

**БЪЛГАРСКА АКАДЕМИЯ НА НАУКИТЕ
ИНСТИТУТ ПО ИНФОРМАЦИОННИ И
КОМУНИКАЦИОННИ ТЕХНОЛОГИИ**

Стефка Стоянова Фиданова

**МЕТОД НА МРАВКИТЕ
ЗА РЕШАВАНЕ НА
КОМБИНАТОРНИ ОПТИМИЗАЦИОННИ
ЗАДАЧИ**

дисертация за придобиване на
научна степен "доктор на науките"
в професионално направление 4.6
"Информатика и компютърни науки"

Съдържание

1	Увод	11
1.1	Актуалност и мотивировка на темата	11
1.2	Цел и задачи на дисертационния труд	12
1.3	Методика на изследването	12
1.4	Представяне на резултатите	13
1.5	Съдържание на дисертацията	14
2	Метод на мравките	17
2.1	Структура на АСО алгоритъма	19
2.2	Варианти на АСО алгоритъма	24
3	Метод на мравките за задачата за раницата	27
3.1	Формулиране на задачата	27
3.2	АСО алгоритъм за МКР	29
3.3	Използване на феромона	29
3.4	Използване на разнообразни евристики	33
3.5	Хибриден АСО	36
3.6	Интеркритериален анализ	40
3.7	Стартови стратегии	49
3.8	Заклучение	54
4	Инспектиране на GPS мрежа	65
4.1	Формулиране на задачата	67
4.2	АСО алгоритъм за инспектиране на GPS мрежа	69
4.3	Хибриден АСО алгоритъм за задачата за инспектиране на GPS мрежа	76
4.4	АСО с промени в средата	78

4.5	ICrA на приложението на ASCO за задачата за инспектиране на GPS мрежа	84
4.6	Заклучение	88
5	Построяване на безжична сензорна мрежа	89
5.1	Въведение	89
5.2	Формулировка на задачата	92
5.3	ASCO за позициониране на безжична сензорна мрежа	93
5.4	Преобразуване на WSN задачата до едноцелева чрез умножение	96
5.5	Преобразуване на WSN задачата до едноцелева чрез сумиране	100
5.6	Многоцелева WSN задача	104
5.7	Прилагане на различни Интеркритериални анализи към ASCO алгоритъма	107
5.8	Заклучение	121
6	Наемане на работна сила	123
6.1	Формулиране на Задачата	124
6.2	ASCO Алгоритъм за Наемане на Работна Сила	126
6.3	Хибриден ASCO алгоритъм за наемане на работна сила	133
6.4	Влияние на ASCO параметрите върху ефективността на алгоритъма	137
6.5	ИнтерКритериален Анализ	138
6.6	Заклучение	144
7	Моделиране на пътничкопоток	145
7.1	Формулиране на задачата	146
7.2	Метод на мравките за моделиране на пътничкопоток	148
7.3	Експериментални резултати	149
7.4	Заклучение	152
8	Заклучение	155
8.1	Списък на публикациите	156
8.2	Списък на цитиранията	159
8.3	Приноси	170
8.4	Декларация за оригиналност	172
8.5	Благодарности	173

Списък на таблиците

3.1	АСО параметри	32
3.2	АСО с различно обновяване на феромона	34
3.3	АСО параметри	39
3.4	Сравняване на АСО алгоритмите	39
3.5	Скала за съгласие и несъгласие [11]	42
3.6	Индексна Матрица за ICrA	43
3.7	Степен на съгласие – $\mu_{C,C'}$ (μ - базиран ICrA)	43
3.8	Степен на несъгласие – $\nu_{C,C'}$ (μ - базиран ICrA)	44
3.9	Степен на неопределеност – $\pi_{C,C'}$ (μ - базиран ICrA)	44
3.10	Степен на съгласие – $\mu_{C,C'}$ (ν -базиран ICrA)	45
3.11	Степен на несъгласие – $\nu_{C,C'}$ -values (ν -базиран ICrA)	45
3.12	Степен на неопределеност – $\pi_{C,C'}$ (ν -базиран ICrA)	46
3.13	Степен на съгласие – $\mu_{C,C'}$ (Безпристрастен ICrA)	46
3.14	Степен на несъгласие – $\nu_{C,C'}$ (Безпристрастен ICrA)	47
3.15	Степен на неопределеност – $\pi_{C,C'}$ (Безпристрастен ICrA)	47
3.16	Степен на съгласие – $\mu_{C,C'}$ (Балансиран ICrA)	48
3.17	Степен на несъгласие – $\nu_{C,C'}$ (Балансиран ICrA)	48
3.18	Степен на неопределеност – $\pi_{C,C'}$ - (Балансиран ICrA)	49
3.19	Оценяване на стратегиите	55
3.20	Интуиционистки размита оценка на стратегиите	56
3.21	Индексна матрица за $\mu_{C,C'}$	57
3.22	Индексна матрица за $\nu_{C,C'}$	58
3.23	Стойности на $\mu_{C,C'}$, $\nu_{C,C'}$, $\pi_{C,C'}$ – част 1	59
3.24	Стойности за $\mu_{C,C'}$, $\nu_{C,C'}$, $\pi_{C,C'}$ – част 2	60
3.25	Индексна матрица за $\mu_{C,C'}$ (интуиционистки размита оценка)	61
3.26	Индексна матрица за $\nu_{C,C'}$	62
3.27	Стойности за $\mu_{C,C'}$, $\nu_{C,C'}$, $\pi_{C,C'}$ – част 1 (Интуиционистки размита оценка)	63

3.28	Стойности за $\mu_{C,C'}$, $\nu_{C,C'}$, $\pi_{C,C'}$ – част 2 (Интуиционистки размита оценка)	64
4.1	Сравнение на MMAS и ACS.	74
4.2	MMAS алгоритъм с процедура за локално търсене	78
4.3	Сравнение между ACO и ϵ ACO за задачата за инспектиране на GSP.	83
4.4	Резултати от 30 пускания	85
4.5	Индексна матрица за ICrA	86
5.1	Параметри на задаата	97
5.2	Сравнение с други методи	98
5.3	Експериментални резултати	98
5.4	Разстояние от разширения фронт	99
5.5	Сравнение между алгоритмите	101
5.6	Парето фронтове	102
5.7	Разстояние от разширения фронт	102
5.8	Недоминирани решения за претеглените суми	103
5.9	Сравнение между алгоритмите	104
5.10	Парето фронтове	105
5.11	Разстояние от Разширения фронт	106
5.12	Сравнение между алгоритмите	106
5.13	Хипер обеми на решенията	107
5.14	Скала за съгласие и несъгласие [11]	108
5.15	Резултати от μ -базиран ICrA – двойки в SPC	110
5.16	Резултати от μ -базиран ICrA – двойки в PC	112
5.17	Резултати от μ -базиран ICrA –двойки в WPC	113
5.18	Резултати от ν -базиран ICrA – двойки в WPC и WD	114
5.19	Резултати от ν -базиран ICrA – двойки в WPC, WD и D	115
5.20	Резултати от ν -базиран ICrA – двойки в WD и D	116
5.21	Резултати от ν -базиран ICrA – двойки в WPC	116
5.22	Резултати от Балансиран ICrA – двойки в WPC и WD	117
5.23	Резултати от Балансиран ICrA – двойки в WPC, PC и WD	118
5.24	Резултати от Балансиран ICrA – двойки в WD и WPC	119
5.25	Резултати от Балансиран ICrA – двойки в PC	120
5.26	Резултати от ν -базиран и Безпристрастен ICrA	121
6.1	Характеристики на тестовите примери	129

6.2	АСО параметри	129
6.3	Средни резултати за структурираните примери	130
6.4	Средни резултати за неструктурираните примери	130
6.5	Параметри на алгоритъма от [3]	131
6.6	Средно време за структурирани примери	132
6.7	Средно време за неструктурирани примери	132
6.8	Ранкиране на хибридните АСО алгоритми	135
6.9	Сравняване на хибридните АСО алгоритми спрямо време- то за изчисление	136
6.10	АСО параметри	138
6.11	Резултати за структурирани примери	139
6.12	Резултати за не структурирани примери	139
6.13	Индексна матрица за ICGA	140
6.14	Скала на съгласие и несъгласие по отношение на [80]	143
6.15	Индексна матрица според разстоянието до точка $\langle 1, 0 \rangle$. .	143
7.1	Експериментални резултати	150
7.2	Експериментални резултати за линията София Варна . . .	151

Списък на фигурите

2.1	Псевдокод описващ метода на мравките	22
3.1	Влияние на позицията на феромона	30
3.2	Сравняване на разнообразни евристики. Пунктир с точка представя евристика А, пунктир - евристика В, точки - евристика С и тънка линия - евристика D.	37
4.1	Сравнение между ACS и MMAS алгоритми за rgb443 тес- това задача.	75
4.2	MMAS с процедура за локално търсене за rgb443 тестова задача	79
4.3	Поведението на логистичната карта за различни стойности на r (по оста x , за $r = 2$ до 4 на фигурата).	80
4.4	Поведението на логистичната карта за стойности на r в интервала $[3.901, 3.908]$. Този регион на логистичната карта като цяло е хаотичен, но регулярните области на привли- чане (с типичните бифуркации) все още могат да бъдат идентифицирани.	81
4.5	GPS връзка в положителна скала за консонанс	87
4.6	GPS отношение в скалата на дисонанс	87
4.7	GPS отношение в отрицателна скала на консонанс	87
5.1	Представяне на ICrA резултатите – μ -базиран ICrA	109
5.2	Представяне на ICrA резултати при ν -базиран ICrA	113
5.3	Резултати от прилагането на Балансиран ICrA	117
5.4	ICrA резултати при Безпристрастен ICrA	120
6.1	Представяне на ICrA резултати, чрез интуиционистки раз- мит триъгълник	142

7.1	Пътници използвали влаковете София Варна	152
-----	--	-----

Глава 1

Увод

1.1 Актуалност и мотивировка на темата

Оптимизацията е ключова тема в информатиката, изкуствения интелект, изследването на операциите и свързаните с тях области. Целта на комбинаторната оптимизация е да намери оптимален обект по някакъв критерий, от крайно множество от обекти. Тя се отнася за тези оптимизационни задачи, за които множеството от валидни решения е дискретно или може да бъде редуцирано до дискретно и целта е да бъде намерено възможно най-доброто решение.

Много проблеми възникващи, в реалния живот и индустрията, могат да бъдат описани като комбинаторни оптимизационни задачи. Например съществуват много начини да се построи комуникационна мрежа: кой е най-добрият? Има много възможности да разпределим последователността от операции на машини при организация на някакво производство: коя ще даде най-добър резултат? Как най-добре да разпределим един бюджет? Когато има твърде много кандидати за решения, отговорите на тези въпроси могат да бъдат трудни и решаването им да изисква твърде много време. За такъв тип задачи обикновено се използват приближени методи.

Комбинаторните оптимизационни задачи се състоят от целева функция, която трябва да се минимизира или максимизира и ограничения. Примери за оптимизационни задачи са задачата за търговския пътник [115], маршрутизация на превозни средства [126], минимално обхващащо дърво [105], удовлетворяване на ограничения [87], задачата за раницата

[30] и други. Това са NP (неполиномиални) задачи и за да бъдат намерени решения близки до оптималните, обикновено се използват метаевристични методи. Един от тях е методът на мравките [26]. Той е много подходящ за решаване на дискретни задачи, защото е конструктивен метод.

1.2 Цел и задачи на дисертационния труд

Основната цел на дисертационния труд е разработването на алгоритми, на основата на метода на мравките, за решаването на задачи от реалния живот и индустрията.

За постигането на тази цел са формулирани следните пет задачи:

- Разработване на алгоритъм за решаване на задачата за раницата;
- Разработване на алгоритъм за инспектиране на GPS мрежа;
- Разработване на алгоритъм за построяване на безжична сензорна мрежа по два критерия, минимален брой сензори и минимална използвана енергия;
- Разработване на алгоритъм за наемане на работна сила;
- Разработване на алгоритъм за моделиране на пътникопоток по два критерия, време за придвижване и цена за придвижване.

1.3 Методика на изследването

Един от най-успешните метаевристични методи за решаване на комбинаторни оптимизационни задачи е методът на мравките. Идеята за него идва от поведението на мравките в природата. Когато търсят храна, мравките в природата маркират своя път оставяйки химическа субстанция наречена феромон. Ако е изолирана, една мравка се движи основно по случаен начин. Ако има предварително поставен феромон, мравката го регистрира и решава да го следва с голяма вероятност и по този начин го засилва с ново количество феромон. Повтарянето на горния механизъм от мравките в природата води до това, че ако по една следа са преминали повече мравки, толкова по-желана става тя за следващите мравки. От друга страна, феромонът на слабо използваните пътища

постепенно намалява заради изпарение. Така природата предпазва мравките да следват стари и неизгодни следи.

С времето феромонът започва да се изпарява, и намалява атрактивността на следата. Колкото повече време е необходимо на една мравка за да измине пътя, толкова по-продължително неговият феромон се изпарява. За сравнение, късите пътища се употребяват по-често и така количеството феромон става повече в сравнение с този по дългите пътища. Изпаряването на феромона има предимството да предотвратява стагнацията и оставането на мравката в някой локален минимум. Ако нямаше изпарение, то пътят, избран от първата мравка ще бъде изключително привлекателен за следващите мравки. В този случай експлоатацията на пространството от решенията ще бъде ограничена.

1.4 Представяне на резултатите

В съответствие с поставените задачи са разработени алгоритми, решаващи конкретни класове проблеми. Направена е програмна реализация на всеки от разработените алгоритми. Програмите са написани на езика C. Направено е изследване за стойностите на управляващите параметри.

Публикации

Авторът на дисертацията има над 200 публикации, като повечето са в областта на комбинаторната оптимизация и прилагането на стохастични методи. Резултатите от дисертацията са публикувани в 19 публикации съдържащи: 1 монография издадена в престижното научно издателство Springer, 1 с импакт фактор в списание в топ 10% на Q1, 11 с импакт ранг, 5 реферирани в световната система за реферирание и индексирание и една публикувана в международно списание. Всички публикации са след 2016, когато дисертантката е придобила званието професор, и не са участвали в предходни процедури.

Цитирания

Авторът на дисертацията има над 1250 цитирания. Публикациите, на основата на които е написана тази дисертация, са цитирани 52 пъти. Използваните публикации и цитирания са след придобиване на предходни степени и звания и не са използвани в други процедури. Индексът на Хирш на автора, спрямо известните ѝ цитирания, е $h=18$.

1.5 Съдържание на дисертацията

Дисертационният труд е разделен на Увод, шест глави и заключение. Дисертацията съдържа:

- 189 страници;
- 76 таблици;
- 16 фигури;
- 127 литературни източника.

Глава втора съдържа структурата на метода на мравките. В нея е разгледана първоначалната идея. Направен е паралел с мравките в природата и как алгоритъмът имитира поведението на мравките. Направен е преглед на съществуващите варианти на метода на мравките.

Глава трета е посветена на задачата за раницата и решаването ѝ с помощта на метода на мравките. Тази задача е представител на задачите за намиране на подмножество. След като бъде представена с граф само част от върховете на графа принадлежат към решението на задачата, като редът в който са избрани не е от съществено значение. Задачата дава възможност за конструиране на разнообразни евристични функции. Често основният алгоритъм се комбинира с процедура за локално търсене с цел подобряване на намерените решения и повишаване бързодействието на алгоритъма. Предложена е подходяща процедура, която е съобразена със спецификата на задачата. Методът на мравките е конструктивен метод. При него е важно решението да започне да се строи от подходящ връх в графа. Предложени са стартови стратегии за подобряване поведението на алгоритъма. Приложен е интеркритериален анализ за сравняване и оценка на работата на хибридният алгоритъм с традиционния метод на мравките.

Глава четвърта решава задачата за инспектиране на GPS мрежа. При тази задача трябва да се обходят наземните станции от мрежата и да се определи последователността на обхождане. Задачата е асиметрична, защото времето за преминаване от станция i до станция j може да е различно от времето за преминаване от станция j до станция i . Предложен е алгоритъм, на основата на метода на мравките, за решаване на задачата. Разработени са няколко варианта на процедури за локално търсене.

Предложен е алгоритъм на основата на метода на мравките, като са добавени промени/шум в средата на търсене. Приложен е интеркритериален анализ върху средните стойности от 2, 3, ..., 30 пускания на алгоритъм върху 8 тестови примера. По този начин е изследвана коректността на предложения алгоритъм.

Глава пета разглежда задачата за построяване на безжична сензорна мрежа. Задачата е двукритериална. Цели се пълно покритие на дадена област с минимум брой сензори, използващи минимум енергия за трансфер на събраните данни. Двете цели са взаимно противоречащи си. За намаляване на използваната енергия от сензор се добавят сензори, което увеличава броя на сензорите. Премахването на сензор води до увеличаване на натоварването и от там на използваната енергия от сензорите, които са били в съседство с премахнатия сензор. Предложени са няколко подхода за решаване на задачата, като многокритериална и като се сведе до еднокритериална. Задачата е сведена до еднокритериална по два начина, чрез умножаване на двете целеви функции и чрез използването на сума с тегла. Приложен е интеркритериален анализ показващ корелацията между различните варианти на алгоритмите, решаващи задачата.

Глава шеста е посветена на решаването на задачата за наемане на работна сила. При тази задача има операции, които трябва да се изпълнят, работници, които са кандидати да бъдат наети. Всеки работник си има цена на наемане. Задачата е с много ограничения, като: минимално време което трябва да работи отделен работник по всяка една операция за която е назначен, за да работи качествено; максимално време според трудовото законодателство, което един работник може да работи; максимален брой работници, които могат да работят едновременно и др. Целта е всички операции да бъдат изпълнени на минимална цена. Предложен е алгоритъм на базата на метода на мравките. Разработена е процедура за локално търсене с цел подобряване работата на алгоритъма. Изследвано е влиянието на стойностите на параметрите на алгоритъма върху неговата ефективност. Приложен е интеркритериален анализ върху алгоритъма с използването на различен брой мравки при фиксирано време за да се извлече допълнително знание за връзката между параметрите на алгоритъма.

В глава седма е предложен алгоритъм, на основата на метода на мравките, за решаване на задачата за моделиране на пътничопоток. Задачата е представена като двукритериална. Превозват се пътници по дадено направление с разнообразни видове транспорт. Началната и крайната

спирки са общи. Част от превозните средства не спират на някои от междинните спирки. Изследва се колко пътници са се качили във всяко от превозните средства, като целите са обща минимална цена на придвижване и общо минимално време за придвижване. Двете цели са противоположни. Превозните средства, които се движат с по-голяма скорост са по-скъпи и обратно, по-евтините са по-бавни.

Глава 2

Метод на мравките

Мравките се срещат на всички континенти с изключение на Антарктида. Семействата на мравките са многогодишна, силно организирана общност. Те живеят в организирани групи, наречени колонии и имат сложни взаимоотношения помежду си. Мравките имат изключително силно обоняние и са в контакт помежду си основно чрез феромони. Секретите са различни – да маркират пътя към храната, полови секрети, сигнали за опасност, феромони за разпознаване на типа дейности, които извършват.

От гледна точка на компютърните науки, интерес представлява поведението на мравките в процеса на локализация и усвояване на хранителни източници за нуждите на оцеляването на колонията като цяло. Отвъд поведението на индивида, индивидите в колонията спазват глобални правила за поведението на мравуняка, които могат да бъдат приложени към конкретни области на математиката и информатиката под формата на оптимизационни алгоритми. Мравките използват много феромони в комуникацията си. Когато една мравка попадне на източник на храна, тя започва да отделя феромон по пътя си обратно към мравуняка и оставя вид миризлива пътека към храната. Всички мравки, намерили храна, оставят след връщането си тази феромонова следа, образувайки нещо като ароматна мрежа, насочваща към източниците на храна. Когато няма следа от феромон, мравката поема в произволна посока, но в присъствието на такава мравката вероятно ще поеме по пътя с най-интензивната феромонова следа. Това води до автокаталитичен процес, при който мравките откриват най-кратките пътища към източниците на храна. С течение на времето феромонът се изпарява и ако не се подновява, количеството му намалява. Това предотвратява възможността

мравките да следват някаква стара следа, сочеща към вече изчерпан източник на храна.

Мравките, имайки ограничени индивидуални възможности, работейки като колектив са в състояние да намерят най-краткия път между своето гнездо и източника на храна. Това се нарича групова интелигентност. Те действат по следния начин:

- Първата мравка намира източника на храна, по някакъв начин, след което се връща в гнездото, оставяйки по пътя следа от феромон;
- Мравките следват възможните пътища следейки концентрацията на феромон и така правят по-късите пътища по-привлекателни.;
- Мравките предпочитат по-късите пътища, по този начин голяма част от по-дългите пътища губят от своя феромон.

Моделът обясняващ това поведение е както следва:

- Мравката се движи повече или по-малко случайно около колонията;
- Ако открие източник на храна, тя се връща в гнездото, оставяйки по своя път следа от феромон;
- Тези феромони привличат другите мравки и те започват да го следват;
- Връщайки се в колонията, мравките ще увеличат концентрацията на феромон по пътя;
- Ако има два пътя стигащи до един и същ източник на храна тогава, за дадено време по-късият път ще бъде изминат от повече мравки отколкото по-дългият;
- По-късият път ще увеличи концентрацията на феромона си и ще стане по-привлекателен.
- Вероятно по-дългият път ще изчезне след време заради изпарението на феромона;

- Евентуално всички мравки ще изберат да следват късия път.

Мравките използват околната среда като посредник за комуникация. Те обменят информация индиректно, чрез поставяне на феромон. Обменената информация има локално значение. Единствено мравка, която е в близост до феромон, има отношение към него. Тази система се нарича „системата на петната” и се появява при много общности на социални животни. Механизмът на решаване на сложни задачи, които са твърде сложни за една мравка е добър пример за самоорганизираща се система. Тази система се базира на положително натрупване (поставянето на феромон привлича другите мравки, които го увеличават от своя страна) и на отрицателно (отпадането на път заради изпарението предпазва системата от разпадане). Теоретично, ако количеството феромон остава едно и също през цялото време по всички пътища, то няма да бъде избран път. Поради предходните натрупвания, има леки промени в количеството феромон и това позволява изборът на път [28].

2.1 Структура на АСО алгоритъма

В експеримент, наречен Double Bridge [21], авторите правят изследвания за това как да поставят феромон и връзката с поведението на мравките. Пръв Marco Dorigo прилага идеи от поведението на мравки за решаване на комбинаторни оптимизационни задачи [22, 26, 28]. Първият алгоритъм на мравките е въведен през 1992 от него в докторската му дисертация [25], която той защитава в Политехниката на Милано, Италия.

Методът на мравките (Ant Colony Optimization - АСО) е част от метаевристичните методи за оптимизация. Метаевристиката е процедура от високо ниво, предназначена за намиране, изграждане или избор на процедура на ниско ниво, която може да гарантира намирането на достатъчно добро решение на оптимизационната задача, особено когато информацията е непълна или компютърните ресурси са ограничени.

Първият АСО алгоритъм има за цел да намери оптималния път в граф, въз основа на поведението на мравки, които търсят път между своята колония и източника на храна [22, 25]. От тогава до днес, оригиналната идея на Дориго е разширявана и модифицирана от различни изследователи, за да се приложи към по-широк клас задачи.

2.1.1 Представяне на задачата с граф

Мравките в природата обменят информация индиректно, като поставят феромон по пътя си. Този механизъм за решаване на задача, която е твърде сложна за индивида, е добър пример за самоорганизираща се система. Дори леки промени във феромона позволяват да се предпочитат маршрут. Това поведение на мравките дава идеята за представяне на задачата с граф, а потенциалните решения като пътища в графа. Оптимизационната задача се свежда до намирането на най-краткия/дългият път в графа при допълнителни ограничения. В зависимост от самата задача, феромонът може да бъде поставен върху дъгите на графа или върху възлите на графа. Представянето на задачата с граф е една от важните точки на алгоритъма. Трябва да се реши кои елементи от задачата да бъдат представени от възлите, какво е значението на дъгите.

2.1.2 Конструирание на решенията

Основната част от един алгоритъм, използващ метода на мравките, е построяване на решението. Методът е итеративен. Накратко алгоритъмът може да бъде представен по следния начин:

- На всяка итерация, всяка мравка започва да строи своето решение от случаен връх в графа. Случайният старт е начин за разнообразяване на търсенето в множеството от решения;
- Мравката избира следващият връх, който да включи в решението с помощта на функция, наречена вероятност на прехода. Тази функция е произведение от количеството феромон съответстващо на прехода (на дъгата свързваща двата върха или на избрания връх), и евристична информация;
- Мравката спира да прибавя нови върхове когато вероятността за добавяне на нов връх стане 0;
- Накрая на всяка итерация феромонът се обновява;
- Алгоритъмът спира когато е достигнато условието за край.

Вероятността на прехода е произведение от количеството феромон съответстващо на прехода (на дъгата свързваща двата върха или на избра-

ния връх), и евристична информация. Евристичната информация зависи от решаваната задача. Количеството феромон представлява опита на мравките, от предните итерации. Евристичната информация е функция, представляваща предварителното познаване на задачата. Подходящото конструиране на евристична информация е особено важно за намирането на добри решения. Обикновено това е подходяща комбинация между параметрите на целевата функция и тези на ограниченията. Мравката добавя към частичното решение върхът с най-висока вероятност на прехода. Ако има повече от един връх с равна вероятност, то се избира един от тях на случаен принцип.

Мравката се придвижва от връх i до връх j от графа на задачата с вероятност:

$$p_{ij} = \frac{\tau_{ij}^{\alpha} \eta_{ij}^{\beta}}{\sum_{\text{all possible } k} \tau_{ik}^{\alpha} \eta_{ik}^{\beta}} \quad (2.1)$$

Където:

- τ_{ij} е количеството феромон съответстващо на прехода от връх i до връх j ;
- α е параметър, контролиращ влиянието на τ_{ij} ;
- η_{ij} е евристичната информация;
- β е параметър, контролиращ влиянието на η_{ij}

При по-високо ниво на феромон или евристична информация, вероятността да бъде избран този преход е по-голяма. Преди първата итерация се поставя начален феромон τ_0 , който има малка положителна стойност.

2.1.3 Обновяване на феромона

Мравките в природата поставят феромон на пътя, който следват за да могат да се върнат в гнездото. По-късите пътища събират повече феромон и така с течение на времето стават по-желани и биват следвани от повече мравки.

В АСО алгоритмите, "изкуствените" мравки имитират мравките от природата. Те поставят феромон на елементите на графа (дъги или върхове), които са свързани с решението им. Преди първата итерация феромонът по всички елементи е еднакъв и е равен на малка положителна

```

Ant Colony Optimization
Initialize number of ants;
Initialize the ACO parameters;
while not end-condition do
    for k=0 to number of ants
        ant k starts from random node;
        while solution is not constructed do
            ant k selects higher probability node;
        end while
    end for
    Update-pheromone-trails;
end while

```

Фигура 2.1: Псевдокод описващ метода на мравките

стойност $0 < \tau_0 < 1$. Можем да разглеждаме феромона като минал опит на колонията от мравки, като нейна глобална памет по отношение на използването на дъгата (i, j) в минали решения. Когато всички мравки са изградили своите решения, феромонните следи се актуализират с оглед на текущия опит. Първо феромонните следи се изпаряват, стойността τ_{ij} се умножава с фактор $1 - \rho$, където $\rho \in [0, 1]$ е скоростта на изпарение. Така мравките частично забравя стария опит и се фокусират върху новата информация. След това мравките добавят новия феромон. Количеството на добавения феромон съответства на качеството на намереното от мравката решение. Елементите на по-добрите решения получават повече феромон от останалите и така вероятността да бъдат предпочетени на следващата итерация е по-голяма.

Правилото за обмяна на феромона е следното:

$$\tau_{ij} = (1 - \rho)\tau_{ij} + \Delta\tau_{ij}, \quad (2.2)$$

Където τ_{ij} е количеството феромон, съответстващо на прехода от връх i до връх j , ρ е скоростта на изпарение на феромона.

2.1.4 Параметри на алгоритъма

АСО има няколко параметъра, от които зависи ефективността на алгоритъма. Те са:

- брой N на използваните мравки;
- скорост на изпарение ρ ;
- начален феромон τ_0 ;
- брой I на итерациите;
- α и β .

АСО е метод основан на множество от индивиди (популация). На всяка итерация всяка от мравките изгражда решение, започвайки от случаен връх на графа на задачата. Случайният старт представлява разнообразяване на процеса на търсене на решения. Благодарение на него, методът използва малка популация, обикновено до 20 мравки са достатъчни. За сравнение генетичният алгоритъм (GA) се нуждае от популация от минимум 100 индивида [41].

Параметърът на изпарение симулира процеса на изпарение в природата. Той показва важността на старата информация по отношение на новата в процеса на търсене на решения. Параметърът на изпарение зависи от решаваната задача. Неговата стойност е в интервала $(0, 1)$.

Методът не е много чувствителен спрямо стойността на началния феромон. По-важното е да е строго положителен. Ако началният феромон е 0, тогава вероятността за преход ще бъде 0 и алгоритъмът ще спре преди края на първата итерация.

АСО е итеративен процес. Един от използваните критерии за край е достигане на максималния брой итерации, който е предварително определен. Друг критерий за край може да бъде определен брой итерации, без подобряване на резултата. Ако имаме някаква предварителна оценка за резултата, като критерий за край може да се използва спиране на алгоритъма когато е достатъчно близо до тази оценка. Може да се използва и комбинация от няколко критерия за край.

Параметрите α и β показват важността на количеството феромон и на евристичната информация съответно за процеса на търсене. Техните стойности зависят от решаваната задача.

2.2 Варианти на АСО алгоритъма

Съществуват няколко варианта на метода на мравките. Основната разлика между тях е начинът по който се обновява феромона. Целта е да се избягват локалните оптимуми и да се подобри сходимостта на алгоритъма.

2.2.1 Ant system

Първият вариант на АСО е предложен от Marco Dorigo и е наречен Ant System [15]. Той следва точно поведението на мравките в дивата природа. При този вариант на метода всички мравки отлагат феромон, пропорционален на качеството на техните решения. Наблюденията са, че елементите на графа натрупват твърде много феромон и алгоритъмът попада в локален оптимум, в много ранен стадий, и не може да излезе от него. Експериментално е установено, че той работи по-лошо от други известни метаевристични методи. Поради това са създадени различни варианти на метода, за да се подобри качеството на намерените решения.

2.2.2 Elitist Strategy for Ant System

Това е първата модификация на АСО, използваща стратегията на елитните индивиди. Модификацията е предложена от Dorigo [28], който е и автор на метода. В този вариант, предварително определен брой мравки (една или повече), постигнали най-добрите решения, се обявяват за елитни и само те имат право да поставят феромон на елементите на техните решения. Целта е да се даде възможност за увеличаване на стойността на феромона, на тези елементи от графа, които са част от добрите, намерени до сега, решения. Това е техника за концентриране на търсенето около вече намерени добри решения, като се предполага, че в близост може да има още по-добри решения. Ако всички мравки бъдат декларирани като елитни, то тогава имаме варианта Ant System.

2.2.3 Ant Colony System

При Ant Colony System основното е, че има два вида обновяване на феромона, глобално и локално. При глобалното обновяване, феромонът на всички елементи на графа на задачата, които са участвали в решение,

конструирано през текущата итерация, се намалява (изпарява), а след това се добавя нов феромон, който е равен на началния. По този начин се гарантира, че феромонът на всеки един елемент от графа ще е повече или равен на началния феромон. При локалното обновяване на феромона, само елементите на най-доброто решение получават феромон и той е пропорционален на стойността на целевата функция.

С глобалното обновяване на феромона се цели разнообразяване на търсенето. Предотвратява се възможността стойността на феромона на някои елементи да стане твърде близка до 0 и по този начин части от множеството на решенията да останат неизследвани. Локалното обновяване на феромона, засилва неговата концентрация около най-доброто решение, намерено до сега, и концентрира търсенето около него.

2.2.4 Max-Min Ant System

Този вариант е известен още като MAX-MIN АСО или ММАС [116]. Алгоритъмът е променен като са фиксирани горна и долна граници на феромона $[\tau_{min}, \tau_{max}]$. Феромон се поставя само на елементите на графа на задачата, принадлежащи на най-доброто намерено решение, глобално или за итерацията. Феромонът на останалите елементи се намалява (изпарява), подобно на алгоритъма с една елитна мравка. Ако количеството феромон на някой елемент стане по-малко от фиксираната долна граница, то то се поставя да бъде равно на τ_{min} . Така количеството феромон на отделните елементи на графа на задачата не може да стане близко до 0 и по този начин части от множеството на решенията да останат неизследвани.

Когато количеството феромон на някой елемент от графа на задачата стане повече от фиксираната горна граница, то се поставя да бъде равно на горната граница τ_{max} . С тези изменения се цели да няма елементи на графа, които имат твърде много феромон и това да доведе до повторението на едни и същи решения, т.е. стагнация на търсенето. От друга страна се предотвратява някои елементи на графа на задачата да имат количество феромон близко до 0 и те да станат нежелани, когато мравката прави своя избор.

2.2.5 ACO with additional reinforcement

В този вариант на метода на мравките се поставя феромон, пропорционален на целевата функция, на елементите на най-доброто решение за текущата итерация [31]. Елементите на графа, които не участват в нито едно решение за текущата итерация, също получават феромон. Елементите на графа на задачата, които принадлежат на останалите решения запазват своя феромон. Идеята е неизползваните елементи да имат повече феромон от елементите на недобрите решения и близко, но по-малко от елементите на най-доброто решение. Така от една страна се разнообразява търсенето, като мравките се стимулират да търсят решения в неизследвани области, като избягват вече намерените недобри решения. От друга страна търсенето се концентрира около най-доброто решение.

2.2.6 ACO with Intuitionistic fuzzy pheromone

В този вариант на метода е предложено обновяването на феромона да бъде интуиционистки размито [62]. Това означава, че новодобавения феромон се умножава с коефициен α , така че $\alpha \leq \rho$, скоростта на изпарение. Коефициентът α се генерира по случаен начин в интервала $[0, \rho]$. Така се получава елемент на неопределеност и по-голямо разнообразяване на търсенето.

За повечето метаевристични методи няма теоретично доказателство, че са сходящи. В [28] е доказана теорема за сходимост на метода на мравките. Най-общо тази теорема гласи, че ако феромонът на никой от елементите на графа не клони към нула, то вероятността за намиране на глобалният оптимум клони към 1, когато броят на итерациите клони към безкрайност. Може да се отбележи, че последните четири, от изложените варианти на метода, отговарят на условията на теоремата.

Глава 3

Метод на Мравките за задачата за раницата

Резултатите от тази глава са публикувани в следните статии [46, 49, 52, 57, 62, 63].

Много мерната задача за раницата (МКР) е сложна комбинаторна оптимизационна задача с широко приложение. Задачи от различни области на индустрията могат да бъдат представени като МКР включително финансово и друг вид управление.

3.1 Формулиране на задачата

МКР има значителен брой приложения както в теорията, така и в практиката. Тя се явява и като подзадача в по-сложни оптимизационни задачи и успешното им решаване зависи от решаването на МКР.

За задачата за раницата можем да си мислим като задача за разпределение на ресурси. Има m ресурса (раници) и n обекта, като обект j носи печалба p_j . Всеки ресурс има свой собствен бюджет c_i (обем на раницата) и обект j консумира количество r_{ij} от ресурс i . Ние се интересуваме от максимизирането на общата печалба и в същото време да останем в рамките на ограничения бюджет. МКР може да се формализира по следния начин:

$$\begin{aligned}
& \max \sum_{j=1}^n p_j x_j \\
& \text{subject to } \sum_{j=1}^n r_{ij} x_j \leq c_i \quad i = 1, \dots, m \\
& x_j \in \{0, 1\} \quad j = 1, \dots, n.
\end{aligned} \tag{3.1}$$

x_j е 1 ако обект j е избран и 0 в другия случай. Нека $I = \{1, \dots, m\}$ и $J = \{1, \dots, n\}$, като $c_i \geq 0$ за всяко $i \in I$. Добре дефинираната МКР предполага, че $p_j > 0$ и $r_{ij} \leq c_i \leq \sum_{j=1}^n r_{ij}$ за всички $i \in I$ и $j \in J$. Отбелязваме, че матрицата $[r_{ij}]_{m \times n}$ и векторът $[c_i]_m$ са не отрицателни.

Можем да отбележим следните важни приложения на МКР: задачата за подреждане на товари, разрязване на запаси, пакетиране, контролиране на бюджет, финансово управление. Sinha и Zoltner [113] предлагат използването на МКР във задачата за отказоустойчивост (fault tolerance), а Diff и Hellman [24] проектират публична криптографска схема, чиято сигурност се постига чрез трудността при решаването на МКР. Martello и Toth [90] отбелязват, че задачата за разпределение върху два процесора може да се реши като МКР. Други приложения са индустриалното управление, управление на колектив, в корабоплаването, космонавтиката, теорията за изчислителната сложност.

Задачата за намиране на най-кратък път в транспортна мрежа води до определяне на подмножество от свързани пътища, които заедно образуват най-късото разстояние или най-краткото разстояние по отношение на времето или най-евтиният начин да стигнем от едно място до друго. Задачата е кое подмножество от линии дава най-бързата връзка за комуникация. Теорията за изчислителната сложност е част от теорията за изчисляването имаща връзка с наличните ресурси по време на изчислението за решаване на дадена задача. В този случай основната задача се формулира като МКР или МКР се явява като под-задача при решаването на по-общата задача за назначаване. Тя отново възниква при задачата за маршрутизация на превозни средства (vehicle routing). Като допълнение МКР може да се разглежда като обобщен модел на двоична задача с положителни коефициенти [80].

3.2 АСО алгоритъм за МКР

МКР е интересна задача както от практическа, така и от теоретическа гледна точка: практическа, защото включва голям брой задачи идващи от реалния живот и индустрията; теоретическа, защото дава разнообразни възможности за моделиране на феромона и евристичната информация. Един от основните елементи на АСО алгоритмите е описването на решаваната задача с граф. Трябва да се реши кой от елементите на задачата ще съответства на върховете на графа и какъв е смисълът на ребрата. Решението се представя като път в графа.

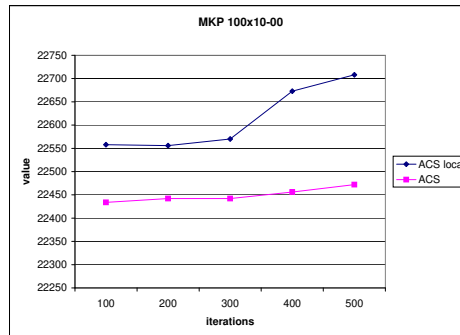
Определяме графа съответстващ на МКР по следния начин: върховете съответстват на обектите и всеки два върха са свързани с ребра. Пълното свързване на графа означава, че след обект i обект j може да бъде избран ако има достатъчно ресурс и ако обект j все още не е избран. Алгоритъмът е итеративен. На всяка итерация всяка мравка конструира решение. Началният обект (връх в графа) се избира по случаен начин. След това се добавят нови обекти без да се нарушават ограниченията на ресурса. След като всички мравки построят своите решения, стойностите на феромона се обновяват. Поради случайния избор на първия обект в решението, може да се използва малък брой мравки. Експериментално е установено, че между 10 и 20 мравки са достатъчни за намиране на добри решения без това да води до необходимост от увеличаване на броя на итерациите.

3.3 Използване на феромона

Методът на мравките използва "феромон" за отбелязване на съответните елементи на графа в решения с какво качество са участвали. Качеството на решението се измерва със стойността на целевата функция.

3.3.1 Позициониране на феромона

Едно от основните неща при решаването на МКР с АСО алгоритъм е да се реши къде да бъде поставян феромона, така че максимално да допринесе за намиране на добри решения. Задачата дава две възможности за поставяне на феромона:



Фигура 3.1: Влияние на позицията на феромона

- Първата и може би по-естествена възможност е феромонът да бъде поставян на върховете на графа. Идеята е да бъде повишена привлекателността на някои върхове (тези които са принадлежали на добри решения) да бъдат по-желани за избор при построяването на нови решения.
- Втората възможност е феромонът да бъде поставен на ребрата на графа. В този случай идеята е да се повиши привлекателността да бъде избран връх j ако последно избраният връх е i , когато е поставен феромон на реброто (i, j) .

Първата възможност изглежда по-естествена за МКР, защото за решението на задачата не е важен в какъв ред са избрани обектите (върховете). За решаването на МКР Leguizamón и Michalevici използват първата възможност [86], докато във [32] е използвана втората.

Двете възможности за поставяне на феромона са тествани върху 20 тестови примера от OR Library, съдържащи 100 обекта и 10 ограничения (<http://mscmga.ms.ic.ac.uk/jeb/orlib>). Броят на използваните итерации е $K = 500$ за всички пускания. Използва се ACS (ant colony system) алгоритъм и всеки тестови пример е пускан 30 пъти. Началната стойност на феромона е фиксирана на $\tau_0 = 0.5$. Параметърът на изпарението е $\rho = 0.1$. Броят на мравките е 10. На фигура 3.1 със синя линия е показано поведението на алгоритъма когато феромонът е на ребрата, а с червена линия - когато феромонът е на върховете. Както е показано на Фигура 3.1 има ранна стагнация на алгоритъма когато феромонът се

поставя на върховете. Този ефект може да бъде обяснен с голямото натрупване на феромон на някои върхове, което води до повтарянето на едни и същи решения и попадането в локален оптимум. Във втория случай феромонът е разпръснат върху ребрата и това води до по-голямо разнообразяване на търсенето. От тук нататък феромонът ще бъде поставян на ребрата на графа, когато АСО се прилага за решаване на задачата за раницата.

