760
mknapcb1_14	100	5	45090	44420	2,437	4,804	1,245	0,976
mknapcb1_15	100	5	42218	41851	1,978	4,445	1,869	1,028
mknapcb1_16	100	5	42927	42759	2,820	5,035	1,786	0,917
mknapcb1_17	100	5	42009	41393	2,573	4,876	1,139	0,595
mknapcb1_18	100	5	45020	44499	2,702	5,050	1,375	0,917
mknapcb1_19	100	5	43441	43135	1,928	4,462	1,105	0,575
mknapcb1_20	100	5	44554	44027	1,888	4,328	1,013	0,589

Тестиране инстанце су димензије 100x5. Резултати генетског алгоритма су блиски најбољим познатим решењима, највећа разлика јавља се у решењу код инстанце mknapcb1_02, а релативна грешка није већа од 0.03, па се генетски алгоритам сматра успешним.

Табела 5. Експериментални резултати за инстанце димензије *(n,m) = (250,5)*

назив инстанце	димензија		најбоље познато	ГА	t (s)	ttot (s)	rel (%)	sigma (%)
	пред.	кар.						
mknapcb2_01	250	5	59312	57068	13,443	21,870	4,137	1,988
mknapcb2_02	250	5	61472	58003	11,364	20,702	2,476	1,788
mknapcb2_03	250	5	62130	58710	12,368	22,452	3,501	1,459
mknapcb2_04	250	5	59446	57582	13,381	22,128	3,437	1,572
mknapcb2_05	250	5	58951	56653	11,188	20,385	3,284	1,932
mknapcb2_06	250	5	60056	56818	14,676	23,282	2,28	1,125
mknapcb2_07	250	5	60414	57211	16,256	24,180	2,339	1,586
mknapcb2_08	250	5	61472	58692	12,793	21,392	2,75	1,712
mknapcb2_09	250	5	61885	58621	14,125	22,647	2,38	1,845
mknapcb2_10	250	5	58959	56352	10,052	20,095	3,407	2,143
mknapcb2_11	250	5	109109	106194	13,944	22,325	1,995	0,95
mknapcb2_12	250	5	109841	106420	13,152	21,442	1,203	0,846
mknapcb2_13	250	5	108489	105243	16,029	23,858	1,515	0,782
mknapcb2_14	250	5	109383	107014	20,339	25,817	1,722	0,963
mknapcb2_15	250	5	110720	107567	13,736	22,824	1,426	0,834
mknapcb2_16	250	5	110256	106562	14,982	23,236	1,028	0,886
mknapcb2_17	250	5	109016	106513	16,767	25,094	1,46	0,952
mknapcb2_18	250	5	109037	105861	11,586	20,984	1,31	1,023
mknapcb2_19	250	5	109957	106705	14,748	23,297	1,55	1,056
mknapcb2_20	250	5	107038	103538	18,757	25,404	1,24	0,637

Тестиране инстанце су димензије 250x5; највећа разлика у решењу ГА и најбољег познатог решења се огледа у инстанци mknapcb2_02; релативна грешка у овом случају износи 5.643.

Табела 6. Експериментални резултати за инстанце димензије *(n,m) = (500,5)*

назив инстанце	димензија		најбоље познато	ГА	t (s)	ttot (s)	rel (%)	sigma (%)
	пред.	кар.						
mknapcb3_01	500	5	120130	111457	62,784	88,876	2,637	1,366
mknapcb3_02	500	5	117837	109630	52,156	83,639	2,812	1,725
mknapcb3_03	500	5	121109	113247	55,000	84,635	2,932	2,044
mknapcb3_04	500	5	120798	112154	59,711	84,585	2,356	1,885
mknapcb3_05	500	5	122319	113575	58,594	86,734	2,505	1,665
mknapcb3_06	500	5	122007	112611	59,888	87,462	2,913	1,965
mknapcb3_07	500	5	119113	109745	52,889	87,491	3,163	2,019
mknapcb3_08	500	5	120568	111231	61,280	83,460	3,343	1,803
mknapcb3_09	500	5	121575	112091	57,344	86,613	2,27	1,775
mknapcb3_10	500	5	120699	113084	54,226	85,487	4,89	2,526
mknapcb3_11	500	5	218422	210108	64,626	88,966	1,531	1,024
mknapcb3_12	500	5	221191	211810	62,889	88,347	1,391	1,05
mknapcb3_13	500	5	217534	208287	69,639	90,100	2,43	1,07
mknapcb3_14	500	5	223558	215285	67,772	91,769	2,1	1,17
mknapcb3_15	500	5	218962	210432	61,836	89,230	2,321	1,131
mknapcb3_16	500	5	220514	211231	70,380	91,441	1,198	0,786
mknapcb3_17	500	5	219987	210961	74,682	96,168	2,027	1,211
mknapcb3_18	500	5	218194	208973	72,038	91,616	1,197	0,847
mknapcb3_19	500	5	216976	207565	67,341	90,966	1,09	0,68
mknapcb3_20	500	5	219693	210575	70,354	94,245	1,436	0,887