3.3.2 Използване на интуиционистки размит феромон

В интуиционистки размитата логика всяко твърдение се оценява със степени на валидност и невалидност. Добавя се и една трета компонента, наречена степен на неопределеност [12, 13]. По този начин сумата от степените на валидност и невалидност е по-малка или равна на 1. Когато степента на неопределеност е 0 имаме традиционната размита логика въведена от Zadeh [124].

В тази секция ще приложим интуиционистка размитост към обновяването на феромона. В началото се поставя еднакъв феромон на всичките ребра, който има малка положителна стойност τ_0 , $\tau_0 \in (0, 1)$. На края на всяка итерация феромонът се обновява съобразно построените, от мравките, решения. Нека ρ е скоростта на изпарение. Правилото за обновяване на феромона е:

$$\tau_{ij} \leftarrow (1 - \rho)\tau_{ij} + \Delta\tau_{ij}. \quad (3.2)$$

В повечето приложения на АСО към МКР, $\Delta\tau_{ij} = \rho F$, където F е стойността на целевата функция за съответното решение [33].

В традиционния метод на мравките, параметърът на изпарение ρ е входен параметър и остава непроменен до края на изпълнението на алгоритъма. Във [62] е предложено използването на *интуиционистки размит феромон*. В случая на интуиционистки размит феромон сме предложили следната формула за обновяване на феромона [62]:

$$\tau_{ij} \leftarrow (1 - \rho)\tau_{ij} + \alpha F, \quad (3.3)$$

където $(1 - \rho) + \alpha \leq 1$, $\alpha \in (0, 1)$.

В случая на интуиционистки размит феромон генерираме параметъра ρ като случайно число в интервала $(0, 1)$; след това параметърът α се

генерира като случайно число в интервала $(0, \rho)$. Така гарантираме, че сумата от тези два параметъра ще е по-малка или равна на 1 и обновяването на феромона ще е интуиционистки размито.

Предложили сме два варианта на прилагането на интуиционистки размито обновяване на феромона. В първия вариант, параметрите ρ и α се генерират в началото на изпълнението на алгоритъма още преди първата итерация. Така те остават непроменени до края на изпълнението на алгоритъма, но са различни за различните изпълнения на алгоритъма.

Във втория вариант, параметрите ρ и α се генерират в началото на всяка итерация. Така техните стойности са различни за всяка итерация на едно изпълнение на алгоритъма. Целта на интуиционистки размитото обновяване на феромона е по-голямо разнообразяване на търсенето, което би довело до намиране на по-добри решения.

Експериментални резултати

Двата предложени варианта за обновяване на феромона са тествани върху 10 МКР тестови примера от Operational Research Library “OR-Library”, <http://people.brunel.ac.uk/~mastjjb/jeb/orlib/mknapinfo.html> (21 Jun 2021).

Всеки тестови пример се състои от 100 обекта и 10 раници/ограничения. За всички тестове АСО алгоритъмът е с едни и същи параметри, Таблица 3.1. Параметрите са фиксирани експериментално. Алгоритъмът е прилаган 30 пъти със всеки един от вариантите за всеки един от тестовите примери. Приложен е ANOVA тест за да се гарантира статистическата разлика между получените осреднени резултати.

Таблица 3.1: АСО параметри

Параметри	Стойност
Брой итерации	300
Брой мравки	20
ρ	0.5
τ_0	0.5
a	1
b	1

Искаме да отбележим, че и двата варианта на интуиционистки раз-

митото обновяване на феромона не оказват влияние на времето за изпълнение на алгоритъма. В Таблица 3.2, са показани средните резултати върху 10 тестови примера, всеки изпълнен 30 пъти. Във втората колона са резултатите, постигнати с традиционния метод на мравките. Третата колона показва резултатите при интуиционистки размито обновяване на феромона, когато параметрите се пресмятат в началото на алгоритъма. Четвъртата колона показва резултатите, когато параметрите на интуиционистки размития феромон се изчисляват в началото на всяка итерация. Последният ред показва средната стойност върху всяка от колоните. Наблюдаваме, че и при двата варианта на интуиционистки размито обновяване на феромона имаме подобряване на постигнатите резултати в 8 от 10 теста, спрямо традиционния АСО, както и че средното върху всички резултати е по-добро. Ако направим сравнение между двата варианта на интуиционистки размит феромон, отбелязваме, че вариантът с изчисляване на параметрите в началото на алгоритъма, работи по-добре, отколкото ако ги пресмятаме на всяка итерация. Първият вариант постига по-добри резултати в 6 от 10 случая и средното върху всички тестови примери е по-добро. Можем да заключим, че интуиционистки размитото обновяване на феромона подобрява действието на алгоритъма и постигнатите резултати, като увеличава разнообразието на търсене и от там вероятността за намиране на добро решение. Това разнообразяване е по-балансирано, когато пресмятането на коефициентите се извършва еднократно в началото на алгоритъма, отколкото на всяка итерация.

Тестван е и вариант на АСО, когато параметърът ρ е фиксиран и само параметърът α се генерира по случаен начин спазвайки условието $(1 - \rho) + \alpha \leq 1$. По този начин се запазва интуиционистката размитост. Получените резултати са статистически идентични с вариантите, когато и двата параметъра ρ и α се генерират по случаен начин. Поради тази причина не е необходимо настройване на параметъра ρ .

3.4 Използване на разнообразни евристики

Когато изчисляваме вероятността за преход, евристичната информация би могла да бъде конструирана по различни начини. Тя може да бъде статична или динамична. Можем да използваме различните параметри на задачата и да ги комбинираме по разнообразен начин. Статичната

Таблица 3.2: АСО с различно обновяване на феромона

Тест	Трад. АСО	Инт Фъзи начало	Инт. Фъзи итер.
МКР 100x10-01	21989.43	22061.00	21992.92
МКР 100x10-02	22059.90	22055.76	22096.00
МКР 100x10-03	20989.00	21020.56	20997.00
МКР 100x10-04	21625.56	21656.03	21666.66
МКР 100x10-05	21718.33	21754.00	21761.63
МКР 100x10-06	21869.73	21891.20	21867.16
МКР 100x10-07	21477.30	21509.53	21502.2
МКР 100x10-08	21573.03	21668.06	21596.16
МКР 100x10-09	22248.43	22229.46	22221.00
МКР 100x10-10	40581.51	40556.50	40578.10
average	23613.22	23640.23	23627.88

евристика остава непроменена до края на изпълнението на алгоритъма, докато динамичната съответства на моментното състояние на алгоритъма, решаващ задачата. Добре е в евристичната информация да бъде включена печалбата, като основна съставна част на целевата функция. Предполагаме, че ще управляваме търсенето по-ефективно, ако в евристиката включим повече информация за решаваната задача. Опитваме се да включим параметрите на ограниченията по разнообразен начин за да разберем тяхното влияние.

3.4.1 Статични евристики

Предложили сме два вида статични евристики, които сме нарекли съответно "евристика А" и "евристика В".

- **Евристика А.**

Нека $s_j = \sum_{i=1}^m r_{ij}$. За евристична информация използваме:

$$\eta_{ij} = p_j^{d_1} / s_j^{d_2}, \quad (3.4)$$

$0 < d_1$ и $0 < d_2$ са параметри. Разходите са включени в евристиката. Така обектите с по-голяма цена и по-малко разходи ще бъдат по-

желани. По този начин се стремим към баланс между разходите и цената (приходите) при избора на даден обект (проект).

- **Евристика В.**

Нека $s_j = \sum_{i=1}^m r_{ij}/c_i$. За евристична информация използваме:

$$\eta_{ij} = p_j^{d_1}/s_j^{d_2}, \quad (3.5)$$

$0 < d_1$ и $0 < d_2$ са параметри. Така евристиката зависи от цената, разходите и наличния бюджет. Обектите с по-голяма цена, на които разходът е по-малка част от бюджета ще бъдат по-желани.

3.4.2 Динамични евристики

Третият и четвъртият тип евристики са динамични. Те съответстват на моментното състояние на алгоритъма. Ще ги наречем "Евристика С" и "Евристика D" съответно.

- **Евристика С [86].**

Нека $b_i = c_i - \sum_{j=1}^n r_{ij}x_j$ е остатъка от бюджета, преди избирането на следващия обект, а $s_j = \sum_{i=1}^m r_{ij}/b_i$ ако $b_i \neq 0$ и $s_j = \sum_{i=1}^m r_{ij}$ ако $b_i = 0$. За евристична информация се използва:

$$\eta_{ij} = p_j^{d_1}/s_j^{d_2}, \quad (3.6)$$

където $d_1 = d_2$. Целта е евристичната информация максимално да съответства на текущото състояние на алгоритъма и така да постигне възможно най-добри решения. В [86] авторите не проверяват дали $b_i \neq 0$, но тъй като това може да се случи и има деление на b_i , ние сме добавили тази проверка в алгоритъма.

- **Евристика D.**

За евристична информация използваме:

$$\eta_{ij} = p_j^{d_1}/s_j^{d_2}, \quad (3.7)$$

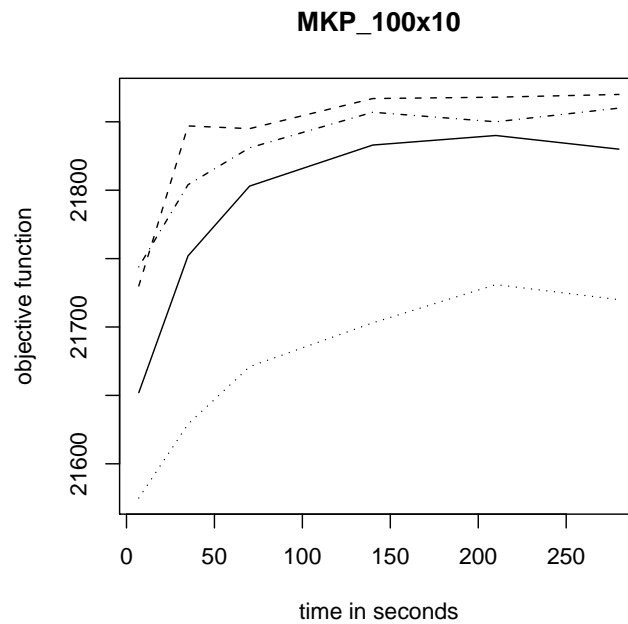
$0 < d_1$ и $0 < d_2$ са параметри и могат да не бъдат равни. Смисълът на s_j е като в Евристика С. Така Евристика С е частен случай на Евристика D.

3.4.3 Сравняване на различните евристики

Предложените евристики са тествани върху тестовите примери от предходната секция. Алгоритъмът е пускан с едни и същи параметри за всички евристики и тестови примери, за да може да се направи сравнение. Резултатите от тестовете са показани на фигура 3.2. Тя показва средните резултати върху 30 пускания на алгоритъма с всеки тестови пример. Наблюдаваме, че Евристика В има предимство пред останалите евристики. Това означава, че е важно да се избере обект с малко разходи, но е още по важно тези разходи да представляват малък дял от бюджета по това перо. Предполагахме, че ще получим по-добри резултати с използването на динамична евристика, защото тя съответства на моментното състояние на алгоритъма. Противно на нашите очаквания, резултатите получени с използването на динамична евристика не са по-добри. Тези резултати могат да се обяснят с това, че статичната евристика дава по-добра цялостна оценка за задачата от динамичната. Относно параметрите d_1 и d_2 , по добри резултати се получават когато $d_1 \neq d_2$.

3.5 Хибриден АСО

Понякога използваният алгоритъм не е достатъчен за постигането на добри решения. Тогава се прави комбинация от няколко методологии, за да могат да се комбинират добрите им качества. Понякога се комбинира точен с метаевристичен метод. Друг път два метаевристични метода. Това са така наречените хибридни методи. Най-често се използва един основен метод и намерените от него решения се подобряват с прилагането на локална оптимизация (локално търсене). Добавянето на процедура за локално търсене може да помогне за излизането от локални оптимуми в търсенето на глобалния. Процедурата за локално търсене може да се прилага след всяка итерация, а ако отнема време може да бъде прилагана веднъж на фиксиран брой итерации или в случаите когато в текущата итерация няма подобряване на решението.



Фигура 3.2: Сравняване на разнообразни евристики. Пунктир с точка представя евристика А, пунктир - евристика В, точки - евристика С и тънка линия - евристика D.

При прилагането на процедура за локално търсене се генерират съседни решения. Най-доброто от съседните решения се сравнява с текущото най-добро решение. То се възприема ако подобрява текущото решение. Основната цел е да се подобри бързодействието на алгоритъма и да се постигат по-добри решения.

В тази глава сме конструирали процедура за локално търсене, свързана със спецификата на МКР. Целта е намиране на по-добри решения [57].

Решенията на МКР се представят като двоичен низ, като 1 съответства на избраните обекти, а 0 на тези, които не са избрани. В нашата процедура за локално търсене избираме по случаен начин две позиции в построеното от мравката решение. Ако една от избраните позиции има стойност 0, я заменяме с 1, а ако им стойност 1 я заменяме с 0. Проверяваме дали новополученото решение е валидно. Ако решението е валидно го сравняваме с текущото (първоначалното) решение. Ако новогенерираното решение е по-добро от текущото, то го заменяме с него.

Прилагаме тази процедура за локално търсене еднократно в края на всяка итерация, независимо дали новото решение е по-добро от текущото или не. По този начин, предложената процедура за локална оптимизация разнообразява търсенето, без съществено да увеличава използваните компютърни ресурси.

Предложеният хибриден АСО алгоритъм е тестван върху същите 10 тестови примера, използвани в предходните секции. За изчисляване на средните стойности сме направили 30 независими пускания на всеки от тестовите примери, тъй като алгоритъмът е стохастичен и за да гарантираме устойчивостта на получените средни стойности. Приложили сме ANOVA тест за статистически анализ, за да сме сигурни, че получената разлика на средните резултати е съществена. Използваните параметри за АСО алгоритъма са показани на Таблица 3.3.

В Таблица 3.4 са показани средните резултати от 30 пускания на тестовите примери, когато е прилаган АСО с локално търсене и АСО без локално търсене. На последния ред е средното време за работа на алгоритъма.

Таблица 3.4 показва, че за 8 от 10 тестови примера хибридният АСО алгоритъм постига по-добри решения от традиционния. За примери МКР $100 \times 10-02$ и МКР $100 \times 10-10$ получените резултати са статистически идентични. Основният проблем на хибридните алгоритми с включено локално търсене е увеличаване на времето за изчисление. Опитали сме

Таблица 3.3: АСО параметри

Параметри	Стойност
Брой итерации	300
ρ	0.5
τ_0	0.5
Брой мравки	20
a	1
b	1
d_1	1
d_2	1

Таблица 3.4: Сравняване на АСО алгоритмите

примери	хибриден АСО	традиционен АСО
МКР 100 × 10-01	22022.73	21989.43
МКР 100 × 10-02	22071.46	22081.36
МКР 100 × 10-03	21089.3	21027.63
МКР 100 × 10-04	21846	21635.3
МКР 100 × 10-05	21814.3	21717.3
МКР 100 × 10-06	21989.26	21869.73
МКР 100 × 10-07	21506.26	21477.3
МКР 100 × 10-08	21672.53	21606.43
МКР 100 × 10-09	22272.36	22257
МКР 100 × 10-10	40626.66	40623.26
време за изчисляване	64.052 s	65.552 s

се да предложим ефективна и в същото време консумираща малко време процедура за локално търсене. Ние генерираме само едно съседно решение. Ако то е по-добро от текущото то го замества и новият феромон се добавя в съответствие с решението, получено след локалното търсене. Тази процедура се прилага към всички получени решения, но еднократно, независимо дали новото решение е по-добро или не. Както се вижда от таблицата, увеличаването на времето за изпълнение на алгоритъма е само 2.34%.

Можем да заключим, че предложената процедура за локално търсене е ефективна и ефикасна. Действието на алгоритъма е подоброено без съществено увеличаване на времето за изпълнението му.

3.6 Интеркритериален анализ

Процедурата за локално търсене е метод за хибридизация и подобряване на основния алгоритъм, при решаването на сложни задачи. Тя помага за избягването на локалните оптимуми и по-бързо да се намерят решения, близки до оптималното. Теорията за интуиционистки размита логика, която е в основата на интеркритериалния анализ (ICrA) е използвана за изучаване на поведението на предложени хибриден алгоритъм. Приложени са разнообразни алгоритми за интеркритериален анализ за сравняване на хибридни АСО алгоритъм с традиционния [63].

Авторите на [9] са предложили нов подход за оценка наречен интеркритериален анализ. Този подход е на основата на индексните матрици [6] и интуиционистки размитите множества [7, 13]. Накратко, интуиционистки размитите двойки са наредени двойки от реални, не отрицателни числа $\langle a, b \rangle$, за които $a, b \in [0, 1]$ и $a + b \leq 1$, които се използват за оценяване на някакъв обект или процес. Компонентите (a и b) на интуиционистки размитата двойка могат да бъдат интерпретирани като степен на "принадлежност" и "непринадлежност" към дадено множество, степен на "съгласие" и "несъгласие" степен на "валидност" и "невалидност" и други.

Нека е дадена Индексна Матрица (ИМ) [6, 10], чието индексно множество се състои от имена на критерии (за редове) и обекти (за колони). Освен това се предполага, че елементите на тази ИМ са реални числа. Ще се получи ИМ с набори от индекси, състоящи се от имената на критериите (за редове и колони) с елементи интуиционистки размити двойки, съответстващи на "съгласие" и "несъгласие" на съответните

критерии. Освен това се предполагат две неща: (i) всички критерии предоставят оценка за всички обекти и всички тези оценки са налични; (ii) всички оценки на дадени критерии могат да бъдат сравнени помежду си. Множеството от всички оценявани обекти се обозначава с O , а наборът от стойности, присвоени по даден критерий C на обектите с $C(O)$.

Нека имаме индексна матрица [10]

$$A = \begin{array}{c|cccccc} & O_1 & \dots & O_k & \dots & O_l & \dots & O_n \\ \hline C_1 & a_{C_1, O_1} & \dots & a_{C_1, O_k} & \dots & a_{C_1, O_l} & \dots & a_{C_1, O_n} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ C_i & a_{C_i, O_1} & \dots & a_{C_i, O_k} & \dots & a_{C_i, O_l} & \dots & a_{C_i, O_n} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ C_j & a_{C_j, O_1} & \dots & a_{C_j, O_k} & \dots & a_{C_j, O_l} & \dots & a_{C_j, O_n} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ C_m & a_{C_m, O_1} & \dots & a_{C_m, O_k} & \dots & a_{C_m, O_l} & \dots & a_{C_m, O_n} \end{array},$$

където за всяко p, q , ($1 \leq p \leq m, 1 \leq q \leq n$):

- C_p е критерий участващ в оценяването;
- O_q е обект, който се оценява;
- a_{C_p, O_q} е реално число или друг обект, който е съпоставим относно връзката R с друг a -обект, така, че за всяко i, j, k : $R(a_{C_k, O_i}, a_{C_k, O_j})$ е дефинирано. Нека \bar{R} е двойствено отношение на R в смисъл на R удовлетворено, тогава \bar{R} не е удовлетворено и обратното. Например, ако " R " е отношението " $<$ " тогава \bar{R} е отношението " $>$ " и обратното.

Нека $S_{k,l}^\mu$ е броят на случаите в които $R(a_{C_k, O_i}, a_{C_k, O_j})$ и $R(a_{C_l, O_i}, a_{C_l, O_j})$ са едновременно удовлетворени. Нека $S_{k,l}^\nu$ е броят на случаите за които $R(a_{C_k, O_i}, a_{C_k, O_j})$ и $\bar{R}(a_{C_l, O_i}, a_{C_l, O_j})$ са едновременно удовлетворени.

Очевидно,

$$S_{k,l}^\mu + S_{k,l}^\nu \leq \frac{n(n-1)}{2}.$$

Завсяко k, l , така че $1 \leq k < l \leq m$ и за $n \geq 2$, дефинираме

$$\mu_{C_k, C_l} = 2 \frac{S_{k,l}^\mu}{n(n-1)}, \quad \nu_{C_k, C_l} = 2 \frac{S_{k,l}^\nu}{n(n-1)}.$$

Тогава, $\langle \mu_{C_k, C_l}, \nu_{C_k, C_l} \rangle$ е интуиционистки размита двойка. Можем да конструираме индексната матрица по следния начин:

$$\begin{array}{c|ccc}
 & C_1 & \dots & C_m \\
\hline
C_1 & \langle \mu_{C_1, C_1}, \nu_{C_1, C_1} \rangle & \dots & \langle \mu_{C_1, C_m}, \nu_{C_1, C_m} \rangle \\
\vdots & \vdots & \vdots & \\
C_m & \langle \mu_{C_m, C_1}, \nu_{C_m, C_1} \rangle & \dots & \langle \mu_{C_m, C_m}, \nu_{C_m, C_m} \rangle
\end{array},$$

която определят степента на съответствие между критериите C_1, \dots, C_m .

Използвана е платформата ICrAData за изчисляване на ICrA [75].

Получените ICrA резултати са анализирани на базата на предложената в [11] скала за съгласие и несъгласие. Схемата за нивото на съгласие и несъгласие между всяка двойка критерии е представено в Таблица 3.5.

Таблица 3.5: Скала за съгласие и несъгласие [11]

Интервал за $\mu_{C, C'}$	Значение
[0.00-0.05]	strong negative consonance (SNC)
(0.05-0.15]	negative consonance (NC)
(0.15-0.25]	weak negative consonance (WNC)
(0.25-0.33]	weak dissonance (WD)
(0.33-0.43]	dissonance (D)
(0.43-0.57]	strong dissonance (SD)
(0.57-0.67]	dissonance (D)
(0.67-0.75]	weak dissonance (WD)
(0.75-0.85]	weak positive consonance (WPC)
(0.85-0.95]	positive consonance (PC)
(0.95-1.00]	strong positive consonance (SPC)

Получените резултати от прилагането на ACO и хибриден ACO са анализирани с използването на разнообразни алгоритми за интеркритериален анализ. Входната матрица за интеркритериалния анализ е представена на Таблица 3.6:

Приложени са няколко разновидности на интеркритериалния анализ.

3.6.1 Резултати от прилагането на μ - базиран ICrA

За изчисляване на интеркритериалния анализ се използва платформата ICrAData [75]. В таблица 3.7 са показани стойностите на μ .

Таблица 3.6: Индексна Матрица за ICrA

	run1	run2	...	run30
$P1_t$	$val_{P1_t,1}$	$val_{P1_t,2}$	\vdots	$val_{P1_t,30}$
$P2_t$	$val_{P2_t,1}$	$val_{P2_t,2}$	\vdots	$val_{P2_t,30}$
\vdots	\vdots	\vdots	\ddots	\vdots
$P10_t$	$val_{P10_t,1}$	$val_{P10_t,2}$	\vdots	$val_{P10_t,30}$
$P1_h$	$val_{P1_h,1}$	$val_{P1_h,2}$	\vdots	$val_{P1_h,30}$
$P2_h$	$val_{P2_h,1}$	$val_{P2_h,2}$	\vdots	$val_{P2_h,30}$
\vdots	\vdots	\vdots	\ddots	\vdots
$P10_h$	$val_{P10_h,1}$	$val_{P10_h,2}$	\vdots	$val_{P10_h,30}$

(3.8)
Таблица 3.7: Степен на съгласие – $\mu_{C,C'}$ (μ - базиран ICrA)

μ	$P1_t$	$P2_t$	$P3_t$	$P4_t$	$P5_t$	$P6_t$	$P7_t$	$P8_t$	$P9_t$	$P10_t$
$P1_h$	1	0.46	0.51	0.5	0.45	0.38	0.45	0.46	0.37	0.55
$P2_h$	0.42	1	0.48	0.3	0.48	0.39	0.36	0.37	0.34	0.36
$P3_h$	0.57	0.57	1	0.5	0.46	0.46	0.46	0.43	0.39	0.47
$P4_h$	0.47	0.39	0.55	1	0.35	0.47	0.4	0.41	0.51	0.61
$P5_h$	0.25	0.34	0.33	0.27	1	0.36	0.35	0.34	0.34	0.34
$P6_h$	0.4	0.45	0.5	0.57	0.43	1	0.44	0.46	0.46	0.48
$P7_h$	0.41	0.3	0.44	0.38	0.4	0.38	1	0.51	0.36	0.38
$P8_h$	0.4	0.42	0.38	0.42	0.4	0.37	0.39	1	0.42	0.29
$P9_h$	0.45	0.38	0.35	0.44	0.49	0.49	0.39	0.39	1	0.41
$P10_h$	0.55	0.52	0.5	0.52	0.46	0.63	0.48	0.54	0.38	1

В Таблица 3.8 са показани стойностите на ν .

В Таблица 3.9 е показана степента на неопределеност.

3.6.2 Резултати от прилагането на ν -базиран ICrA

В Таблица 3.10 е показана стойността на μ .

Таблица 3.8: Степен на несъгласие – $\nu_{C,C'}$ (μ - базиран ISrA)

ν	$P1_t$	$P2_t$	$P3_t$	$P4_t$	$P5_t$	$P6_t$	$P7_t$	$P8_t$	$P9_t$	$P10_t$
$P1_h$	0	0.51	0.47	0.48	0.52	0.54	0.52	0.5	0.4	0.43
$P2_h$	0.44	0	0.38	0.56	0.37	0.4	0.48	0.46	0.38	0.49
$P3_h$	0.43	0.4	0	0.49	0.52	0.47	0.5	0.54	0.4	0.51
$P4_h$	0.49	0.55	0.42	0	0.59	0.43	0.54	0.53	0.26	0.36
$P5_h$	0.46	0.36	0.38	0.43	0	0.35	0.37	0.34	0.27	0.37
$P6_h$	0.51	0.45	0.41	0.33	0.46	0	0.44	0.42	0.31	0.43
$P7_h$	0.41	0.5	0.37	0.43	0.41	0.4	0	0.28	0.34	0.43
$P8_h$	0.37	0.35	0.39	0.35	0.36	0.36	0.37	0	0.24	0.47
$P9_h$	0.47	0.53	0.57	0.48	0.41	0.4	0.52	0.51	0	0.51
$P10_h$	0.43	0.44	0.48	0.45	0.5	0.29	0.47	0.42	0.39	0

Таблица 3.9: Степен на неопределеност – $\pi_{C,C'}$ (μ - базиран ISrA)

π	$P1_t$	$P2_t$	$P3_t$	$P4_t$	$P5_t$	$P6_t$	$P7_t$	$P8_t$	$P9_t$	$P10_t$
$P1_h$	0	0.03	0.02	0.02	0.03	0.08	0.03	0.04	0.23	0.02
$P2_h$	0.14	0	0.14	0.14	0.15	0.21	0.16	0.17	0.28	0.15
$P3_h$	0	0.03	0	0.01	0.02	0.07	0.04	0.03	0.21	0.02
$P4_h$	0.04	0.06	0.03	0	0.06	0.1	0.06	0.06	0.23	0.03
$P5_h$	0.29	0.3	0.29	0.3	0	0.29	0.28	0.32	0.39	0.29
$P6_h$	0.09	0.1	0.09	0.1	0.11	0	0.12	0.12	0.23	0.09
$P7_h$	0.18	0.2	0.19	0.19	0.19	0.22	0	0.21	0.3	0.19
$P8_h$	0.23	0.23	0.23	0.23	0.24	0.27	0.24	0	0.34	0.24
$P9_h$	0.08	0.09	0.08	0.08	0.1	0.11	0.09	0.1	0	0.08
$P10_h$	0.02	0.04	0.02	0.03	0.04	0.08	0.05	0.04	0.23	0

В Таблица 3.11 е показана стойността на ν .

В Таблица 3.12 е показана стойността на π .

Таблица 3.10: Степен на съгласие – $\mu_{C,C'}$ (ν -базиран ICrA)

μ	$P1_t$	$P2_t$	$P3_t$	$P4_t$	$P5_t$	$P6_t$	$P7_t$	$P8_t$	$P9_t$	$P10_t$
$P1_h$	1	0.46	0.51	0.50	0.45	0.38	0.45	0.46	0.37	0.55
$P2_h$	0.42	1	0.48	0.30	0.48	0.39	0.36	0.37	0.30	0.36
$P3_h$	0.57	0.57	1	0.50	0.46	0.46	0.46	0.43	0.39	0.47
$P4_h$	0.47	0.39	0.55	1	0.35	0.47	0.40	0.41	0.50	0.60
$P5_h$	0.25	0.33	0.33	0.27	1	0.32	0.33	0.34	0.29	0.34
$P6_h$	0.40	0.44	0.50	0.57	0.43	1	0.44	0.46	0.42	0.47
$P7_h$	0.41	0.30	0.44	0.38	0.39	0.36	1	0.50	0.31	0.38
$P8_h$	0.40	0.41	0.38	0.41	0.40	0.36	0.38	1	0.37	0.29
$P9_h$	0.45	0.38	0.35	0.44	0.49	0.47	0.38	0.39	1	0.41
$P10_h$	0.55	0.52	0.50	0.52	0.46	0.63	0.48	0.54	0.38	1

Таблица 3.11: Степен на несъгласие – $\nu_{C,C'}$ -values (ν -базиран ICrA)

ν	$P1_t$	$P2_t$	$P3_t$	$P4_t$	$P5_t$	$P6_t$	$P7_t$	$P8_t$	$P9_t$	$P10_t$
$P1_h$	0	0.51	0.47	0.48	0.52	0.54	0.52	0.50	0.40	0.43
$P2_h$	0.44	0	0.38	0.56	0.37	0.40	0.49	0.46	0.41	0.49
$P3_h$	0.43	0.40	0	0.49	0.52	0.47	0.50	0.54	0.40	0.51
$P4_h$	0.49	0.55	0.42	0	0.59	0.43	0.54	0.53	0.27	0.36
$P5_h$	0.47	0.36	0.38	0.43	0	0.38	0.38	0.34	0.32	0.38
$P6_h$	0.51	0.46	0.41	0.33	0.46	0	0.44	0.42	0.34	0.43
$P7_h$	0.41	0.50	0.37	0.43	0.41	0.41	0	0.29	0.38	0.43
$P8_h$	0.37	0.36	0.39	0.36	0.36	0.37	0.37	0	0.30	0.48
$P9_h$	0.47	0.53	0.57	0.48	0.41	0.41	0.52	0.51	0	0.51
$P10_h$	0.43	0.44	0.48	0.45	0.50	0.29	0.47	0.42	0.39	0

3.6.3 Резултати от прилагането на Безпристрастен ICrA

В Таблица 3.13 е показана стойността на μ .

В Таблица 3.14 е представена стойността на ν .

В Таблица 3.15 е представена степента на неопределеност.

Таблица 3.12: Степен на неопределеност – $\pi_{C,C'}$ (ν -базиран ICrA)

π	$P1_t$	$P2_t$	$P3_t$	$P4_t$	$P5_t$	$P6_t$	$P7_t$	$P8_t$	$P9_t$	$P10_t$
$P1_h$	0	0.03	0.01	0.02	0.03	0.08	0.03	0.04	0.22	0.02
$P2_h$	0.14	0	0.14	0.14	0.16	0.21	0.16	0.17	0.29	0.15
$P3_h$	0.00	0.03	0	0.02	0.03	0.08	0.03	0.04	0.22	0.01
$P4_h$	0.03	0.05	0.03	0	0.05	0.10	0.06	0.06	0.23	0.04
$P5_h$	0.29	0.30	0.29	0.30	0	0.29	0.29	0.32	0.39	0.28
$P6_h$	0.09	0.10	0.09	0.10	0.11	0	0.12	0.12	0.23	0.09
$P7_h$	0.18	0.20	0.19	0.19	0.20	0.23	0	0.21	0.30	0.19
$P8_h$	0.23	0.23	0.23	0.23	0.24	0.27	0.25	0	0.33	0.23
$P9_h$	0.07	0.09	0.08	0.09	0.09	0.12	0.09	0.11	0	0.08
$P10_h$	0.01	0.03	0.02	0.03	0.03	0.09	0.04	0.05	0.23	0

Таблица 3.13: Степен на съгласие – $\mu_{C,C'}$ (Безпристрастен ICrA)

μ	$P1_t$	$P2_t$	$P3_t$	$P4_t$	$P5_t$	$P6_t$	$P7_t$	$P8_t$	$P9_t$	$P10_t$
$P1_h$	1	0.46	0.51	0.50	0.45	0.38	0.45	0.46	0.37	0.55
$P2_h$	0.42	1	0.48	0.30	0.48	0.39	0.36	0.37	0.30	0.36
$P3_h$	0.57	0.57	1	0.50	0.46	0.46	0.46	0.43	0.39	0.47
$P4_h$	0.47	0.39	0.55	1	0.35	0.47	0.40	0.41	0.50	0.60
$P5_h$	0.25	0.33	0.33	0.27	1	0.32	0.33	0.34	0.29	0.34
$P6_h$	0.40	0.44	0.50	0.57	0.43	1	0.44	0.46	0.42	0.47
$P7_h$	0.41	0.30	0.44	0.38	0.39	0.36	1	0.50	0.31	0.38
$P8_h$	0.40	0.41	0.38	0.41	0.40	0.36	0.38	1	0.37	0.29
$P9_h$	0.45	0.38	0.35	0.44	0.49	0.47	0.38	0.39	1	0.41
$P10_h$	0.55	0.52	0.50	0.52	0.46	0.63	0.48	0.54	0.38	1

3.6.4 Резултати от прилагането на Балансиран ICrA

В Таблица 3.16 е представена стойността на μ .

Таблица 3.14: Степен на несъгласие – $\nu_{C,C'}$ (Безпристрастен ICrA)

ν	$P1_t$	$P2_t$	$P3_t$	$P4_t$	$P5_t$	$P6_t$	$P7_t$	$P8_t$	$P9_t$	$P10_t$
$P1_h$	0	0.51	0.47	0.48	0.52	0.54	0.52	0.50	0.40	0.43
$P2_h$	0.44	0	0.38	0.56	0.37	0.40	0.48	0.46	0.38	0.49
$P3_h$	0.43	0.40	0	0.49	0.52	0.47	0.50	0.54	0.40	0.51
$P4_h$	0.49	0.55	0.42	0	0.59	0.43	0.54	0.53	0.26	0.36
$P5_h$	0.46	0.36	0.38	0.43	0	0.35	0.37	0.34	0.27	0.37
$P6_h$	0.51	0.45	0.41	0.33	0.46	0	0.44	0.42	0.31	0.43
$P7_h$	0.41	0.50	0.37	0.43	0.41	0.40	0	0.28	0.34	0.43
$P8_h$	0.37	0.35	0.39	0.35	0.36	0.36	0.37	0	0.24	0.47
$P9_h$	0.47	0.53	0.57	0.48	0.41	0.40	0.52	0.51	0	0.51
$P10_h$	0.43	0.44	0.48	0.45	0.50	0.29	0.47	0.42	0.39	0

Таблица 3.15: Степен на неопределеност – $\pi_{C,C'}$ (Безпристрастен ICrA)

π	$P1_t$	$P2_t$	$P3_t$	$P4_t$	$P5_t$	$P6_t$	$P7_t$	$P8_t$	$P9_t$	$P10_t$
$P1_h$	0	0.03	0.01	0.02	0.03	0.08	0.04	0.04	0.22	0.02
$P2_h$	0.14	0	0.14	0.15	0.16	0.21	0.16	0.17	0.32	0.15
$P3_h$	0.00	0.03	0	0.02	0.03	0.08	0.03	0.04	0.22	0.01
$P4_h$	0.03	0.05	0.03	0	0.05	0.10	0.06	0.06	0.24	0.04
$P5_h$	0.29	0.31	0.29	0.30	0	0.33	0.30	0.32	0.45	0.29
$P6_h$	0.09	0.10	0.09	0.10	0.11	0	0.12	0.12	0.27	0.10
$P7_h$	0.18	0.20	0.19	0.19	0.20	0.24	0	0.21	0.35	0.19
$P8_h$	0.23	0.24	0.23	0.23	0.25	0.29	0.25	0	0.39	0.24
$P9_h$	0.07	0.09	0.08	0.09	0.09	0.13	0.10	0.11	0	0.08
$P10_h$	0.01	0.03	0.02	0.03	0.03	0.09	0.04	0.05	0.23	0

В Таблица 3.17 е представена стойността на ν .

В Таблица 3.18 е представена степента на неопределеност.

Таблица 3.16: Степен на съгласие – $\mu_{C,C'}$ (Блансиран ICrA)

μ	$P1_t$	$P2_t$	$P3_t$	$P4_t$	$P5_t$	$P6_t$	$P7_t$	$P8_t$	$P9_t$	$P10_t$
$P1_h$	1	0.46	0.51	0.50	0.45	0.38	0.45	0.46	0.37	0.55
$P2_h$	0.42	1	0.48	0.30	0.48	0.39	0.36	0.37	0.32	0.36
$P3_h$	0.57	0.57	1	0.50	0.46	0.46	0.46	0.43	0.39	0.47
$P4_h$	0.47	0.39	0.55	1	0.35	0.47	0.40	0.41	0.50	0.61
$P5_h$	0.25	0.34	0.33	0.27	1	0.34	0.34	0.34	0.31	0.34
$P6_h$	0.40	0.45	0.50	0.57	0.43	1	0.44	0.46	0.44	0.47
$P7_h$	0.41	0.30	0.44	0.38	0.39	0.37	1	0.50	0.34	0.38
$P8_h$	0.40	0.41	0.38	0.41	0.40	0.36	0.39	1	0.40	0.29
$P9_h$	0.45	0.38	0.35	0.44	0.49	0.48	0.38	0.39	1	0.41
$P10_h$	0.55	0.52	0.50	0.52	0.46	0.63	0.48	0.54	0.38	1

Таблица 3.17: Степен на несъгласие – $\nu_{C,C'}$ (Балансиран ICrA)

ν	$P1_t$	$P2_t$	$P3_t$	$P4_t$	$P5_t$	$P6_t$	$P7_t$	$P8_t$	$P9_t$	$P10_t$
$P1_h$	0	0.51	0.47	0.48	0.52	0.54	0.52	0.50	0.40	0.43
$P2_h$	0.44	0	0.38	0.56	0.37	0.40	0.48	0.46	0.40	0.49
$P3_h$	0.43	0.40	0	0.49	0.52	0.47	0.50	0.54	0.40	0.51
$P4_h$	0.49	0.55	0.42	0	0.59	0.43	0.54	0.53	0.27	0.36
$P5_h$	0.47	0.36	0.38	0.43	0	0.36	0.37	0.34	0.30	0.38
$P6_h$	0.51	0.46	0.41	0.33	0.46	0	0.44	0.42	0.33	0.43
$P7_h$	0.41	0.50	0.37	0.43	0.41	0.40	0	0.28	0.36	0.43
$P8_h$	0.37	0.35	0.39	0.36	0.36	0.37	0.37	0	0.27	0.47
$P9_h$	0.47	0.53	0.57	0.48	0.41	0.40	0.52	0.51	0	0.51
$P10_h$	0.43	0.44	0.48	0.45	0.50	0.29	0.47	0.42	0.39	0

В Таблицы 3.7, 3.10, 3.13 и 3.16 (стойностите на μ), Таблица 3.8, 3.11, 3.14 и 3.17 (стойностите на ν) са показани резултатите от четирите варианта на интеркритериален анализ приложен към МКР. Стойностите на π са представени в Таблицы 3.9, 3.12, 3.15 и 3.18. Представени са решения на едни и същи тестови примери при прилагане на различни варианти на АСО. Според критериите от [7], резултатите от прилагането на интеркритериален анализ показват, че двойките критерии са в несъответствие или в строго несъответствие. Това означава, че двата разглеждани алго-

Таблица 3.18: Степен на неопределеност – $\pi_{C,C'}$ - (Балансиран ICrA)

π	$P1_t$	$P2_t$	$P3_t$	$P4_t$	$P5_t$	$P6_t$	$P7_t$	$P8_t$	$P9_t$	$P10_t$
$P1_h$	0	0.03	0.01	0.02	0.03	0.08	0.03	0.04	0.22	0.02
$P2_h$	0.14	0	0.14	0.14	0.16	0.21	0.16	0.17	0.29	0.15
$P3_h$	0.00	0.03	0	0.02	0.03	0.08	0.03	0.04	0.22	0.01
$P4_h$	0.03	0.05	0.03	0	0.05	0.10	0.06	0.06	0.23	0.04
$P5_h$	0.29	0.30	0.29	0.30	0	0.29	0.29	0.32	0.39	0.28
$P6_h$	0.09	0.10	0.09	0.10	0.11	0	0.12	0.12	0.23	0.09
$P7_h$	0.18	0.20	0.19	0.19	0.20	0.23	0	0.21	0.30	0.19
$P8_h$	0.23	0.23	0.23	0.23	0.24	0.27	0.25	0	0.33	0.23
$P9_h$	0.07	0.09	0.08	0.09	0.09	0.12	0.09	0.11	0	0.08
$P10_h$	0.01	0.03	0.02	0.03	0.03	0.09	0.04	0.05	0.23	0

ритъма (традиционен АСО и хибриден АСО) работят по различен начин.

Можем да кажем, че резултатите получени от прилагането на интеркритериален анализ са надеждни, имайки предвид стойностите на $\pi_{C,C'}$, показващи нивото на неопределеност. Само за връзките между $P5_t - P5_h$, $P8_t - P8_h$ and $P9_t - P9_h$, има няколко по-високи стойности за $\pi_{C,C'}$, съответно 0.31, 0.25 и 0.24.

Изводът, че хибридният АСО алгоритъм работи по-добре се потвърждава недвусмислено от четирите различни алгоритъма за интеркритериален анализ. Наблюдават се някои разлики в случаите $P9$ и $P6$, които са пренебрежимо малки.