Инстанце су димензије 500x5. С обзиром на величину инстанце, релативна грешка је прилично мала и њена највећа вредност се огледа у инстанци mknapcb3_09. У овом „најгорем“ случају релативна грешка није већа од 0.079%, па се сматра да је ГА опет био успешан.

Табела 7. Експериментални резултати за инстанце димензије $(n,m) = (100,10)$

назив инстанце	димензија		најбоље познато	ГА	t (s)	ttot (s)	rel (%)	sigma (%)
	пред.	кар.						
mknapcb4_01	100	10	23064	22477	1,263	3,932	2,623	1,881
mknapcb4_02	100	10	22801	22095	1,501	4,109	2,899	1,414
mknapcb4_03	100	10	22131	21445	3,093	5,242	1,743	1,273
mknapcb4_04	100	10	22772	22270	2,142	4,737	2,016	1,271
mknapcb4_05	100	10	22751	22209	1,937	4,604	2,559	1,534
mknapcb4_06	100	10	22777	22111	2,599	4,862	2,008	1,187
mknapcb4_07	100	10	21875	20966	1,686	4,127	2,126	1,362
mknapcb4_08	100	10	22635	21998	2,048	4,540	2,627	1,574
mknapcb4_09	100	10	22511	21251	1,646	4,300	2,145	1,450
mknapcb4_10	100	10	22702	22074	1,813	4,353	3,277	1,663
mknapcb4_11	100	10	41395	40652	2,343	4,856	1,319	0,761
mknapcb4_12	100	10	42344	41585	2,805	4,976	0,807	0,507
mknapcb4_13	100	10	42401	41258	1,934	4,547	1,457	0,870
mknapcb4_14	100	10	45624	44976	2,736	5,120	1,692	0,793
mknapcb4_15	100	10	41884	41308	2,386	4,764	1,937	1,094
mknapcb4_16	100	10	42995	42139	2,945	5,154	1,473	0,960
mknapcb4_17	100	10	43559	42887	2,576	5,002	1,106	1,066
mknapcb4_18	100	10	42970	42160	3,907	5,975	1,321	1,037
mknapcb4_19	100	10	42212	41424	2,672	4,942	1,390	1,031
mknapcb4_20	100	10	41207	40405	2,168	4,556	1,639	1,115

Као што се види из **Табеле 7**, генетски алгоритам је добио решења блиска оптималним. Највећа разлика у решењу је у инстанцама mknapcb4_09 и mknapcb4_13, релативна грешка је око 0.056, (тачније 0,055973) па се резултати сматрају добрим, а алгоритам успешан и у овом случају.