3.7 Стартови стратегии

Случайният старт е един от основните елементи на метода на мравките. Той дава възможност да се използва относително малък брой мравки. Има задачи, за които е важен изборът на начален връх. Между тях са задачите за избор на подмножество, каквато е задачата за раницата (МКР). Ако мравката започва строежа на решение от връх, който не е част от добро решение, то тя няма шанс да постигне добро решение.

За по-добро управление на процеса на построяване на решение, сме включили полу-случаен старт на мравките. Нашата цел е да използваме опита на мравките и да направим алгоритъма по-ефективен. Разделяме

множеството от върхове на графа на задачата на няколко подмножества. Въвеждаме оценка за това колко е добре и колко не е добре мравката да започне построяването на решение от връх принадлежащ на дадено множество, съобразно броя на добрите и на лошите решения стартирали от върхове, принадлежащи на съответното множество [46, 52].

$$D_j(i) = \phi.D_j(i-1) + (\psi - \phi).F_j(i), \quad (3.9)$$

$$E_j(i) = \phi.E_j(i-1) + (\psi - \phi).G_j(i), \quad (3.10)$$

където $i \geq 1$ е номера на текущата итерация и за всяко j ($1 \leq j \leq N$):

$$F_j(i) = \begin{cases} f_{j,A}/n_j & \text{if } n_j \neq 0 \\ F_j(i-1) & \text{otherwise} \end{cases}, \quad (3.11)$$

$$G_j(i) = \begin{cases} g_{j,B}/n_j & \text{if } n_j \neq 0 \\ G_j(i-1) & \text{otherwise} \end{cases}, \quad (3.12)$$

$f_{j,A}$ е броя на решенията между най-добрите $A\%$, $g_{j,B}$ е броя на решенията между най-лишите $B\%$, където $A + B \leq 100$, $i \geq 2$ и

$$\sum_{j=1}^N n_j = n, \quad (3.13)$$

където n_j ($1 \leq j \leq N$) е броя на построените от мравките решения, когато са започнали от връх принадлежащ на подмножеството от върхове j , n е броят на мравките. Началните стойности на тегловите коефициенти е: $D_j(1) = 1$ и $E_j(1) = 0$. Параметрите ϕ и ψ , $0 \leq \phi, \psi \leq 1$ и $\psi \geq \phi$, показват теглото на информацията от предходните итерации и от последната итерация съответно. Когато $\phi = 0$ под внимание се взема само информацията от последната итерация. Ако $\phi = 0.5 \times \psi$, влиянието на информацията от предходните итерации спрямо информацията от последната итерация е еднакво. Когато $\phi = \psi$, под внимание се взема само информацията от предходните итерации. Когато $\phi = 0.25 \times \psi$, влиянието на информацията от предишните итерации е три пъти по-малко отколкото информацията от последната итерация. Когато $\phi = 0.75 \times \psi$, влиянието на информацията от предходните итерации е три пъти по-голямо от влиянието на информацията от последната итерация. Когато сумата

от тегловите коефициенти е по-малка или равна на 1, тогава оценката е интуиционистки размита, а когато е равна на 1 е размита [7].

Предложени са няколко стартови стратегии и комбинации от тях. За всяко множество j , $D_j(i)$ е оценката доколко е добре началният връх на решението да е от това множество, а оценката $E_j(i)$ показва доколко не е добре началният връх на решението да е от това множество, където i е номерът на текущата итерация. Определяме граница D за това дали оценката е добра и граница E под която оценката е лоша. Предложени са следните стартови стратегии [40]:

- 1) Ако $\frac{E_j(i)}{D_j(i)} > E$ тогава за текущата итерация подмножество j е забранено. Началният връх се избира по случаен начин от $\{j \mid j \text{ не е забранено}\}$;
- 2) Ако $\frac{E_j(i)}{D_j(i)} > E$ тогава до края на алгоритъма подмножество j е забранено. Началният връх се избира по случаен начин от множество $\{j \mid j \text{ не е забранено}\}$;
- 3) Ако $\frac{E_j(i)}{D_j(i)} > E$ тогава за K_1 поредни итерации подмножество j е забранено. Началният връх се избира по случаен начин от $\{j \mid j \text{ не е забранено}\}$;
- 4) Нека $r_1 \in [\frac{1}{2}, 1)$ и $r_2 \in [0, 1]$ са случайни числа. Ако $r_2 > r_1$ избираме по случаен начин връх от подмножество $\{j \mid D_j(i) > D\}$, в противен случай се избира връх от незабранено подмножество. r_1 е избрано и фиксирано в началото на алгоритъма.
- 5) Нека $r_1 \in [\frac{1}{2}, 1)$ и $r_2 \in [0, 1]$ са случайни числа. Ако $r_2 > r_1$ избираме случаен връх от подмножество $\{j \mid D_j(i) > D\}$, в противен случай се избира връх от незабранено подмножество. r_1 се избира в началото на алгоритъма и нараства с r_3 на всяка итерация.

$K_1, K_1 \in [0, \text{брой итерации}]$ е параметър.

Прилагаме интеркритериален анализ към резултатите, постигнати от АСО с прилагането на разнообразни стартови стратегии [40]. Използвани са същите тестови примери като в предходните секции. Използвани са следните параметри за стартовите стратегии: $A = 30$, $B = 30$, $D = 1$, $E = 0.1$, $K_1 = K_2 = 5$, $r_3 = 0.01$. Броят на върховете в подмножествата е

един и същ за всички подмножества и варира от 1 до 10, по-точно 1, 2, 4, 5 и 10. Пресметнатата е средната стойност на целевата функция получена от 30 пускания на 10 тестови примера. Използвани са оценки от 0 до 10. Получените резултати са показани в Таблица 3.19.

От Таблица 3.19 може да се види, че интеркритериалните обекти (O_1, O_2, \dots, O_{20}) са различния брой върхове в подмножествата от върхове и четири случая за стойността на параметъра φ , $\varphi = [0 \ 0.25 \ 0.5 \ 0.75]$. Интеркритериалните критерии (C_1, C_2, \dots, C_{12}) са 12 различни стартови стратегии и техните комбинации. В случая на интуиционистки размита оценка резултатите са показани на Таблица 3.20.

Аналогично подмножествата от върхове съдържат равен брой върхове, който варира от 1 до 10 или по-точно 1, 2, 4, 5 и 10. Пресметнат е средният резултат върху всичките 10 тестови примера, като всеки от тях е пуснат 30 пъти с всяка една от стартовите стратегии. Получените резултати са класирани с оценки от 10 до 100. Стойността на параметъра φ се движи между 0 и 0.75, а стойността на параметъра ψ се движи между 0.25 и 0.825.

За анализ на получените резултати ще използваме същата както в предходните секции скала за принадлежност и непринадлежност. В Таблица 3.5 са посочени стойностите за определяне нивото на принадлежност и непринадлежност.

След прилагането на интеркритериален анализ са получени две индексни матрици, показващи връзките между разглежданите 12 критерия (стартови стратегии). Матриците за стойностите на $\mu_{C,C'}$ и $\nu_{C,C'}$ са показани в Таблица 3.21 и Таблица 3.22.

От таблиците виждаме, че някои от резултатите имат много високо ниво на неопределеност. Това се получава, защото за някои стартови стратегии имаме едни и същи резултати за различни интеркритериални обекти. Например резултатите за Стратегия 5 и Стратегия 6 в Таблица 3.19.

За по-доброто възприемане и анализиране на получените стойности за $\mu_{C,C'}$, $\nu_{C,C'}$ и $\pi_{C,C'}$, те са сортирани спрямо стойностите на $\mu_{C,C'}$. Те са представени в Таблица 3.23 и Таблица 3.24. Таблица 3.23 показва двойките критерии с високо ниво на "съгласие" ($\mu_{C,C'}$) и ниско ниво на "несъгласие" ($\nu_{C,C'}$). Таблица 3.24 показва двойките критерии с високо ниво на "неопределеност".

От Таблицы 3.23 и 3.24 виждаме, че отношението между критерий C_1 и критерии C_5, C_6 има най-висока стойност за $\mu_{C,C'}$ ($\mu_{C,C'} = 1$), т.е.

тези критерии са в строго положително съгласие. Това означава, че АСО алгоритъмът работи по подобен начин когато е с напълно случаен избор на начален връх и когато използва Стратегии 4 и 5.

В Стратегии 4 и 5 няма забранени подмножества от върхове, подобно на напълно случайния избор. В тези два случая се използва различно ниво на вероятност за избор при различните подмножества от върхове. Други двойки критерии с най-висока стойност за $\mu_{C,C'}$ ($\mu_{C,C'} = 1$) са $C_7 - C_8$ и $C_{11} - C_{12}$. Това означава, че когато АСО алгоритъмът използва Стратегии 1 - 4, Стратегии 1 - 5, Стратегии 3 - 4 и Стратегии 3 - 5 също имат много сходно поведение.

Критерии, които също са в съгласие, макар и в по-ниска степен (строго положително съгласие, положително съгласие и слабо положително съгласие) са между двойките критерии C_2, C_3, C_4 и C_7, C_8, \dots, C_{12} . Те съответстват на Стратегии 1, 2 и 3 и комбинации между тях. При всички тези стратегии има забранени подмножества от върхове, за това АСО алгоритъма работи по подобен начин прилагайки тези стратегии.

Двойките критерии със стойност $\mu_{C,C'} = [0.75 - 0.25]$ са в несъгласие, т.е. няма зависимост между критериите. В зависимост от разглежданата скала (Таблица 3.5) тези критерии са независими. АСО алгоритмите използват само стратегии за случаен избор и тези използващи забрани работят по напълно различен начин и за това няма никаква връзка между тях.

След прилагането на интеркритериален анализ върху резултати получени от прилагането на разнообразни стартови стратегии с интуиционистки размита оценка сме получили две индексни матрици за връзката между разглежданите 12 критерия (стратегии). В резултат сме получили индексни матрици за стойностите на $\mu_{C,C'}$ и $\nu_{C,C'}$, които са показани в Таблица 3.25 и Таблица 3.26.

Стойностите за $\pi_{C,C'}$ са високи, когато една от стратегиите е използвала само случаен избор на начален връх (контролиран или не), а другата използва само забрани. Когато и двете стратегии са само със случаен избор или само със забрани, то стойността на $\mu_{C,C'}$ е близка до 1. Това означава, че стратегиите със забрани работят по съвсем различен начин от стратегиите само със случаен избор. Когато едната от стратегиите е комбинация от случаен избор и забрана, тогава стойността на $\pi_{C,C'}$ не е толкова голяма, както в предишните случаи. Това означава, че все пак има някакво подобие в поведението на двата варианта.

Отново и в този случай на оценяване, за по-доброто възприемане

и анализиране на получените резултати стойностите за $\mu_{C,C'}$, $\nu_{C,C'}$, и $\pi_{C,C'}$ на двойките критерии са подредени по отношение на стойността за $\mu_{C,C'}$. Списъкът е представен в Таблицы 3.27 и 3.28. Таблица 3.27 показва двойките критерии започвайки с високо ниво на съгласие ($\mu_{C,C'}$) и ниско ниво на несъгласие ($\nu_{C,C'}$).

Виждаме, че за всички двойки критерии нивото на несъгласие е ниско. Това означава, че няма двойки критерии при които алгоритъмът да работи по противоположен начин. Таблица 3.28 показва двойките критерии с високо ниво на неопределеност. Сравнявайки резултатите от Таблица 3.25 и Таблица 3.26 можем да направим извода, че интуиционистки размитата оценка е по-реалистична.

3.8 Заключение

В тази глава беше разгледано прилагането на метода на мравките за решаване на задачата за раницата. Първо е приложен традиционен метод на мравките като са изследвани разнообразни варианти за конструиране на функцията на евристиката. Предложен е вариант на АСО с използване на два варианта на размит феромон. Направено е сравнение между вариантите и е показано, че при задачи за подмножества е по-добре коефициентите на размития феромон да се преизчисляват в началото на всяка итерация. Разработена е процедура за локално търсене, съобразена със спецификата на задачата. Тази процедура е комбинирана с АСО. Тя се прилага еднократно на всяка итерация, независимо дали подобрява решението или не. Така времето за работа на алгоритъма се увеличава незначително. Направен е интеркритериален анализ, който показва, че прилагането на локално търсене съществено допринася за разнообразяване на избора на следващ връх. Предложени са стартови стратегии за избор на начален връх на решението. Предложена е фиксирана и интуиционистки размита оценка на стратегиите. Приложен е интеркритериален анализ който показва връзката между стратегиите. От него можем да заключим, че когато изборът е със забраняване на подмножества от върхове, алгоритъмът работи доста по-различно от варианта на случаен избор (контролиран или не).

Таблица 3.19: Оценяване на стратегиите

Критерии	Стратегии	$\varphi = 0$					$\varphi = 0.25$				
		10	5	4	2	1	10	5	4	2	1
		O_1	O_2	O_3	O_4	O_5	O_6	O_7	O_8	O_9	O_{10}
C_1	Random	32	32	32	32	32	32	32	32	32	32
C_2	Strategy 1	84	84	87	83	83	83	88	86	90	90
C_3	Strategy 2	33	31	36	53	74	32	31	36	61	81
C_4	Strategy 3	79	86	86	88	86	62	86	84	84	96
C_5	Strategy 4	86	86	86	86	86	86	86	86	86	86
C_6	Strategy 5	86	86	86	86	86	86	86	86	86	86
C_7	Strategies 1-4	83	89	84	81	89	84	91	87	92	96
C_8	Strategies 1-5	83	89	84	81	89	84	91	87	92	96
C_9	Strategies 2-4	33	36	35	53	82	34	33	35	59	85
C_{10}	Strategies 2-5	33	36	35	63	82	34	33	35	59	85
C_{11}	Strategies 3-4	69	89	88	87	90	69	83	86	84	97
C_{12}	Strategies 3-5	69	89	88	87	90	69	83	86	84	97
Критерии	Стратегии	$\varphi = 0.5$					$\varphi = 0.75$				
		10	5	4	2	1	10	5	4	2	1
		O_{11}	O_{12}	O_{13}	O_{14}	O_{15}	O_{16}	O_{17}	O_{18}	O_{19}	O_{20}
C_1	Random	32	32	32	32	32	32	32	32	32	32
C_2	Strategy 1	78	86	88	92	96	71	81	85	89	92
C_3	Strategy 2	34	35	38	51	78	35	55	52	60	87
C_4	Strategy 3	61	86	88	94	97	56	76	88	95	95
C_5	Strategy 4	86	86	86	86	86	86	86	86	86	86
C_6	Strategy 5	86	86	86	86	86	86	86	86	86	86
C_7	Strategies 1-4	79	90	87	94	97	67	83	89	94	95
C_8	Strategies 1-5	79	90	87	94	97	67	83	89	94	95
C_9	Strategies 2-4	35	40	44	56	83	39	47	48	58	85
C_{10}	Strategies 2-5	35	40	44	56	83	39	47	48	58	85
C_{11}	Strategies 3-4	68	92	88	92	96	56	81	87	94	97
C_{12}	Strategies 3-5	68	92	88	92	96	56	81	87	94	97

Таблица 3.20: Интуиционистки разпита оценка на стратегиите

Критерии	Стратегии	φ/ψ				
		0.125/0.25	0.125/0.5	0.125/0.75	0.125/0.875	0.25/0.5
		O_1	O_2	O_3	O_4	O_5
C_1	Random	32	32	32	32	32
C_2	Strategy 1	95	93	93	93	93
C_3	Strategy 2	82	79	79	79	79
C_4	Strategy 3	93	92	92	92	92
C_5	Strategy 4	83	83	83	83	83
C_6	Strategy 5	83	83	83	83	83
C_7	Strategies 1-4	96	96	96	96	96
C_8	Strategies 1-5	96	96	96	96	96
C_9	Strategies 2-4	84	83	83	83	83
C_{10}	Strategies 2-5	84	83	83	83	83
C_{11}	Strategies 3-4	94	93	93	93	93
C_{12}	Strategies 3-5	94	93	93	93	93

Критерии	Стратегии	φ/ψ					
		0.25/0.75	0.25/0.875	0.5/0.75	0.5/0.875	0.75/0.875	0.5/1
		O_6	O_7	O_8	O_9	O_{10}	O_{11}
C_1	Random	32	32	32	32	32	32
C_2	Strategy 1	92	92	94	93	93	96
C_3	Strategy 2	85	85	82	77	83	78
C_4	Strategy 3	93	94	99	94	93	97
C_5	Strategy 4	83	83	83	83	83	83
C_6	Strategy 5	83	83	83	83	83	83
C_7	Strategies 1-4	96	96	92	94	95	97
C_8	Strategies 1-5	96	96	92	94	95	97
C_9	Strategies 2-4	83	83	82	81	86	83
C_{10}	Strategies 2-5	83	83	82	81	86	83
C_{11}	Strategies 3-4	94	94	93	93	97	96
C_{12}	Strategies 3-5	94	94	93	93	97	96

Таблица 3.21: Индексна матрица за $\mu_{C,C'}$

	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}	C_{12}
C_1	1.000	0.042	0.016	0.079	1.000	1.000	0.037	0.037	0.026	0.026	0.026	0.026
C_2	0.042	1.000	0.642	0.742	0.042	0.042	0.826	0.826	0.637	0.621	0.737	0.737
C_3	0.016	0.642	1.000	0.670	0.016	0.016	0.653	0.653	0.889	0.884	0.695	0.695
C_4	0.079	0.742	0.670	1.000	0.079	0.079	0.758	0.758	0.705	0.700	0.821	0.821
C_5	1.000	0.042	0.016	0.079	1.000	1.000	0.037	0.037	0.026	0.026	0.026	0.026
C_6	1.000	0.042	0.016	0.079	1.000	1.000	0.037	0.037	0.026	0.026	0.026	0.026
C_7	0.037	0.826	0.653	0.758	0.037	0.037	1.000	1.000	0.695	0.679	0.800	0.800
C_8	0.037	0.826	0.653	0.758	0.037	0.037	1.000	1.000	0.695	0.679	0.800	0.800
C_9	0.026	0.637	0.889	0.705	0.026	0.026	0.695	0.695	1.000	0.984	0.753	0.753
C_{10}	0.026	0.621	0.884	0.700	0.026	0.026	0.679	0.679	0.984	1.000	0.747	0.747
C_{11}	0.026	0.737	0.695	0.821	0.026	0.026	0.800	0.800	0.753	0.747	1.000	1.000
C_{12}	0.026	0.737	0.695	0.821	0.026	0.026	0.800	0.800	0.753	0.747	1.000	1.000

Таблица 3.22: Индексна матрица за $\nu_{C,C'}$

	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}	C_{12}
C_1	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
C_2	0.000	0.000	0.300	0.137	0.000	0.000	0.095	0.095	0.295	0.311	0.195	0.195
C_3	0.000	0.300	0.000	0.237	0.000	0.000	0.295	0.295	0.079	0.079	0.263	0.263
C_4	0.000	0.137	0.237	0.000	0.000	0.000	0.137	0.137	0.189	0.195	0.084	0.084
C_5	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
C_6	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
C_7	0.000	0.095	0.295	0.137	0.000	0.000	0.000	0.000	0.242	0.258	0.137	0.137
C_8	0.000	0.095	0.295	0.137	0.000	0.000	0.000	0.000	0.242	0.258	0.137	0.137
C_9	0.000	0.079	0.079	0.189	0.000	0.000	0.242	0.242	0.000	0.016	0.205	0.205
C_{10}	0.000	0.084	0.084	0.195	0.000	0.000	0.258	0.258	0.016	0.000	0.211	0.211
C_{11}	0.000	0.263	0.263	0.084	0.000	0.000	0.137	0.137	0.205	0.211	0.000	0.000
C_{12}	0.000	0.263	0.263	0.084	0.000	0.000	0.137	0.137	0.205	0.211	0.000	0.000

Таблица 3.23: Стойности на $\mu_{C,C'}$, $\nu_{C,C'}$, $\pi_{C,C'}$ – част 1

Двойки критерии	$\mu_{C,C'}$	$\nu_{C,C'}$	$\pi_{C,C'}$
$C_1 - C_5$	1.000	0.000	0.000
$C_1 - C_6$	1.000	0.000	0.000
$C_5 - C_6$	1.000	0.000	0.000
$C_7 - C_8$	1.000	0.000	0.000
$C_{11} - C_{12}$	1.000	0.000	0.000
$C_9 - C_{10}$	0.984	0.016	0.000
$C_3 - C_9$	0.889	0.079	0.032
$C_3 - C_{10}$	0.889	0.079	0.032
$C_2 - C_7$	0.826	0.095	0.079
$C_2 - C_8$	0.826	0.095	0.079
$C_4 - C_{11}$	0.821	0.084	0.095
$C_4 - C_{12}$	0.821	0.084	0.095
$C_7 - C_{11}$	0.800	0.137	0.063
$C_7 - C_{12}$	0.800	0.137	0.063
$C_8 - C_{11}$	0.800	0.137	0.063
$C_8 - C_{12}$	0.800	0.137	0.063
$C_4 - C_7$	0.758	0.137	0.105
$C_4 - C_8$	0.758	0.137	0.105
$C_9 - C_{11}$	0.753	0.205	0.042
$C_9 - C_{12}$	0.753	0.205	0.042
$C_{10} - C_{11}$	0.747	0.211	0.042
$C_{10} - C_{12}$	0.747	0.211	0.042
$C_2 - C_4$	0.742	0.137	0.121
$C_2 - C_{11}$	0.737	0.195	0.068
$C_2 - C_{12}$	0.737	0.195	0.068
$C_4 - C_9$	0.705	0.189	0.105
$C_4 - C_{10}$	0.700	0.195	0.105
$C_3 - C_{11}$	0.695	0.263	0.042
$C_3 - C_{12}$	0.695	0.263	0.042
$C_7 - C_9$	0.695	0.242	0.063
$C_8 - C_9$	0.695	0.242	0.063
$C_8 - C_{10}$	0.679	0.258	0.063
$C_7 - C_{10}$	0.679	0.258	0.063
$C_3 - C_4$	0.670	0.237	0.084
$C_3 - C_7$	0.653	0.295	0.053
$C_3 - C_8$	0.653	0.295	0.053
$C_2 - C_3$	0.642	0.300	0.058
$C_2 - C_9$	0.637	0.295	0.068
$C_2 - C_{10}$	0.621	0.311	0.068

Таблица 3.24: Стойности за $\mu_{C,C'}$, $\nu_{C,C'}$, $\pi_{C,C'}$ – част 2

Двойки критерии	$\mu_{C,C'}$	$\nu_{C,C'}$	$\pi_{C,C'}$
$C_1 - C_4$	0.079	0.000	0.921
$C_4 - C_5$	0.079	0.000	0.921
$C_4 - C_6$	0.079	0.000	0.921
$C_1 - C_2$	0.042	0.000	0.958
$C_2 - C_5$	0.042	0.000	0.958
$C_2 - C_6$	0.042	0.000	0.958
$C_1 - C_7$	0.037	0.000	0.963
$C_1 - C_8$	0.037	0.000	0.963
$C_5 - C_7$	0.037	0.000	0.963
$C_5 - C_8$	0.037	0.000	0.963
$C_6 - C_7$	0.037	0.000	0.963
$C_6 - C_8$	0.037	0.000	0.963
$C_1 - C_9$	0.026	0.000	0.974
$C_1 - C_{10}$	0.026	0.000	0.974
$C_1 - C_{11}$	0.026	0.000	0.974
$C_1 - C_{12}$	0.026	0.000	0.974
$C_5 - C_9$	0.026	0.000	0.974
$C_5 - C_{10}$	0.026	0.000	0.974
$C_5 - C_{11}$	0.026	0.000	0.974
$C_5 - C_{12}$	0.026	0.000	0.974
$C_6 - C_9$	0.026	0.000	0.974
$C_6 - C_{10}$	0.026	0.000	0.974
$C_6 - C_{11}$	0.026	0.000	0.974
$C_6 - C_{12}$	0.026	0.000	0.974
$C_1 - C_3$	0.016	0.000	0.984
$C_3 - C_5$	0.016	0.000	0.984
$C_3 - C_6$	0.016	0.000	0.984

Таблица 3.25: Индексна матрица за $\mu_{C,C'}$ (интуиционистки размита оценка)

	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}	C_{12}
C_1	1.000	0.291	0.145	0.182	1.000	1.000	0.382	0.382	0.382	0.382	0.327	0.327
C_2	0.291	1.000	0.327	0.509	0.291	0.291	0.364	0.364	0.364	0.364	0.509	0.509
C_3	0.145	0.327	1.000	0.564	0.145	0.145	0.382	0.382	0.564	0.564	0.600	0.600
C_4	0.182	0.509	0.564	1.000	0.182	0.182	0.309	0.309	0.291	0.291	0.564	0.564
C_5	1.000	0.291	0.145	0.182	1.000	1.000	0.382	0.382	0.382	0.382	0.327	0.327
C_6	1.000	0.291	0.145	0.182	1.000	1.000	0.382	0.382	0.382	0.382	0.327	0.327
C_7	0.382	0.364	0.382	0.309	0.382	0.382	1.000	1.000	0.600	0.600	0.473	0.473
C_8	0.382	0.364	0.382	0.309	0.382	0.382	1.000	1.000	0.600	0.600	0.473	0.473
C_9	0.382	0.364	0.564	0.291	0.382	0.382	0.600	0.600	1.000	1.000	0.527	0.527
C_{10}	0.382	0.364	0.564	0.291	0.382	0.382	0.600	0.600	1.000	1.000	0.527	0.527
C_{11}	0.327	0.509	0.600	0.564	0.327	0.327	0.473	0.473	0.527	0.527	1.000	1.000
C_{12}	0.327	0.509	0.600	0.564	0.327	0.327	0.473	0.473	0.527	0.527	1.000	1.000

Таблица 3.26: Индексна матрица за $\nu_{C,C'}$

	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}	C_{12}
C_1	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
C_2	0.000	0.000	0.492	0.236	0.000	0.000	0.218	0.218	0.218	0.218	0.273	0.273
C_3	0.000	0.491	0.000	0.327	0.000	0.000	0.345	0.345	0.164	0.164	0.182	0.182
C_4	0.000	0.236	0.327	0.000	0.000	0.000	0.382	0.382	0.364	0.364	0.182	0.182
C_5	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
C_6	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
C_7	0.000	0.218	0.345	0.382	0.000	0.000	0.000	0.000	0.182	0.182	0.145	0.145
C_8	0.000	0.218	0.345	0.382	0.000	0.000	0.000	0.000	0.182	0.182	0.145	0.145
C_9	0.000	0.218	0.164	0.364	0.000	0.000	0.182	0.182	0.000	0.000	0.018	0.018
C_{10}	0.000	0.218	0.164	0.364	0.000	0.000	0.182	0.182	0.000	0.000	0.018	0.018
C_{11}	0.000	0.273	0.182	0.182	0.000	0.000	0.145	0.145	0.018	0.018	0.000	0.000
C_{12}	0.000	0.273	0.182	0.182	0.000	0.000	0.145	0.145	0.018	0.018	0.000	0.000

Таблица 3.27: Стойности за $\mu_{C,C'}$, $\nu_{C,C'}$, $\pi_{C,C'}$ – част 1 (Интуиционистки размита оценка)

Двойки критериуми	$\mu_{C,C'}$	$\nu_{C,C'}$	$\pi_{C,C'}$
$C_1 - C_5$	1.000	0.000	0.000
$C_1 - C_6$	1.000	0.000	0.000
$C_5 - C_6$	1.000	0.000	0.000
$C_7 - C_8$	1.000	0.000	0.000
$C_9 - C_{10}$	1.000	0.000	0.000
$C_{11} - C_{12}$	1.000	0.000	0.000
$C_3 - C_{11}$	0.600	0.182	0.218
$C_3 - C_{12}$	0.600	0.182	0.218
$C_7 - C_9$	0.600	0.182	0.218
$C_7 - C_{10}$	0.600	0.182	0.218
$C_8 - C_9$	0.600	0.182	0.218
$C_8 - C_{10}$	0.600	0.182	0.218
$C_3 - C_4$	0.564	0.327	0.109
$C_3 - C_9$	0.564	0.164	0.273
$C_3 - C_{10}$	0.564	0.164	0.273
$C_4 - C_{11}$	0.564	0.182	0.255
$C_4 - C_{12}$	0.564	0.182	0.255
$C_9 - C_{11}$	0.527	0.018	0.455
$C_9 - C_{12}$	0.527	0.018	0.455
$C_{10} - C_{11}$	0.527	0.018	0.455
$C_{10} - C_{12}$	0.527	0.018	0.455
$C_2 - C_4$	0.509	0.236	0.255
$C_2 - C_{11}$	0.509	0.273	0.218
$C_2 - C_{12}$	0.509	0.273	0.218
$C_7 - C_{11}$	0.473	0.145	0.382
$C_7 - C_{12}$	0.473	0.145	0.382
$C_8 - C_{11}$	0.473	0.145	0.382
$C_8 - C_{12}$	0.473	0.145	0.382
$C_1 - C_7$	0.382	0.000	0.618
$C_1 - C_8$	0.382	0.000	0.618
$C_1 - C_9$	0.382	0.000	0.618
$C_1 - C_{10}$	0.382	0.000	0.618
$C_3 - C_7$	0.382	0.345	0.273

Таблица 3.28: Стойности за $\mu_{C,C'}$, $\nu_{C,C'}$, $\pi_{C,C'}$ – част 2 (Интуиционистки размита оценка)

Двойки критерии	$\mu_{C,C'}$	$\nu_{C,C'}$	$\pi_{C,C'}$
$C_3 - C_8$	0.382	0.345	0.273
$C_5 - C_7$	0.382	0.000	0.618
$C_5 - C_8$	0.382	0.000	0.618
$C_5 - C_9$	0.382	0.000	0.618
$C_5 - C_{10}$	0.382	0.000	0.618
$C_6 - C_7$	0.382	0.000	0.618
$C_6 - C_8$	0.382	0.000	0.618
$C_6 - C_9$	0.382	0.000	0.618
$C_6 - C_{10}$	0.382	0.000	0.618
$C_2 - C_7$	0.364	0.218	0.418
$C_2 - C_8$	0.364	0.218	0.418
$C_2 - C_9$	0.364	0.218	0.418
$C_2 - C_{10}$	0.364	0.218	0.418
$C_1 - C_{11}$	0.327	0.000	0.673
$C_1 - C_{12}$	0.327	0.000	0.673
$C_2 - C_3$	0.327	0.491	0.182
$C_5 - C_{11}$	0.327	0.000	0.673
$C_5 - C_{12}$	0.327	0.000	0.673
$C_6 - C_{11}$	0.327	0.000	0.673
$C_6 - C_{12}$	0.327	0.000	0.673
$C_4 - C_7$	0.309	0.382	0.309
$C_4 - C_8$	0.309	0.382	0.309
$C_1 - C_2$	0.291	0.000	0.709
$C_2 - C_5$	0.291	0.000	0.709
$C_2 - C_6$	0.291	0.000	0.709
$C_4 - C_9$	0.291	0.364	0.345
$C_4 - C_{10}$	0.291	0.364	0.345
$C_1 - C_4$	0.182	0.000	0.818
$C_4 - C_5$	0.182	0.000	0.818
$C_4 - C_6$	0.182	0.000	0.818
$C_1 - C_3$	0.145	0.000	0.855
$C_3 - C_5$	0.145	0.000	0.855
$C_3 - C_6$	0.145	0.000	0.855

Глава 4

Инспектиране на GPS мрежа

Съвременните космически технологии са доказали, че са подходящ инструмент за решаването на проблеми от реалния живот в области като прогнозата за времето, управление на природни и причинени от хората бедствия, картографията, геодезията и др. За това е необходимо разработването и оптимизирането на ефективни сателитни системи, които да предлагат надеждна навигационна услуга. Глобалната система за позициониране (GPS) има нужда от периодично проследяване, състоящо се от поставяне на GPS приемници последователно на определени точки. Целта на тази разработка е да предостави нови и потенциално мощни метаевристични алгоритми за разработване и подобряване на това космическо приложение. При проектирането на наблюдение на GPS мрежа, даден набор от наземни точки трябва да се обходи последователно (да се направи график). Цената на обхождането е сумата от времето, необходимо за преминаване от една точка до друга. Проблемът е да се търси най-добрият ред, в който се извършва това наблюдение. Минимизирането на разходите по графика е целта на тази работа. Резултатите от тази глава са публикувани в [64].

GPS играе важна роля в много съвременни навигационни системи и започва да играе все по-голяма роля в предоставянето на точни сигнали за времето за много индустрии. Приложенията включват геодезия, автономно управление на превозни средства, включително системата за интелигентни магистрали, морска навигация, контрол на въздушното движение. Услугата е винаги достъпна по всяко време и при всякакви метеорологични условия. Освен това системите за сателитна навигация оказват влияние в геологията, по-специално върху геодезическата работа

при бързо и ефективно определяне на позиции и промени в мрежите за позиции. Измерването изисква наблюдението да премине физически през целия наблюдаван терен, за да измери разстоянието между две съседни точки. Методите за геодезия са претърпели еволюционна промяна през последните няколко години с внедряването на сателитни навигационни системи. Най-известните космически системи са: американската глобална система за позициониране NAVSTAR, руската навигационна спътникова система GLObal (GLO NASS) и европейската система за сателитна навигация (GALILEO).

Глобалната навигационна система има принос в повечето начини за позициониране и направляване. Също така GPS вече има огромен принос в инспектирането, основно като технология за земемерно инспектиране [85]. GPS навигацията подпомага безопасното преминаване на плавателните съдове, самолетите и автомобилите, докато GPS инспектирането се свързва повече с традиционните функции на установяване на геодезен контрол, подпомагащ инженерните конструкции, кадастралните инспекции и картографирането. Доказано е, че единично GPS наблюдение дава някакво отклонение. За това се правят подходящи комбинации от измервания и обработки, за да се намали ефекта на отклонението в резултата. Изчислявайки разликата между едновременно събраните данни от два или повече приемника до няколко GPS спътника позволява да се елиминира или съществено да се намали отклонението. На практика GPS технологиите могат да бъдат приложени за широк клас дейности и са конкурентни на наземните технологии за инспектиране.

Ще покажем някои основни приложения на GPS инспектирането:

- Геодезен контрол: GPS вече е изместил другите методи за установяване, поддържане и гъстота на геодезната мрежа. На разстояние от няколко хиляди километра, методология използваща множество станции дава позициониране с точност до няколко дециметра. За сравнение GPS може да даде точност от 1 ppm даже за разстояния от няколко километра. Още повече, че GPS е много по-бърз [85].
- Научен контрол: Измерването на деформацията на кристали е в центъра за разбирането на земетресенията, движението на земните плочи, механизмът на изграждане на планините, действието на вулканите в близост до земната кора. В тези случаи се измерва промяната в позициите, преместването или огъването с времето.

За това е необходимо измерванията да бъдат с много голяма точност [85].

Свършена е доста научна работа с използването на GPS, като са използвани постоянните GPS приемници [92, 97], които не изискват наземни инспектори да придвижват приемниците между точките за да наблюдават сесиите. Въпреки всичко понякога се налага такъв вид измерване. Конкретни примери са дадени в [18, 72]. GPS спътниковата технология е по-свършена от традиционните и от теодолитните процедури. Тя е лесна за използване и не е необходимо потребителят да има предварителен опит в сателитното инспектиране.

Наблюдението се състои от съставяне на списък на сесиите, които трябва да бъдат наблюдавани. Цената на сесиите е цената на преместването на приемниците между точките. За малки мрежи е възможно да се намерят оптимални решения, но за големи мрежи трябва да се приемат близки до оптималните решения. Има нужда от специални алгоритми за точна и надеждна обработка на GPS сигналите. Приемник с неизвестна позиция на Земята трябва да открие и преобразува сигналите, получени от всички спътници. Тези измервания ще позволят на потребителя да изчисли триизмерна координатна позиция: местоположение на приемника.

Намирането на оптималното решение изисква голям изчислителен ресурс. Следователно са необходими нови методи за осигуряване на почти оптимални решения за големи мрежи в рамките на приемливо количество изчислителни усилия. Тези техники обикновено се основават на структурирани метаевристики [19, 35, 112].

4.1 Формулиране на задачата

GPS мрежата може да бъде определена като набор от станции (a_1, a_2, \dots, a_n) , които се координират чрез поставяне на приемници (X_1, X_2, \dots) върху тях, за да определят сесиите $(a_1a_2, a_1a_3, a_2a_3, \dots)$ между тях. Задачата е да се намери най-добрият ред, в който тези сесии могат да бъдат организирани, за да се даде най-добрият график. По този начин графикът може да се определи като редица от сесии, които да се спазват последователно. Решението е представено чрез линеен граф с претеглени ребра. Възлите представляват станциите, а ребрата представляват разходите за преместване. Целта на задачата е намаляване на

цената на решението, която е сумата от разходите (времето) за преминаване от една точка в друга, $C(V) = \sum C(a_i, a_j)$, където $a_i a_j$ е сесия в решение V . Например, ако броят на точките (станциите) е 4, възможно решение е $V = (a_1, a_3, a_2, a_4)$ и то може да бъде представено с линейния граф $a_1 \rightarrow a_3 \rightarrow a_2 \rightarrow a_4$. Цената за придвижване е както следва: $C(a_1, a_3), C(a_3, a_2), C(a_2, a_4)$. Така цената на решението е $C(V) = C(a_1, a_3) + C(a_3, a_2) + C(a_2, a_4)$.

На практика задачата е да се определи как всеки GPS приемник трябва да се премества между станциите, за да бъде извършено наблюдението по ефективен начин, като се вземат предвид някои важни фактори като време, разходи и т.н. Проблемът е да се търси най-добрият ред по отношение на времето, през което тези сесии могат да бъдат наблюдавани, за да се даде най-евтиният график или да се сведе до минимум $C(V)$. Първоначалните данни са матрица на разходите, която представлява разходите за преместване на приемник от една точка в друга. Цената може да бъде оценена чисто по времето или чисто по разстоянието; за повече подробности вижте Dage [19].

Тази глава разглежда проблема с определянето на оптималния график на сесията, според някои критерии за разходите. Ако цената на придвижване между две точки е независима от посоката, тогава ще се получи симетрична матрица на разходите, но реалистичният случай е несиметричен и задачата става по-трудна. Несиметричната матрица на разходите е по-реалистична, тъй като движенията между точките обикновено изискват комбинация от шофиране и ходене и следователно пътуванията нагоре обикновено изискват повече време от пътуванията надолу.

Ако S представлява броя на сесиите, тогава броят на възможните графици на сесиите се дава от $S!$, Коего очевидно е много голям брой за големи стойности на S . Следователно е невъзможно да се намери оптимално решение за големи мрежи в разумен срок. В този случай се прилагат приближени методи, които намират бързо решения, близки до оптималното.

Целта на настоящата глава е да покаже как АСО може да се използва за подреждане на наблюдаваните GPS сесии, като се имат предвид разходите за преместване на приемниците между точките в мрежата. Графикът на сесиите се определя като последователност от сесии, които трябва да се наблюдават. АСО е техника за определяне на почти оптимални решения с разумно количество изчислителни ресурси. Използва

се за решаване на големи задачи, които не могат да бъдат решени оптимално за разумно време.

4.2 АСО алгоритъм за инспектиране на GPS мрежа

В предложения АСО-GPS алгоритъм се извършва инициализация, по време на която мравките се позиционират на различни възли (сесии) и първоначален феромон, разпределен еднакво по пътища, свързващи тези сесии. Мравките актуализират нивото на феромон, докато изграждат своите графици, като добавят нови сесии към текущия частичен график. На всяка времева стъпка мравките изчисляват набор от възможни ходове и избират най-добрия според някои вероятностни правила въз основа на евристичната информация и нивото на феромона. Колкото по-висока е стойността на феромона и евристичната информация, толкова по-изгодно е да бъде избран този ход. Избраният възел се поставя в списък, свързан с мравката, за да се предотврати повторният му избор. Евристичната информация представлява по-близките сесии около текущата сесия, докато „паметта“ на ниво феромон на всеки път представлява използваемостта на този път в миналото за намиране на добри графици. В края на всяка итерация списъкът с върхове за всяка мравка ще бъде пълен и полученият най-евтин график се изчислява и запамятава. За следващата итерация, списъците с върхове ще бъдат изпразнени, готови за употреба, а нивото на феромона ще бъде актуализирано. Този процес се повтаря, докато се достигне зададеният брой итерации (критерии за край).

Важно е при прилагането на предложената техника да се осигурят подходящи и внимателно подбрани структурни и контролни компоненти според размера и вида на приложената GPS мрежа. Структурните елементи определят процедурата, при която се моделира задачата за обхождане и наблюдение на GPS мрежа, за да се впише в рамката на АСО, както следва:

- *Текуща ценова матрица:* Тази матрица $C[i, j]$ показва цената за преминаване от края на сесия i до началото на наблюдението на сесия j . Размерът на тази матрица зависи от броя на сесиите за наблюдение на цялата мрежа [19].

- *Начален феромон:* Началното количество феромон τ_0 , което е равно на интензивността на пътя във време 0, трябва да бъде някаква малка стойност ($0 \leq \tau_0 < 1$). По време на изпълнението на алгоритъма, този начален феромон се обновява на всяка итерация и накрая получаваме матрица на феромона $\tau[i, j]$. Тази матрица дава важна информация за по-бързите и за по-бавните пътища, която насочва наблюдаващите мрежата да избират коя да е следващата сесия за наблюдение.
- *Списък с обходените сесии:* Към всяка мравка е свързан списък с обходените сесии, който предотвратява тяхното повторно обхождане. Когато се избира следваща сесия за наблюдение, се проверява дали тя е в списъка. След края на всяка итерация, списъкът с обходените сесии се изпразва и мравките отново започват да строят своя път, започвайки от случайна сесия за начална.

Контролните параметри ръководят работата на ACO-GPS алгоритъма. Те основно се отнасят до използването на феромона.