Табела 8. Експериментални резултати за инстанце димензије *(n,m) = (250,10)*

назив инстанце	димензија		најбоље познато	ГА	t (s)	ttot (s)	rel (%)	sigma (%)
	пред.	кар.						
mknapcb5_01	250	10	59187	55102	12,945	22,169	2,976	1,656
mknapcb5_02	250	10	58662	55520	12,950	22,779	1,909	0,911
mknapcb5_03	250	10	58094	53974	14,723	23,928	3,195	1,843
mknapcb5_04	250	10	61000	57062	12,189	21,600	3,211	2,060
mknapcb5_05	250	10	58092	54304	10,959	21,073	1,924	1,592
mknapcb5_06	250	10	58803	55247	11,243	20,348	2,431	1,575
mknapcb5_07	250	10	58607	55739	11,785	21,069	2,889	1,446
mknapcb5_08	250	10	58917	56044	9,954	20,665	3,527	2,362
mknapcb5_09	250	10	59384	56534	10,790	20,785	3,929	2,718
mknapcb5_10	250	10	59193	55960	11,379	21,276	2,857	1,992
mknapcb5_11	250	10	110863	107276	13,855	23,949	1,456	0,878
mknapcb5_12	250	10	108659	104850	10,976	21,028	1,861	1,037
mknapcb5_13	250	10	108932	105668	16,301	24,488	1,842	0,891
mknapcb5_14	250	10	110037	105891	17,208	25,046	1,672	0,820
mknapcb5_15	250	10	108423	105356	14,554	23,511	1,693	0,860
mknapcb5_16	250	10	110841	106947	18,679	26,256	1,223	0,671
mknapcb5_17	250	10	106075	102586	11,772	21,655	1,960	1,124
mknapcb5_18	250	10	106686	102551	15,577	23,555	1,281	1,067
mknapcb5_19	250	10	109825	106573	14,580	23,158	2,052	0,897
mknapcb5_20	250	10	106723	102184	15,305	24,331	1,160	0,895

Иако генетски алгоритам није достигао најбоље познато решење ни у једном од двадесет тестираних инстанци, опет, решења су довољно блиска познатима, и највећа релативна грешка је у случају инстанце mknapcb5_03 0.07092, што је одступање од 7,091% одступања.

Табела 9. Експериментални резултати за инстанце димензије *(n,m) = (500,10)*

назив инстанце	димензија		најбоље познато	ГА	t (s)	ttot (s)	rel (%)	sigma (%)
	пред.	кар.						
mknapcb6_01	500	10	117726	108547	59,954	89,705	2,860	1,793
mknapcb6_02	500	10	119139	108407	54,191	84,975	2,715	1,582
mknapcb6_03	500	10	119159	110926	56,451	87,706	2,499	1,800
mknapcb6_04	500	10	118802	109063	51,257	83,829	2,095	1,380
mknapcb6_05	500	10	116434	109909	59,226	91,543	3,518	1,661
mknapcb6_06	500	10	119454	111293	61,246	88,934	3,384	1,511
mknapcb6_07	500	10	119749	111450	55,430	88,443	3,124	2,011
mknapcb6_08	500	10	118288	111172	61,695	91,809	3,380	1,560
mknapcb6_09	500	10	117779	109738	74,648	92,739	2,343	1,742
mknapcb6_10	500	10	119125	110114	50,598	83,997	2,914	1,587
mknapcb6_11	500	10	217318	207459	52,891	86,138	1,722	0,968
mknapcb6_12	500	10	219022	210411	72,813	95,973	2,687	1,000
mknapcb6_13	500	10	217772	206930	74,877	94,640	1,493	1,146
mknapcb6_14	500	10	216802	205493	74,338	94,978	1,175	0,940
mknapcb6_15	500	10	213809	203360	65,840	92,725	1,409	0,948
mknapcb6_16	500	10	215013	206048	62,282	89,456	2,186	1,143
mknapcb6_17	500	10	217896	208487	73,442	94,321	2,016	0,992
mknapcb6_18	500	10	219949	210592	76,882	96,848	1,958	0,986
mknapcb6_19	500	10	214332	204923	53,473	83,958	2,353	1,229
mknapcb6_20	500	10	220833	209737	65,209	89,780	1,499	1,213

Инстанце које имају највећу разлику у решењу су инстанце mknapcb6_13, mknapcb6_20 и mknapcb6_14. Највећа разлика је 11309, што је неких 5,21% одступања од најбољег решења.

Табела 10. Експериментални резултати за инстанце димензије *(n,m) = (100,30)*

назив инстанце	димензија		најбоље познато	ГА	t (s)	ttot (s)	rel (%)	sigma (%)
	пред.	кар.						
mknapcb7_01	100	30	21946	21243	2,662	5,155	3,479	1,942
mknapcb7_02	100	30	21716	21023	2,003	4,760	2,722	1,433
mknapcb7_03	100	30	20754	19955	1,761	4,503	2,884	1,562
mknapcb7_04	100	30	21464	20787	1,485	4,306	2,081	1,277
mknapcb7_05	100	30	21814	21222	2,357	4,969	2,732	1,272
mknapcb7_06	100	30	22176	21412	1,905	4,553	2,402	1,820
mknapcb7_07	100	30	21799	20945	1,510	4,321	1,741	1,480
mknapcb7_08	100	30	21397	20829	2,374	4,993	2,313	1,849
mknapcb7_09	100	30	22493	21805	2,289	5,004	2,792	1,110
mknapcb7_10	100	30	20983	20265	2,157	4,924	3,040	1,787
mknapcb7_11	100	30	40767	39598	2,253	5,194	0,784	0,660
mknapcb7_12	100	30	41304	40516	3,061	5,806	2,141	1,317
mknapcb7_13	100	30	41560	40678	2,961	5,677	1,196	0,791
mknapcb7_14	100	30	41041	40263	2,408	5,481	1,894	0,915
mknapcb7_15	100	30	40872	40034	2,529	5,338	1,055	0,743
mknapcb7_16	100	30	41058	40090	3,157	6,008	1,260	1,005
mknapcb7_17	100	30	41062	40187	2,792	5,710	2,069	1,156
mknapcb7_18	100	30	42719	41826	2,594	5,383	2,049	1,102
mknapcb7_19	100	30	42230	41416	1,976	4,967	1,993	1,046
mknapcb7_20	100	30	41700	40759	3,785	6,235	1,476	0,992

Инстанце димензије 100 и 30: већа димензија инстанце захтева и време извршавања, чије су вредности у колони 5 прилично мале. Још, највећа релативна грешка у ових двадесет инстанци није већа од 0,0385, чиме се потврђује успех алгоритма.

Табела 11. Експериментални резултати за инстанце димнзије *(n,m) = (250,30)*

назив инстанце	димензија		најбоље познато	ГА	t (s)	ttot (s)	rel (%)	sigma (%)
	пред.	кар.						
mknapcb8_01	250	30	56693	54162	12,286	23,215	4,302	1,866
mknapcb8_02	250	30	58318	55744	16,510	24,447	3,369	2,303
mknapcb8_03	250	30	56553	53368	14,942	24,607	3,153	1,825
mknapcb8_04	250	30	56863	53407	16,183	25,350	3,071	2,172
mknapcb8_05	250	30	56629	52514	9,048	20,673	2,993	1,530
mknapcb8_06	250	30	57119	54345	17,175	26,496	2,235	1,888
mknapcb8_07	250	30	56292	53206	12,422	22,783	2,777	2,022
mknapcb8_08	250	30	56403	52969	9,796	21,319	2,929	1,920
mknapcb8_09	250	30	57442	54267	12,123	22,397	2,756	1,697
mknapcb8_10	250	30	56447	53680	15,935	24,566	3,732	1,892
mknapcb8_11	250	30	107689	104365	16,247	26,072	2,260	1,442
mknapcb8_12	250	30	108338	103645	14,941	24,828	0,981	0,773
mknapcb8_13	250	30	106385	103569	18,362	27,692	2,596	1,173
mknapcb8_14	250	30	106796	102601	19,470	27,176	1,177	0,763
mknapcb8_15	250	30	107396	103898	17,799	27,189	2,245	1,110
mknapcb8_16	250	30	107246	103977	19,298	27,969	1,816	1,086
mknapcb8_17	250	30	106308	102497	17,981	27,581	1,726	1,360
mknapcb8_18	250	30	103993	100768	16,996	27,631	1,003	0,840
mknapcb8_19	250	30	106835	103409	21,122	29,161	1,669	0,871
mknapcb8_20	250	30	105751	101395	15,435	25,639	1,303	1,094

Слично као у претходној табели, алгоритам је достигао задовољавајућа решења, инстанца која се издваја је mknapcb8_07, са релативном грешком од 0,073, док остале имају релативну грешку мању од 0.06.