- *Параметър на интензитета ($\alpha \geq 0$):* Този параметър контролира относителното тегло на интензитета на феромона τ_{ij} , по време на процеса на подбор на следващите сесии, които да бъдат наблюдавани. Ако $\alpha = 0$ (мравките не общуват), най-близките сесии е по-вероятно да бъдат избрани от мравките. От друга страна, високите стойности на α (мравките комуникират) означава, че следата е много важна и следователно настоящите мравки са склонни да избират пътища, избрани преди това от други мравки.
- *Параметърът за контрол на видимостта ($\beta \geq 0$):* Функцията на параметъра β е да постави повече или по-малко акцент върху разстоянието спрямо феромона. Този параметър има положително влияние върху качеството на разписанието и това може да се види при коригиране на относителната важност на видимостта ν_{ij} при оценка на цената на път в разписанието. Ако $\beta = 0$, функционира само интензитетът на феромона и това ще доведе до бърз избор на разписания, които може да не са оптимални.
- *Параметърът за контрол на изпарението ($0 \leq \rho < 1$):* Този параметър се използва за регулиране на количеството на феромона, поставен от мравката при конструирането на нейното решение.

- *Критериите за спиране:* Бяха приети няколко критерия за прекратяване и най-простият е да се прекрати процесът след предварително дефиниран брой итерации. Всяка итерация на АСО-GPS процеса изисква две стъпки, за създаване на разписание; изграждане на разписание и актуализиране на нивото на феромона. Друг критерий за спиране е да се прекрати търсенето, когато не може да се наблюдава допълнително подобрене за фиксиран брой итерации.

4.2.1 Прилагане на АСО алгоритъма

В нашата реализация използваме MAX-MIN Ant System (MMAS) [116] и Ant Colony System (ACS) [26], които са едни от най-добрите варианти на метода на мравките. В MMAS основното е използването на фиксирана горна граница τ_{max} и долна граница τ_{min} на количеството на феромона. По този начин частично се предотвратява натрупването на голямо количество феромон от част от възможните придвижвания и повторението на едни и същи решения е частично предотвратено. Основните характеристики на MMAS са:

- Концентрация на търсенето в близост до най-доброто намерено решение досега. Това може да се постигне, като се позволи само на една мравка да добави феромон след всяка итерация, най-добрата.
- Задълбочено изследване на най-доброто решение. След първата итерация следите на феромоните се инициализират отново с τ_{max} . В следващата итерация само движенията, които принадлежат на най-доброто решение, получават феромон, докато други стойности на феромона само се изпаряват.

Целта на използването само на едно решение е елементите на решението, които често се срещат в най-добрите намерени решения, да получат по-голямо подсилване. За да се избегне стагнация на търсенето, диапазонът на възможната стойност на феромона при всяко движение е ограничен до интервал $[\tau_{min}, \tau_{max}]$. τ_{max} е асимптотичен максимум от τ_{ij} и $\tau_{max} = 1/(1 - \rho)C(V^*)$, докато $\tau_{min} = 0,087\tau_{max}$. Където V^* е оптималното решение, а V_{best} е итерационното най-добро решение и $i, j = 1, \dots, n$. V^* е неизвестно, затова използваме V_{best} вместо V^* .

Основните характеристики на ACS са:

- Концентрация на търсенето в близост до най-доброто намерено решение досега.
- Феромонът на по-лошите решения намалява и частично се предотвратява повторението им.

Феромонът, съответстващ на конструираните от мравки решения, се актуализира с помощта на локални и глобални правила за актуализиране. Докато мравките изграждат своето решение в същото време, те локално актуализират нивото на феромона на посетените пътища. Целта е да се намали нивото на феромона на използваните пътища, докато това при неизползваните остава непроменено. По този начин частично предотвратяваме натрупването на голямо количество феромони от една страна, а нивото на феромона никога не става по-малко от τ_0 от друга. Това е вид разнообразяване на търсенето в пространството за търсене.

Когато всички мравки завършат своите решения, нивото на феромона се актуализира чрез прилагане на глобалното правило за актуализиране. Увеличава се само феромона, съответстващ на най-доброто намерено решение. Глобалното правило за актуализиране има за цел да осигури по-голямо количество феромон по пътя на най-доброто решение. Това е един вид засилване на търсенето около най-доброто намерено решение. И в двете реализации използваме евристична информация, равна на едно върху цената на сесията.

4.2.2 Експериментални Резултати

Едно от основните неща при прилагането на АСО алгоритмите е графът на задачата. В нашата реализация възлите на графа съответстват на станциите, а дъгите съответстват на цената на сесиите. Мравките отлагат феромона върху дъгите.

За да постигнем добри резултати, трябва внимателно да изберем настройките на АСО параметрите и да определим структурните елементи на алгоритмите. При всяка итерация мравките започват да конструират своето решение от произволен възел на графа, следователно броят на мравките може да бъде много по-малък от броя на възлите (станциите). По този начин намаляваме необходимите изчислителни ресурси като време и памет. Експериментално установихме, че 10 мравки са достатъчни.

Използвайки повече мравки, увеличаваме времето за изчисление, без да подобряваме постигнатите резултати, когато броят на итерациите е същият.

Контролните параметри управляват работата на алгоритмите и се отнасят главно до информацията за феромона и вероятността за преход. Параметърът α е параметърът на интензитета на феромона, а β е евристичен контролен параметър. Когато стойността на α е висока, информацията за феромона е по-важна, при избора на мравката за следващ възел, към който да се придвижи. Когато β е високо, евристична информация е по-важна. След няколко теста установихме, че най-добрите стойности за параметри за контрол на вероятността за преход за задачата за инспектиране на GPS мрежа са $\alpha = 1$ и $\beta = 2$.

Параметърът ρ се използва в правилата за актуализиране на феромона за разнообразяване на търсенето чрез регулиране на влиянието на стария феромон. $\rho \in (0, 1)$, когато ρ е близо до 0 само новият добавен феромон е важен в следващата итерация. Когато ρ е близо до 1, влиянието на стария феромон (опит от предишни итерации) е голямо. Установихме, че нашите алгоритми постигат най-добри резултати, когато $\rho = 0.2$.

Началната стойност на феромона τ_0 не оказва влияние върху работата на алгоритъма. Важното е τ_0 да бъде по-малко от $\Delta\tau_{ij}$. Затова избираме началната стойност на феромона да бъде $\tau_0 = 0.005$.

За реалистично сравнение на производителността на ACS и MMAS алгоритмите за инспектиране на GPS мрежа, ние използваме чисти АСО алгоритми, без да включваме никакви процедури за локално търсене. Броят на итерациите е равен на броя на станциите.

В този раздел анализираме експерименталните резултати, получени с помощта на ACS и MMAS алгоритми, описани в предишния раздел. Като тестови задачи ние използваме реални данни от GPS мрежите на Малта и Сейшелите. GPS мрежата на Малта се състои от 38 сесии, а GPS мрежата на Сейшелите състои от 71 сесии. Използваме 6 по-големи тестови примера от <http://www.informatik.uni-heidelberg.de/groups/comopt/software/TSLIB95/ATSP.html>. Тестовите примери варират от 100 до 443 сесии.

За всеки експеримент резултатите се получават чрез извършване на 30 независими пускания на алгоритъма, след което се усредняват получените стойности на целевата функция, за да се осигури статистическа достоверност на наблюдаваната разлика. Анализът на данните с помощта на ANOVA/Kruskal-Wallis тест е използван за получаване на статис-

тическа увереност на нивото 95% от резултатите.

Таблица 4.1: Сравнение на MMAS и ACS.

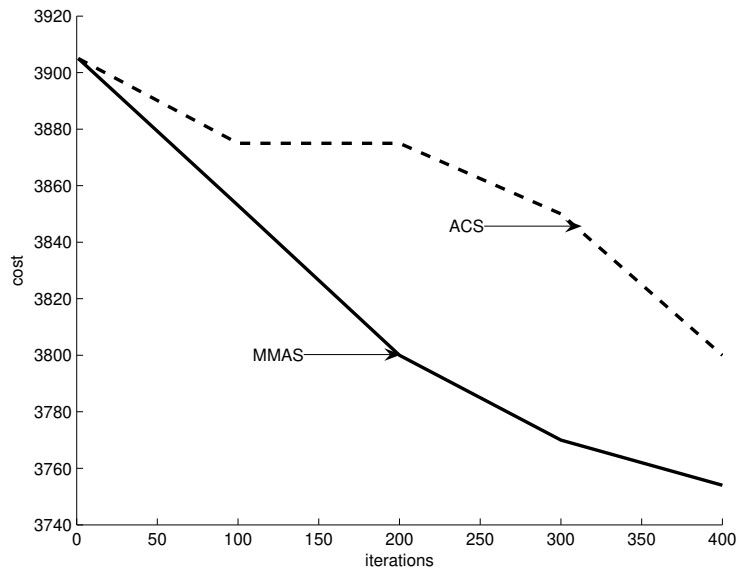
Tests	sessions	ACS	ACS-best	MMAS	MMAS-best
Malta	38	902	900	902	900
Seychelles	71	920	853	912	852
rro124	100	41584	40423	41276	39723
ftv170	170	3395	3362	3356	3115
rgb323	323	1688	1680	1665	1625
rgb358	358	1734	1711	1675	1646
rgb403	403	3499	3490	3428	3362
rgb443	443	3806	3795	3754	3705

В таблица 4.1 показваме постигнатата цена за всеки тестов пример. За сравнение използваме едни и същи настройки на параметрите и за двата алгоритъма. Показани са средната и минималната цена, достигнати от всеки алгоритъм. С удебелен шрифт е най-добрата постигнатата цена.

Анализирайки резултатите, стигаме до заключението, че MMAS алгоритъмът превъзхожда ACS алгоритъма. Минимални разходи се получават от MMAS за повечето тестове. За Malta и ftv170 няма относителна разлика между двата алгоритъма.

Нека сравним нашите резултати с резултатите в [111]. Като тестове, те използват само малки примери за Малта и Сейшелите и за да подобрят производителността на своя АСО алгоритъм, те го комбинират с процедура за локално търсене. Те съобщават само най-добрите резултати, които са открили, както следва: 895 за Малта и 853 за Сейшелите с 200 итерации. По този начин можем да заключим, че нашият алгоритъм е по-добър, защото постигаме сходна най-добра цена за същите тестови примери, но без локално търсене и с по-малък брой итерации. Освен това наблюдаваме по-голяма разлика между алгоритмите при по-големи тестови задачи.

На фиг. 4.1 изобразяваме сравнението на цената на решението за тес-



Фигура 4.1: Сравнение между ACS и MMAS алгоритми за rgb443 тестова задача.

тов пример rbg443. Наблюдаваме подобряването на резултатите по време на процесите на търсене и можем да заключим, че MMAS алгоритъмът превъзхожда ACS за задачата за инспектиране на GPS мрежа.

4.3 Хибриден АСО алгоритъм за задачата за инспектиране на GPS мрежа

Много често АСО алгоритмите се комбинират с подходяща процедура за локално търсене (ЛТ), за да се подобри ефективността на алгоритъма и качеството на постигнатите решения.

4.3.1 Стратегии за локално търсене

Основната концепция на ЛТ е търсене в локалното съседство на текущото решение. Методът за локално търсене разбърква дадено решение за генериране на различни съседни решения с помощта на механизъм за генериране на преместване. ЛТ се опитва да подобри първоначалното решение чрез серия промени за локално подобряване. Генерирането на преместване е преход от решение S към решение $S' \in V(S)$ в една стъпка. Тези решения са избрани и приети по някои предварително определени критерии. Върнато решение S' може да не е оптимално, но е най-доброто решение в близко съседство $V(S)$. Локално оптимално решение е решение с минимална възможна стойност на цената измежду съседните решения. Познаването на пространството за решение е основният ключ към по-ефективни стратегии за търсене. Тези стратегии са предназначени да използват предварителни познания за задачата и за преодоляване на сложността на пълното изчерпване се организира търсене в алтернативите, предлагани от конкретно представяне на решението. Основната цел на прилагането на ЛТ е ускоряване и подобряване на решенията, създадени от основния алгоритъм, АСО. В този раздел, различни локални структури за търсене $L(k, l)$, които отговарят на изискванията на обхождането на GPS мрежа, са разработени и приложени, за по-ефективно търсене, където k е броят на генерираните съседни решения и l е номерът на използвания метод за разбъркване на решенията. Задачата е представена с граф.

Предложени са следните ЛТ процедури:

4.3. Хибриден АСО алгоритъм за задачата за инспектиране на GPS мрежа 77

1. Последователна размяна на възли: **for** $i = 1$ **to** $n - 1$; **for** $j = i + 1$ **to** n ; разменя a_i и a_j , [112];
2. Размяна на случайни възли: два възела се избират на случаен принцип и се разменят;
3. Изтриване на случайно ребро: нека текущото решение е $(a_1, a_2, \dots, a_i, a_{i+1}, \dots, a_n)$. Реброто $(i, i + 1)$ се избира на случаен принцип и се изтрива. Новото решение е $(a_{i+1}, \dots, a_n, a_1, \dots, a_i)$;
4. Жадно изтриване на ребро: Най-дългото (най-скъпо) ребро се изтрива. Новото решение е конструирано както в предходния случай;
5. Изтриване на произволни 2 ребра: Нека текущото решение е $(a_1, a_2, \dots, a_i, a_{i+1}, \dots, a_j, a_{j+1}, \dots, a_n)$. Ребрата $(i, i + 1)$ и $(j, j + 1)$ се избират на случаен принцип и се изтриват. Новите решения са $(a_{i+1}, \dots, a_j, a_1, \dots, a_i, a_{j+1}, \dots, a_n)$, $(a_{j+1}, \dots, a_n, a_{i+1}, \dots, a_j, a_1, \dots, a_i)$, $(a_1, \dots, a_i, a_{j+1}, \dots, a_n, a_{i+1}, \dots, a_j)$;
6. Жадно изтриване на 2 ребра: Двете най-дълги ребра се изтриват. Новите решения се построяват както в горния случай.

Когато мравките конструират своите решения, ние подобряваме текущото най-добро решение, чрез прилагане на процедура за ЛТ. Актуализацията на феромона е по отношение на подобрените след ЛТ решения. Можем да изтрием повече от две ребра, но след няколко теста установихме, че времето за работа нараства пропорционално на броя на изтритите ребра без подобряване на решението сравнено с изтриване на две ребра.

4.3.2 Експериментални резултати

В този раздел са сравнени резултатите от АСО (вариант ММАС), комбиниран с различни процедури за локално търсене. За тестовете се използват същите примери като в предишния раздел. За да се поддържа ниско време за работа, съседният набор се състои от толкова решения, колкото е броят на сесиите (n). Броят на итерациите е равен на размера на тестовия пример.

В таблица 4.2 показваме постигнатите цени за всички тестови примери. С удебелен шрифт са минималните постигнати цени. След анализ на резултатите заключаваме, че процедурата за локално търсене $L(n, 5)$

Таблица 4.2: ММАС алгоритъм с процедура за локално търсене

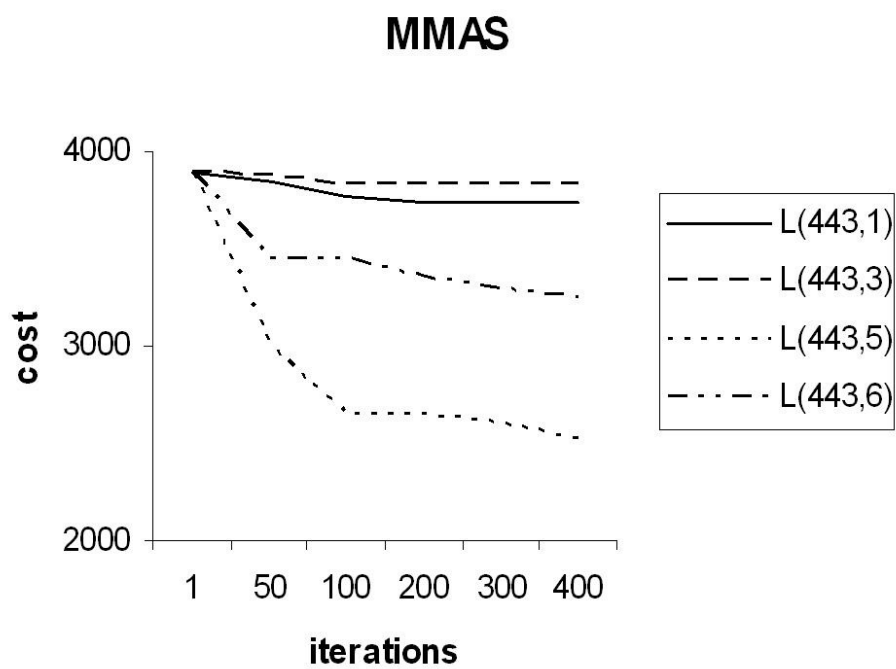
Tests	LS sessions	L(n,1)	L(n,2)	L(n,3)	L(n,4)	L(n,5)	L(n,6)
Malta	38	902	902	895	895	872	902
Seychelles	71	920	920	915	915	851	893
rro124	100	41276	41276	41004	41004	39971	41281
ftv170	170	3356	3356	3229	3290	3266	3320
rgb323	323	1661	1661	1691	1666	1378	1423
rgb358	358	1680	1680	1689	1702	1477	1549
rgb403	403	3428	3428	3401	3401	2408	2710
rgb443	443	3751	3751	3838	3838	2631	3349

превъзхожда другите. Сравнявайки ММАС, комбиниран с различни ЛТ процедури, най-добрите резултати се получават от $L(n, 5)$ за повечето тестови примери с изключение на ftv170. За този тестови пример няма относителна разлика между $L(n, 3)$ и $L(n, 5)$. Забелязваме, че няма относителна разлика между постигнатите резултати с $L(n, 1)$, $L(n, 2)$ и постигнатите резултати без ЛТ. Заключаваме, че $L(n, 1)$ и $L(n, 2)$ не подобряват постигнатите от ММАС решения.

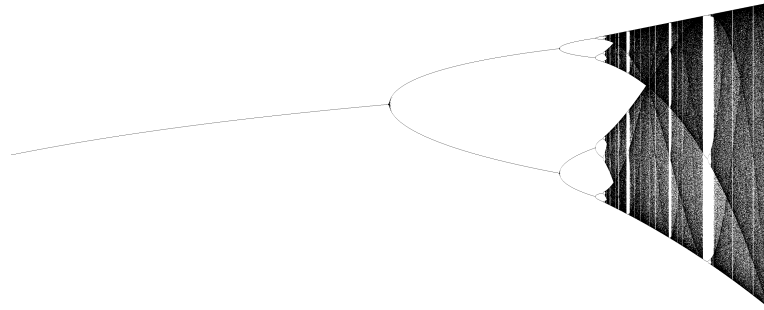
На фигура 4.2 изобразяваме сравнението на цената на решението за тестови пример rgb443, прилагайки $L(443, 1)$, $L(443, 3)$, $L(443, 5)$ и $L(443, 6)$. Наблюдаваме, че $L(443, 5)$ и $L(443, 6)$ постигат по-добри резултати от другите. Най-добрата цена се постига с $L(443, 5)$. Резултатите, постигнати от $L(443, 1)$, са статистически подобни на резултатите, постигнати без LS, така че на фигура 4.2 наблюдаваме подобряването на резултатите при използване на ЛТ. Процедурата за локално търсене $L(n, 5)$ се представя по-добре от $L(n, 6)$, тъй като случайното изтриване води до голямо разнообразие.

4.4 АСО с промени в средата

Логистичната карта е квадратно динамично уравнение, предложено през 1938 г. като демографски модел [117]. Това е доста прост квадратен по-



Фигура 4.2: MMAS с процедура за локално търсене за rbg443 тестова задача



Фигура 4.3: Поведението на логистичната карта за различни стойности на r (по оста x , за $r = 2$ до 4 на фигурата).

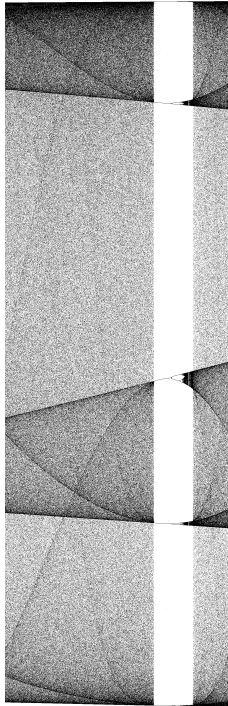
ЛИНОМ:

$$x_{n+1} = rx_n(1 - x_n), \quad n > 0, \quad (4.1)$$

където x_n представлява размера на популацията в момент n и r е константа, наречена коефициент на нарастване. Като се има предвид, че $x_0 \in [0, 1]$ и $r \in [0, 4]$, това динамично уравнение може или да се сближи, или да бъде хаотично. В първия случай, при всяко $x_0 \in [0, 1]$, x_n кло-ни към така наречената „област на привличане“. Във втория случай x_n никога не се сближава, но по-скоро може да приеме случайна стойност, всяка възможна стойност в диапазона $[0, 1]$. Fig 4.3 показва поведението на логистичната карта за различни стойности на r в диапазона $[2, 4]$ (поведението му е линейно в диапазона $[0, 2]$).

По оста x разглеждаме дискретно подмножество от 3000 равно отдалечени стойности за r между 2 и 4; по оста y , за всяка разглеждана стойност за r , ние отчитаме съответната област на привличане. За да идентифицираме областите на привличане, вземаме 1500 равно отдалечени точки в интервала $[0, 1]$ и прилагаме уравнение (4.1) 1000 пъти за всяко от тях. За малки стойности на r , логистичната картата винаги се сближава до една единствена точка, т.е. областта на привличане се състои от една точка само. Първата бифуркация се появява, когато $r = 3$, където областта на привличане се състои от 2 точки; след това има друга бифуркация, когато $r = 1 + \sqrt{6}$, така че областта на привличане се състои от 4 точки. За по-големи стойности за r , логистичната карта изпитва други бифуркации и може да бъде хаотична за някои под интер-

вали от r . Въпреки това, в тези хаотични региони, все още е възможно да се идентифицират регулярни области на привличане. Например, на Фиг. 4.4, същата графика, изобразена на Фиг. 4.3 е увеличена в областта $r = [3.901, 3.908]$, където това явление е ясно показано. Регулярните региони, които могат да се видят на фиг. 4.3, все още съдържат бифуркации. Освен това можем да забележим, че цялата графика на фиг. 4.3 се появява отново в нашата увеличена област. Други области на привличане могат да бъдат идентифицирани чрез разглеждане на по-тесни под интервали от r , както и други копия на цялата графика. Фиг. 4.3 всъщност е фрактал, поради самоподобие то му.



Фигура 4.4: Поведението на логистичната карта за стойности на r в интервала $[3.901, 3.908]$. Този регион на логистичната карта като цяло е хаотичен, но регулярните области на привличане (с типичните бифуркации) все още могат да бъдат идентифицирани. .

Ние симулираме регулярни и хаотични промени на средата в АСО, като въвеждаме логистичната карта в уравнение (4.1), което се използва в АСО за актуализиране на феромоните. В хипотезата целевата функция на разглежданият проблем е положителна и по-голяма от 1, изразът $1/f(X)$ в уравнение (4.2) винаги има стойности вариращи между 0 и 1. Следователно той може да заеме мястото на x_0 в логистичната карта, така че стойността x_1 може да се изчисли за дадена стойност от $r \in [0, 4]$. Следователно уравнението за актуализиране на феромона става:

$$\tau_{uv} = \tau_{uv} + r \cdot \frac{1}{f(X)} \cdot \left(1 - \frac{1}{f(X)}\right). \quad (4.2)$$

С тази проста промяна в правилото за актуализиране на феромона, ние изкуствено нарушаваме околната среда на мравките, което иначе би зависело само от стойностите на целевата функция, съответстваща на това решение. Различните стойности за r могат да произведат различни промени в средата в зависимост от поведението на логистичната карта. За стойности на r , за които логистичната карта се сближава, нивата на феромоните, добавени към τ_{uv} , са постоянни, намалявайки по този начин ефектите на висококачествените решения, които биха могли да подведат мравките към локален оптимум. За стойности на r , за които вместо това логистичната карта се държи хаотично, средата е доминираща върху избора на мравките, тъй като актуализацията на феромоните зависи най-вече от симулираната среда, а не от действителната стойност на целевата функция. Ние наричаме АСО с промени в средата *environment* АСО (*eACO*). В тази работа представяме някои предварителни експерименти, при които *eACO* се използва за решаване на задачата за инспектиране на GSP мрежа.

4.4.1 Числени експерименти

Като тестови случаи разглеждаме данни от две реални мрежи: Малта [111], съставена от 38 сесии и Сейшелите [111], съставена от 71 сесии. Разглеждаме и по-големи примери предназначени за тестване на ATSP, които са свободно достъпни в Интернет.¹

¹<http://www.informatik.uni-heidelberg.de/groups/comopt/software/TSLIB95/ATSP.html>

<i>instances</i>	$ V $	АСО	<i>e</i> АСО			
			$r = 1$	$r = 2$	$r = 3$	$r = 4$
Malta	38	899.50	897.00	897.00	897.00	900.33
Seyshels	71	922.06	905.73	905.60	887.33	906.73
kro124p	100	40910.60	40725.40	40799.30	40753.00	40803.76
ftv170	171	3341.93	3314.20	3313.76	3319.83	3338.53
rgb323	323	1665.90	1654.40	1648.66	1649.43	1649.53
rgb358	358	1692.66	1679.63	1689.00	1682.80	1685.95
rgb403	403	3428.56	3413.63	3392.23	3393.76	3386.10
rgb443	443	3765.80	3749.93	3742.86	3742.43	3754.50

Таблица 4.3: Сравнение между АСО и *e*АСО за задачата за инспектиране на GSP.

В нашата *e*АСО реализация актуализацията на феромона се извършва чрез прилагане на уравнение (4.2). И накрая, стойностите за параметрите на вероятността за преход α и β са фиксирани съответно на 1 и 2. Тези стойности са идентифицирани като оптимални за АСО при решаването на задачата за инспектиране на GSP мрежа [38]. В следващите експерименти ще се съсредоточим върху качеството на намерените решения, а не на представянето на алгоритмите. Всъщност увеличението на сложността за използване на уравнение (4.2), може да се пренебрегне при разглеждане на цялостната сложност на алгоритмите.

Таблица 4.3 показва някои изчислителни експерименти за различни стойности на r . Средните стойности над 30 пускания на алгоритъма са отчетени в таблицата. *e*АСО може да идентифицира по-добре качествени решения във всички експерименти и за почти всички използвани стойности за r . Това показва, че всъщност променлива среда за мравките, вместо постоянна, дава предимства на търсенето. За стойности на r , равни на 1, 2 и 3, логистичната карта доближава до една уникална стойност; докато за $r = 4$ е хаотично. Следователно изглежда, че най-добри резултати могат да бъдат постигнати, когато околната среда има тенденция да хомогенизира феромоновите следи. Забележете обаче, че само един експеримент с $r = 4$, когато средата е хаотична, не е в състояние да осигури по-добро решение (срещу този осигурено от стандартния АСО).

4.5 ICrA на приложението на АСО за задачата за инспектиране на GPS мрежа

В тази секция е приложен InterCriteria Analysis (ICrA) за анализ на алгоритъма на мравки (АСО), използван за намиране на близки до оптималните решения за проблема за инспектиране на GPS мрежа.

4.5.1 ICrA Резултати

В таблица 6.14 са представени резултатите от всеки 30 пускания за осемте GPS тестови задачи. Средните резултати от първите 5 пускания, първите 10 пускания и т.н. и накрая от всичките 30 пускания са изчислени и представени в таблица 4.5. За тези резултати е приложен интеркритериален анализ.

Получените степени на "съгласие" ($\mu_{C,C'}$) са както следва:

ρ	GPS_1	GPS_2	GPS_3	GPS_4	GPS_5	GPS_6	GPS_7	GPS_8
GPS_1	1	0.60	0.27	0.67	0.73	0.67	0.33	0.87
GPS_2	0.60	1	0.27	0.80	0.73	0.53	0.47	0.73
GPS_3	0.27	0.27	1	0.07	0	0.20	0.40	0.13
GPS_4	0.67	0.80	0.07	1	0.93	0.73	0.53	0.80
GPS_5	0.73	0.73	0	0.93	1	0.80	0.60	0.87
GPS_6	0.67	0.53	0.20	0.73	0.80	1	0.67	0.80
GPS_7	0.33	0.47	0.40	0.53	0.60	0.67	1	0.47
GPS_8	0.87	0.73	0.13	0.80	0.87	0.80	0.47	1

(4.3)

Получените степени на "несъгласие" ($\nu_{C,C'}$) са както следва:

4.5. ICRA на приложението на АСО за задачата за инспектиране на GPS мрежа85

Таблица 4.4: Резултати от 30 пускания

	GPS_1	GPS_2	GPS_3	GPS_4	GPS_5	GPS_6	GPS_7	GPS_8
run1	905	897	41773	3333	1670	1711	3434	3786
run2	900	914	40486	3115	1636	1648	3412	3791
run3	900	934	41005	3279	1663	1668	3454	3731
run4	895	910	41311	3230	1652	1657	3416	3754
run5	895	927	42107	3267	1660	1684	3384	3729
run6	895	908	40089	3422	1641	1697	3397	3723
run7	900	963	41384	3299	1665	1699	3486	3723
run8	895	915	40894	3448	1679	1702	3454	3779
run9	900	865	40471	3314	1681	1688	3396	3769
run10	895	923	41004	3326	1661	1681	3474	3772
run11	900	911	40776	3450	1692	1695	3447	3776
run12	895	918	41299	3408	1615	1708	3484	3784
run13	900	959	40541	3265	1691	1689	3435	3720
run14	900	930	41046	3391	1661	1715	3382	3794
run15	900	963	40693	3358	1692	1719	3442	3750
run16	895	934	42219	3405	1679	1664	3364	3733
run17	900	918	40947	3290	1646	1690	3422	3717
run18	900	914	40602	3398	1654	1673	3395	3803
run19	895	930	40827	3630	1677	1684	3431	3827
run20	905	963	39244	3263	1684	1703	3428	3749
run21	920	916	41137	3341	1694	1726	3464	3701
run22	900	914	41773	3370	1659	1693	3430	3778
run23	900	916	39477	3291	1646	1675	3422	3754
run24	900	963	39096	3338	1678	1735	3446	3758
run25	895	913	40604	3409	1698	1652	3437	3819
run26	900	964	41147	3302	1642	1699	3393	3791
run27	900	865	40984	3286	1641	1692	3407	3785
run28	905	897	41311	3333	1657	1705	3472	3803
run29	895	928	41747	3335	1674	1705	3405	3796
run30	900	890	41324	3362	1689	1723	3444	3779

Таблица 4.5: Индексна матрица за ICrA

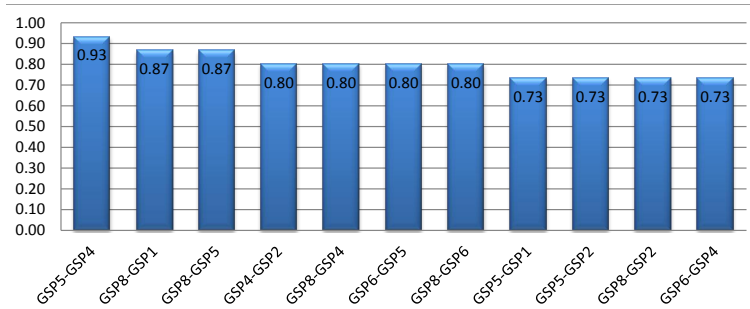
	C_1	C_2	C_3	C_4	C_5	C_6
GPS_1	899.00	898.00	898.33	898.50	899.40	899.50
GPS_2	916.40	915.60	922.47	924.80	924.72	922.07
GPS_3	41336.40	41052.40	40991.93	40935.90	40832.20	40910.60
GPS_4	3244.80	3303.30	3327.00	3344.55	3345.60	3341.93
GPS_5	1656.20	1660.80	1663.93	1664.95	1666.96	1665.90
GPS_6	1673.60	1683.50	1690.73	1688.75	1690.24	1692.67
GPS_7	3420.00	3430.70	3433.13	3426.85	3429.44	3428.57
GPS_8	3758.20	3755.70	3758.73	3760.50	3760.80	3765.80

σ	GPS_1	GPS_2	GPS_3	GPS_4	GPS_5	GPS_6	GPS_7	GPS_8
GPS_1	0	0.40	0.73	0.33	0.27	0.33	0.67	0.13
GPS_2	0.40	0	0.73	0.20	0.27	0.47	0.53	0.27
GPS_3	0.73	0.73	0	0.93	1	0.80	0.60	0.87
GPS_4	0.33	0.20	0.93	0	0.07	0.27	0.47	0.20
GPS_5	0.27	0.27	1	0.07	0	0.20	0.40	0.13
GPS_6	0.33	0.47	0.80	0.27	0.20	0	0.33	0.20
GPS_7	0.67	0.53	0.60	0.47	0.40	0.33	0	0.53
GPS_8	0.13	0.27	0.87	0.20	0.13	0.20	0.53	0

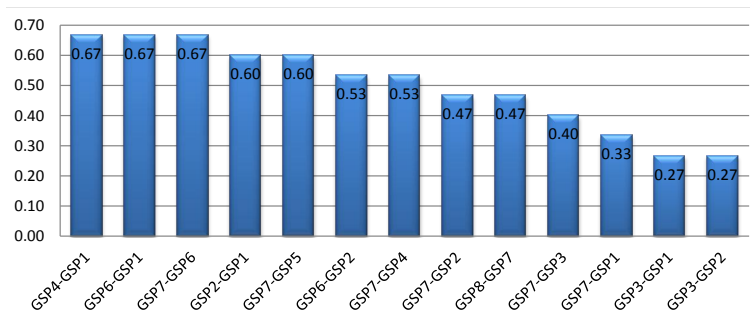
(4.4)

По отношение на InterCriteria анализ, ние забелязваме, че ACO алгоритъмът работи по подобен начин за GPS_2 , GPS_4 , GPS_5 и GPS_8 . Те са GPS мрежи с различен брой сесии, но може би тези мрежи имат сходна структура, поради което степента на "съгласие" е висока. За други мрежи от тестовите можем да заключим, че те имат много различна структура. GPS, които са в положителен консонанс, са показани на фиг. 6.1. За тези GPS ACO алгоритъмът получава сходни резултати, т.е. изпълнението на алгоритъма е идентично при решаване на поставените задачи. На фиг. 4.6 са показани тези двойки GPS тестове, за които ACO алгоритъмът няма идентична производителност, т.е. тези GPS тестове са в дисонанс. Накрая на фиг. 4.7 са представени двойки GPS, които са в отрицателно съгласие. Определението за положително/отрицателно съгласие и дисонанс е според [80].

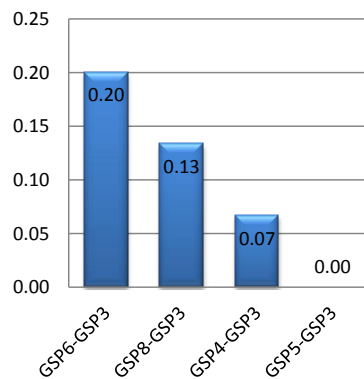
4.5. ICRA на приложението на ACO за задачата за инспектиране на GPS мрежа87



Фигура 4.5: GPS връзка в положителна скала за консонанс



Фигура 4.6: GPS отношение в скалата на дисонанса



Фигура 4.7: GPS отношение в отрицателна скала на консонанс

4.6 Заключение

В тази глава е разгледана задачата за инспектиране на GPS мрежа. Предложени са два варианта на АСО алгоритъма за решаване на задачата, MMAS и ACS. Алгоритмите са тествани с тестови примери съдържащи от 38 до 443 сесии. Докладвано е сравнение на производителността на двата АСО алгоритъма, приложени за различни GPS мрежи. MMAS алгоритъма превъзхожда ACS алгоритъма. Получените резултати са обнадеждаващи и се вижда способността на разработените техники бързо да генерират висококачествени решения за инспектиране на GPS мрежи. Проблемът е важен, защото възниква при безжични комуникации като GPS и мобилен телефон и може да подобри услугите в мрежите. Следователно задачата е актуална и има икономическо значение. Върху получените резултати е приложен Интеркритериален анализ. Чрез ICrA можем да проверим коректността на изпълнението на алгоритъма. Показва дали конструкцията на различните GPS мрежи са сходни.

Глава 5

Построяване на безжична сензорна мрежа

Пространствено разпределени сензори, които комуникират безжично, образуват безжична сензорна мрежа (WSN). Тази мрежа следва физическите условия или условията на заобикалящата ни среда. Специален, мощен възел, наречен високо енергиен комуникационен възел (HECN) събира данните от всички сензори и ги изпраща до централния компютър, където се обработват. Трябва да се използват възможно по-малко сензори и енергия, при условие, че наблюдавания терен има пълно покритие. Задачата е многоцелева. Предложен е алгоритъм, на основата на метода на мравките, за решаване на задачата като многоцелева и два АСО алгоритъма за решаването ѝ като едноцелева. Резултатите от тази глава са публикувани в [51, 54, 58, 108, 64].

5.1 Въведение

Развитието на новите технологии, през последните десетилетия, дава възможност за безжично предаване на данни. Така се създава нов тип мрежи, наречени безжични мрежи.

Сензор е всяко устройство, което събира и предава данни. Данните могат да бъдат движение, светлина, изображение, химическа субстанция, звук и други. Безжичните сензорни мрежи позволяват наблюдение на големи площи без намесата на оператор. Тяхната работа се основава на обмен на локална информация между възлите с цел постигане на

глобална цел. Съвместната работа на сензорните възли е важна характеристика при решаване на сложни задачи. Безжичните мрежи обикновено се използват в случаите, когато не е възможно или не е подходящо използването на традиционните мрежи. Те намират приложения в различни области като наблюдение на климата, военна употреба, промишлеността, отчитане на информация от трудно достъпни места. За разлика от други мрежи, сензорните мрежи зависят от разполагането на сензори на физическо място, за да изпълнят желаната задача. Понякога се налага използването на стотици или хиляди сензорни възли в малки области, за да се осигури ефективен обхват на покритие в географски терен.

За много приложения безжичните сензори предлагат възможности с по-ниска цена за събиране на данни за състоянието на системата за намаляване на използваната енергия и по-добро управление на ресурсите. Безжичните сензори се използват за ефективно наблюдение на магистрали, мостове и тунели. Други приложения са непрекъснато наблюдение на офис сгради, болници, летища, фабрики, електроцентрали или производствени съоръжения. Сензорите могат да отчитат температура, напрежение или химически вещества.

WSN позволява автоматично наблюдение на почти всяко явление. WSN дава много възможности и предлага решения за голям брой нови проблеми. WSN са били използвани във военни дейности като разузнаване, проследяване [20], екологични дейности като предотвратяване на горски пожари, геофизични дейности като изследване на изригвания на вулкани [119], наблюдение на здравни данни [123] или гражданско строителство [100].

WSN възел съдържа няколко компонента, включващи предавател, батерия, микроконтролер, аналогова верига и сензорен интерфейс. В системите, захранвани от батерии, по-високите скорости на данни и по-честата употреба на предавателя консумират повече енергия. Има няколко отворени въпроса, касаещи сензорните мрежи, като обработка на сигнала [103], позициониране [121], оперативни разходи, локализация и оценка на местоположението.

Безжичните сензори имат две основни функции: събиране и предаване на данни. Събраните данни могат да са от различен тип (сеизмични, акустични, химически, оптични и др.). Сензорите, които са далеч от високо енергийния комуникационен възел (HECN) не може да общуват директно с него. За това сензорите предават своите данни към този възел на стъпки (хопове), като използват близки сензори като комуникационни

релета.

При планирането на WSN позиционирането на сензорните възли е един от основните проблеми. Покритието, получено с мрежата и икономическата цена на мрежата зависят пряко от него. Тъй като WSN могат да имат голям брой възли, задачата за избор на географските позиции на възлите, за оптимално проектирана мрежа, може да бъде много сложна. За това, метаевристиката изглежда интересен вариант за решаване на този проблем.

В тази глава предлагаме алгоритъм за решаване на задачата за разполагането на WSN, който е базиран на метода на мравките. Ние се съсредоточаваме както върху минимизирането на изчерпването на енергията на възлите в мрежата, така и върху минимизирането на броя на възлите, докато пълното покритие на мрежата и свързаността се смятат за ограничения. Задачата е многоцелева. Решаваме я с няколко подхода: като многоцелева и като я сведем до едноцелева. Преобразуваме задачата в едноцелева по два начина: чрез умножаване на целевите функции и чрез претеглена сума от целевите функции. Сравнили сме получените резултати с многоцелевия вариант. Изследвали сме работата на алгоритъма и влиянието на броя на мравките върху постигнатите решения.

Jourdan [78] решава вариант на задачата за WSN позициониране, използвайки многоцелеви генетичен алгоритъм. В неговата формулировка, трябва да се позиционират точно определен брой сензори, така че да се получи максимално покритие на зададената област. В други приложения се набляга на енергията, използвана от мрежата. В [79] е предложен АСО алгоритъм, а в [120] е предложен еволюционен алгоритъм за този вариант на задачата. АСО алгоритъмът в [39] взема под внимание само минимизиране броя на сензорите. В [96] са предложени няколко еволюционни алгоритъма за решаване на задачата. Генетичният алгоритъм от [81] постига подобни решения като алгоритмите от [96], но е тестван върху малки тестови задачи.

Броят на сензорните възли трябва да се поддържа възможни по-малък, по икономически причини, а от друга страна мрежата трябва да бъде свързана. И накрая, енергията на мрежата е ключов въпрос, който трябва да се вземе предвид, тъй като от това зависи продължителността на живота на мрежата.