Табела 12. Експериментални резултати за инстанце димензије *(n,m) = (500,30)*

назив инстанце	Димензија		најбоље познато	ГА	t (s)	ttot (s)	rel (%)	sigma (%)
	пред.	кар.						
mknapcb9_01	500	30	115868	107419	59,797	90,734	2,289	1,482
mknapcb9_02	500	30	114667	107435	66,914	97,448	3,753	2,907
mknapcb9_03	500	30	116661	108539	69,446	93,982	2,465	1,601
mknapcb9_04	500	30	115237	106833	51,358	89,666	2,031	1,480
mknapcb9_05	500	30	116353	108027	51,077	83,825	4,168	2,155
mknapcb9_06	500	30	115604	107098	60,606	88,410	2,565	1,749
mknapcb9_07	500	30	113952	105875	70,356	97,372	2,580	1,251
mknapcb9_08	500	30	114199	105568	53,166	86,013	2,194	1,825
mknapcb9_09	500	30	115247	107400	56,290	87,116	3,975	2,627
mknapcb9_10	500	30	116947	108578	52,652	84,422	2,930	1,658
mknapcb9_11	500	30	217995	206418	64,622	93,222	1,834	1,023
mknapcb9_12	500	30	214534	203537	69,833	95,617	1,582	1,414
mknapcb9_13	500	30	215854	205166	80,103	99,975	1,793	0,901
mknapcb9_14	500	30	217836	209219	72,377	99,663	2,128	1,386
mknapcb9_15	500	30	215566	204030	80,310	101,684	1,443	1,078
mknapcb9_16	500	30	215762	206269	71,004	99,524	2,117	0,845
mknapcb9_17	500	30	215772	205743	68,648	96,208	2,076	1,324
mknapcb9_18	500	30	216336	205035	62,380	92,958	1,304	0,872
mknapcb9_19	500	30	217290	204688	64,010	93,314	1,133	0,962
mknapcb9_20	500	30	214624	203386	74,688	102,011	1,451	0,830

И последње инстанце облика mknapcb и димензија 500 и 30. У тринаест од двадесет случајева, релативна грешка је мања од 0,06, време добијања најбоље јединке и извршавања читаве инстанце је задовољавајућег квалитета.

5. Закључак

У раду је описан генетски алгоритам на решавање проблема вишедимензионог ранца, једног од најпознатијих проблема у комбинаторној оптимизацији.

Примењено је целобројно кодирање јединки, где се прво генерише пермутација од n предмета. Затим се на основу пермутације, а имајући у виду ограничења ранца по сваком предмету, одређује да ли је предмет изабран или није. На тај начин је свако решење допустиво, па се не појављује проблем некоректних јединки. Пошто нема недоспустивих решења, омогућена је примена стандардних генетских оператора: фино градирана турнирска селекција, једнопозиционо укрштање и проста мутација са залеђеним генима.

У оквиру генетског алгоритма, такође су примењене неке од раније познатих техника за побољшавање решења и убрзавање рада. Стационарни генетски алгоритам са елитистичком стратегијом очувава добра решења и значајно доприноси добром квалитету коначног решења. Уклањање вишеструке појаве истих јединки и ограничавање појаве различитих јединки са истом вредношћу функције циља ефективно доприноси избегавању преурањене конвергенције у неком локалном максимуму. Кеширање ГА у доброј мери смањује време извршавања, а не утиче на остале аспекте рада генетског алгоритма.

Експериментални резултати су извршени на инстанцама предложеним у литератури. За инстанце веће димезије не знају се оптимална решења, али су обично познати најбољи досадашњи резултати. Као што се може видети из табела са резултатима, генетски алгоритам је добио решења задовољавајућег квалитета, пошто је достигао скоро сва оптимална решења. И на већим инстанцама за које се не зна оптимално решење, ГА је био врло успешан. Време извршавања је релативно кратко, осим у случају једне инстанце екстремно велике димензије.

Даљи рад се може одвијати у следећим правцима:

- решавање проблема помоћу других метахеуристика,
- модификација генетског алгоритма за решавање сродних проблема,
- паралелизација и извршавање на вишепроцесорском рачунару.

У раду осим стручног постоји и научни допринос који се огледа у новом начину кодирања за дати проблем. При томе је предложена фунцкија циља где су све јединке коректне, чиме је омогућена ефикасна имплементација.

Као што се види из резултата, овакав комбинаторни приступ је ефикасан и успешан у решавању проблема вишедимензионог ранца. Због свега горе описаног, у раду је дат допринос већем броју области и то: решавању проблема ранца, метахеуристика и комбинаторној оптимизацији.