5.2 Формулировка на задачата

Безжичната сензорна мрежа се състои от пространствено разпределени сензори, които наблюдават условията на зоната за събиране на информация, като температура, звук, вибрации, налягане, движение, замърсители и др. [1, 110]. Развитието на безжичните сензорни мрежи е било мотивирано от военни приложения като наблюдение и проследяване, но сега се използват в много индустриални и граждански области на приложение, включително наблюдение и контрол на промишлени процеси, наблюдение на здравето на машините, мониторинг на околната среда и местообитанията, приложения за здравеопазване, домашна автоматизация и контрол на трафика [110].

Сензорният възел може да варира по размер от този на кутия до размера от няколко милиметра [110]. Цената на сензорните възли е също така променлива, варираща от стотици долари до няколко цента, в зависимост от тяхната сложност и изискванията към отделните сензорни възли [110]. Ограниченията за размера и цената на сензорните възли водят до съответни ограничения върху ресурси като енергия, памет, изчислителна скорост и честотна лента [110].

Безжичната сензорна мрежа се състои от сензорни възли. Всеки сензорен възел събира данни от област около себе си, наречена област на наблюдение. Радиусът на наблюдение определя големината на областта, наблюдавана от сензора и така определя областта на наблюдение. Радиусът на комуникация определя колко далече възелът може да изпрати събраните данни. Специален възел в безжичната сензорна мрежа, наречен високо енергиен комуникационен възел (HECN), отговаря за външният достъп до мрежата. Така, всеки сензорен възел от мрежата трябва да може да изпрати данните си до HECN. Тъй като радиусът за комуникация обикновено е много по-малък от размера на мрежата, то е невъзможна директна връзка с отдалечените възли. Сензорните възли, които не са в обхвата на HECN, установяват многостъпков комуникационен път. Те предават събраните данни чрез другите сензори, които се намират в по-голяма близост до HECN.

Задачата за построяване на безжична сензорна мрежа цели намирането на географското разположение на сензорните възли, които оформят мрежата. В нашата формулировка сензорните възли трябва да бъдат поставени в терен, така че да осигурят пълно покритие на терена, с минимален брой сензори и минимизиране на енергията, изразходва-

на в комуникациите от всеки един възел, като същевременно се запазва свързаността на мрежата. Минимален брой сензори означава по евтина мрежа за построяване. Минимална енергия означава по-евтина мрежа за експлоатация. Енергията на мрежата определя живота на мрежата, колко често трябва да се сменят батериите. Това са противоположни цели, тъй като колкото повече възли има, толкова по-малък е дялът на препредаванията, които носят и обратното, когато се опитваме да намалим консумацията на енергия, обикновено добавяме нови възли. В тази задача се търси баланс между броя на използваните сензори и консумираната енергия от всеки един от тях.

За да се определи изразходваната от комуникации енергия, се изчислява броят на предаванията, които всеки възел извършва. WSN работи на цикли: В един цикъл всеки възел събира данните и ги изпраща до HESN. Всеки възел предава информационните пакети на съседен възел, който е най-близо до HESN, или самия HESN, ако е в рамките на комуникационния обхват. Когато няколко съседни възела са на равно разстояние от HESN, трафикът се разпределя между тях. Тоест, ако възел има n съседи, които са на най-краткото разстояние от HESN, всеки получава $1/n$ от своя трафик. Следователно, всеки възел има трафик, равен на 1 (съответстващ на собствените му изпратени данни) плюс сумата от всички трафик натоварвания, получени от съседи, които са по-далеч от HESN. Под енергия на мрежата разбираме енергията, използвана от най-натоварения възел. Това определя живота на батерията на този възел, а от там и живота на цялата мрежа.

От една страна наблюдаваната зона трябва да бъде напълно покрита. От друга страна, броят на сензорните възли трябва да се поддържа възможно най-нисък, тъй като използването на много възли представлява висока цена на мрежата, вероятно влияние на околната среда. Целите на тази задача е да се сведе до минимум мрежовата енергия и броят на разположените сензори, докато зоната е напълно покрита и свързана.

5.3 АСО за позициониране на безжична сензорна мрежа

Задачата за позициониране на безжична сензорна мрежа е трудна комбинаторна оптимизационна задача, която се нуждае от експоненциално

количество изчислителни ресурси (NP-hard). Точните методи, както и традиционните числени методи са неподходящи за решаването на такъв тип задачи. За това, такъв тип задачи се решават с прилагането на метаевристични методи. Много от съществуващите решения за позициониране на WSN идват от областта на еволюционните алгоритми [4, 96]. Трябва да отбележим, че методът на мравките е конструктивен метод, за разлика от повечето метаевристики, които се основават на подобряването на едно или повече предходни решения. Поради тази причина методът на мравките е много подходящ за задачи със строги ограничения.

В нашето приложение използваме MAX-MIN Ant System (MMAS) [116], който е един от най-популярните варианти на метода на мравките. Основната характеристика на MMAS е използването на фиксирана горна граница τ_{max} и долна граница τ_{min} на феромона. По този начин натрупването на големи количества феромон от част от възможните движения и повторение на едни и същи решения е частично предотвратено от едната страна, а количеството на феромона да намалее много и да стане близо до нула и съответният елемент да не бъде използван, е частично предотвратено, от друга страна.

Един от най-важните моменти на АСО алгоритмите е изграждането на графа на задачата. Трябва да изберем кои елементи от задачата ще отговарят на възлите и значението на дъгите, къде е по-подходящо да депозираме феромона - върху възлите или върху дъгите. В нашата реализация WSN задачата е представена с два графа, което е един от приносите ни. Теренът е моделиран чрез правоъгълна решетка $G = \{g_{ij}\}_{N \times M}$, където M и N са размерите на наблюдаваната област. Чрез граф G се изчислява покритието на областта. Използваме друг граф $G1_{N1 \times M1}$, във върховете на който поставяме сензорите, $N1 \leq N$ и $M1 \leq M$. Параметрите $N1 \leq N$ и $M1 \leq M$ зависят от радиусите на наблюдение и комуникация. По този начин се намаляваме броят на изчисленията, които алгоритъмът извършва, съответно се намалява времето за изпълнение. Феромонът се свързва с мястото на поставяне $Ph = \{ph_{ij}\}_{N1 \times M1}$, началният феромон има малка стойност, например $1/n_{ants}$. Мястото, където се намира НЕСН, е първата позиция в решението (нулева позиция).

Конструирането на евристичната информация е друг важен момент в АСО алгоритмите. Евристичната информация трябва да бъде конструирана така, че да управлява мравките да търсят по-добри решения и да избягват невалидни решения. За някои алгоритми не е очевидно как тя да бъде конструирана. Параметрите на задачата трябва да бъдат

комбинирани по възможно най-подходящия начин.

Предложената от нас евристична информация е произведение от три параметъра както следва:

$$\eta_{ij}(t) = s_{ij}l_{ij}(1 - b_{ij}), \quad (5.1)$$

където s_{ij} е броят на непокритите точки, които новият сензор ще покрие,

$$l_{ij} = \begin{cases} 1 & \text{ако има комуникация} \\ 0 & \text{ако няма комуникация} \end{cases} \quad (5.2)$$

b е матрицата на решенията а матричният елемент $b_{ij} = 1$, когато има поставен сензор на възел (i, j) от граф $G1$, в противен случай $b_{ij} = 0$. Чрез s_{ij} се опитваме да увеличим локално покритите точки, повече ново покрити точки може да доведе до по-малък брой сензори. С l_{ij} гарантираме свързаността на мрежата. Поставянето на сензори започва от НЕСН към периферията. Всеки нов сензор се поставя така, че да може да предаде събраната информация до НЕСН. Чрез изразът $(1 - b_{ij})$ се гарантира, че на един възел от граф $G1$ ще се постави най-много един сензор, т.е. в една и съща позиция няма да има два или повече сензора. Когато вероятността за преход $p_{ij} = 0$ за всички стойности на i и j , търсенето на нови позиции за поставяне на сензори спира. Така изграждането на решението спира, ако няма повече свободни позиции, или всички точки са покрити или нова комуникация е невъзможна.

Многоцелевата оптимизация (МОП) е процесът на едновременно оптимизиране на две или повече противоречиви цели, обект на определени ограничения. Ако многоцелевият проблем е добре структуриран, не трябва да има едно-единствено решение, което едновременно да оптимизира всяка цел в крайна степен.

Многоцелевата оптимизация (МОП) има своите корени през деветнадесети век в работата на Еджуърт и Парето в икономиката [101]. Оптималното решение за МОП не е едно-единствено решение както за едноцелевите оптимизационни задачи, а множество от решения, дефинирани като оптимални решения по Парето. Решението е оптимално по Парето, ако не е възможно да се подобри дадена цел, без да се влоши поне една от другите цели. Основната цел на разрешаването на многоцелеви задачи е да се получи оптималното множество на Парето, наречено Парето фронт. Едно решение доминира над друго, ако поне един от неговите компоненти е по-добър от същия компонент на другото решение,

а другите компоненти не са по-лоши. Парето фронтът е множеството от недоминирани решения.

5.4 Преобразуване на WSN задачата до едноцелева чрез умножение

Задачата за позициониране на WSN е NP-трудна многоцелева оптимизационна задача. Опростяваме я като я преобразуваме до едноцелева. Обикновено преобразуването от многоцелева до едноцелева задача влошава намерените решения. За това е много важно как ще бъде конструирана новата целева функция. Ние предлагаме следната целева функция:

$$C(V_k) = f_1(V_k) \times f_2(V_k), \quad (5.3)$$

където $f_1(V_k)$ е броя на сензорите поставени от мравка k , а $f_2(V_k)$ е енергията на решението намерено от мравка k . V_k е решението, намерено от мравка k . Целта ни е да минимизираме и двете целеви функции f_1 и f_2 , което води до минимизиране на новата целева функция C .

С нашия алгоритъм можем да решим задачата за позициониране на WSN за произволна правоъгълна област и HECN може да бъде фиксиран на произволна точка от областта. Алгоритъмът е тестван върху тестови пример използван и от други автори [4, 96]. Примерът се състои от област състояща се от 500×500 точки за покриване с използването на сензори с радиус на наблюдение равен на радиуса на комуникация, покриващ 30 точки (виж Таблица 5.1). Използваме този пример за да можем да направим сравнение на получените резултати от нашия алгоритъм с другите. Областта е с площ от 250 000 точки общо, и всеки сензор покрива 2827 точки, което означава, че при идеални условия ще са необходими само 89. Тези идеални условия не съществуват, защото това означава да няма припокриване на наблюдаваните от сензорите области, което е невъзможно поради тяхната геометрична форма (кръг). Следователно очакваният минимален брой възли за пълно покритие е по-голям от 89.

Когато прилагаме нашия ACO алгоритъм към този пример граф G ще съдържа 500×500 върха. За $G1$ използваме 100 пъти по-малък граф, той се състои от 50×50 върха. Броят на върховете на граф $G1$ е пропорционален на броя на върховете на граф G и на големината (броя

5.4. Преобразуване на WSN задачата до едноцелева чрез умножение 97

покриващи точки) на радиусите на наблюдение и на комуникация. Така възлите на граф $G1$ се позиционират върху възлите на граф G и радиусите на наблюдение и на комуникация са в размер 3 точки от граф $G1$. В използвания пример HECN се намира в центъра на областта.

Таблица 5.1: Параметри на задаата

Област	500X500
Радиус на наблюдение	30
Радиус на комуникация	30

Пример за решение, което постига пълно покритие на областта, е квадратна решетка, образувана от сензорите, разделени с 30 точки. Започвайки от HECN, трябва да бъдат покрити 250 точки от всяка страна на областта, което изисква 8 сензора. Следователно мрежата има 17 ($8 + 8 + 1$) реда и 17 колони, като по този начин използва 289 сензора, включително HECN. В тази симетрична конфигурация има четири възела, директно свързани към HECN, така че целият трафик на мрежовите 288 съобщения на рунд се разделя равномерно между тях. Така най-натовареният възел предава 72 съобщения. Резултатът от това решение е $(288, 72)$, където първото число е броят използвани сензори, а второто е енергията на мрежата. Ще наричаме това решение симетрично решение. Ще смятаме едно решение за добро, ако то доминира симетричното решение.

Приложен е MAX-MIN метод на мравките (MMAS) със следните параметри: $\alpha = \beta = 1$, $\rho = 0.5$. Когато броят на мравките се удвои, времето за работа се удвоява, както и използваната памет. За това е важно да се намери най-добрият брой мравки, за да се използват по-малко компютърни ресурси. Използвали сме от 1 до 10 мравки, като максималният брой итерации е фиксиран на 60. Изследваме влиянието на броя на мравките върху качеството на получените решения.

В Таблица 5.2 сравняваме резултатите, постигнати от нашия АСО алгоритъм и от еволюционните алгоритми използвани в [96] (по отношение на сензорите и по отношение на енергията). Наблюдаваме, че нашият едноцелеви АСО алгоритъм постига решения с по-малък брой сензори, отколкото еволюционните алгоритми. Енергията на нашето решение с най-малко енергия е малко повече от това, постигнато чрез еволюционни алгоритми, но броят на сензорите е много по-малък. Ще отбележим, че

Таблица 5.2: Сравнение с други методи

Алгоритми	мин. сензори	мин. енергия
Symmetric	(288,72)	(288,72)
МОЕА	(260, 123)	(291,36)
NSGA-II	(262,83)	(277,41)
IBEA _{HD}	(265,83)	(275,41)
АСО	(225,63)	(237,51)

решението с най-малка енергия, постигнато от еволюционните алгоритми, не доминира симетричното решение защото има повече сензори от него.

Таблица 5.3: Експериментални резултати

мравки	Парето фронт (енергия)														
1	53													58	63
2	53			54			55			57					
3			53			54			55			57			
4	52				53			55				57			
5	52					53	54	55					57		
6			51	53			54	55					57		
7			51			53	54					57			
8			51	53			54					57			
9			51	53			54					57			
10			51	53			54					57			
сензори	239	238	237	236	235	234	233	232	231	230	229	228	227	226	225

Нашият АСО алгоритъм е пуснат 30 пъти със всяка от бройките мравки. Извлекли сме Парето фронта от оптималните решения от тези 30 пускания. На Таблица 5.3 са показани намерените недоминирани решения (Парето фронт). След анализ на резултатите в таблицата, стигаме до заключението, че Парето фронтът постигнат с използването на 6 мравки доминира тези постигнати с 1,2,3 и 4 мравки. Парето Фронтите пос-

5.4. Преобразуване на WSN задачата до едноцелева чрез умножение 99

тигнати с 8, 9 и 10 мравки са идентични. Парето фронтът постигнат от 7 мравки доминира този постигнат от 8 мравки, съответно от 9 и 10. Парето фронтът постигнат от 4 мравки е част от Парето фронтът постигнат от 5 мравки, но съдържа едно решение по-малко.

Направили сме Парето фронт получен от всички пускания на алгоритъма с различния брой мравки (от 1 до 10) и ще го наричаме общ Парето фронт. Нека имаме брой сензори от 225 до 239. Ако за някой брой сензори няма съответна енергия в общия Парето фронт (т.е. нямаме постигнато такова решение) то слагаме енергията да бъде равна на енергията на решението с по-малко сензори. Можем да направим това, защото ако вземем едно решение и добавим сензор в обхвата на НЕСН, то при новополученото решение енергията няма да бъде по-голяма (дори би могла да се намали) единствено броят на сензорите ще се увеличи с 1. Така на всеки брой сензори имаме съответстваща енергия. Този фронт ще наричаме Разширен фронт. За нашия пример разширеният фронт е $\{(239, 51), (238, 51), (237, 51), (236, 53), (235, 53), (234, 53), (233, 54), (232, 55), (231, 57), (230, 57), (229, 57), (228, 57), (227, 57), (226, 57), (225, 57)\}$.

За да можем да оценим кой Парето фронт е по-добър сме добавили допълнителен критерий. Ще изчислим разстоянието между Паретофронта и Разширения фронт. За да можем да изчислим разстоянието, разширяваме всеки от Парето фронтите по начин подобен на общия Парето фронт. Разстоянието между Парето фронта и Разширения фронт е сумата от разстоянията между точките с еднакъв брой сензори, или това е разликата между тяхната енергия. Тази разлика е винаги положителна, защото Разширеният фронт доминира Парето фронтите. Според този критерий, най-добрият Парето фронт ще е най-близкият до Разширения фронт.

Таблица 5.4: Разстояние от разширения фронт

брой мравки	5	6	7
разстояние	11	2	2

В Таблица 5.4 показваме разстоянието между Разширения фронт и Парето фронтите постигнати с 5, 6 и 7 мравки. Анализирайки резултата можем да заключим, че Парето фронтите постигнати с 6 и 7 мравки са по-добри от този постигнат от 5 мравки, защото са по-близо до Раз-

ширения фронт. Парето фронтите с използването на 6 и на 7 мравки са еднакво добри, защото са еднакво отдалечени от Разширения фронт. Имайки в предвид, че използването на повече мравки увеличава времето за изпълнение и използваната памет, заключаваме, че най-добрия брой мравки за решаването на нашата задача е 6.

5.5 Преобразуване на WSN задачата до едноцелева чрез сумиране

Първо ще разгледаме случая когато целевата функция е сума от броя на сензорите и енергията на мрежата и ние търсим решение, което да я минимизира. Новата целева функция е както следва:

$$C(V_k) = \frac{f_1(V_k)}{\max_i f_1(V_i)} + \frac{f_2(V_k)}{\max_i f_2(V_i)} \quad (5.4)$$

където V_k е решението, конструирано от мравка k и $f_1(V_k)$ и $f_2(V_k)$ са броя на сензорите и енергията съответстващи на решението V_k . Разделяйки f_1 на $\max_i f_1$ и f_2 на $\max_i f_2$ е начин на нормализиране на стойността съответно на f_1 и f_2 . Максимумът е спрямо решенията от първата итерация. По този начин, когато енергията и/или броят на сензорите намалее, стойността на целевата функция ще намалее и двата компонента имат еднакво влияние.

И в този случай използваме тестовия пример от предната секция. След няколко пускания на алгоритъма сме определили най-подходящите стойности на неговите параметри. Прилагаме MMAS варианта на метода на мравките със следните стойности на параметрите: $\alpha = \beta = 1$, $\rho = 0.5$. Отново използваме брой на мравките от 1 до 10 като максималния брой итерации е фиксиран на 60.

В Таблица 5.5 сравняваме резултатите, постигнати от нашия АСО алгоритъм и от еволюционните алгоритми използвани в [96] и симетричното решение (по отношение на сензорите и по отношение на енергията). Посочените еволюционни алгоритми са решавали задачата като многоцелева. АСОprod е АСО алгоритъма от предходната секция, когато целевата функция е произведение от двете целеви функции на задачата, а АСОsum е вариантът в който целевата функция е сума от нормализираните стойности на двете целеви функции. И двата варианта на АСО

5.5. Преобразуване на WSN задачата до едноцелева чрез сумиране 101

намират по-добри решение, сравнени с другите алгоритми. ACOprod намира решение с по-малък брой сензори от ACOsum, докато ACOsum намира решение с по-малко енергия.

Таблица 5.5: Сравнение между алгоритмите

Алгоритъм	мин. сензори	мин. енергия
Symmetric	(288,72)	(288,72)
MOEA	(260, 123)	(291,36)
NSGA-II	(262,83)	(277,41)
IBEA _{HD}	(265,83)	(275,41)
ACOprod	(225,63)	(237,51)
ACOsum	(226,58)	(234,48)

Извършени са 30 пускания на алгоритъма ACOsum. Броят на сензорите на получените решения е в интервала [226, 247]. Така решението с най-големият брой сензори, съдържа по-малко сензори от решението с най-малък брой сензори на еволюционните алгоритми. Решението с най-малка енергия използва малко по-голяма енергия от еволюционните алгоритми, но за сметка на това с много по-малко сензори.

Изучаваме влиянието на броя на мравките върху работата на алгоритъма и качеството на постигнатите решения. Нашата цел е да намерим оптимален брой мравки, първо според качеството на решенията и второ - според изчислителните ресурси.

Построили сме парето фронт, на основата на 30 пускания на алгоритъма, за всеки един брой мравки, като сме отделили недоминираните решения на постигнатите оптимални решения. В Таблица 5.6 са показани достигнатите недоминирани решения (Парето фронтове). Най-лявата колона е броя сензори, а останалите колони показват постигнатата енергия от съответния брой мравки. Наблюдаваме, че Парето фронтът постигнат с използването на 8 мравки доминира Парето фронтите постигнати от 1,2,3,4,5,6, и 7 мравки. Парето фронтите постигнати от 8, 9 и 10 мравки не се доминират един друг. За да ги оценим сме построили Разширен фронт, подобно на предходната секция. Така на всеки брой сензори има съответстваща енергия. За нашият тестови пример разши-

Таблица 5.6: Парето фронтове

сензори	мравки									
	1	2	3	4	5	6	7	8	9	10
239	55			50						
238										
237	56									
236										
235		52	52	52					52	
234					51	51	51	48		51
233									54	54
232								51	55	55
231	58	58			56	56	56		51	
230		59	57	57	57	57	57	54	57	
229		60		60	60	60	58	56		
228	60									
227			60	61				57	58	58
226								58		

реният фронт е $\{(226, 58), (227, 58), (228, 58), (229, 58), (230, 57), (231, 56), (232, 55), (233, 54), (234, 51), (235, 51)\}$. Изчислили сме разстоянието между Парето фронтите и Разширения фронт.

Таблица 5.7: Растояние от разширения фронт

мравки	8	9	10
разстояние	3	4	3

Анализирайки Таблица 5.7 наблюдаваме, че Парето фронтите достигнати с 8 и с 10 мравки са на разстояние 3 от Разширения фронт, докато Парето фронтът достигнат с 9 мравки е на разстояние 4. Това означава, че Парето фронтите достигнати с 8 и с 10 мравки са еднакво добри и са по-добри от Парето фронта достигнат с 9 мравки. Заключаваме, че 8 мравки е оптималният брой мравки за този вариант на

5.5. Преобразуване на WSN задачата до едноцелева чрез сумиране 103

алгоритъма, защото с по-малко мравки Парето фронтът се достига за по-кратко време и се използва по-малко памет.

До тук разгледахме вариантът когато целевата функция е сума от двете целеви функции на задачата. Друг вариант е целевата функция да бъде претеглена сума. В този случай целевата функция е както следва:

$$C(V_k) = \lambda \frac{f_1(V_k)}{\max_i f_1(V_i)} + (1 - \lambda) \frac{f_2(V_k)}{\max_i f_2(V_i)} \quad (5.5)$$

В този случай е много важно правилно да бъде определен тегловият коефициент λ . Ще изследваме влиянието на параметъра λ , когато той е в множеството $\{0.125, 0.250, 0.375, 0.5, 0.625, 0.750, 0.875\}$. В Таблица 5.8 са показани недоминираните решения за всяка стойност на λ , а в Таблица 5.9 е направено сравнение с останалите алгоритми.

Таблица 5.8: Недоминирани решения за претеглените суми

λ	Недоминирани решения
0.125	(238,49) (235,51) (229,58) (227,60)
0.250	(238,49) (235,51) (228,58)
0.375	(238,49) (235,51) (229,57) (227,58) (226,60)
0.500	(238,49) (235,56) (229,57) (228,58) (226,60)
0.625	(238,49) (235,51) (229,57) (227,68) (226,60)
0.750	(238,49) (235,51) (229,57) (227,68) (226,60)
0.875	(238,49) (230,51) (225,57)

АСО алгоритъмът с претеглени суми (АСОweight) намира най-добро решение когато $\lambda = 0.875$ броят на сензорите има по-голямо влияние върху решението, отколкото енергията. По отношение на решението с най-малко сензори, решението достигнато от АСОweight доминира решенията достигнати от АСОprod и АСОsum. По отношение на минималната енергия, решението достигнато от АСОsum доминира тези достигнати от АСОprod и АСОweight.

Таблица 5.9: Сравнение между алгоритмите

Алгоритъм	мин. сенсори	мин. енергия
Symmetric	(288,72)	(288,72)
МОЕА	(260, 123)	(291,36)
NSGA-II	(262,83)	(277,41)
IBEA _{HD}	(265,83)	(275,41)
ACOp _{rod}	(225,63)	(237,51)
ACOs _{um}	(226,58)	(234,48)
ACOw _{eight}	(225,57)	(238,49)

5.6 Многоцелева WSN задача

В тази секция решаваме задачата за позициониране на WSN като многоцелева задача. За целта сме разработили алгоритъм на основата на метода на мравките наречен ACOf_{multi}. За разлика от едноцелевия алгоритъм, за обновяване на феромона използваме фитнес функция вместо целевата функция. Ролята на фитнес функцията е намерените решения да бъдат оценени. Целта е да се добави феромон на елементите на недоминираните решения и да насърчим мравките да търсят други недоминирани решения около тях. Предлагаме следната фитнес функция:

$$F(V_k) = \frac{f_1(V_k)}{\max_i f_1(V_i)} + \frac{f_2(V_k)}{\max_i f_2(V_i)} \quad (5.6)$$

Изследвали сме влиянието на броя на мравките върху качеството на намерените решения. Фиксирали сме броя на итерациите да бъде 60 а броя на мравките да принадлежи на следното множество {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}. Използваме същият тестови пример както в предходните секции за да можем да направим сравнение. Направени са 30 пускания на алгоритъма. Намереният брой сензори е в интервала [223, 247]. Отново решението с най-големия брой сензори съдържа по-малко сензори от решенията с най-малко сензори намерени от еволюционните алгоритми.

В Таблица 5.10 са показани намерените от ACOf_{multi} недоминирани решения (Парето фронт). В най-лявата колона е броят сензори, а в следващите колони е енергията постигната от различния брой мравки за този

Таблица 5.10: Парето фронтове

сензори	мравки									
	1	2	3	4	5	6	7	8	9	10
244					52					
243										
242										
241										
240	53	53								
239	56			50						
238			53							
237										
236										
235			54						50	
234					53			48	53	
233							51			
232			55	51	54	50	52	51		48
231		55			55		53			
230	57					52		54		
229	58			55				56		56
228									54	
227		57	57		57	56		57		
226	59	95	73	57	59	57	56			
225				58	60	58	57	58		57
224	61				88	65	61	59	57	71
223					89	81				

брой сензори. Анализирайки Таблица 5.10 наблюдаваме, че Парето фронтът, получен при използването на 6 мравки доминира Паретофронтите, получени при използването на 1,2,3,4, и 5 мравки. Парето фронтите, получени при използването на 6,7,8,9 и 10 мравки не се доминират един друг и не можем да кажем кой от тях е по-добър.

Както в предните секции, построяваме общ Парето фронт от обединяване на Парето фронтите, постигнати от 6, 7, 8, 9 и 10 мравки. Общият Парето фронт е $\{(232,48), (230,52), (228,54), (226,56), (224,57),$

(223,81)}. След това построяваме и Разширения фронт, който има видът: $\{(234,48), (233,48), (232,48), (231,52), (230,52), (229,54), (228,54), (227,56), (226,56), (225,57), (224,57), (223,81)\}$. Изчисляваме разстоянието от Паретофронтовете до Разширения фронт. Така оценяваме Парето фронтовете. Най-добрият трябва да е най-близо до Разширения фронт.

Таблица 5.11: Разстояние от Разширения фронт

мравки	6	7	8	9	10
разстояние	20	23	21	22	29

В Таблица 5.11 са показани разстоянията до Парето фронтовете постигнати от 6, 7, 8, 9, и 10 мравки до Разширения фронт. Анализирайки Таблица 5.11, можем да заключим, че Парето фронтът, получен от 6 мравки, е на най-малко разстояние от Разширения фронт, следователно е най-добър.

В Таблица 5.12 е направено сравнение между резултатите постигнати от нашия многоцелеви АСО алгоритъм, тези постигнати от еволюционните алгоритми в [96], едноцелевите АСО алгоритми от предходните секции и симетричното решение.

Таблица 5.12: Сравнение между алгоритмите

Алгоритъм	мин. сензори	мин енергия
Symmetric	(288,72)	(288,72)
MOEA	(260,123)	(291,36)
NSGA-II	(262,83)	(277,41)
IBEA _{HD}	(265,83)	(275,41)
ACOprod	(225,63)	(237,51)
ACOsum	(226,58)	(234,48)
ACOweight	(225,57)	(238,49)
ACOmulti	(224,57)	(234,48)

Виждаме, че многоцелевият АСО алгоритъм постига по-добри решения от едноцелевите АСО алгоритми. Решението с най-малко сензори

5.7. Прилагане на различни Интеркритериални анализи към АСО алгоритъма¹⁰⁷

е (224, 57). АСОmulti дори постига решение със 223 сензора, но то има много висока енергия, по-висока от тази на симетричното решение.

Използвали сме също така и хипер обем (Hyper volume [4]) като оценка за качество на намерените решения. Математически, за всяко решение се построява хиперкуб с референтна точка W и решението като диагонално противоположна точка на хиперкуба. Референтната точка се намира като се вземе векторът с най-лошите стойности на целевите функции. След това се намира обединението на всички хиперкубове и се пресмята техният хипер обем (INV). Най-добър е алгоритъмът постигнал по-голям хипер обем.

Таблица 5.13: Хипер обеми на решенията

Алгоритъм	среден Хипер обем	макс. Хипер обем	мин. Хипер обем
МОЕА	0.7388	0.7847	–
NSGA-II	0.7306	0.7868	–
IBEA _{HD}	0.7280	0.7704	–
АСОmulti	0.9440	0.9900	0.7769

В Таблица 5.13 са показани хипер обемите на фронтите, намерени от различните алгоритми. Наблюдаваме, че по отношение на хипер обема, АСОmulti е значително по-добър от останалите алгоритми. Трите еволюционни алгоритъма имат статистически сходни резултати. Минималният хипер обем, отнасящ се за алгоритъма АСОmulti е подобен на максималните хипер обеми, отнасящи се за трите еволюционни алгоритъма. Така по отношение на критерия с хипер обеми, алгоритъмът АСОmulti постига по-добри решения (е по-добър) от еволюционните алгоритми.

5.7 Прилагане на различни Интеркритериални анализи към АСО алгоритъма

В предходните секции на тази глава бяха предложени разнообразни варианти на основата на метода на мравките, за решаване на задачата за позициониране на WSN: многоцелеви АСО алгоритъм; едноцелеви АСО алгоритъм с умножение на двете целевите функции, едноцелеви

АСО алгоритъм със сумиране на двете целеви функции. Алгоритмите са тествани върху пример състоящ се от правоъгълна област съдържаща 500×500 точки и радиуси на наблюдение и комуникация покриващи 30 точки. Броят на използваните мравки е от 1 до 10. В тази глава ще бъдат приложени няколко варианта на Интеркритериален анализ (ICrA) върху вариантите на алгоритъма.

Входната индексна матрица (IM) и пълното множество от получени резултати могат да бъдат намерени на <http://intercriteria.net/studies/aco/>.

ICrA е приложен върху данни от индексната матрица, където различните АСО алгоритми (*АСО_u* – едноцелеви с умножение, *АСО_s* – едноцелеви със сумиране и *АСО_m* – многоцелеви) са представени като критерии а броя на сензорите [223, 224, 225, 226, 227, 228, 229, 230, 231, 232, 233, 234, 235, 236, 237, 238, 239, 240, 241, 242, 243, 244] – са обекти. Използвана е платформата ICrAData за изчисляване на ICrA [75].

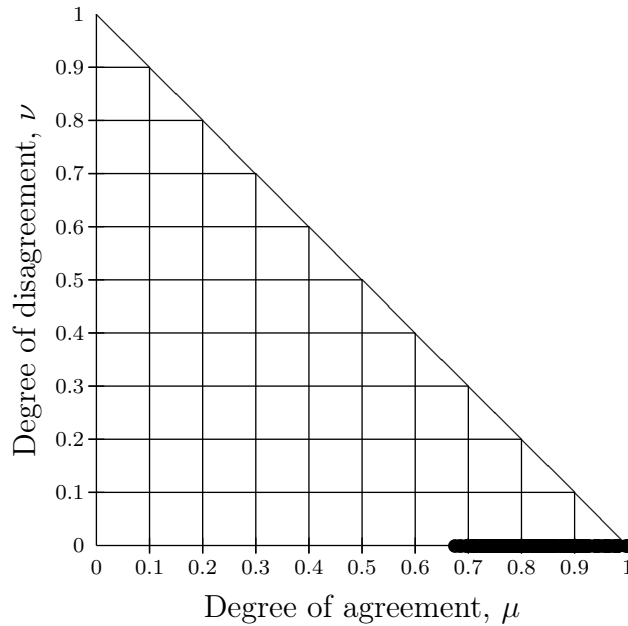
Получените ICrA резултати са анализирани на базата предложената в [11] скала за съгласие и несъгласие. Схемата за нивото на съгласие и несъгласие между всяка двойка критерии е представено в Таблица 5.14.

Таблица 5.14: Скала за съгласие и несъгласие [11]

Интервал за $\mu_{C,C'}$	Значение
[0.00-0.05]	strong negative consonance (SNC)
(0.05-0.15]	negative consonance (NC)
(0.15-0.25]	weak negative consonance (WNC)
(0.25-0.33]	weak dissonance (WD)
(0.33-0.43]	dissonance (D)
(0.43-0.57]	strong dissonance (SD)
(0.57-0.67]	dissonance (D)
(0.67-0.75]	weak dissonance (WD)
(0.75-0.85]	weak positive consonance (WPC)
(0.85-0.95]	positive consonance (PC)
(0.95-1.00]	strong positive consonance (SPC)

5.7.1 Резултати от прилагането на μ -базиран ICrA

Всички резултати от прилагането на μ -базиран ICrA са визуализирани на Фигура 5.1 със специфична триъгълна геометрична интерпретация на интуиционистки размитите множества (IFS), по този начин ни позволява да подредим тези резултати едновременно според степените на "съгласие" $\mu_{C,C'}$ и "несъгласие" $\nu_{C,C'}$ на интуиционистки размити двойки.



Фигура 5.1: Представяне на ICrA резултатите – μ -базиран ICrA

Както може да се види на Фигура 5.1, стойностите на ν за всички разглеждани двойки са 0 ($\nu_{C,C'} = 0$). Има няколко резултата за $\pi_{C,C'}$ които са между $\pi_{C,C'} = 0.02$ и $\pi_{C,C'} = 0.32$.

Таблица 5.15 показва двойките критерии, които принадлежат на SPC. Показана е единствено връзката между алгоритмите с различна целева функция. Има още 24 двойки критерии, които също са в SPC. Тези двойки критерии съответстват на алгоритми с една и съща целева функция и сходен брой мравки, като например $ACOu8 - ACOu9$ или $ACOs2 - ACOs3$. Наблюдават се и някои изключения, като двойките критерии $ACOu1 - ACOu2$ и $ACOs4 - ACOs5$ които са WPC и

$ACOu5 - ACOu6$ които са WD. Трябва да отбележим, че тези двойки принадлежат на едноцелевите варианти на задачата. Тези двойки не са включени в таблицата.

Таблица 5.15: Резултати от μ - базиран ICrA – двойки в SPC

Двойки критерии от ACO алгоритми	стойност на $\mu_{C,C'}$
$ACOs8 - ACOm8$	0.98
$ACOs5 - ACOm8$	0.96
$ACOs6 - ACOm8$	0.96
$ACOs7 - ACOm8$	0.96
$ACOu3 - ACOm3$	0.95

Таблица 5.15 показва, че SPC се наблюдава в четири случая между $ACOs - ACOm$ и в един между $ACOu - ACOm$. Не се наблюдава строга връзка между $ACOu$ и $ACOs$. Като заключение ще кажем, че между едноцелеви алгоритми SPC се наблюдава само когато те са от един тип и са с близък брой мравки.

В Таблица 5.16 са показани резултати за алгоритми с различни целеви функции които са в PC. На този случай отговарят двойки от многоцелевия алгоритъм, които използват сходен брой мравки, както и двойки от едноцелевите алгоритми с доста различен брой мравки ($ACOs2 - ACOs10$, $ACOs3 - ACOs10$, $ACOu2 - ACOu10$, $ACOu7 - ACOu10$). Пълното множество от получените числени резултати може да се види на <http://intercriteria.net/studies/aco/>.

От общо 84 случая на двойки критерии между различните алгоритми, 19 двойки са между $ACOs$ и $ACOu$ (едноцелеви), а останалите са между едноцелеви и многоцелеви ACO алгоритми. Това означава, че връзката между едноцелевите варианти на алгоритъма и многоцелевия вариант е по-слаба отколкото между различните едноцелеви варианти. Последните са предимно в SPC.

Показаните в Таблица 5.16 резултати показват, че алгоритмите $ACOs - ACOu$ са много по-сходни, отколкото $ACOu - ACOm$.

163 двойки критерия са в WPC. 157 от тях са между едноцелеви и многоцелеви ACO алгоритми. Само 6 двойки са между $ACOs - ACOs$, а 4 са между $ACOu - ACOu$. 137 от двойките критерии са между $ACOu -$

5.7. Прилагане на различни Интеркритериални анализи към АСО алгоритъма¹¹¹

АСОs и *АСОu – АСОm*. Това показва, че двата едноцелеви АСО алгоритъма работят по доста различен начин.

64 двойки критерии са в WD. При тях няма двойки критерии между *АСОs – АСОs*.

В Таблица 5.17 са дадени двойки критерии между алгоритми с една и съща целева функция, които са в слабо положително съгласие.

Имайки в предвид връзката между алгоритмите с една и съща целева функция, можем да оценим влиянието на броя на мравките върху поведението на алгоритъма. Резултатите показват, че *АСОs* е по-малко чувствителен към използвания брой мравки, в сравнение с *АСОu* и *АСОm*. По-голяма разлика в поведението на алгоритъма се наблюдава при *АСОu* и *АСОm* – 53% от всички двойки критерии. В 14% от случаите се наблюдава WD за алгоритмите *АСОu – АСОu* и *АСОm – АСОm*, при голяма разлика между броя на използваните мравки (повече от три), което означава, че *АСОu* и *АСОm* силно се влияят от броя на използваните мравки.

5.7.2 Резултати от използването на ν -базиран ICrA

Всички резултати, основани на ν -базиран ICrA, са визуализирани на Фигура 5.2.

Представените резултати показват, че в този случай степента на неопределеност ($\nu_{C,C'}$) не е нула. В представените резултати ще бъде включена и стойността на $\nu_{C,C'}$.

В Таблица 5.18 са представени резултати, когато е използван ν -базиран ICrA. В сравнение с μ -базиран ICrA, разглежданите двойки критерии не са в SPC. Има двойки които са в WD (*АСОs5 – АСОm8*, *АСОs6 – АСОm8* и *АСОs7 – АСОm8*) или в WPC (*АСОs8 – АСОm8* и *АСОu3 – АСОm3*).

Представените резултати показват, че в случая на ν -базиран ICrA се наблюдава по-слаба корелация в сравнение с μ -базиран ICrA. Основната причина за това е начинът на интерпретиране на еднаквите резултати в индексната матрица (виж <http://intercriteria.net/studies/aco/>). Еднаквите стойности се причисляват към степента на "несъгласие" и така стойността на $\nu_{C,C'}$ нараства, а стойността на $\mu_{C,C'}$ намалява.

В Таблица 5.19 са представени резултати от различни целеви функции разгледани в секция 5.7.1, заедно със съответстващата $\nu_{C,C'}$ стойност.

Двойките критерии, които са в слабо положително съгласие са представени с удебелен шрифт. Както може да се види, само 18 двойки крите-

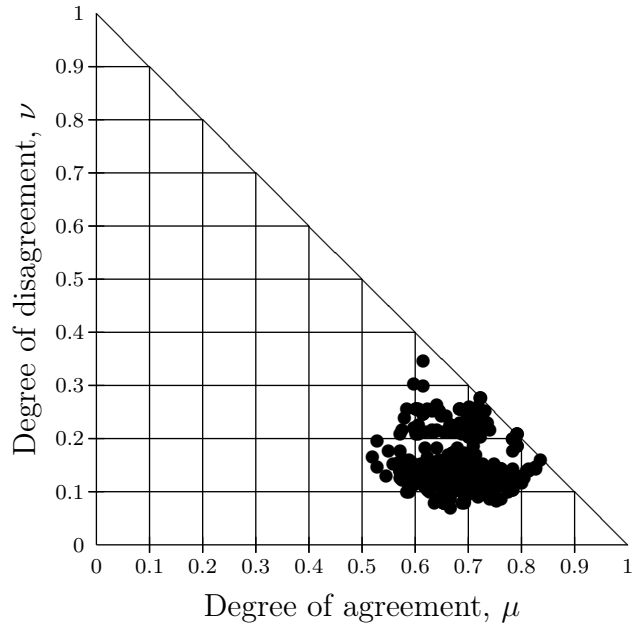
Таблица 5.16: Резултати от μ -базиран ICGA – двойки в РС

Двойки критерии от ACO алгоритми	Стойност на $\mu_{C,C'}$	Двойки критерии от ACO алгоритми	Стойност на $\mu_{C,C'}$
<i>ACOu6 – ACOm3</i>	0.93	<i>ACOs8 – ACOm9</i>	0.88
<i>ACOs9 – ACOm8</i>	0.92	<i>ACOu2 – ACOm2</i>	0.88
<i>ACOu4 – ACOm3</i>	0.92	<i>ACOu4 – ACOs1</i>	0.88
<i>ACOs8 – ACOm5</i>	0.92	<i>ACOu6 – ACOs9</i>	0.88
<i>ACOu1 – ACOm1</i>	0.92	<i>ACOs9 – ACOm3</i>	0.87
<i>ACOu3 – ACOs4</i>	0.92	<i>ACOu2 – ACOm4</i>	0.87
<i>ACOu4 – ACOm4</i>	0.92	<i>ACOu5 – ACOs1</i>	0.87
<i>ACOu5 – ACOm3</i>	0.92	<i>ACOs9 – ACOm7</i>	0.87
<i>ACOu5 – ACOm4</i>	0.91	<i>ACOu6 – ACOs10</i>	0.87
<i>ACOu3 – ACOm4</i>	0.91	<i>ACOs7 – ACOm6</i>	0.86
<i>ACOs7 – ACOm5</i>	0.90	<i>ACOs8 – ACOm10</i>	0.86
<i>ACOs9 – ACOm9</i>	0.90	<i>ACOu7 – ACOs10</i>	0.86
<i>ACOu10 – ACOm3</i>	0.90	<i>ACOs7 – ACOm10</i>	0.86
<i>ACOu7 – ACOm3</i>	0.90	<i>ACOu10 – ACOs9</i>	0.86
<i>ACOu8 – ACOm3</i>	0.90	<i>ACOu5 – ACOs10</i>	0.86
<i>ACOu9 – ACOm3</i>	0.90	<i>ACOu7 – ACOs9</i>	0.86
<i>ACOs7 – ACOm7</i>	0.90	<i>ACOu8 – ACOs9</i>	0.86
<i>ACOu1 – ACOm4</i>	0.90	<i>ACOu9 – ACOs9</i>	0.86
<i>ACOs7 – ACOm9</i>	0.89	<i>ACOu1 – ACOs4</i>	0.85
<i>ACOs8 – ACOm7</i>	0.89	<i>ACOu2 – ACOm5</i>	0.85
<i>ACOu4 – ACOs4</i>	0.89	<i>ACOu2 – ACOs10</i>	0.85
<i>ACOu2 – ACOm3</i>	0.89	<i>ACOu4 – ACOs10</i>	0.85
<i>ACOu3 – ACOs1</i>	0.89	<i>ACOs9 – ACOm6</i>	0.85
<i>ACOu5 – ACOs4</i>	0.89	<i>ACOu2 – ACOs4</i>	0.85
<i>ACOs8 – ACOm6</i>	0.88	<i>ACOu3 – ACOs9</i>	0.85
<i>ACOs9 – ACOm5</i>	0.88	<i>ACOu6 – ACOs1</i>	0.85

5.7. Прилагане на различни Интеркритериални анализи към АСО алгоритъма113

Таблица 5.17: Резултати от μ -базиран ICrA –двойки в WPC

Двойки критерии от АСО алгоритми от АСО алгоритми	Стойности на $\mu_{C,C'}$
$ACOs_4 - ACOs_{10}$	0.77
$ACOs_4 - ACOs_5$	0.82
$ACOs_4 - ACOs_6$	0.82
$ACOs_4 - ACOs_7$	0.82
$ACOs_4 - ACOs_8$	0.81
$ACOs_4 - ACOs_9$	0.84
$ACOu_1 - ACOu_2$	0.78
$ACOu_1 - ACOu_3$	0.81
$ACOu_1 - ACOu_4$	0.82
$ACOu_1 - ACOu_5$	0.82



Фигура 5.2: Представяне на ICrA резултати при ν -базиран ICrA

Таблица 5.18: Резултати от ν -базиран ICrA – двойки в WPC и WD

Двойки критерии от ACO алгоритми	стойност на $\mu_{C,C'}$	стойност на $\nu_{C,C'}$
$ACOs8 - ACOm8$	0.83	0.25
$ACOs5 - ACOm8$	0.71	0.25
$ACOs6 - ACOm8$	0.71	0.25
$ACOs7 - ACOm8$	0.71	0.25
$ACOu3 - ACOm3$	0.81	0.14

рии са в WPC. Останалите двойки критерии са в слабо несъгласие (WD) или в несъгласие (D).

В Таблица 5.20 са представени двойките критерии, които са в WPC при μ -базиран ICrA. В случая на ν -базиран ICrA тези двойки критерии са в WD и D (за двойките критерии $ACOs4 - ACOs10$ и $ACOu1 - ACOu2$).

В Таблица 5.21 са представени само двойките критерии които са в WPC (ν -базиран ICrA). Ниската стойност на $\mu_{C,C'}$ показва, че едноцелевите ACO алгоритми със сумиране ($ACOs$) са по-малко чувствителни към броя на използваните мравки, в сравнение с едноцелевите ACO алгоритми с умножение ($ACOu$) и многоцелевите ACO алгоритми ($ACOm$).

5.7.3 Резултати от прилагането на Балансиран ICrA

Всички резултати, на основата на прилагането на Балансиран ICrA, са визуализирани на Фигура 5.3.

В сравнение с резултатите от ν базиран ICrA, в този случай корелацията между двойките критерии запазва стойностите от $\mu_{C,C'}$. Стойностите $\nu_{C,C'}$ се намират между стойността на $\pi_{C,C'}$ и $\mu_{C,C'}$.

Отново сравняваме получените резултати с резултатите от секция 5.7.1. Таблица 5.22 показва стойностите $\mu_{C,C'}$ и $\nu_{C,C'}$ на двойките критерии, разгледани в μ -базиран ICrA. В сравнение с резултатите от секция 5.7.1, разглежданите двойки критерии не са в SPC, те са в WD ($ACOs5 - ACOm8$, $ACOs6 - ACOm8$ and $ACOs7 - ACOm8$) или в WPC ($ACOs8 - ACOm8$ и $ACOu3 - ACOm3$).

В Таблица 5.23 са показани резултатите от алгоритми с различни

5.7. Прилагане на различни Интеркритериални анализи към АСО алгоритъма115

Таблица 5.19: Резултати от ν -базиран ICrA – двойки в WPC, WD и D

Двойки критерии от АСО алгоритми	стойност на $\mu_{C,C'}$	стойност на $\nu_{C,C'}$	Двойки критерии от АСО алгоритми	стойност на $\mu_{C,C'}$	стойност на $\nu_{C,C'}$
<i>ACOu6 – ACOm3</i>	0.78	0.14	<i>ACOs8 – ACOm9</i>	0.67	0.21
<i>ACOs9 – ACOm8</i>	0.72	0.20	<i>ACOu2 – ACOm2</i>	0.73	0.15
<i>ACOu4 – ACOm3</i>	0.79	0.13	<i>ACOu4 – ACOs1</i>	0.77	0.10
<i>ACOs8 – ACOm5</i>	0.71	0.20	<i>ACOu6 – ACOs9</i>	0.73	0.15
<i>ACOu1 – ACOm1</i>	0.71	0.21	<i>ACOs9 – ACOm3</i>	0.74	0.14
<i>ACOu3 – ACOs4</i>	0.80	0.12	<i>ACOu2 – ACOm4</i>	0.76	0.11
<i>ACOu4 – ACOm4</i>	0.78	0.13	<i>ACOu5 – ACOs1</i>	0.77	0.10
<i>ACOu5 – ACOm3</i>	0.73	0.11	<i>ACOs9 – ACOm7</i>	0.65	0.21
<i>ACOu5 – ACOm4</i>	0.78	0.13	<i>ACOu6 – ACOs10</i>	0.69	0.17
<i>ACOu3 – ACOm4</i>	0.79	0.13	<i>ACOs7 – ACOm6</i>	0.61	0.25
<i>ACOs7 – ACOm5</i>	0.70	0.20	<i>ACOs8 – ACOm10</i>	0.61	0.25
<i>ACOs9 – ACOm9</i>	0.70	0.21	<i>ACOu7 – ACOs10</i>	0.68	0.18
<i>ACOu10 – ACOm3</i>	0.76	0.14	<i>ACOs7 – ACOm10</i>	0.60	0.26
<i>ACOu7 – ACOm3</i>	0.76	0.14	<i>ACOu10 – ACOs9</i>	0.71	0.15
<i>ACOu8 – ACOm3</i>	0.76	0.14	<i>ACOu5 – ACOs10</i>	0.70	0.16
<i>ACOu9 – ACOm3</i>	0.76	0.14	<i>ACOu7 – ACOs9</i>	0.71	0.15
<i>ACOs7 – ACOm7</i>	0.65	0.25	<i>ACOu8 – ACOs9</i>	0.71	0.15
<i>ACOu1 – ACOm4</i>	0.71	0.19	<i>ACOu9 – ACOs9</i>	0.71	0.15
<i>ACOs7 – ACOm9</i>	0.67	0.23	<i>ACOu1 – ACOs4</i>	0.70	0.15
<i>ACOs8 – ACOm7</i>	0.65	0.24	<i>ACOu2 – ACOm5</i>	0.73	0.12
<i>ACOu4 – ACOs4</i>	0.78	0.11	<i>ACOu2 – ACOs10</i>	0.70	0.16
<i>ACOu2 – ACOm3</i>	0.77	0.11	<i>ACOu4 – ACOs10</i>	0.70	0.16
<i>ACOu3 – ACOs1</i>	0.78	0.10	<i>ACOs9 – ACOm6</i>	0.63	0.22
<i>ACOu5 – ACOs4</i>	0.78	0.10	<i>ACOu2 – ACOs4</i>	0.76	0.09
<i>ACOs8 – ACOm6</i>	0.63	0.25	<i>ACOu3 – ACOs9</i>	0.73	0.12
<i>ACOs9 – ACOm5</i>	0.71	0.17	<i>ACOu6 – ACOs1</i>	0.75	0.10

Таблица 5.20: Резултати от ν -базиран ICrA – двойки в WD и D

Двойки критерии от ACO алгоритми	стойност на $\mu_{C,C'}$	стойност на $\nu_{C,C'}$
$ACO_{s4} - ACO_{s10}$	0.66	0.11
$ACO_{s4} - ACO_{s5}$	0.69	0.13
$ACO_{s4} - ACO_{s6}$	0.69	0.13
$ACO_{s4} - ACO_{s7}$	0.69	0.13
$ACO_{s4} - ACO_{s8}$	0.69	0.12
$ACO_{s4} - ACO_{s9}$	0.73	0.12
$ACOU_1 - ACOU_2$	0.66	0.12
$ACOU_1 - ACOU_3$	0.68	0.13
$ACOU_1 - ACOU_4$	0.68	0.14
$ACOU_1 - ACOU_5$	0.68	0.13

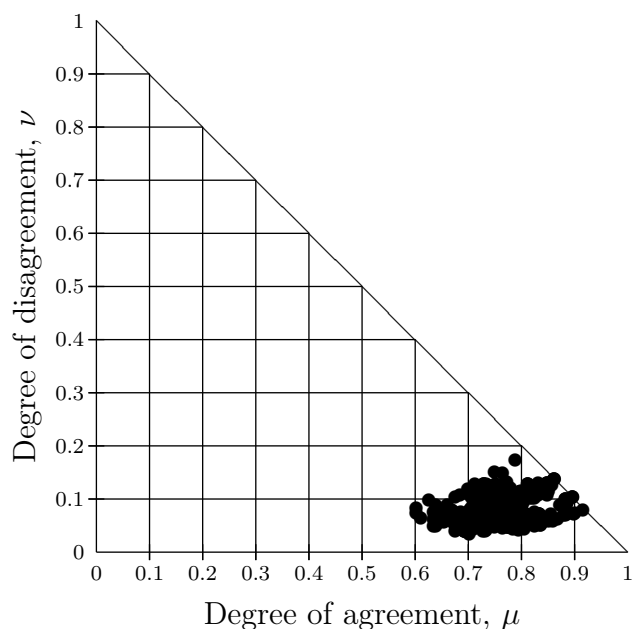
Таблица 5.21: Резултати от ν -базиран ICrA – двойки в WPC

Двойки критерии от ACO алгоритми	стойности на $\mu_{C,C'}$	стойности на $\nu_{C,C'}$
$ACOU_4 - ACOU_5$	0.84	0.16
$ACOU_3 - ACOU_4$	0.83	0.15
$ACOU_3 - ACOU_5$	0.83	0.14
$ACOU_2 - ACOU_4$	0.81	0.14
$ACOU_2 - ACOU_5$	0.81	0.14
$ACOU_3 - ACO_{m3}$	0.81	0.14
$ACOU_2 - ACOU_3$	0.81	0.13

целеви функции разгледани в секция 5.7.1 и съответните стойности на $\nu_{C,C'}$.

Двойките критерии, които са в WC и в PC са представени с удебелен шрифт. В сравнение с ν -базиран ICrA с 34 двойки критерии в WD и D, тук само 6 двойки критерии са в WD. В случая на μ -базиран ICrA всички показани в Таблица 5.23 двойки критерии са в PC. Така, Балансирани-

5.7. Прилагане на различни Интеркритериални анализи към АСО алгоритъма117



Фигура 5.3: Резултати от прилагането на Балансиран ІСгА

Таблица 5.22: Резултати от Балансиран ІСгА – двойки е WPC и WD

Двойки критерии от АСО алгоритми	стойности на $\mu_{C,C'}$	стойности на $\nu_{C,C'}$
$ACO_{s8} - ACO_{m8}$	0.86	0.13
$ACO_{s5} - ACO_{m8}$	0.84	0.12
$ACO_{s6} - ACO_{m8}$	0.84	0.12
$ACO_{s7} - ACO_{m8}$	0.84	0.12
$ACO_{u3} - ACO_{m3}$	0.88	0.07

ят ІСгА показва подобно поведение като μ -базиран ІСгА, но взема под внимание повтарящите се числови резултати. Стойностите $\pi_{C,C'}$ варират от $\pi_{C,C'} = 0.02$ до $\pi_{C,C'} = 0.33$.

В Таблица 5.24 са представени двойките критерии разгледани в случая μ -базиран ІСгА с WPC. В случая на Балансиран ІСгА, тези двойки критерии са в WD с изключение на три двойки, $ACO_{s4} - ACO_{s5}$,

Таблица 5.23: Резултати от Балансиран ICGA – двойки в WPC, PC и WD

Двойки критерии на ACO алгоритмис	стойн. на $\mu_{C,C'}$	стойн. на $\nu_{C,C'}$	Двойки критерии на ACO алгоритмите	стойн. на $\mu_{C,C'}$	стойн. на $\nu_{C,C'}$
<i>ACOu6 – ACOm3</i>	0.85	0.07	<i>ACOs8 – ACOm9</i>	0.67	0.21
<i>ACOs9 – ACOm8</i>	0.76	0.07	<i>ACOu2 – ACOm2</i>	0.73	0.15
<i>ACOu4 – ACOm3</i>	0.82	0.10	<i>ACOu4 – ACOs1</i>	0.83	0.05
<i>ACOs8 – ACOm5</i>	0.82	0.10	<i>ACOu6 – ACOs9</i>	0.81	0.07
<i>ACOu1 – ACOm1</i>	0.81	0.11	<i>ACOs9 – ACOm3</i>	0.81	0.07
<i>ACOu3 – ACOs4</i>	0.86	0.06	<i>ACOu2 – ACOm4</i>	0.82	0.06
<i>ACOu4 – ACOm4</i>	0.85	0.07	<i>ACOu5 – ACOs1</i>	0.82	0.05
<i>ACOu5 – ACOm3</i>	0.85	0.06	<i>ACOs9 – ACOm7</i>	0.76	0.11
<i>ACOu5 – ACOm4</i>	0.85	0.06	<i>ACOu6 – ACOs10</i>	0.78	0.09
<i>ACOu3 – ACOm4</i>	0.85	0.06	<i>ACOs7 – ACOm6</i>	0.74	0.12
<i>ACOs7 – ACOm5</i>	0.80	0.10	<i>ACOs8 – ACOm10</i>	0.73	0.13
<i>ACOs9 – ACOm9</i>	0.80	0.10	<i>ACOu7 – ACOs10</i>	0.77	0.09
<i>ACOu10 – ACOm3</i>	0.83	0.07	<i>ACOs7 – ACOm10</i>	0.73	0.13
<i>ACOu7 – ACOm3</i>	0.83	0.07	<i>ACOu10 – ACOs9</i>	0.78	0.07
<i>ACOu8 – ACOm3</i>	0.83	0.07	<i>ACOu5 – ACOs10</i>	0.78	0.08
<i>ACOu9 – ACOm3</i>	0.83	0.07	<i>ACOu7 – ACOs9</i>	0.78	0.07
<i>ACOs7 – ACOm7</i>	0.77	0.13	<i>ACOu8 – ACOs9</i>	0.78	0.07
<i>ACOu1 – ACOm4</i>	0.80	0.09	<i>ACOu9 – ACOs9</i>	0.78	0.07
<i>ACOs7 – ACOm9</i>	0.78	0.11	<i>ACOu1 – ACOs4</i>	0.78	0.08
<i>ACOs8 – ACOm7</i>	0.77	0.12	<i>ACOu2 – ACOm5</i>	0.79	0.06
<i>ACOu4 – ACOs4</i>	0.84	0.05	<i>ACOu2 – ACOs10</i>	0.77	0.08
<i>ACOu2 – ACOm3</i>	0.83	0.06	<i>ACOu4 – ACOs10</i>	0.77	0.08
<i>ACOu3 – ACOs1</i>	0.84	0.05	<i>ACOs9 – ACOm6</i>	0.74	0.11
<i>ACOu5 – ACOs4</i>	0.84	0.05	<i>ACOu2 – ACOs4</i>	0.81	0.04
<i>ACOs8 – ACOm6</i>	0.76	0.13	<i>ACOu3 – ACOs9</i>	0.79	0.06
<i>ACOs9 – ACOm5</i>	0.80	0.08	<i>ACOu6 – ACOs1</i>	0.80	0.05

ACOs4 – ACOs6 и *ACOs4 – ACOs7*. Тези двойки критерии са в WPC. В сравнение със случая на μ -базиран ICGA, всички двойки критерии в

5.7. Прилагане на различни Интеркритериални анализи към АСО алгоритъма¹¹⁹

Таблица 5.24 са в WPC. При прилагането на Балансиран ICrA алгоритъм е показано, че само АСОs запазват РС между някои от двойките критерии. Можем да заключим, че АСОs алгоритмите са по-малко чувствителни към броя на използваните мравки, в сравнение с АСОu и АСОm алгоритмите.

Таблица 5.24: Резултати от Балансиран ICrA – двойки в WD и WPC

Двойки критерии от АСО алгоритми	стойност на $\mu_{C,C'}$	стойност на $\nu_{C,C'}$
<i>АСОs4 – АСОs10</i>	0.71	0.06
<i>АСОs4 – АСОs5</i>	0.76	0.06
<i>АСОs4 – АСОs6</i>	0.76	0.06
<i>АСОs4 – АСОs7</i>	0.76	0.06
<i>АСОs4 – АСОs8</i>	0.75	0.06
<i>АСОs4 – АСОs9</i>	0.79	0.06
<i>АСОu1 – АСОu2</i>	0.72	0.06
<i>АСОu1 – АСОu3</i>	0.75	0.06
<i>АСОu1 – АСОu4</i>	0.75	0.07
<i>АСОu1 – АСОu5</i>	0.75	0.07

В случая на Балансиран ICrA алгоритъм има няколко двойки критерии с една и съща целева функция, които са в РС. Резултатите са показани в Таблица 5.25.

В случая на ν -базиран ICrA тези двойки критерии са в WPC, а в случая на μ -базиран ICrA те са в РС и дори в SPC. Ако корелацията между две двойки критерии е различна при прилагането на различни ICrA варианти, това означава, че това е наистина силна зависимост на двойката.

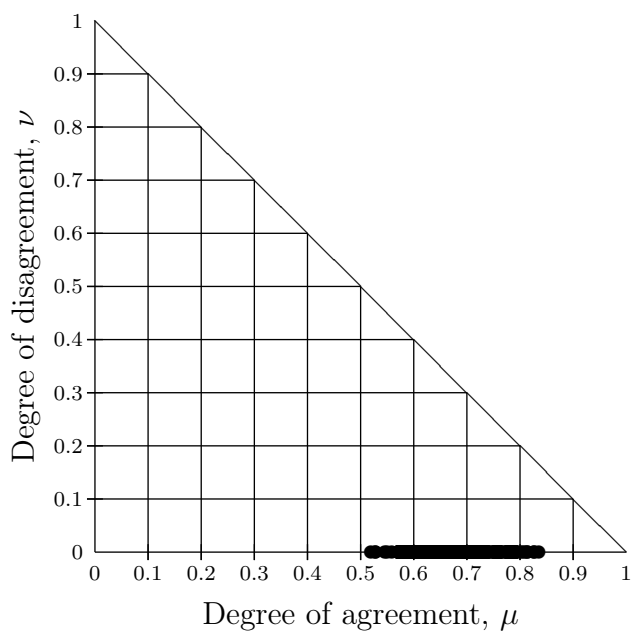
5.7.4 Резултати от прилагането на Безпристрастен ICrA

Резултатите от прилагането на Безпристрастен ICrA са визуализирани на Фигура 5.4.

В този случай, получените $\mu_{C,C'}$ стойности са същите като тези получени в случая на ν -базиран ICrA (секция 5.7.2). Разликата в резултатите

Таблица 5.25: Резултати от Балансиран ICrA – двойки в РС

Двойки критерии от АСО алгоритми	стойност на $\mu_{C,C'}$	стойност на $\nu_{C,C'}$
$ACOu4 - ACOu5$	0.92	0.08
$ACOu3 - ACOu4$	0.90	0.07
$ACOu3 - ACOu5$	0.90	0.07
$ACOu2 - ACOu4$	0.89	0.07
$ACOu2 - ACOu5$	0.88	0.07
$ACOu3 - ACOm3$	0.88	0.07
$ACOu2 - ACOu3$	0.87	0.06



Фигура 5.4: ICrA резултати при Безпристрастен ICrA

е в оценката на стойностите $\nu_{C,C'}$. Всички $\nu_{C,C'}$ стойности са нула, което води до не нулеви стойности на степента на неопределеност $\pi_{C,C'}$. Стойностите $\pi_{C,C'}$ варират от $\pi_{C,C'} = 0.16$ до $\pi_{C,C'} = 0.48$. В случай на повтарящи се стойности във входната индексна матрица, Безпристраст-

ният ICGA увеличава степента на неопределеност $\pi_{C,C'}$. Това е причината за нулевите оценки за стойностите $\nu_{C,C'}$.

В Таблица 5.26 е направено сравнение на резултатите получени при различни целеви функции (двойките критерии в SPC в случая на μ -базиран ICGA), с резултатите получени в случая на ν -базиран ICGA алгоритъм.

Таблица 5.26: Резултати от ν -базиран и Безпристрастен ICGA

Двойки критерии от ACO алгоритми	ν -базиран ICGA			Безпристрастен ICGA		
	$\mu_{C,C'}$	$\nu_{C,C'}$	$\pi_{C,C'}$	$\mu_{C,C'}$	$\nu_{C,C'}$	$\pi_{C,C'}$
$ACOs8 - ACOm8$	0.83	0.25	0.08	0.83	0.00	0.17
$ACOs5 - ACOm8$	0.71	0.25	0.04	0.71	0.00	0.29
$ACOs6 - ACOm8$	0.71	0.25	0.04	0.71	0.00	0.29
$ACOs7 - ACOm8$	0.71	0.25	0.04	0.71	0.00	0.29
$ACOu3 - ACOm3$	0.81	0.14	0.05	0.81	0.00	0.19

Както може да се види, разликата между различните варианти е интерпретацията на повтарящите се числови резултати във входната индексна матрица. В случая на μ -базиран ICGA алгоритъм, повтарящите се числови стойности се интерпретират като степен на несъгласие, а в случая на Безпристрастен ICGA алгоритъм се интерпретират като степен на неопределеност.

5.8 Заключение

В тази глава са предложени многоцелеви както и няколко варианта на едноцелеви ACO алгоритми за решаване на задачата за позициониране на безжична сензорна мрежа. Задачата се състои в предоставянето на броя и местоположението на сензорните възли, които образуват WSN с пълно покритие на сензорното поле и минимален брой сензори и енергия. Сравняваме постигнатите резултати с най-добрите намерени резултати за тази задача в литературата. Показваме, че предложените ACO алгоритми се представят по-добре от съществуващите. Едноцелевият алгоритъм с претеглена сума постига подобни решения като многоцелевия ACO алгоритъм. Изследвахме най-добрата стойност на теглата и установихме,

че е по-добре, когато броят на сензорите има по-голямо влияние върху целевата функция от енергията. Изследвали сме влиянието на броя на мравките върху работата на АСО алгоритъма, приложен към задачата за позициониране на безжична сензорна мрежа. По-малкият брой мравки води до по-кратко време на работа и свежда до минимум използването на паметта, което е важно за сложни / големи случаи. Променяхме броя на мравките, като същевременно фиксирахме броя на итерациите. Освен това, ние включихме концепцията за Разширен фронт, като допълнителен инструмент за сравняване на фронтове на Парето, които не доминират един върху друг.

Интеркритериалният анализ е мощен инструмент за изследване на връзките между различни обекти. Изследвахме три варианта на алгоритъма АСО, приложен към WSN задачата. Всеки вариант се тества с различен брой мравки, между 1 и 10. Търсим връзката между вариантите на АСО и броя на мравките. WSN задачата е многоцелева. Когато се преобразува в едноцелева, чрез сумиране на двете целеви функции, алгоритъмът е по-малко чувствителен към броя на използваните мравки. Когато задачата е решена като многоцелева, ние наблюдаваме по-голяма разлика в производителността на алгоритъма според броя на мравките. Има по-голямо сходство между поведението на двата едноцелеви варианта, отколкото между някои от едноцелевите и многоцелевите варианти. Приложен е Интеркритериален анализ, за да се изследват връзките между разглежданите АСО алгоритми. Анализирано е влиянието на броя на мравките. Приложени са четири различни варианта на Интеркритериален анализ, μ -базиран, Балансиран, ν -базиран и Безпристрастен. Получените резултати показват, че при корелация между дадени АСО алгоритми тази зависимост се появява независимо от използвания от нас ICA алгоритъм.

Глава 6

Наемане на работна сила

Управлението на човешките ресурси е една от основните части при организацията на производството. Планирането на работната сила е важна индустриална задача. Тя включва няколко нива на трудност: подбор и възлагане. Първо наетите трябва да бъдат избрани от голям брой налични работници. След това избраните работници трябва да бъдат разпределени към задачите, които ще се изпълняват. Целта е да се изпълнят изискванията за работа с минимални разходи за възлагане.

Това е една трудна задача за вземане на решение, която не би могла да бъде решена с използването на точни методи в реалистични случаи. В [73, 99] се изучава детерминистичен вариант на задачата за наемане на работна сила. В [73] моделите за планиране на работната сила, които съдържат нелинейни модели на човешко обучение, са преформулирани в термините на целочисленото програмиране. Авторите показват, че е много по-лесно задачата да бъде решена когато е дефинирана в термините на целочисленото програмиране, отколкото в термините на нелинейното програмиране. Модел на планиране на работна сила е разгледан и в [99]. Моделът включва различията на работниците, както и възможността за обучение и надграждане на работниците. Вариант на задачата със случайни поръчки е предложен в [16, 102]. В [16] е предложен двуфазен стохастичен метод за планиране и разпределяне на работници с няколко квалификации, като се има предвид среда, предлагаща разнообразни услуги със случайни поръчки. В някои варианти на задачата са включена възможност за неопределеност [74, 89, 98, 122, 128]. В този случай съответната целева функция и дадените ограничения се преобразуват в ясни еквиваленти и след това моделът се решава чрез традиционни методи

[98] или разглежданият модел с неопределености се трансформира в еквивалентна детерминирана форма, както е показано в [122]. Повечето от авторите опростяват задачата, като пропускат някои от ограниченията. Някои конвенционални методи могат да бъдат приложени по отношение на проблема с планирането на работната сила като смесено линейно програмиране [29], метод на декомпозиция [102]. За по-сложните проблеми с нелинейното планиране на работната сила изтъкнатите методи не са приложими. Има някои приложения на евристични методи, включително генетични алгоритми [3, 88], меметични алгоритми [114] и др. Обикновено метаевристичните алгоритми се прилагат, когато задачата е сложна и има строги ограничения [76, 125]. В тази работа е предложен алгоритъм на базата на метода на мравките за решаване на задачата за наемане на работна сила [47, 107, 64, 109]. Чрез много приложения е показано, че АСО алгоритъмът е много ефективен при решаването на различни сложни задачи за оптимизация [44, 70].

6.1 Формулиране на Задачата

В тази глава се използва дефиницията на задачата за наемане на работна сила, дадена от Glover [68]. Дадено е множество от задания $J = \{1, \dots, m\}$, които трябва да бъдат изпълнени за фиксиран период от време. Всяко задание j изисква d_j часа за да бъде завършено. $I = \{1, \dots, n\}$ е множеството от налични работници. Всеки работник трябва да работи върху всяко едно от назначените му задания минимум h_{min} часа, за да работи ефективно. Работник i е наличен s_i часа. Максималният брой задания, назначени на един работник е s_i часа. Работниците имат различни умения, множеството A_i показва за кои задания работник i е квалифициран. Максималният брой работници, който може да бъде назначен за планирания период е t или най-много t работника могат да бъдат избрани от множеството I на наличните работници и назначените работници трябва да бъдат способни да приключат всички задания. Целта е да се намери валидно решение, което да има минимална цена за назначаване.

Всеки работник i и задание j са свързани с цена c_{ij} за назначаване на работника за изпълнение на заданието. Математическото представяне на задачата за наемане на работна сила е както следва:

$$x_{ij} = \begin{cases} 1 & \text{ако работник } i \text{ е назначен за задание } j \\ 0 & \text{в противен случай} \end{cases}$$

$$y_i = \begin{cases} 1 & \text{ако работник } i \text{ е избран} \\ 0 & \text{в противен случай} \end{cases}$$

z_{ij} = брой часов за които работник i
е назначен за задание j

Q_j = множество от работници с квалификация за задание j

$$\text{Minimize } \sum_{i \in I} \sum_{j \in A_i} c_{ij} \cdot x_{ij} \quad (6.1)$$

При условие:

$$\sum_{j \in A_i} z_{ij} \leq s_i \cdot y_i \quad i \in I \quad (6.2)$$

$$\sum_{i \in Q_j} z_{ij} \geq d_j \quad j \in J \quad (6.3)$$

$$\sum_{j \in A_i} x_{ij} \leq j_{max} \cdot y_i \quad i \in I \quad (6.4)$$

$$h_{min} \cdot x_{ij} \leq z_{ij} \leq s_i \cdot x_{ij} \quad i \in I, j \in A_i \quad (6.5)$$

$$\sum_{i \in I} y_i \leq t \quad (6.6)$$

$$x_{ij} \in \{0, 1\} \quad i \in I, j \in A_i$$

$$y_i \in \{0, 1\} \quad i \in I$$

$$z_{ij} \geq 0 \quad i \in I, j \in A_i$$

Цената за наемане е целевата функция за тази задача. Броят на часовете, които един работник може да работи е ограничен (неравенство 6.2). Всички задания трябва да бъдат изцяло завършени (неравенство 6.3). Броят на заданията, върху които един работник може да работи е

ограничен (неравенство 6.4). За да може един работник да работи ефективно, той трябва да изпълнява дадено задание не по-малко от фиксиран минимален брой часове (неравенство 6.5). Броят на наетите работници, работещи едновременно е ограничен (неравенство 6.6).

Този модел може да се използва за оптимизирането на различни целеви функции. В тази работа целта е минимизирането на общата цена за наемане. Ако \tilde{c}_{ij} е цената на работник i да работи върху задание j в продължение на един час, тогава целевата функция може да минимизира цената за изпълнението на всички задания на часова база.

$$f(x) = \text{Min} \sum_{i \in I} \sum_{j \in A_i} \tilde{c}_{ij} \cdot x_{ij} \cdot z_{ij} \quad (6.7)$$

Някои работници могат да имат предпочитания да работят върху част от заданията, за които са най-квалифицирани. Тогава целевата функция може да бъде максимално удовлетворяване на предпочитанията на работниците или да се намери максимумът от минималната стойност на предпочитанията на подбраните работници.

Както беше споменато, в тази глава ще се минимизира цената за наемане (уравнение 6.1). Задачата за наемане/планиране на работна сила е трудна оптимизационна задача поради строгите ограничения, влияние оказва и връзката между параметрите h_{min} и d_j . Когато задачата е структурирана (d_j е кратно на h_{min}) е по-лесно да бъде намерено валидно решение, отколкото при неструктурираните задачи (d_j и h_{min} не са свързани).

6.2 АСО Алгоритъм за Наемане на Работна Сила

Една от основните точки на метода на мравките е правилното представяне на решаваната задача чрез граф. В разглеждания случай задачата е представена чрез тримерен граф, където връх (i, j, z) означава работник i да бъде нает да работи по задание j за време z . В началото на всяка итерация, всяка мравка започва да строи решение от случаен връх на графа на задачата. За всяка една от мравките се генерират три случайни числа. Първото случайно число е в интервала $[0, \dots, n]$ и съответства на работника, който се наема. Второто случайно число е в интервала

$[0, \dots, m]$ и съответства на заданието, върху което този работник трябва да работи. Третото случайно число е в интервала $[h_{min}, \dots, \min\{d_j, s_i\}]$ и съответства на броя часове, които работник i е нает да работи върху задание j . След това се прилага вероятностното правило за преход за да се добави следващ връх в частичното решение. Този процес се повтаря, докато решението бъде построено.

Евристичната информация се изчислява по следната формула:

$$\eta_{ijl} = \begin{cases} l/c_{ij} & l = z_{ij} \\ 0 & otherwise \end{cases} \quad (6.8)$$

С тази евристична информация насърчаваме назначаването на поевтините работници за възможно по-дълго време. Мравката добавя към своето решение, върхът с най-голяма вероятност. Когато една мравка има няколко, равно вероятни възможности за следващ връх, тя избира по случаен начин между тях.

Когато се добавя нов връх в частичното решение, се вземат в предвид ограниченията на задачата или по-точно: колко работника вече са наети; за колко часа е нает всеки един от работниците; за всяко от заданията за колко от необходимите за изпълнението му часове вече има назначени работници. Ако добавянето на някой връх нарушава някое от тези ограничения, то вероятността за добавянето му се поставя да бъде 0. Ако е невъзможно в решението да бъде добавен нов връх от графа на задачата (вероятността за всички върхове е 0), конструирането на решението спира. Когато конструираното решение е валидно решение, то стойността на целевата функция е равна на сумата от цената за наемане на наетите работници. Ако конструираното решение е невалидно решение, то стойността на целевата функция се поставя да бъде -1 .

Само мравките, които построят валидно решение имат право да добавя феромон на елементите му. Феромонът, който се добавя е равен на реципрочната стойност на целевата функция, отговаряща на това решение.

$$\Delta\tau_{i,j} = \frac{\rho - 1}{f(x)} \quad (6.9)$$

Така, върховете от графа на задачата, които принадлежат на подобри решения (по-малка стойност на целевата функция) получават повече феромон от останалите върхове и стават по желани на следващата итерация.

В края на всяка итерация се прави сравнение между най-доброто намерено решение през текущата итерация и най-доброто намерено решение от началото на алгоритъма (глобално най-добро). Ако най-доброто решение от текущата итерация е по-добро от глобално най-доброто решение, то най-доброто решение от текущата итерация става най-добро глобално решение. Като условие за край на алгоритъма се използва фиксиран брой итерации.

Разработена е програма на езика C, която реализира алгоритъма. Използваме изкуствено генерираните тестови задачи, разгледани в [3]. Характеристиките на тези задачи са дадени в таблица 7.1.

Множеството от тестови примери включва десет структурирани и десет неструктурирани задачи. Структурираните задачи са с номера от $S01$ до $S10$. Неструктурираните задачи са с номера от $U01$ до $U10$.

Броят на итерациите е фиксиран на 100. Стойностите на управляващите параметри на предложеният АСО алгоритъм са показани в таблица 7.2. Тези стойности са получени експериментално, след няколко тествания на алгоритъма.

Алгоритъмът е стохастичен и от статистическа гледна точка трябва да се изпълни минимум 30 пъти, за да се гарантира стабилността на средните резултати. Извършени са 30 независими пускания на алгоритъма със всеки един от тестовите примери. След това е направен статистически анализ на резултатите, използвайки ANOVA тест. Тестът показва, че има значителна разлика между резултатите, постигнати чрез различни методи, или резултатите не са статистически еднакви.

Нека бъдат сравнени числените резултати постигнати от предложеният тук АСО алгоритъм и тези получени от генетичния алгоритъм (GA) и scatter search (SS) представени в [3]. Таблица 6.3 показва постигнатите резултати при структурирани тестови примери, докато таблица 6.4 показва постигнатите резултати при неструктурирани тестови примери.

Наблюдаваме, че АСО алгоритъмът превъзхожда другите два алгоритъма. АСО е конструктивен метод и когато графът на задачата и евристичната информация са подходящо подбрани и описват задачата по добър начин, това може много да помогне за по-доброто поведение на алгоритъма и от там за постигане на добри решения и добра производителност. Предложеният в тази работа граф на задачата има звездовидна структура. Всеки работник и всяко задание са свързани с няколко върха от графа, съответстващи на времето, за което работникът е назначен да изпълнява това задание. Предложената евристична информация стиму-

Таблица 6.1: Характеристики на тестовите примери

Параметри	Стойности
n	20
m	20
t	10
s_i	[50,70]
j_{max}	[3,5]
h_{min}	[10,15]

Таблица 6.2: АСО параметри

Параметри	Стойности
Брой итерации	100
ρ	0.5
τ_0	0.5
Брой мравки	20
a	1
b	1

Таблица 6.3: Средни резултати за структурираните примери

Пример	Стойност на целевата функция		
	SS	GA	ACO
<i>S01</i>	936	963	807
<i>S02</i>	952	994	818
<i>S03</i>	1095	1152	882
<i>S04</i>	1043	1201	849
<i>S05</i>	1099	1098	940
<i>S06</i>	1076	1193	869
<i>S07</i>	987	1086	812
<i>S08</i>	1293	1287	872
<i>S09</i>	1086	1107	793
<i>S10</i>	945	1086	825

Таблица 6.4: Средни резултати за неструктурираните примери

Пример	Стойност на целевата функция		
	SS	GA	ACO
<i>U01</i>	1586	1631	814
<i>U02</i>	1276	1264	845
<i>U03</i>	1502	1539	906
<i>U04</i>	1653	1603	869
<i>U05</i>	1287	1356	851
<i>U06</i>	1193	1205	873
<i>U07</i>	1328	1301	828
<i>U08</i>	1141	1106	801
<i>U09</i>	1055	1173	768
<i>U10</i>	1178	1214	818

Таблица 6.5: Параметри на алгоритъма от [3]

Параметри	Genetic algorithm
размер на популацията	400
crossover rate	0.8
mutation rate	0.2
Параметри	Scatter search
начална популация	15
reference set update and creation	8
subset generated	all 2-elements subsets
ρ_i	0.1

лира по-евтините работници да бъдат назначавани за възможно по-дълго време. Това е алчна (greedy) стратегия. След първата итерация нивото на феромона отразява опита на мравките по време на процеса на търсене, като по този начин влияе върху стратегията за търсене. Елементите на добрите решения натрупват повече феромон по време на изпълнението на алгоритъма от останалите и стават по-желани при следващите итерации.

Сега ще бъде направено сравнение между предложения ACO алгоритъм и други два метаевристични алгоритъма GA и SS [3]. Алгоритмите са изпълнени на сходни компютри. В таблица 6.5 са дадени параметрите на използваните GA и SS.

Средното време за изпълнение на всеки от алгоритмите е дадено в таблици 6.6 and 6.7. Средното време е изчислено върху 30 пускания на алгоритъма за всеки от тестовите примера.

Вижда се, че ACO алгоритъмът е по-бърз от GA и SS. По отношение на времето за изпълнение GA и SS сходно поведение. От резултатите представени в таблици 6.3, 6.4, 6.6 и 6.7 може да се направи заключението, че ACO алгоритъмът дава доста обещаващи резултати. Той постига по-добри решения за по-кратко време в сравнение с другите два алгоритъма, GA и SS. Ако сравним използваната памет, ACO използва по-малко памет от GA (популацията при GA е 400 индивида) и сходно количество памет с SS (размерът на първоначалната популация е 15 а референтното множество е 8 индивида) [3].

Таблица 6.6: Средно време за структурирани примери

Пример	Време, s		
	SS	GA	ACO
S01	72	61	26
S02	49	32	21
S03	114	111	22
S04	86	87	25
S05	43	40	21
S06	121	110	23
S07	52	49	23
S08	46	42	24
S09	70	67	20
S10	105	102	22

Таблица 6.7: Средно време за неструктурирани примери

Пример	Време, s		
	SS	GA	ACO
U01	102	95	22
U02	94	87	20
U03	58	51	20
U04	83	79	20
U05	62	57	23
U06	111	75	22
U07	80	79	21
U08	123	89	20
U09	75	72	26
U10	99	95	20

6.3 Хибриден АСО алгоритъм за наемане на работна сила

Основният принос в тази секция е хибридизацията на АСО алгоритъм с процедура за локално търсене. Основната цел на локалното търсене е да намали времето за намиране на най-доброто решение и евентуално да подобри намерените решения.

Процедурата за локално търсене е прилагана само върху намерените невалидни решения и то еднократно, независимо дали новото решение е валидно или не. По този начин предложената процедура за локално търсене не отнема много време. Задачата за наемане на работна сила има строги ограничения и част от мравките не успяват да намерят валидни решения. С предложената процедура за локално търсене се увеличава възможността за намиране на валидни решения и от там се увеличава възможността за подобряване на текущото решение.

Ако намереното от мравката решение не е валидно, то се премахват част от назначените работници, след което се назначават нови работници на тяхно място. Премахнатите работници биват избирани по случаен начин. Така полученото частично решение бива довършено с назначаване на нови работници следвайки правилата на АСО алгоритъма. АСО алгоритъмът е стохастичен и за това новопостроените решения с прилагането на локално търсене с голяма вероятност ще бъдат различни от изходните. Направено е сравнение между поведението на алгоритъма с локално търсене и без локално търсене. Използвани са тестовите примери от предходната секция.