Литература

[Alp03] **Alp O., Erkut E.**, “An Efficient Genetic Algorithm for the p-Median Problem”, Annals of Operations Research, Kluwer Academic Publishers (2003).

[Bäc00a] **Bäck T., Fogel D. B., Michalewicz Z.**, "Basic Algorithms and Operators", in: Evolutionary Computation 1, Institute of Physics Publishing, Bristol-Philadelphia (2000).

[Bäc00b] **Bäck T., Fogel D. B., Michalewicz Z.**, "Advanced Algorithms and Operators", In: Evolutionary Computation 2, Institute of Physics Publishing, Bristol-Philadelphia, (2000), http://msmga.ms.ic.ac.uk/jeb/orlib/info.html

[Bea93a] **Beasley D., Bull D.R., Martin R.R.**, "An Overview of Genetic Algorithms, Part 1, Fundamentals", University Computing, Vol. 15, No. 2, pp.58-69 (1993) ftp://ralph.cs.cf.ac.uk/pub/papers/GAs/ga_overview1.ps

[Bea93b] **Beasley D., Bull D.R., Martin R.R.**, "An Overview of Genetic Algorithms, Part 2, Research Topics", University Computing, Vol. 15, No. 4, pp. 170-181 (1993).

[Čan96] **Čangalović M.**, "Opšte heuristike za rešavanje problema kombinatorne optimizacije", u: Kombinatorna optimizacija: Matematička teorija i algoritmi, str. 320-350 (1996).

[Chu97] **Chu P.C.H., Beasley J.E.**''A genetic algorithm for the generalised assignment Problem'', Computers & Operations Research, Vol. 24, pp. 17–23 (1997).

[Chu98] **P.C. Chu and J.E. Beasley**. “A genetic algorithm for the multidimentional knapsack problem. Journal of heuristic”,4: 63-86, (1998).

[Dre03] **Drezner Z., Erkut E.,** “An Efficient Genetic Algorithm for the p-Median Problem”, Annals of Operations Research, Vol. 122, pp. 21-42 (2003).

[Đur08] **Đurić B., Kratica J., Tošić D., Filipović V.** ''Solving the maximally balanced connected partition problem in graphs by using genetic algorithm'',Computing and Informatics, Vol. 27, No. 3, pp. 41-354 (2008)

[Fil98] **Filipović V.,** "Predlog poboljšanja operatora turnirske selekcije kod genetskih algoritama", Magistarski rad, Univerzitet u Beogradu, Matematički fakultet (1998).

[Fil00] **Filipović V., Kratica J., Tošić D., Ljubić I.,** "Fine Grained Tournament Selection for the Simple Plant Location Problem", Proceedings of the 5th Online

World Conference on Soft Computing Methods in Industrial Applications - WSC5, pp. 152-158 (2000).

[Fil06] **Filipović, V.,** "Operatori selekcije i migracije i WEB servisi kod paralelnih evolutivnih algoritama", Doktorska disertacija, Matematički fakultet, Beograd (2006).

[Gil66] **Gilmore, P.C., R. Gomory**. The theory and computation of knapsack functions. *Operations Research* 14 1045–1075. (1966).

[Gol89] **Goldberg D.E.**, "Genetic Algorithms in Search, Optimization and Machine Learning", Addison-Wesley Publ. Comp., Reading, Mass., (1989).

[Gol09] **Goldbarg M.C., Bagi L.B., Goldbarg E.F.G.** ''Transgenetic algorithm for the Traveling Purchaser Problem'', European Journal of Operational Research,Vol. 199, pp. 36–45 (2009).

[Hol75] **Holland J.H.**, "Adaptation in Natural and Artificial Systems", The University of Michigan Press, Ann Arbor (1975).

[Juh06] **Juhos, I., Van Hemert, J.I.**, "Improving graph colouring algorithms and heuristics using a novel representation", Lecture Notes in Computer Science, Vol. 3906, pp. 123-134 (2006).

[Kar72] **Richard M. Karp,** "Reducibility Among Combinatorial Problems". In R. E. Miller and J. W. Thatcher (editors). Complexity of Computer Computations. New York: Plenum. pp. 85–103 (1972)

[Kom96] **Cvetković D., Čangalović M., Dugošija Đ., Kovačević- Vujičić V., Simić S., Vuleta J.,** „Kombinatorna optimizacija – matematička teorija i algoritmi", Društvo operacionih istraživača Jugoslavije, Beograd, (1996).

[Kra98] **Kratica J., Filipović V., Tošić D.**, "Solving the Uncapacitated Warehouse Location problem by SGA Eith Add-Heuristic", XV ECPD International Conference on Material Handling and Warehousing, University of Belgrade, Faculty of Mechanical Engineering, Materials Handing Institute, Belgrade (1998).

[Kra99a] **Kratica J.**, "Improving Performances of the Genetic Algorithm by Caching", Computers and Artificial Inteligence, Vol. 18, No. 3, pp. 271-283 (1999).

[Kra99b] **Kratica J.**, "Improvement of Simple Genetic Algorithm for Solving the Uncapacitated Warehouse Location Problem", Advances in Soft Computing - Engineering Design and Manufacturing, R.Roy, T. Furuhashi and P.K. Chawdhry (Eds), Springer-Verlag London Limited, pp. 390-402 (1999)

[Kra00] **Kratica J.,** "Paralelizacija genetskih algoritama za rešavanje nekih NP-kompletnih problema", Doktorska disertacija, Matematički fakultet, Beograd (2000).

[Kra01b] **Kratica J., Tošić D.**, "Introduction to Genetic Algorithms and Some Applications", Proceedings of a Workshop on Computational Intelligence - Theory and Applications, pp. 57-68, Niš, Yugoslavia, February 27, (2001).

[Kra01] **Kratica J., Tošić D., Filipović V., Ljubić I.** „Solving the Simple Plant Location Problem by Genetic Algorithms", RAIRO - Operations Research, Vol. 35, pp. 127-142 (2001).

[Kra02] **Kratica J., Tošić D., Filipović V., Ljubić I.** „A Genetic Algorithm for the Uncapacitated Network Design Problem", Soft Computing in Industry – Recent Applications, Engineering series, pp. 329-338 (2002).

[Kra03] **Kratica J., Ljubić I., Tošić D.** „A Genetic Algorithm for the Index Selection Problem". Springer Lecture Notes in Computer Science, Vol. 2611, pp.281-291 (2003).

[Kra05] **Kratica J., Stanimirović Z., Tošić D., Filipović V.** „Genetic Algorithm for Solving Uncapacitated Multiple Allocation Hub Location Problem", Computers and Informatics, Vol. 24, pp. 415-426 (2005).

[Kra06] **Kratica J., Stanimirović Z.** „Solving the uncapacitated multiple allocation p-hub center problem by genetic algorithm", Asia Pacific Journal of Operational Research 23, pp. 425-437 (2006)

[Kra07] **Kratica J., Stanimirović Z., Tošić D., Filipović V.** „Two genetic algorithms for solving the uncapacitated single allocation p-hub median problem", European Journal of Operational Research Vol. 182, No. 1, pp. 15-28 (2007)

[Kra08a] **Kratica J., Kovačević-Vujčić V., Čangalović M.** „Computing the metric dimension of graphs by genetic algorithms", Computational Optimization and Applications, DOI 10.1007/s10589-007-9154-5, (in press)

[Kra08b] **Kratica J., Kovačević-Vujčić V., Čangalović M**. „Computing strong metric dimension of some special classes of graphs by genetic algorithms", Yugoslav Journal of Operations Research, Vol. 18, No. 2, pp.~143-151 (2008)

[Kra08d] **Kratica J., Tošić D., Filipović V., Dugošija Đ.,** "Two hybrid genetic algorithms for solving the super-peer selection problem", WSC 2008.

[Kra11] **Kratica J., Kostić T, Tošić D., Dugošija Đ., Filipović V.** „A genetic algorithm for the routing and carrier selection problem". Computer Science and Information Systems-COMSIS in press 2011.

[Lor55] **Lorie, J., L.J. Savage.** “Three problems in capital rationing”. *The Journal of Business* 28 229–239., (1955).