Задачата за наемане на работна сила има много строги ограничения. Понякога само 2-3 мравки успяват да намерят валидни решения на итерация. Има итерации без нито едно валидно решение. Това усложнява процеса на търсене. Целта ни е да се намали броя на невалидните решения и с това да се увеличи възможността за намиране на добри решения. Това може да доведе до намаляване броя на итерациите за намиране на най-доброто решение. Предложената процедура за локално търсене не изисква много време, защото тя се прилага еднократно само върху невалидните решения, без значение дали новополученото решение е валидно или не. Наблюдава се, че след прилагане на процедурата за локално търсене на първата итерация, броят на невалидните решения на следващите итерации силно намалява. По този начин процедурата се прилага все по

рядко с увеличаване на броя на итерациите. Това е още една причина, предложеното локално търсене да не увеличава съществено времето за изпълнение на алгоритъма. Разгледали сме и сме сравнили 4 случая: АСО алгоритъм без локално търсене; локално търсене при което броят на премахнатите работници е една четвърт от общия брой назначени работници (АСО четвърт); локално търсене при което броят на премахнатите работници е половината от общия брой назначени работници (АСО половина); локално търсене при което се премахват всички назначени работници и решението се построява отново (АСО рестарт).

Извършени са 30 независими пускания на хибридният алгоритъм със всеки един от тестовите примери. След това е направен статистически анализ на резултатите, използвайки ANOVA тест. Тестът показва, че има значителна разлика между резултатите, постигнати чрез различни методи, или резултатите не са статистически еднакви.

Интересуваме се от броя итерации за намиране на най-доброто решение. Необходимият брой итерации може да се различава много за отделните тестови примери. За това ще се използва класиране (ранкиране) на алгоритмите. Вариантът на нашия хибриден алгоритъм е на първо място, ако постигне най-доброто решение с по-малък среден брой итерации върху 30 пускания на програмата, по отношение на останалите варианти, тогава му присвояваме ранг 1. Ако алгоритъмът е на второ място му присвояваме ранг 2 и му присвояваме 3 когато се класира на 3-то място. Присвояваме ранг 4 когато алгоритъмът се нуждае от най-много итерации за постигане на най-доброто решение. На някои варианти може да се присвои един и същи ранг, ако броят на итерациите за намиране на най-доброто решение е един и същ. Ранговете върху всичките 20 тестови примера се сумират, за да се получи рангът на хибридният алгоритъм.

В таблица 6.8 отчитаме постигнатото ранкиране на различните случаи на нашия хибриден алгоритъм. Както беше споменато и преди, с "АСО четвърт" ще назоваваме случаят, в който се премахва една четвърт от назначените работници. "АСО половина" е случаят, когато се премахват половината от назначените работници. "АСО рестарт" е случаят в който се премахват всички назначени работници. Това отговаря на рестартиране на процедурата за намиране на решение или решението да бъде конструирано отново.

Един от основните въпроси е колко от работниците, от невалидното решение, да бъдат премахнати, така че ново конструираното решение да бъде валидно и близко до оптималното (най-доброто). Ние изчисля-

Таблица 6.8: Ранкиране на хибридните АСО алгоритми

	АСО	АСО четвърт	АСО половина	АСО рестарт
първо място	4 пъти	4 пъти	8 пъти	8 пъти
второ място	4 пъти	4 пъти	7 пъти	6 пъти
трето място	8 пъти	6 пъти	4 пъти	3 пъти
четвърто място	4 пъти	6 пъти	1 пъти	3 пъти
ранк	52	54	38	41

ваме класирането по отношение на средния брой итерации, за да намерим най-доброто решение за 30 повторения на теста. Когато повече от половината от мравките намерят невалидни решения, отклонението от средното е по-голямо в сравнение с тестовете, когато повечето мравки постигат валидни решения. Таблица 6.8 показва, че процедурата за локално търсене намалява броя на итерациите, необходими за намиране на най-доброто решение, когато повече от половината от работниците на невалидното решение бъдат отстранени. Традиционният АСО алгоритъм и хибридният АСО с отстраняване на четвърт от работниците са 4 пъти на първо място. Това е в случаите когато или случайно алгоритъмът намери най-доброто решение при първата итерация, или всички мравки намират възможни решения. Наблюдаваме, че и двата случая са на трето и четвърто място 12 пъти. Това означава, че отстраняването на по-малко от половината от работниците не е достатъчно, за да се създаде валидно решение. АСО алгоритъмът с отстраняване на половината от работниците 15 пъти е на първо или второ място и само един път е на четвърто място, което означава, че се представя много по-добре от предишните два случая. Когато всички работници бъдат отстранени, постигнатото класиране е подобно на случая, когато половината от работниците са отстранени. Нека максималният брой на назначените работници е t . Така всяко едно от намерените решения съдържа приблизително t работници. Ако всички работници бъдат отстранени, мравката трябва да добави нови работници на тяхно място, чийто брой е около t . Когато половината от работниците бъдат отстранени, тогава мравката ще добави около $t/2$ нови работници. Времето за изчисление за премахване и добавяне на около $t/2$ работници е около два пъти по-малко, отколкото за премахване

Таблица 6.9: Сравняване на хибридните АСО алгоритми спрямо времето за изчисление

	АСО	АСО четвърт	АСО половина	АСО рестарт
първо място	4 пъти	3 пъти	10 пъти	6 пъти
второ място	7 пъти	5 пъти	5 пъти	5 пъти
трето място	8 пъти	4 пъти	3 пъти	4 пъти
четвърто място	1 пъти	8 пъти	2 пъти	5 пъти
средно време	82.244 с.	93.98 с.	79.63 с.	103.012 с

не и добавяне на около t работници. По този начин можем да заключим, че процедурата за локално търсене с отстраняване на половината от работниците е по-добра от другите случаи.

Друг начин за сравнение е времето за изчисление. За всеки тестови пример и за всеки един от разглежданите случаи се изчислява средно време за постигане на най-добрия резултат на базата на 30 пускания на програмата. В Таблица 6.9 е направена подобна класификация като в Таблица 6.8, но тя е по отношение на времето за изчисляване вместо по отношение на броя на итерациите. След това сумираме средното време за намиране на най-доброто решение за всеки случай по отношение на всички тестови примери. Класирането в Таблица 6.9 според времето за изчисление е сходно с това в Таблица 6.8, според броя на използваните итерации. Най-добри резултати са постигнати, когато бъдат отстранени половината от назначените работници в процедурата за локално търсене, а най-лоши резултати се постигат, когато се отстранява една четвърт от назначените работници. Процедурата за локално търсене с отстраняване на половината от работниците е на първо място 10 пъти, а на четвърто място само 2 пъти. Процедурата за локално търсене с отстраняване на една четвърт от работниците е на първо място 3 пъти и на четвърто място 8 пъти. Що се отнася до времето за изчисляване, процедурата за локално търсене с отстраняване на половината от работниците, отново е най-добра, най-лоша е процедурата за локално търсене с отстраняване на всички работници. Реконструирането на решение от самото начало отнема повече време, отколкото реконструкцията на частично решение, поради което АСО алгоритъмът с локална процедура за търсене, премах-

6.4. Влияние на АСО параметрите върху ефективността на алгоритъма¹³⁷

Вайки всички работници, се представя най-лошо. Резултатите от таблица 6.9 показват, че премахването само на една четвърт от работниците от решението не е достатъчно за изграждането на добро решение и отнема много време в сравнение с традиционния АСО алгоритъм. Според двата вида сравнение, времето за класиране и изчисляване АСО алгоритъмът с локална процедура за търсене, премахвайки половината от назначените работници, се представя най-добре.

6.4 Влияние на АСО параметрите върху ефективността на алгоритъма

В този раздел анализираме работата на АСО по отношение на броя на мравките и качеството на постигнатите решения. Използваме същите изкуствено генерирани тестови примери, разгледани в [3]. Ако броят на мравките на АСО алгоритъма се увеличи, времето за изчисляване и използваната памет се увеличават пропорционално. Ако броят на итерациите се увеличи, се увеличава само времето за изчисляване. Ако времето за изчисляване е фиксирано и ние променяме само броя на мравките, това означава, че променяме и броя на итерациите, но в обратна посока, или ако времето е фиксирано, това е еквивалентно на фиксирането на производението на броя на мравките и броя на итерации.

Прилагаме брой мравки от множеството $\{5, 10, 20, 40\}$ и съответно брой итерации от множеството $\{400, 200, 100, 50\}$. Поради стохастичния характер на алгоритъма изпълняваме АСО алгоритъма за всичките 20 тестови примера с всеки един от четирите варианта на АСО алгоритъма (АСО 5×400 , АСО 10×200 , АСО 20×100 и АСО 40×50) 30 пъти. Търсим максималния брой итерации в рамките на фиксираното време за изчисления, което е необходимо, за да намерим най-доброто решение. Сравняваме производението мажду броя на мравките и броя на необходимите итерации. Стойностите на параметрите за нашия АСО алгоритъм са дадени в Таблица 6.10.

Таблицы 6.11 и 6.12 показват производението между броя на мравките и броя на итерациите, които са използвани за намирането на най-доброто решение. Когато производението е едно и също за различни АСО алгоритми (с различен брой мравки), за най-добър се смята този резултат, получен с по-малък брой мравки, тъй като в този случай използваната

Таблица 6.10: АСО параметри

Параметри	Стойности
Брой итерации	400, 200, 100, 50
ρ	0.5
τ_0	0.5
Брой мравки	5, 10, 20, 40
a	1
b	1

памет е по-малко. Най -добрите резултати са показани с удебелен шрифт.

Нека обсъдим резултатите, отчетени в таблици 6.11 и 6.12. По отношение на структурираните тестови примери, за 8 от тях най -доброто време за изпълнение е, когато броят на мравките е 5. Само за две от тестовите задачи ($S01$ и $S05$) времето за изпълнение е по-добро съответно за 40 и 10 мравки, но резултатът е близо до този, постигнат с 5 мравки. По отношение на неструктурирани тестови задачи, за 6 от тях най -доброто време за изпълнение се постига, когато броят на използваните мравки е 5. За три тестови задачи ($U03$, $U05$ и $U06$) най -добри резултати се постигат с помощта на АСО с 20 мравки. За тест $U05$, резултатът е доста близък до резултата с 5 мравки. Само за тест $U03$ разликата с резултатите при използване на 5 мравки е съществена. По този начин можем да заключим, че за този проблем най-доброто поведение на алгоритъма, използващо по-малко изчислителни ресурси е, когато броят на използваните мравки е 5.

6.5 ИнтерКритериален Анализ

В този раздел използваме ИнтерКритериален Анализ (ICrA), за да получим някои допълнителни познания за разглежданите четири варианта на АСО алгоритъма. Въз основа на резултатите в Таблицы 6.11 и 6.12 конструираме следната индексна матрица Таблица 6.10:

От Таблица 6.10 може да се види, че ICrA обектите ($S01$, $S02$, ..., $S10$, $U01$, ..., $U05$) са различните тестови примери, а ICrA критериите ($АСО_{5 \times 400}$, $АСО_{10 \times 200}$, $АСО_{20 \times 100}$ и $АСО_{40 \times 50}$) са АСО алгоритмите с

Таблица 6.11: Резултати за структурирани примери

Примери	Произведение мравка × итерации			
	ACO _{5×400}	ACO _{10×200}	ACO _{20×100}	ACO _{40×50}
S01	195	200	260	120
S02	195	330	340	640
S03	475	490	580	1160
S04	1540	1540	1540	1560
S05	415	320	420	920
S06	165	250	420	520
S07	570	720	860	880
S08	1125	1130	1140	1120
S09	855	860	720	880
S10	230	230	520	840

Таблица 6.12: Резултати за не структурирани примери

Примери	Произведение мравки × итерации			
	ACO _{5×400}	ACO _{10×200}	ACO _{20×100}	ACO _{40×50}
U01	330	340	340	440
U02	160	340	340	400
U03	1000	1000	560	640
U04	760	760	820	820
U05	295	295	280	280
U06	945	940	720	920
U07	550	550	580	760
U08	160	300	220	120
U09	570	950	920	840
U10	485	550	600	1120

Таблица 6.13: Индексна матрица за ICSA

	$ACO_{5 \times 400}$	$ACO_{10 \times 200}$	$ACO_{20 \times 100}$	$ACO_{40 \times 50}$	
S01	195	200	260	120	
S02	195	330	340	640	
S03	475	490	580	1160	
S04	1540	1540	1540	1560	
S05	415	320	420	920	
S06	165	250	420	520	
S07	570	720	860	880	
S08	1125	1130	1140	1120	
S09	855	860	720	880	
S10	230	230	520	840	
U00	775	780	780	2060	(6.10)
U01	330	340	340	440	
U02	160	340	340	400	
U03	1000	1000	560	640	
U04	760	760	820	820	
U05	295	295	280	280	
U06	945	940	720	920	
U07	550	550	580	760	
U08	160	300	220	120	
U09	570	950	920	840	
U10	485	550	600	1120	

различен брой мравки.

След прилагане на ICrA, използвайки софтуера ICrAData [75], за индексни матрици Таблица 6.10 получихме двете индексни матрици с връзките между разглежданите четири критерия. Получените индексни матрици за $\mu_{C,C'}$, $\nu_{C,C'}$ и $\pi_{C,C'}$ стойности са показани в Таблица 6.11, Таблица 6.12 и Таблица 6.13.

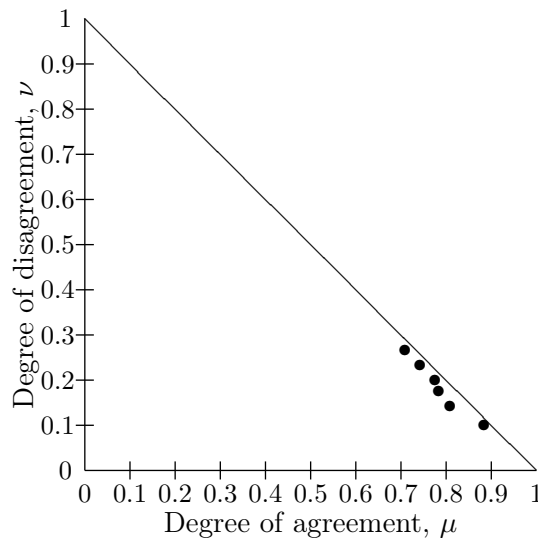
$\mu_{C,C'}$	ACO _{5×400}	ACO _{10×200}	ACO _{20×100}	ACO _{40×50}	
ACO _{5×400}	1	0.88	0.78	0.74	(6.11)
ACO _{10×200}	0.88	1	0.78	0.71	
ACO _{20×100}	0.78	0.78	1	0.81	
ACO _{40×50}	0.74	0.71	0.81	1	

$\nu_{C,C'}$	ACO _{5×400}	ACO _{10×200}	ACO _{20×100}	ACO _{40×50}	
ACO _{5×400}	0	0.10	0.18	0.23	(6.12)
ACO _{10×200}	0.10	0	0.20	0.27	
ACO _{20×100}	0.18	0.20	0	0.14	
ACO _{40×50}	0.23	0.27	0.14	0	

$\pi_{C,C'}$	ACO _{5×400}	ACO _{10×200}	ACO _{20×100}	ACO _{40×50}	
ACO _{5×400}	0	0.02	0.04	0.03	(6.13)
ACO _{10×200}	0.02	0	0.03	0.03	
ACO _{20×100}	0.04	0.03	0	0.05	
ACO _{40×50}	0.03	0.03	0.05	0	

Получените резултати от ICrA се визуализират на Фигура 6.1 в рамките на специфичната триъгълна геометрична интерпретация на IFS. Резултатите показват, че следните двойки критерии, според [80], са:

- положителен консонанс:
ACO_{5×400}-ACO_{10×200} – с ниво на "съгласие"
 $\mu_{C,C'} = 0.88$;



Фигура 6.1: Представяне на ИСгА резултати, чрез интуиционистки размит триъгълник

- слаб положителен консонанс:
 $АСО_{20 \times 100} - АСО_{40 \times 50}$,
 $АСО_{20 \times 100} - АСО_{5 \times 400}$ и
 $АСО_{20 \times 100} - АСО_{10 \times 200}$ – с ниво на "съгласие"
 $\mu_{C,C'} = 0.81$, $\mu_{C,C'} = 0.78$ и $\mu_{C,C'} = 0.78$, съответно;
- слаб дисонанс:
 $АСО_{40 \times 50} - АСО_{5 \times 400}$ и
 $АСО_{40 \times 50} - АСО_{10 \times 200}$ – с ниво на "съгласие"
 $\mu_{C,C'} = 0.74$ и $\mu_{C,C'} = 0.71$, съответно.

АСО алгоритмите с близки стойности на броя на мравките (5 и 10, 10 и 20, 20 и 40) показват сходно поведение. Същият резултат се получава за АСО алгоритмите с 5 и 20 мравки. АСО алгоритмите с по-голяма разлика в броя на мравките (5 и 40, 10 и 40) са в слаб дисонанс, тоест тяхното поведение не е сходно.

В [9] авторът предлага да се класират двойките критерии в двете измерения едновременно (степени на "съгласие" $\mu_{C,C'}$ и "несъгласие" $\nu_{C,C'}$ на интуиционистки размитите двойки). Това може да бъде направено чрез изчисляването за всяка точка от Фигура. 6.1 на нейното разстояние

Таблица 6.14: Скала на съгласие и несъгласие по отношение на [80]

Интервал от $\mu_{C,C'}$	Значение
[0.00-0.05]	строг отрицателен consonance
(0.05-0.15]	отрицателен consonance
(0.15-0.25]	слаб отрицателен consonance
(0.25-0.33]	слаб dissonance
(0.33-0.43]	dissonance
(0.43-0.57]	строг dissonance
(0.57-0.67]	dissonance
(0.67-0.75]	слаб dissonance
(0.75-0.85]	слаб положителен consonance
(0.85-0.95]	положителен consonance
(0.95-1.00]	строг положителен consonance

от точката $\langle 1, 0 \rangle$. Формулата за изчисляване на разстоянието $d_{C,C'}$ на двойката C, C' до точка $\langle 1, 0 \rangle$ е:

$$d_{C,C'} = \sqrt{(1 - \mu_{C,C'})^2 + \nu_{C,C'}^2} \quad (6.14)$$

Резултатите са представени в таблица Table 6.15.

В този случай двойките критерии (различни АСО алгоритми) са подредени според техните $d_{C,C'}$ сортирани в низходящ ред, както следва:

- АСО_{5×400}-АСО_{10×200};

Таблица 6.15: Индексна матрица според разстоянието до точка $\langle 1, 0 \rangle$

	АСО _{5×400}	АСО _{10×200}	АСО _{20×100}	АСО _{40×50}
АСО _{5×400}	0	0.154	0.279	0.348
АСО _{10×200}	0.154	0	0.301	0.395
АСО _{20×100}	0.279	0.301	0	0.238
АСО _{40×50}	0.348	0.395	0.238	0

- $ACO_{20 \times 100} - ACO_{40 \times 50}$;
- $ACO_{5 \times 400} - ACO_{20 \times 100}$;
- $ACO_{10 \times 200} - ACO_{20 \times 100}$;
- $ACO_{5 \times 400} - ACO_{40 \times 50}$;
- $ACO_{40 \times 50} - ACO_{10 \times 200}$.

От тези резултати също се вижда сходното поведение на АСО алгоритъма при 5 и 10, и при 20 и 40 мравки. Отново се вижда сходно поведение при АСО алгоритъма с 5 и с 20 мравки. Следващите три двойки критерии са класирани по същия начин. Така получените ИСГА резултати се потвърждават от два различни подхода – при използване на скалата, предложена в [80] и в съответствие със степента на "съгласие" $\mu_{C,C'}$ и "несъгласие" $\nu_{C,C'}$ на интуиционистки размити двойки [9].

От друга страна, ИСГА потвърждава заключението, че за тази задача най-добрата производителност на алгоритъма, т.е., използвайки по-малко изчислителни ресурси, е с използването на пет мравки.

6.6 Заключение

В тази глава предлагаме АСО алгоритъм за решаване на задачата за планиране на работна сила. Ние сравняваме ефективността на нашия алгоритъм с други два метаевристични метода, GA и SS алгоритмите. Сравнението се извършва по различни критерии. Наблюдавахме, че АСО алгоритъмът постига по-добри решения от другите два алгоритъма. Що се отнася до времето за изпълнение, АСО алгоритъмът е по-бърз. Популацията при АСО алгоритъма се състои от 20 индивида и паметта, използвана от алгоритъма, е подобна на тази, използвана от SS и по-малка от паметта, използвана от GA. Ние предлагаме подходяща процедура за локална търсене, която намалява времето за постигане на най-доброто решение. Изследвано е влиянието на АСО параметрите върху работата на алгоритъма. Показано е, че алгоритъмът е най-ефективен при използването на 5 мравки. Приложен е интеркритериален анализ за да се извлече допълнително знание за връзката между параметрите на алгоритъма.

Глава 7

Моделиране на пътникопоток

Общественият транспорт е много важен за нашето ежедневие. За това оперативната ефективност на обществения транспорт трябва да бъде оптимизирана. Междуселищният обществен транспорт до 1000 км се състои предимно от влакове и автобуси. Важно е транспортните оператори да знаят колко хора ще го използват и какви са техните предпочитания, за да могат транспортните фирми да бъдат печеливши и да се направят разписания, които са удобни за пътниците. Предложили сме модел в който задачата е дефинирана като оптимизационна задача с две целеви функции, минимално време за транспортиране на всички пътници и минимална цена. Приложили сме метода на мравките за моделиране на пътникопотока. Моделът показва колко пътници ще използват влак и колко автобус според това кое е по-важно за тях, времето за транспортиране или цената.

Хората използват разнообразни начини за транспортиране от едно населено място до друго. Удобството на транспорта е много важно, за това е от съществено значение да се създават добри модели, чрез които той да може да бъде оптимизиран. Обикновено влаковете (с изключение на високоскоростните, които се движат с над 200 км/ч) са по-евтини, но и по-бавни. Скоростните влакове и автобусите са по-бързи, но и по-скъпи. В един модел на транспорт всичко това трябва да бъде взето под внимание. За това задачата е представена като оптимизационна задача с две целеви функции, цена за превоз и време за придвижване, като целта е и двете да бъдат минимизирани. Двете цели са взаимно противоречащи си. За да бъде намалена общата цена за придвижване трябва да се използват бавни влакове, което увеличава времето за превоз. За да

се намали времето за превоз трябва да се използват по-бързи превозни средства, които са по-скъпи и води до увеличаване на общата цена за превоз.

Когато задачата е с повече от една целеви функции нямаме едно оптимално решение, а множество от недоминирани решения. Те могат да бъдат анализирани по отношение на предпочитанията на потенциалните пътници и така да определим колко от тях ще използват влак и колко автобус (или високоскоростен влак, ако съществува).

Железопътният транспорт е най-старият обществен транспорт, който се използва и до днес. В днешно време автобусният транспорт конкурира железопътният, особено там където има магистрала. За това аналитичните модели са много важни за по-нататъшното планиране и вземане на решения при развитието на транспорта.

Съществуват различни видове транспортни модели [5]. Значението и ролята на всеки тип модели се обсъждат във връзка с неговата функция. Някои от моделите са съсредоточени върху планирането [2]. Други модели са фокусирани върху симулация, за да се анализира нивото на използване на различни видове транспорт [95]. Има модели, чиято цел е оптимален дизайн на транспортната мрежа [67]. Нашият модел е фокусиран върху моделиране на пътничкопотока според предпочитанията на участниците [43]. Чрез модела могат да се видят слабите места на всеки вид превоз. Моделът може да покаже разпределението на пътничкопотока при промяна на графика или при промяна на капацитета на превозните средства. Резултатите от тази глава са публикувани в [43, 59, 65]

Многоцелевата оптимизация е трудна от изчислителна гледна точка. Ние предлагаме алгоритъм на основата на метода на мравките за решаване на задачата. Моделът е тестван върху един изкуствено направен тестови пример с един влак и един автобус, които тръгват от първата гара по едно и също време. След това моделът е тестван върху реални данни, транспортът между София и Варна, една от най-дългите дестинации в България.

7.1 Формулиране на задачата

В транспортирането на пътници възникват разнообразни проблеми. Някои от тях се отнасят до оптимизиране на разписанието на превозните средства [71]. Други моделират потока от пътници и неговото оптимално

управление [94]. Някои задачи се опитват за случаи където има един тип превозни средства, докато при други има комбинация от разнообразни типове превозни средства [94]. Общото между тях е, че са трудни от изчислителска гледна точка. Нашата задача се отнася за пътници, пътуващи в дадено посока, покрита с няколко вида превозни средства, влак и автобус.

В нашия случай има дестинация от спирка A до спирка B . Има няколко вида превозни средства, разнообразни влакове и автобуси, които пътуват между спирка A и спирка B . Всяко превозно средство има множество от междинни спирки, на които спира, между двете крайни спирки. Някои от междинните спирки могат да са общи за някои от превозните средства. Нека множеството от всички спирки да бъде $S = \{s_1, \dots, s_n\}$ и на всяка спирка s_i , $i = 1, \dots, n - 1$, n е броят на спирките, във всеки времеви слот има брой пътници, които искат да пътуват до гара s_j , $j = i + 1, \dots, n$. Всяко превозно средство може да има различна скорост и цена за придвижване от спирка s_i до спирка s_j .

Входните данни са множеството от спирки S , начален час на всяко превозно средство от първата спирка, време за преминаване на всяко превозно средство от спирка s_i до спирка s_j , капацитет на всяко превозно средство, цена за пътуване с превозното средство от една гара до друга, брой пътници през отделните времеви слотове, които искат да пътуват от една спирка до друга. Нашият алгоритъм изчислява колко пътници ще се качат на всяко от превозните средства и ще пътуват от спирка s_i до спирка s_j във всеки времеви слот. Дефинирали сме две целеви функции, сумата от цените на всички продадени билети и общото време за превозване на пътниците. Ако някое от превозните средства неспира на някоя от спирките, то поставяме времето за пътуване и цената до тази спирка да бъдат 0.

Исходните данни са броя пътници във всяко превозно средство от спирка s_i до спирка s_j , за $i = 1, \dots, n - 1$ и s_j , $j = i + 1, \dots, n$, във всеки времеви интервал и стойностите на двете целеви функции.

Предложеният проблем е много труден от изчислителна гледна точка. Точните методи, както и традиционните числени методи изискват твърде много изчислителни ресурси за да намерят оптималното или близко да него решение. Такъв тип проблеми обикновено се решават с някакъв метаевристичен метод. За решаването на тази задача сме приложили метода на мравките, защото е един от най-добрите метаевристични методи за решаване на дискретни оптимизационни задачи. Решаваната задача

е с повече от една целеви функции и затова сме предложили вариант на метода на мравките за многоцелева оптимизация. Многоцелевата оптимизация има своите корени през XIX век в работата на Edgeworth и Pareto в икономиката [101]. Оптималното решение на задачи с повече от една целеви функции не е единствено, както това е при задачите с една целева функция. В случая на няколко целеви функции имаме множество от недоминирани решения наричани Парето оптимални решения. Решението е Парето оптимално, ако не е възможно да се подобри някоя от компонентите му, без това да доведе до влошаване на някоя от другите компоненти. Основната цел при решаването на многоцелеви задачи е да се намери множеството от Парето оптимални решения, образуващи Парето фронт. Едно решение доминира друго, ако поне една от компонентите му е по-добра, а останалите не са по-лоши. След намирането на Парето фронта, потребителите решават кое от решенията да изберат, евентуално използвайки допълнителни критерии. Когато се използват приближени методи, целта е да се намерят решения, близки до Парето фронта.

7.2 Метод на мравките за моделиране на пътничопоток

При прилагане на метода на мравките за задачата за моделиране на пътничопоток времето се разделя на времеви интервали/слотове ($N \times 24$ времеви интервали, съответстващи на $60/N$ минути; например $2 \times 24 = 48$ времеви интервала, съответстват на 30 минути). Решаваме задачата за едно направление. Множество от $N \times 24$ върха, съответства на всяка една от спирките, и времевите интервали. Така графът на задачата показва дали има превозно средство на спирка $s_i, i = 1, \dots, n$ във фиксиран времеви момент. Феромонът ще бъде поставян на върховете на графа, при решаването на тази задача. Мравките започват да строят решението от първата спирка. Те избират по случаен начин колко пътника ще се качат във всяко от превозните средства. Горната граница на пътниците, които могат да се качат в превозното средство е минимумът от разликата между капацитетът на превозното средство и пътниците, които вече са в него и броя на пътниците, които искат да пътуват. Ако на дадена спирка в даден момент време има само едно превозно средство, то в

него се качват максимално възможно пътници. Прилага се функцията за вероятност на прехода за да се моделира колко пътници ще се качат в превозното средство на следващата спирка. Евристичната информация която конструираме е сума от реципрочните стойности на двете целеви функции.

В нашия вариант на прилагане на метода на мравките, първо феромонът на всички върхове се намалява, като се умножава с коефициента на изпарение. След това се добавя нов феромон на върховете на доминираните решения. Добавеният феромон е равен на единица върху сумата от реципрочните стойности на двете целеви функции.

7.3 Експериментални резултати

Направена е програмна реализация на предложения алгоритъм. След направени тестове, са фиксирани следните стойности за параметрите на алгоритъма: параметър на изпарението $\rho = 0.5$; $\alpha = \beta = 1$; началният феромон е поставен да бъде $\tau_0 = 0.5$.

Подготвили сме малък пример с който да тестваме алгоритъма. Примерът съдържа един влак и един автобус, които тръгват от началната спирка. Има общо 4 спирки. Автобусът е по-бърз, а влакът е по-евтин. Можем да видим в различните случаи колко от пътниците биха използвали автобус и колко влак, в зависимост от техните предпочитания да платят по-малко или да пристигнат по-бързо.

В този пример са използвани 5 мравки и алгоритъмът е работил 5 итерации. Таблица 7.1 показва получените резултати, подредени по нарастващ ред на общата цена за придвижване и намаляващ ред на общото време за придвижване. Четвъртата колона показва броят на пътниците използвали влак. Наблюдаваме, че когато общата цена за придвижване е по-голяма, то броят на пътниците във влака е по-малък. Има няколко изключения. При решение No 6 общата цената за придвижване е по-малка отколкото при решения No 7 и No 8, но броят на пътниците използвали влак е по-малък отколкото при тези решения. При подробен анализ на решението се вижда, че пътниците са предпочитали влак за дългите разстояния. При решение No 16, пътниците използвали влак са по-малко от пътниците използвали влак при решения No 17 и No 18. В този случай пътниците са използвали влак за късите разстояния.

Алгоритъмът е приложен и за реален пример. Началната спирка е

Таблица 7.1: Експериментални резултати

No	Цена	Време	Влак
1	265	451	86
2	270	450	79
3	272	448	86
4	274	446	79
5	277	443	76
6	282	438	66
7	289	431	78
8	291	429	69
9	292	428	64
10	295	425	66
11	301	419	65
12	302	418	60
13	307	413	62
14	309	411	64
15	313	407	67
16	321	399	60
17	324	396	63
18	331	389	61

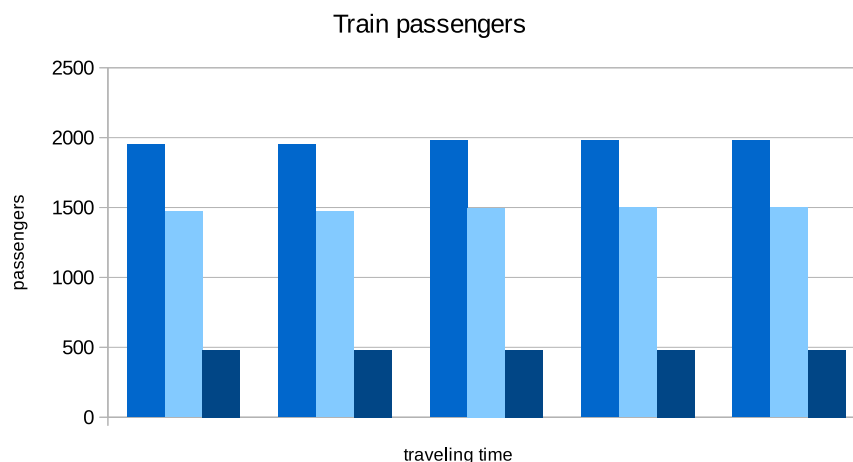
Таблица 7.2: Експериментални резултати за линията София Варна

No	Цена	Време	Влак
1	51843	25840	1951
2	51797	25842	1952
3	51579	25862	1978
4	51571	25869	1979
5	51563	25870	1980

София, столицата на България, а крайната е Варна, морската столица на България. Това е едно от най-дългите железопътни направления в България, с дължина около 450 километра. По това направление има 5 влака и 23 автобуса за едно денонощие. Те се движат с различна скорост, цените за придвижване от една спирка до друга са различни и имат разлики в междинните спирки. Нямаме точни данни за броя на пътниците, които пътуват от една спирка до друга по линията София Варна. По тази причина сме направили оценка на броя на пътниците, имайки в предвид броя на жителите на населените места в които спират превозните средства.

В този пример са използвани 10 мравки и алгоритъмът е работил 100 итерации. Таблица 7.2, подобно на Таблица 7.1, показва получените решения, подредени по нарастващ ред на общата цена за придвижване и намаляващ ред на общото време за придвижване. Четвъртата колона показва броя на пътниците използвали влака във всеки един от случаите. Броят на пътниците използвали влак, съответно автобус, се променя, само когато на една и съща спирка, по едно и също време има повече от едно превозно средство. Както и в предния случай наблюдаваме, че когато общата цена намалява, общият брой пътници, използвали влак се увеличава. Съответно когато общото време за придвижване нараства, броят на пътниците използвали влак намалява, което означава, че броят на пътниците използвали автобус се увеличава.

На фигура 7.1 е показан броят на пътниците използвали влак. Всяка серия от колони съответства на едно от получените решенията, лявата серия от колони съответства на решение 1 от Таблица 7.2, а дясната серия от колони съответства на решение 5 от Таблица 7.2. Всяка серия



Фигура 7.1: Пътници използвали влаковете София Варна

се състои от три колони. Лявата колона от всяка серия съответства на общият брой пътници използвали влак по разглежданото направление. Средната колона съответства на броя на пътниците използвали влак на дълго разстояние, повече от 300 км. Лявата колона съответства на броя на пътниците използвали влак на къси разстояния, по-малко от 300 км. Наблюдаваме, че при решенията с по-ниска обща цена за пътуване и по-дълго общо време за придвижване, нараства общият брой на пътниците използвали влак заради нарастване на пътниците използвали влак на дълги разстояния. Броят на пътниците използвали влак на къси разстояния е почти еднакъв за всички решения. Разликата в цената за пътуване и времето за придвижване на къси разстояния е сходна при влак и автобус. Поради това не се наблюдава преминаване на пътници от единия вид транспортно средство към другия.

7.4 Заключение

В тази глава беше предложен модел на пътничопоток между две населени места при наличието на няколко вида транспорт. Задачата е дефинирана като оптимизационна с две целеви функции. Едната цел е намаляване общото време за придвижване, а другата е намаляване на

общата цена за пътуване. Предложили сме алгоритъм, базиран на метода на мравките, за решаването на тази задача. С помощта на този модел може да се прави анализ на съществуващият междуселищен транспорт. Може да се предвиди как ще се измени пътничкопотокът, при промяна на разписанието на превозните средства. Друга възможност е да се моделира пътничкопотока при добавяне на превозни средства или премахване на някои от тях по разглежданото направление. По този начин може да се оптимизира междуселищният транспорт по дадено направление, като това доведе до подобряване на предлаганата услуга и намаляване на разходите. Когато има локален транспорт по късите дестинации на избрано дълго направление, алгоритъмът първо може да се прилага за моделиране и оптимизиране на превозът по дългото направление и след неговото уточняване, той да бъде приложен и за локалния транспорт от междинните направления.

Глава 8

Заклучение

Комбинаторните оптимизационни задачи са изключително трудна от изчислителна гледна точка. Не съществува универсален метод за тяхното решаване. Обикновено прилагането на даден метод за решаването на такъв тип задачи зависи от самата задача и може да е различно за различни варианти на една и съща задача.

Фокусът в тази дисертация е към прилагането на метода на мравките. Този метод е измежду най-добрите за решаване на комбинаторни оптимизационни задачи. Методът на мравките се различава от останалите методи по това, че е конструктивен метод и превъзхожда повечето от останалите методи в голяма част от приложенията.

В настоящата дисертация са събрани резултатите на автора, в областта на метода на мравките и неговите приложения, постигнати през последните 7 години. В началото е дадено описание на метода и неговите разновидности.

Отделните глави представляват приложението на метода на мравките към отделни задачи. Това са задачата за раницата, задачата за инспектиране на GPS мрежа, построяване на безжична сензорна мрежа, задачата за наемане на работна сила, моделиране на пътничкопоток. Всички те са сложни комбинаторни оптимизационни задачи изискващи неполиномиален брой изчисления. Изследвано е влиянието на параметрите на разработените алгоритми. Приложен е интеркритериален анализ. Направена е програмна реализация на разработените алгоритми.

8.1 Списък на публикациите

Монографии

- 1 Fidanova S.. Ant Colony Optimization and Applications. Studies in Computational Intelligence, 947, Springer, 2021, ISBN:978-3-030-67380-2, DOI:<https://doi.org/10.1007/978-3-030-67380-2>, 142 pages.

Списания с импакт фактор

- 2 Fidanova S., Atanassov K.. ACO with Intuitionistic Fuzzy Pheromone Updating Applied on Multiple Knapsack Problem. Mathematics, 9, 13, MDPI, 2021, ISSN:2227-7390, DOI:10.3390/math9131456, 1-7. IF 2.9,Q1.

Издания с импакт ранг

- 3 Fidanova S., Metaheuristic Method for Transport Modelling and Optimization Studies in Computational Intelligence, 648, Springer, 2016, 295 - 302. SJR 0.235
- 4 Roeva O., Fidanova S., Paprzycki M.. Comparison of Different ACO Start Strategies Based on InterCriteria Analysis. Recent Advances in Computational Optimization, Studies in Computational Intelligence, 717, Springer, 2018, ISBN:978-3-319-59860-4, 53-72. SJR 0.235
- 5 Fidanova S., Shindarov M., Marinov P.. Wireless Sensor Positioning Using ACO Algorithm. Studies in Computational Intelligence, 657, Springer, 2017, ISBN:978-3-319-41437-9, ISSN:1860-949X, 33-44. SJR 0.235
- 6 Fidanova S., Roeva O.. InterCriteria Analyzis of Differen Variants of ACO algorithm for Wireless Sensor Network Positioning. Lecture Notes in Computer Science, 11189, Springer, 2019, 88-96.SJR 0.407
- 7 Fidanova S., Roeva O.. Multi-Objective ACO Algorithm for WSN Layout: InterCriteria Analisys. Lecture Notice in Computer Science, 11958, Springer, 2020, ISBN:978-3-030-410315, 474-481. SJR 0.407

- 8 Roeva O., Fidanova S.. Different InterCriteria Analysis of Variants of ACO algorithm for Wireless Sensor Network Positioning. *Studies in Computational Intelligence*, 838, Springer, 2020, ISBN:978-3-030-22723-4, 83-103. SJR 0.237
- 9 Roeva O., Fidanova S., Luque G., Paprzycki M., InterCriteria Analysis of ACO Performance for Workforce Planing Problem, *Studiec in Computational Intelligence* 795, 2019, 47-68. SJR 0.237
- 10 Fidanova S., Luque G., New Local Search Procedure for Workforce Planning Problem, *Cybernetics and Information Technologies*, Vol. 20(6), 2020, 40-48. SJR 420
- 11 Roeva O., Fidanova S., Ganzha M., InterCriteria Analysis of the Evaporation Parameter Influence on Ant Colony Optimization Algorithm: A Workforce Planing Problem, *Studies in Computational Intelligence* 920, 2021, 89-110. SJR 0.237
- 12 Fidanova S., Roeva O., Influence of ACO Evaporation Parameter for Unstructured Workforce Planning Problem, *Large Scale Scientific Computing, Lecture Notes in Computer Science* 13127, 2022, 234-241. SJR 0.407
- 13 Fidanova S., Roeva O., Ganzha M., Ant Colony Algorithm for Fuzzy Transport Modelling: InterCriteriaAnalysis, *Studies in Computational Intelligence* 986, 2022, 123-138. SJR 0.237

Издания реферирани в Scopus

- 14 Fidanova S., Roeva O., Paprzycki M., Gepner P.. InterCriteria Analysis of ACO Start Startegies. *IEEE Xplorer*, 2016, ISBN:ISBN 978-83-60810-90, DOI:ISBN 978-83-60810-90-3, 547-550
- 15 Fidanova S.. Hybrid Ant Colony Optimization Algorithm for Multiple Knapsack Problem. 5th IEEE International Conference on Recent Advances and Innovations in Engineering (ICRAIE), *IEEE Xplorer*, 2020, DOI:10.1109/ICRAIE51050.2020.91-5
- 16 Fidanova S., Roeva O., Ganzha M.. InterCriteria Analyzis of Hybrid Ant Colony Optimization Algorithm for Multiple Knapsack Problem. *Annals of Computer Science and Information Systems*, 25, *IEEE Xplorer*, 2021, ISBN:978-83-959183-6-0, ISSN:2300-5963, DOI:10.15439/2021F22, 173-180

- 17 Fidanova, S., Luquq, G., Roeva, O., Paprzycki, M., Gepner, P.: Ant Colony Optimization Algorithm for Workforce Planning. FedCSIS'2017, pp. 415–419. IEEE Xplorer (2017)
- 18 Fidanova S., Roeva O., Ganzha M., Ant Colony Algorithm for Fuzzy Transport Modelling, In proc of FedCSIS, ACSIS Vol. 21, IEEE Xplorer, 2020, 237-240.