[Lju04] **Ljubić I.** “Exact and memetic algorithms for two network design problems“, PhD thesis, Institute of Computer Graphics, Vienna University of Technology (2004).

[Man57] **Manne, A.S., H.M. Markowitz.** “On the solution of discrete programming problems”. *Econometrica* 25 84–110, (1957).

[Mil08] **Milanović M.**, "Solving the generalized vertex cover problem by genetic algorithm", Computing and Informatics (2008).

[Mat09] **Marie E.M.** „A genetic algorithm for the proportionate multiprocessor open Shop“, Computers & Operations Research, Volume 36, Issue 9, pp. 2601-2618 (2009).

[Mic96] **Michalewicz Z.**, "Genetic Algorithms + Data Structures = Evolution Programs", Third Edition, Springer Verlag, Berlin Heideleberg (1996).

[Mit99] **Mitchell M.,** "An Introduction to Genetic Algorithms", MIT Press, Cambridge, Massachusetts (1999).

[Müh97] **Mühlenbein H.**, "Genetic Algorithms", Local Search in Combinatorial Optimization, eds. Aarts E.H.L., Lenstra J.K., John Wiley & Sons Ltd., pp. 137-172 (1997).

[Ognj01] **Ognjanović Z., Kratica J., Milovanović M.**, “A Genetic Algorithm for Satisfiability Problem in a Probabilistic Logic: A First Report”, Springer Lecture Notes in Artificial Intelligence - LNAI, Vol. 2143, pp. 805-816 (2001).

[Ognj04b] **Ognjanović Z., Midić U., Kratica J.**, “A Genetic Algorithm for Probabilistic SAT Problem”, Lecture Notes in Artificial Intelligence - LNAI, Vol. 3070, pp. 462–467, (2004).

[Rai05] **Raidl, G.R., Gottlieb, J.**, "Empirical analysis of locality, heritability and heuristic bias in evolutionary algorithms: A case study for the multidimensional knapsack problem", Evolutionary Computation, Vol. 13, No. 4, pp. 441-475 (2005).

[Rom09] **Rom W.O., Slotnick S.A.** “Order acceptance using genetic algorithms”, Computers & Operations Research, Volume 36, Issue 6, pp. 1758-1767 (2009).

[Sav97] **Savelsbergh M.W.P.** “A branch-and-cut algorithm for the generalized assignment Problem”, Intractability: A Guide to the Theory of Operations Research,Vol. 45, No. 6, pp. 831-841 (1997).

[Sav08a] **Savić A., Tošić D., Marić M, Kratica J.,** "An genetic algorithm approach for solving the task assignment problem", Serdica Journal of Computing (2008).

[Sav08b] **Savić A.,** "An genetic algorithm approach for solving the machine-job assignment with controllable processing times", Computing and Informatics (2008).

[Sav09] **Savić, A.** "On solving of Maximum Betweenness Problem using Genetic Algorithms", Serdica Journal of Computing, 3, pp. 299-308, (2009)

[Sav10] **Savić, A.** ''Kombinatorni pristup rešavanju nekih problema pridruživanja" , Doktorska disertacija, Beograd (2010).

[Shi79] **Shih, W.** "A branch and bound method for the multiconstraint zero-one knapsack problem". *Journal of the Operational Research Society* 30 369–378., (1979).

[Sta07a] **Stanimirović Z. Kratica J. Dugošija Đ.**, "Genetic algorithms for solving the discrete ordered median problem", European Journal of Operational Research, Vol. 182, No. 3, pp. 983-1001 (2007).

[Sua09] **Souai N., Teghem J.** "Genetic algorithm based approach for the integrated airline crew-pairing and rostering problem", European Journal of Operational Research, Volume 199, Issue 3, pp. 674-683 (2009)

[Tot90] **Martello S. & P. Toth.** "*Knapsack Problems: Algorithms and Computer Implementations*." John Wiley & Sons. (1990).

[Trm00] **Trmčić (Ljubić) I.**, "Primena genetskih algoritama na probleme povezanosti grafova", Magistarski rad, Univerzitet u Beogradu, Matematički fakultet (2000).

[Wei67] **Weingartner, H. M., D. N. Ness.** "Methods for the solution of the multidimensional 0/1 knapsack problem". *Operations Research* 15 83–103., (1967).