Други издания

- 19 Fidanova S., Atanassov K.. Flying Ant Colony Optimization Algorithm for Combinatorial Optimization. Studia Informatica, 38, 4, Polish Information Sociaty, 2017, ISSN:1642-0489, 31-40

8.2 Списък на цитиранията

Fidanova S.. Ant Colony Optimization and Applications. Studies in Computational Intelligence, 947, Springer, 2021, ISBN:978-3-030-67380-2, DOI:<https://doi.org/10.1007/978-3-030-67380-2>, 142 pages.

Е цитирана в:

1. Adamuthe, A.C., Kagwade, S.M., Hybrid and adaptive harmony search algorithm for optimizing energy efficiency in VMP problem in cloud environment (2022) Decision Science Letters, 11 (2), pp. 113-126. <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85122887825&doi=10.5267%2fj.ds1.2022.1.001&partnerID=40&md5=05f414055439af43fe3ef889b601cdb6>
2. Ali M.A., Balamurugan B., Sharma V., IoT and Blockchain Based Intelligence Security System for Human Detection using an Improved ACO and Heap Algorithm (2022) 2022 2nd International Conference on Advance Computing and Innovative Technologies in Engineering, ICACITE 2022, pp. 1792 - 1795, DOI: 10.1109/ICACITE53722.2022.9823827 <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85135471099&doi=10.1109%2fICACITE53722.2022.9823827&partnerID=40&md5=fd8ed18b46bf70a9e986de04>
3. Alobaedy M.M., Khalaf A.A., Fazea Y., Distributed Multi-Ant Colony System Algorithm using Raspberry Pi Cluster for Travelling Salesman Problem (2022) Iraqi Journal of Science, 63 (9), pp. 4067 - 4078, DOI: 10.24996/ij.s.2022.63.9.35 <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85139562253&doi=10.24996%2fij.s.2022.63.9.35&partnerID=40&md5=354de00a4bd89e50b4645da024b2bb12>
4. Belkhadir A., Belkhayat D., Zidani Y., Pusca R., Romary R., Torque Ripple Minimization Control of Permanent Magnet Synchronous Motor using Adaptive Ant Colony Optimization (2022) 2022 8th International Conference on Control, Decision and Information Technologies, CoDIT 2022, pp. 629 - 635, DOI: 10.1109/CoDIT55151.2022.9804127 <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85134345047&doi=10.1109%2fCoDIT55151.2022.9804127&partnerID=40&md5=6fdf53f5970034691d53c077b831c>
5. Guvenc M.A., Bilgic H.H., Cakir M., Mistikoglu S., The prediction of surface roughness and tool vibration by using metaheuristic-based ANFIS during dry turning of Al alloy (AA6013) (2022) Journal of

- the Brazilian Society of Mechanical Sciences and Engineering, 44 (10), art. no. 474, DOI: 10.1007/s40430-022-03798-z, IF 2.361 <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85138621510&doi=10.1007%2fs40430-022-03798-z&partnerID=40&md5=73514312166ea6828adc02f389fb60>
6. Kaidi, W., Khishe, M., Mohammadi, M. (2022). Dynamic levy flight chimp optimization. *Knowledge-Based Systems*, 235, 107625. <https://doi.org/10.1016/j.kbs.2022.107625>. IF 8.139 <https://www.sciencedirect.com/science/article/pii/S0950705121008871>
casa_token=tgWguv8sDt4AAAAA:7Fz4WEDDf6CAQpahCIbnxJ_QVQrkRTqGh3d_ckBs-xz2XDc09DgzusSFikRWc-4H-AAGKsA
 7. Kim G., S. Kim, I. G. Jang, "Loopwise Route Representation-Based Topology Optimization for the Shortest Path Problems," in *IEEE Access*, vol. 10, pp. 128835-128846, 2022, doi: 10.1109/ACCESS.2022.3227388. <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85144789774&doi=10.1109%2fACCESS.2022.3227388&partnerID=40&md5=c210a08c695436a0e44e45>
 8. Kumar, N., Kaswan, K. S. (2022). Soft Hunting Algorithm for Auto-Tuning Software Reliability Growth Models. *Wireless Communications and Mobile Computing*, 2022. Article ID 3752264. <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85137127959&doi=10.1155%2f2022%2f3752264&partnerID=40&md5=11cee5115550595d36a5bb7b730ec03d>
 9. Mirrashid N., Alibeiki E., Rakhtala S.M., Development and Control of an upper Limb Rehabilitation Robot via Ant Colony Optimization - PID and Fuzzy-PID Controllers (2022) *International Journal of Engineering, Transactions B: Applications*, 35 (8), DOI: 10.5829/IJE.2022.35.08B.04 <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85130637106&doi=10.5829%2fIJE.2022.35.08B.04&partnerID=40&md5=cd3fbb9b5ab295a6ef24941>
 10. Murinto, P.A., Ujianto E.I.H., Multilevel Thresholding Image Segmentation Based-Logarithm Decreasing Inertia Weight Particle Swarm Optimization (2022) *International Journal of Advances in Soft Computing and its Applications*, 14 (3), pp. 64 - 77, DOI: 10.15849/IJASCA.221128.05 <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85142303135&doi=10.15849%2fIJASCA.221128.05&partnerID=40&md5=b2ce41ddae161ca9c8b2744c>
 11. Qian Q., Yu K., Yadav P.K., Dhal S., Kalafatis S., Thomasson J.A., Hardin R.G., Cotton Crop Disease Detection on Remotely Collected Aerial Images with Deep Learning, (2022) *Proceedings of SPIE - The*

International Society for Optical Engineering, 12114, art. no. 1211405, DOI: 10.1117/12.2623039 <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85135802427&doi=10.1117%2f12.2623039&partnerID=40&md5=c52283acee4ad46d54cffbf4fd6d79a9>

12. Rajani, Kumar N., Kaswan K.S., Soft Hunting Algorithm for Auto-Tuning Software Reliability Growth Models (2022) *Wireless Communications and Mobile Computing*, 2022, art. no. 3752264, DOI: 10.1155/2022/3752264 <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85137127959&doi=10.1155%2f2022%2f3752264&partnerID=40&md5=11cee5115550595d36a5bb7b730ec03d>
13. Varol Altay E., Hybrid Archimedes optimization algorithm enhanced with mutualism scheme for global optimization problems (2022) *Artificial Intelligence Review*, DOI: 10.1007/s10462-022-10340-z, IF 9,588 <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85143290279&doi=10.1007%2fs10462-022-10340-z&partnerID=40&md5=dbda36ef0755e71f685838666ca02a57>
14. Kaur G., Chanak P., Bhattacharya M., Obstacle-Aware Intelligent Fault Detection Scheme for Industrial Wireless Sensor Networks (2022) *IEEE Transactions on Industrial Informatics*, 18 (10), pp. 6876 - 6886, DOI: 10.1109/TII.2021.3133347 <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85121387038&doi=10.1109%2fTII.2021.3133347&partnerID=40&md5=4945cb61fda4fbc7189a4d943ed4aa7b>
15. Safaeian Hamzehkolaei, N., MiarNaeimi, F., A new hybrid multi-level cross-entropy-based moth-flame optimization algorithm (2021) *Soft Computing*, Vol. 23(18),pp. 14701-14718, IF 3.643 . <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85115987838&doi=10.1007%2fs00500-021-06109-1&partnerID=40&md5=9ff8d4cee4775ac185a532df27ef9364>

Fidanova S., Metaheuristic Method for Transport Modelling and Optimization Studies in Computational Intelligence, 648, Springer, 2016, 295 - 302.

Е цитирана в:

16. Ribagin S., Lyubenova V. (2021) Metaheuristic Algorithms: Theory and Applications. In: Atanassov K.T. (eds) *Research in Computer Science in the Bulgarian Academy of Sciences. Studies in Computational Intelligence*, vol 934. Springer, Cham. https://doi.org/10.1007/978-3-030-72284-5_18 https://link.springer.com/chapter/10.1007/978-3-030-72284-5_18

Roeva O., Fidanova S., Paprzycki M.. Comparison of Different ACO Start Strategies Based on InterCriteria Analysis. Recent Advances in Computational Optimization, Studies of Computational optimization, 717, Springer, 2018, ISBN:978-3-319-59860-4, 53-72.

Е цитирана в:

17. Chorukova E., Marinov P., Umlenski I. (2021) Survey on Theory and Applications of InterCriteria Analysis Approach. In: Atanassov K.T. (eds) Research in Computer Science in the Bulgarian Academy of Sciences. Studies in Computational Intelligence, vol 934. Springer, Cham. https://doi.org/10.1007/978-3-030-72284-5_20 https://link.springer.com/chapter/10.1007/978-3-030-72284-5_20
18. Ribagin S., Lyubenova V. (2021) Metaheuristic Algorithms: Theory and Applications. In: Atanassov K.T. (eds) Research in Computer Science in the Bulgarian Academy of Sciences. Studies in Computational Intelligence, vol 934. Springer, Cham. https://doi.org/10.1007/978-3-030-72284-5_18 https://link.springer.com/chapter/10.1007/978-3-030-72284-5_18
19. Atanassov, Krassimir T. "Applications of IVIFSs." Interval-Valued Intuitionistic Fuzzy Sets. Springer, Cham, 2020. 131-194. https://link.springer.com/chapter/10.1007/978-3-030-32090-4_6
20. Mello-Roman, J.D., Hernandez, A., KPLS optimization with nature-inspired metaheuristic algorithms (2020) IEEE Access, 8, art. no. 9178802, pp. 157482-157492. IF 3.74 <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85091217211&doi=10.1109%2fACCESS.2020.3019771&partnerID=40&md5=cf86a1c4e4a2bec5d81e077d070903b4>
21. Atanassov K, Marinov P, Atanassova V. InterCriteria Analysis with Interval-Valued Intuitionistic Fuzzy Evaluations, Flexible Query Answering Systems, Lecture Notes in Computer Science 11529, 2019 (pp. 329-338). Springer, Cham. https://link.springer.com/chapter/10.1007/978-3-030-27629-4_30
22. Mansour, Imen Ben, Ines Alaya, and Moncef Tagina. "A gradual weight-based ant colony approach for solving the multiobjective multidimensional knapsack problem." Evolutionary Intelligence 12.2 (2019): 253-272. <https://link.springer.com/article/10.1007/s12065-019-00222-9>

Fidanova S.. Hybrid Ant Colony Optimization Algorithm for Multiple Knapsack Problem. 5th IEEE International Conference on Recent Advances and Innovations in Engineering (ICRAIE), IEEE, 2020, DOI:10.1109/ICRAIE51050.2020.9358351, 1-5

Е цитирана в:

23. Abdel-Basset, M., Mohamed, R., Abouhawwash, M., Chang, V., Askar, S.S., A local search-based generalized normal distribution algorithm for permutation flow shop scheduling (2021) Applied Sciences (Switzerland), 11 (11), art. no. 4837, .IF 2.679 <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85107451707&doi=10.3390%2fapp11114837&partnerID=40&md5=b5a465074032557746a4e7744eef634b>
24. Kostyikova, M., Kozachok, L., Levterov, A., Plekhova, A., Shevchenko, V., Okun, A., The use of the heuristic method for solving the knapsack problem (2021) 2021 IEEE 2nd KhPI Week on Advanced Technology, KhPI Week 2021 - Conference Proceedings, pp. 177-180. <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85118916396&doi=10.1109%2fKhPIWeek53812.2021.9570025&partnerID=40&md5=f326e577eedec696e16bd71d58b68>
25. Yang Q., Haomiao L., Xiao W., Omar D., On the adoption of Metaheuristics for Solving 0-1 Knapsack Problems, Proceedings - International Symposium on Parallel Architectures, Algorithms and Programming, PAAPVolume 2021, ISBN 978-166549639-1, Pages 56 - 61. DOI 10.1109/PAAP54281.2021.9720471 https://www.scopus.com/record/display.uri?eid=2-s2.0-85126875329&origin=resultslist&sort=plf-f&src=s&st1=Fidanova&sid=409193dd157582b11c8c4ab6af8sot=b&sdt=b&rr=7&sl=13&s=REF%28Fidanova%29&relpos=2&citeCnt=0&searchTerm=&featureToggles=FEATURE_NE

Fidanova S., Shindarov M., Marinov P.. Wireless Sensor Positioning Using ACO Algorithm. Studies in Computational Intelligence, 657, Springer, 2017, ISBN:978-3-319-41437-9, ISSN:1860-949X, 33-44.

Е цитирана в:

26. Jackman, P. (2022). You Got Data, Now What: Building the Right Solution for the Problem. In: Bochtis, D.D., Moshou, D.E., Vasileiadis, G., Balafoutis, A., Pardalos, P.M. (eds) Information and Communication Technologies for Agriculture—Theme II: Data. Springer Optimization and Its Applications, vol 183. Springer, Cham. <https://doi.org/10.>

1007/978-3-030-84148-5_1 https://www.scopus.com/record/display.uri?eid=2-s2.0-85127162881&origin=resultslist&sort=plf-f&src=s&st1=Fidanova&sid=c88732e8149c133647c7354b6579723f&sot=b&sdt=b&rr=7&sl=13&s=REF%28Fidanova%29&relpos=4&citeCnt=0&searchTerm=&featureToggles=FEATURE_NE

27. Ribagin S., Lyubanova V. (2021) Metaheuristic Algorithms: Theory and Applications. In: Atanassov K.T. (eds) Research in Computer Science in the Bulgarian Academy of Sciences. Studies in Computational Intelligence, vol 934. Springer, Cham. https://doi.org/10.1007/978-3-030-72284-5_18, ISSN: 1860949X https://www.scopus.com/inward/record.uri?eid=2-s2.0-85111092796&doi=10.1007%2f978-3-030-72284-5_18&partnerID=40&md5=8a0ceb89b608f04773d4365d662028d8
28. Bureva, V., Traneva, V., Sotirova, E., Atanassov, K. Index matrices and olap-cube part 5: Index matrix operations over olap-cube (2020) Advanced Studies in Contemporary Mathematics (Kyungshang), 30 (1), pp. 69-88. DOI: 10.17777/ascm2020.30.1.69, ISSN: 12293067 <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85096143035&doi=10.17777%2fascm2020.30.1.69&partnerID=40&md5=38a6393e4603881b8c6385f4f249>

Fidanova S., Roeva O.. InterCriteria Analyzis of Differen Variants of ACO algorithm for Wireless Sensor Network Positioning. Lecture Notes in Computer Science, 11189, Springer, 2019, 88-96.

Е цитирана в:

29. Chorukova, E., Marinov, P., Umlenski, I., Survey on Theory and Applications of InterCriteria Analysis Approach, (2021) Studies in Computational Intelligence, 934, pp. 453-469. DOI:10.1007/978-3-030-72284-5_20 https://www.scopus.com/inward/record.uri?eid=2-s2.0-85111122908&doi=10.1007%2f978-3-030-72284-5_20&partnerID=40&md5=9426c5440afaebae760a8
30. Videv, T., Sotirov, S., Bozveliev, B., Generalized Net Model of the Network for Automatic Turning and Setting the Lighting in the Room with Intuitionistic Fuzzy Estimations, (2020) Studies in Computational Intelligence, 862, pp. 83-90. https://www.scopus.com/inward/record.uri?eid=2-s2.0-85080931227&doi=10.1007%2f978-3-030-35445-9_7&partnerID=40&md5=fce77b43f5c151766ec7582417a2f9db

Fidanova S., Roeva O.. Multi-Objective ACO Algorithm for WSN Layout: InterCriteria Analysis. Lecture Notice in Computer Science, 11958, Springer, 2020, ISBN:978-3-030-410315, 474-481.

Е цитирана в:

31. Todorov T., Dimov D., Ostromski Tz., Zlatev Z. Georgieva R., Poryazov S., Sensitivity Study of a Large-Scale Air Pollution Model by Using Optimized Latin Hyprecube Sampling, Recent Advances in Computational Optimization, Studies in Computational Intelligence 986, 2022, 371-387. SJR 185 https://link.springer.com/chapter/10.1007%2F978-3-030-82397-9_19
32. Chorukova, E., Marinov, P., Umlenski, I., Survey on Theory and Applications of InterCriteria Analysis Approach, (2021) Studies in Computational Intelligence, 934, pp. 453-469. DOI:10.1007/978-3-030-72284-5_20 https://www.scopus.com/inward/record.uri?eid=2-s2.0-85111122908&doi=10.1007%2f978-3-030-72284-5_20&partnerID=40&md5=9426c5440afaebae760a84cbb188

Roeva O., Fidanova S.. Different InterCriteria Analysis of Variants of ACO algorithm for Wireless Sensor Network Positioning. Studies in Computational Intelligence, 838, Springer, 2020, ISBN:978-3-030-22723-4, 83-103.

Е цитирана в:

33. Chorukova, E., Marinov, P., Umlenski, I., Survey on Theory and Applications of InterCriteria Analysis Approach, (2021) Studies in Computational Intelligence, 934, pp. 453-469. DOI:10.1007/978-3-030-72284-5_20 https://www.scopus.com/inward/record.uri?eid=2-s2.0-85111122908&doi=10.1007%2f978-3-030-72284-5_20&partnerID=40&md5=9426c5440afaebae760a84cbb188
34. Afradi, A., Ebrahimabadi, A., Hallajian, T. Prediction of tunnel boring machine penetration rate using ant colony optimization, bee colony optimization and the particle swarm optimization, case study: Sabzkooh water conveyance tunnel (2020) Mining of Mineral Deposits, 14 (2), pp. 75-84. <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85084131408&doi=10.33271%2fmining14.02.075&partnerID=40&md5=3aa4f6f6813f26f8a58f3e7ed39efa30>

Roeva O., Fidanova S., Luque G., Paprzycki M., InterCriteria Analysis of ACO Performance for Workforce Planing Problem, Studiec in Computational Intelligence 795, 2019, 47-68.

Е цитирана в:

35. Chorukova, E., Marinov, P., Umlenski, I., Survey on Theory and Applications of InterCriteria Analysis Approach, (2021) *Studies in Computational Intelligence*, 934, pp. 453-469. DOI:10.1007/978-3-030-72284-5_20 https://www.scopus.com/inward/record.uri?eid=2-s2.0-85111122908&doi=10.1007%2f978-3-030-72284-5_20&partnerID=40&md5=9426c5440afaebae760a8
36. Sallam, K.M.aEmail Author, Turan, H.H.bEmail Author, Chakraborty, R.K.bEmail Author, Elsayah, S.bEmail Author, Ryan, M.J.b, A Differential Evolution Algorithm for Military Workforce Planning Problems: A Simulation-Optimization Approach, *EEE Symposium Series on Computational Intelligence*, SSCI 20201 December 2020, Article number 9308566, Pages 2504-2509 <https://www.scopus.com/record/display.uri?eid=2-s2.0-85099715211&origin=resultslist&sort=plf-f&src=s&st1=&st2=&sid=031beda27901ae541dd219a145b1bcc3&sot=b&sdt=b&rr=30&sl=14&s=REF+%28Fidanova%29&relpos=5&citeCnt=0&searchTerm=>
37. Dezert, Jean, Albena Tchamova, Deqiang Han, and Jean-Marc Tacnet. "Simplification of Multi-Criteria Decision-Making Using Inter-Criteria Analysis and Belief Functions." In 2019 22th International Conference on Information Fusion (FUSION), pp. 1-8. IEEE, 2019. <https://ieeexplore.ieee.org/abstract/document/9011326>

Fidanova S., Roeva O., Paprzycki M., Gepner P.. InterCriteria Analysis of ACO Start Strategies. IEEE Xplorer, 2016, ISBN:ISBN 978-83-60810-90, DOI:ISBN 978-83-60810-90-3, 547-550

Е цитирана в:

38. Ignatova V., Todorova L. (2022) Computer-Based Rehabilitation of Cognitive Impairments in Patients with Multiple Sclerosis. In: Sotirov S.S., Pencheva T., Kacprzyk J., Atanassov K.T., Sotirova E., Staneva G. (eds) *Contemporary Methods in Bioinformatics and Biomedicine and Their Applications*. BioInfoMed 2020. Lecture Notes in Networks and Systems, vol 374. Springer, Cham. https://doi.org/10.1007/978-3-030-96638-6_4 https://www.scopus.com/record/display.uri?eid=2-s2.0-85127061211&origin=resultslist&sort=plf-f&src=s&st1=Fidanova&sid=409193dd157582b11c8c4ab6af886ca4&sot=b&sdt=b&rr=7&sl=13&s=REF%28Fidanova%29&relpos=0&citeCnt=0&searchTerm=&featureToggles=FEATURE_NE

39. Chorukova, E., Marinov, P., Umlenski, I., Survey on Theory and Applications of InterCriteria Analysis Approach, (2021) Studies in Computational Intelligence, 934, pp. 453-469. DOI:10.1007/978-3-030-72284-5_20 https://www.scopus.com/inward/record.uri?eid=2-s2.0-85111122908&doi=10.1007%2f978-3-030-72284-5_20&partnerID=40&md5=9426c5440afaebae760a84cbb188
40. Ribagin S., Lyubenova V. (2021) Metaheuristic Algorithms: Theory and Applications. In: Atanassov K.T. (eds) Research in Computer Science in the Bulgarian Academy of Sciences. Studies in Computational Intelligence, vol 934. Springer, Cham. https://doi.org/10.1007/978-3-030-72284-5_18 https://link.springer.com/chapter/10.1007/978-3-030-72284-5_18
41. Antonov, A., Analysis and detection of the degrees and direction of correlations between key indicators of physical fitness of 10-12-year-old hockey players (2019) International Journal Bioautomation, 23 (3), pp. 303-314. <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85074957141&doi=10.7546%2fijba.2019.23.3.000709&partnerID=40&md5=8e7ddc65c910092393247111a9b>
42. Sotirova, E., Vasilev, V., Bozova, G., Bozov, H., Sotirov, S., Application of the InterCriteria Analysis Method to a Dataset of Malignant Neoplasms of the Digestive Organs for the Burgas Region for 2014-2018 (2019) 2019 Big Data, Knowledge and Control Systems Engineering, BdkCSE 2019, art. no. 9010609, . <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85082306089&doi=10.1109%2fBdkCSE48644.2019.9010609&partnerID=40&md5=596e7a002f5f80aa64797b074ed8d8e9>
43. Sotirova E, Bureva V, Markovska I, Sotirov S, Vankova D. Application of the InterCriteria Analysis Over Air Quality Data. In International Conference on Flexible Query Answering Systems , LNCS 10333, Springer, SJR 0.323, 2017, pp. 226-235.(SCOPUS) https://rd.springer.com/chapter/10.1007/978-3-319-59692-1_20

Fidanova, S., Luquq, G., Roeva, O., Paprzycki, M., Gepner, P.: Ant Colony Optimization Algorithm for Workforce Planning. FedCSIS'2017, pp. 415–419. IEEE Xplorer (2017)

Е цитирана в:

44. Ribagin S., Lyubenova V. (2021) Metaheuristic Algorithms: Theory and Applications. In: Atanassov K.T. (eds) Research in Computer Science

in the Bulgarian Academy of Sciences. Studies in Computational Intelligence, vol 934. Springer, Cham. https://doi.org/10.1007/978-3-030-72284-5_18 https://link.springer.com/chapter/10.1007/978-3-030-72284-5_18

Fidanova S., Roeva O., Ganzha M., Ant Colony Algorithm for Fuzzy Transport Modelling, In proc of FedCSIS, ACSIS Vol. 21, IEEE Explorer, 2020, 237-240.

Е цитирана в:

45. Akanskha, E., Sahoo, A., Gulati, K., Jyoti, Sharma, N., Hybrid Classifier Based on Binary Neural Network and Fuzzy Ant Colony Optimization Algorithm (2021) Proceedings of the 5th International Conference on Trends in Electronics and Informatics, ICOEI 2021, art. no. 9453013, pp. 1613-1619. <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85113465293&doi=10.1109%2fIC0EI51242.2021.9453013&partnerID=40&md5=a8144c3d5061cc89d25ae61549353f55>
46. Baiou, M., Mombelli, A., Quilliot, A., Adouane, L., Zhu, Z., Algorithms for the Safe Management of Autonomous Vehicles (2021) Proceedings of the 16th Conference on Computer Science and Intelligence Systems, FedCSIS 2021, pp. 153-162. <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85117767486&doi=10.15439%2f2021F18&partnerID=40&md5=d61f582a1a1c96896611254cbb55e026>
47. Hou F., Information Flow Optimization for Practice of University Mathematics Interaction Platform Construction, Proceedings of the 5th International Conference on Electronics, Communication and Aerospace Technology, ICECA 2021Pages 1149 - 11522021 5th International Conference on Electronics, Communication and Aerospace Technology, ICECA 2021Coimbatore2 December 2021 through 4 December 2021Code 176530, DOI https://www.scopus.com/record/display.uri?eid=2-s2.0-85125370862&origin=resultslist&sort=plf-f&src=s&st1=Fidanova&sid=d3c94b946ef339479699dce872f&sot=b&sdt=b&rr=14&sl=13&s=REF%28Fidanova%29&relpos=1&citeCnt=0&searchTerm=&featureToggles=FEATURE_N

Fidanova S., Atanassov K.. Flying Ant Colony Optimization Algorithm for Combinatorial Optimization. Studia Informatica, 38, 4, Polish Information Society, 2017, ISSN:1642-0489, 31-40

Е цитирана в:

48. Cheriguene Y., Bousbaa F.Z., Kerrache C.A., Djellikh S., Lagraa N., Lahby M., Lakas A., COCOMA: a resource-optimized cooperative UAVs communication protocol for surveillance and monitoring applications (2022) *Wireless Networks*, DOI: 10.1007/s11276-022-03031-8, IF 2.701 <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85133245052&doi=10.1007%2fs11276-022-03031-8&partnerID=40&md5=212fe930e04d910c7cf4e39328a927>
49. Dafina Zoteva, N. Angelova, Generalized Nets. An Overview of the Main Results and Applications, *Studies in Computational Intelligence* 943, 2021, 177-126. <https://link.springer.com/book/10.1007/978-3-030-72284-5>
50. Ribagin S., Lyubenova V. (2021) Metaheuristic Algorithms: Theory and Applications. In: Atanassov K.T. (eds) *Research in Computer Science in the Bulgarian Academy of Sciences. Studies in Computational Intelligence*, vol 934. Springer, Cham. https://doi.org/10.1007/978-3-030-72284-5_18 https://link.springer.com/chapter/10.1007/978-3-030-72284-5_18
51. Bousbaa, F.Z., Kerrache, C.A., Mahi, Z., Tahari, A.E.K., Lagraa, N., Yagoubi, M.B. GeoUAVs: A new geocast routing protocol for fleet of UAVs (2020) *Computer Communications*, 149, pp. 259-269. <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85074193722&doi=10.1016%2fj.comcom.2019.10.026&partnerID=40&md5=325769433aacc18dce1fa7cf9ad60a65>
52. Cheriguene, Y., Djellikh, S., Bousbaa, F.Z., Lagraa, N., Lakas, A., Kerrache, C.A. and Tahari, A.E.K., 2020, September. SEMRP: an Energy-efficient Multicast Routing Protocol for UAV Swarms. In 2020 IEEE/ACM 24th International Symposium on Distributed Simulation and Real Time Applications (DS-RT) art. no. 9213700. <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85094813992&doi=10.1109%2fDS-RT50469.2020.9213700&partnerID=40&md5=1b5e43d3a51deece772e60ce0fcdf7f3>

8.3 Приноси

Приносите в тази дисертация могат да бъдат разделени на научни и научноприложни. Научните приноси касаят разработването на алгоритми на базата на метода на мравките. Научно-приложните приноси се отнасят към програмната реализация на разработените алгоритми.

Научните приноси са:

- Разработен е хибриден алгоритъм за решаване на задачата за раницата като комбинация между прилагане на метода на мравките и подходяща процедура за локално търсене. Въведено е използването на два варианта на интуиционистки размит феромон. С помощта на интеркритериален анализ е направено сравнение между вариантите на използваните алгоритми, както и на използването на стартови стратегии;
- Разработен е алгоритъм за инспектиране на GPS мрежа на основата на метода на мравките. Добавени са промени в средата на търсене. Изследвани са коректността на алгоритъма и сходството между мрежите с помощта на интеркритериален анализ;
- Разработени са алгоритми за решаване на задачата за построяване на безжична сензорна мрежа на основата на метода на мравките. Направен е анализ на чувствителността спрямо броя на използваните мравки. Анализирани са сходствата и различията на отделните алгоритми с помощта на интеркритериален анализ;
- Разработен е алгоритъм за решаване на задачата за наемане на работна сила на основата на метода на мравките. Разработени са няколко варианта на процедури за локално търсене с цел подобряване работата на алгоритъма. Направен е анализ на чувствителността на алгоритъма спрямо неговите параметри. Приложен е интеркритериален анализ;
- Разработен е алгоритъм за моделиране на пътничкопоток при наличие на разнообразни видове транспорт в едно направление. Алгоритъмът е основан на метода на мравките.

Научно-приложните приноси са:

- Направена е софтуерна реализация на хибридният алгоритъм за решаване на задачата за раницата;
- Направена е софтуерна реализация на алгоритъма за инспектиране на GPS мрежа с промени в средата на търсене;
- Направена е софтуерна реализация на алгоритъма за решаване на задачата за наемане на работна сила;
- Направена е софтуерна реализация на алгоритъм за моделиране на пътникопоток.

Резултатите от настоящата дисертация могат да се използват в най-различни области на науката, индустрията и практиката. Разработените алгоритми и тяхната програмна реализация се отнасят за практически задачи и могат да бъдат внедрени в различни отрасли на икономиката.

8.4 Декларация за оригиналност

Декларация за оригиналност на резултатите

Декларирам, че настоящата дисертация съдържа оригинални резултати, получени при проведени от мен научни изследвания. Резултатите, които са получени, описани и/или публикувани от други учени, са надлежно и подробно цитирани в библиографията.

Настоящата дисертация не е прилагана за придобиване на научна степен в друго висше училище, университет или научен институт.

Подпис:

8.5 Благодарности

Искам да изкажа моите най-искрени благодарности на всички мои колеги от Института по Информационни и Комуникационни Технологии както и на акад. Красимир Атанасов, проф. Олимпия Роева и доц. Вася Атанасова от Института по Биофизика и Биомедицинско Инженерство с които съм работила и заедно сме постигнали тези резултати.

Благодаря на колегите от секция Паралелни Алгоритми за добрата работна атмосфера, за проектите които работихме заедно, за ценните дискусии в които се родиха оригинални идеи, за постигнатите резултати в следствие на колективната ни работа.

Благодаря и на чуждестранните учени, с които съм работила. Ще спомена Енрике Алба и Габриел Люк от Испания; Марчин Папржицки и Мария Ганжа от Полша; Марко Дориго от Белгия, Антонио Мучерино от Франция.

И накрая, но не на последно място искам да благодаря на моето семейство, които винаги са ме подкрепяли, и специално на съпруга ми, който винаги е насърчавал и стимулирал всичките ми научни изследвания.

Библиография

- [1] Akuldiz I.F., Su W., Sankarasubramaniam Y., Cayrci E.: Wireless Sensor Networks: A Survey, *Computer Networks*, **38(4)**, Elsevier, 393–422, (2001)
- [2] A. EL AMARAOUI A., K. MESGHOUNI, *Train Scheduling Networks under Time Duration Uncertainty*, In proc. of the 19th World Congress of the Int. Federation of Automatic Control, 2014, 8762 - 8767.
- [3] Alba, E., Luque, G., Luna, F.: Parallel Metaheuristics for Workforce Planning. *J. Mathematical Modelling and Algorithms* **6(3)**, Springer, 509-528.(2007)
- [4] Alba E., Molina G.: Optimal Wireless Sensor Layout with Metaheuristics: Solving a Large Scale Instance, *Large-Scale Scientific Computing, Lecture Notes in Computer Science*, **4818**, Springer, 527–535, (2008)
- [5] A. A. ASSAD, *Models for Rail Transportation*, Transportation Research Part A General, **143**, 1980, 205 - 220.
- [6] Atanassov, K.: On Index Matrices, Part 1: Standard Cases. *Advanced Studies in Contemporary Mathematics* 20(2), 291–302 (2010)
- [7] Atanassov, K.: *On Intuitionistic Fuzzy Sets Theory*. Springer, Berlin (2012)
- [8] Atanassov, K., Szmidt, E., Kacprzyk, J.: On Intuitionistic Fuzzy Pairs. *Notes on IFS* 19(3), 1–13 (2013)
- [9] Atanassov, K., Mavrov, D., Atanassova, V.: Intercriteria Decision Making: A New Approach for Multicriteria Decision Making, Based

- on Index Matrices and Intuitionistic Fuzzy Sets. Issues in IFSs and GNs **11**, 1–8 (2014)
- [10] Atanassov, K., Index Matrices: Towards an Augmented Matrix Calculus, Springer, Cham, 2014.
- [11] Atanassov, K., Atanassova, V., Gluhchev, G.: InterCriteria Analysis: Ideas and Problems. Notes on Intuitionistic Fuzzy Sets 21(1), 81–88 (2015)
- [12] Atanassov, K. *Intuitionistic Fuzzy Sets*; Studies in Fuzziness and Soft Computing, 123; Springer: Heidelberg, Germany, 1999.
- [13] Atanassov, K. *Intuitionistic Fuzzy Logics*; Studies in Fuzziness and Soft Computing, 351; Springer: Heidelberg, Germany, 2017.
- [14] T. BENZAO AND Z. SHAORONG, *Optimal Design of Monitoring Networks with Prior Deformation Information*, Surv. Rev. 33, (1995), pp. 231–246.
- [15] E. BONABEAU, M. DORIGO, G. THERAULAZ, *Swarm Intelligence: From Natural to Artificial Systems*, Oxford University Press, 1999.
- [16] Campbell, G.: A two-stage stochastic program for scheduling and allocating cross-trained workers. J. Operational Research Society **62(6)**, 1038–1047. (2011)
- [17] P. CROSS AND K. THAPA, *The Optimal Design of Leveling Networks*, Surv. Rev. 25, (1979) pp. 68–79.
- [18] C. CHU, F. NING, P. HUNG AND S. CHU, *The Study of Orthometric Heighting Using GPS*, Geomat. Res. Aus. 66, (1997) pp. 55–76.
- [19] P.J. DARE AND H.A. SALEH, *GPS Network Design: Logistics Solution Using Optimal and Near-Optimal Methods*, J. of Geodesy, Vol 74, (2000) pp. 467–478.
- [20] Deb K., Pratap A., Agrawal S., Meyarivan T.: A Fast and Elitist Multiobjective Genetic Algorithm: Nsga-ii, IEEE Transactions on Evolutionary Computation, **6(2)**, 182–197, (2002).

- [21] Deneubourg J.L., Aron S., Goss S., Pasteels J.M.: The Self-Organising Exploatory Pattern of the Argentine Ant, *J. of Insect Behavior* 3, 159–168 (1990).
- [22] Dorigo, M., Di Caro, G.: The Ant Colony Optimization Meta-heuristic, In: Corne, D, Dorigo, M., Glover, F. (eds): *New Idea in Optimization*, McGraw-Hill, pp. 11–32 (1999).
- [23] O. DIAZ-PARRA, J. A. RUIZ-VANOYE, B. B. LORANCA, A. FUENTES-PENNA, R.A. BARRERA-CAMARA, *A Survey of Transportation Problems* *Journal of Applied Mathematics* Volume 2014 (2014), Article ID 848129, 17 pages.
- [24] Diffe W., Hellman M. E. : New direction in cryptography. *IEEE Trans, Inf. Theory*, IT - 36,(1976), 644–654.
- [25] *Dorigo, M. Optimization, Learning and Natural Algorithms.* Ph.D.Thesis. Politecnico di Milano, Italy, 1992 (in Italian).
- [26] *Dorigo, M., L. M. Gambardella.* Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem. – *IEEE Transactions on Evolutionary Computation*. Vol. 1, 1997, 53-66.
- [27] *Dorigo, M., V. Maniezzo, A. Colorni.* The Ant System: Optimization by a Colony of Cooperative Agents. – *IEEE Transactions on Systems, Man and Cybernetics–Part B*, Vol. 26, 1996, No 1, 29-41.
- [28] *Dorigo, M., T. Stutzle.* Ant Colony Optimization. MIT Press, 2004.
- [29] Easton, F.: Service completion estimates for cross-trained workforce schedules under uncertain attendance and demand. *Production and Operational Management* **23(4)**, 660–675. (2014)
- [30] *Fidanova, S.* Evolutionary Algorithm for Multiple Knapsack Problem. – In: *Int. Conference Parallel Problems Solving from Nature, Real World Optimization Using Evolutionary Computing*, ISBN No 0-9543481-0-9, Granada, Spain, 2002.
- [31] *Fidanova, S.* ACO Algorithm with Additional Reinforcement. M. Dorigo, G. Di Caro, M. Sampels (Eds). – In: *From Ant Colonies to Artificial Ants, Lecture Notes in Computer Science*. No 2463. ISSN 0302-9743. Germany, Springer, 2002, 292-293.

- [32] Fidanova, S. (2003). ACO algorithm for MKP using various heuristic information. In I. Dimov, I. Lirkov, S. Margenov, Z. Zlatev (Eds.), Numerical methods and applications (pp. 434-330) (LNCS 2542). Berlin: Springer-Verlag.
- [33] Fidanova S.. Ant Colony Optimization and Pheromone Importance. Computer Science, Engineering and Applications, 2003, 408-413
- [34] Fidanova S.. Ant Colony Optimization for Multiple Knapsack Problem and Model Bias. Lecture Notes in Computer Science, 3401, Springer, 2005, ISSN:0377-0427, 280-287. SJR:0.339
- [35] Fidanova, S.: Hybrid Heuristics Algorithm for GPS Surveying Problem. Numerical Methods and Applications, Lecture Notes in Computer Science **4310**, 239–248. Springer (2007)
- [36] Fidanova S.. Probabilistic Model of Ant Colony optimization for Multiple Knapsack Problem. Lecture Notes in Computer Science, 4818, Springer, 2008, 545-552. SJR:0.339
- [37] S. FIDANOVA, K. ATANASOV *Generalized Net Model for the Process of Hibrde Ant Colony Optimization* Comptes Rendus de l'Academie Bulgare des Sciences, **62**(3), 2009, 315 - 322.
- [38] Fidanova, S., Alba, E., Molina, G.: Hybrid ACO Algorithm for the GPS Surveying Problem. Large Scale Scientific Computing, Lecture Notes in Computer Science **5910**, 318–325. Springer (2010)
- [39] Fidanova S., Marinov P., Alba E.: ACO for Optimal Sensor Layout, In Proc. of Int. Conf. on Evolutionary Computing, Valencia, Spain, Joaquim Filipe and Janus Kacprzyk eds., SciTePress-Science and Technology Publications portugal, ISBN 978-989-8425-31-7, 5–9, (2010)
- [40] S. Fidanova, K. Atanassov, P. Marinov, Generalized Nets and Ant Colony Optimization, Academic Publishing House, Bulgarian Academy of Sciences, 2011.
- [41] Fidanova, S., Paprzycki, M., Roeva, O.: Hybrid GA-ACO Algorithm for a Model Parameter Identification Problem: In proc. of FedCSIS 2014 conference, pp. 413–420. IEEE Xplorer (2014)

- [42] Fidanova, S., Ilcheva, Z.: Application of Ants Ideas on Image Edge Detection. Large Scale Scientific Computing, Lecture Notes in Computer Science **9374**, pp. 200–207. Springer, (2016)
- [43] Fidanova S., Metaheuristic Method for Transport Modelling and Optimization Studies in Computational Intelligence, 648, Springer, 2016, 295 - 302.
- [44] Fidanova, S., Roeva, O., Paprzycki, M., Gepner, P.: InterCriteria Analysis of ACO Start Strategies. Proceedings of the 2016 Federated Conference on Computer Science and Information Systems, 547–550. (2016)
- [45] Fidanova, S., Roeva, O.: InterCriteria Analysis of Ant Colony Optimization Application to GPS Surveying Problems. Issues in Intuitionistic Fuzzy Sets and Generalized Nets **12**, 20–38. (2016)
- [46] Fidanova S., Roeva O., Paprzycki M., Gepner P.. InterCriteria Analysis of ACO Start Strategies. IEEE Xplorer, 2016, ISBN:ISBN 978-83-60810-90, DOI:ISBN 978-83-60810-90-3, 547-550
- [47] Fidanova, S., Luquq, G., Roeva, O., Paprzycki, M., Gepner, P.: Ant Colony Optimization Algorithm for Workforce Planning. FedCSIS'2017, pp. 415–419. IEEE Xplorer (2017)
- [48] Fidanova, S., Roeva, O., Mucherino, A., Kapanova, K.: InterCriteria Analysis of ANT Algorithm with Environment Change for GPS Surveying Problem. Artificial Intelligence:Methodology, Systems and Applications, Ch. Dachev, G. Agre eds., Lecture Notes in Artificial Intelligence **9883**, pp. 271–278. Springer (2016)
- [49] Fidanova S., Atanassov K.. Flying Ant Colony Optimization Algorithm for Combinatorial Optimization. Studia Informatica, 38, 4, Polish Information Society, 2017, ISSN:1642-0489, 31-40
- [50] Fidanova, S., Roeva, O., Atanasova, V.: Ant Colony Optimization Application to GPS Surveying Problems: InterCriteria Analysis. Intuitionistic Fuzzy Sets and Generalized Nets, Advances in Intelligent Systems and Computing (559), pp. 251–264. Springer (2018)

- [51] Fidanova S., Shindarov M., Marinov P.. Wireless Sensor Positioning Using ACO Algorithm. *Studies in Computational Intelligence*, 657, Springer, 2017, ISBN:978-3-319-41437-9, ISSN:1860-949X, 33-44.
- [52] Roeva O., Fidanova S., Paprzycki M.. Comparison of Different ACO Start Strategies Based on InterCriteria Analysis. *Recent Advances in Computational Optimization, Results of the Workshop on Computational Optimization WCO 2016, Studies of Computational optimization*, 717, Springer, 2018, ISBN:978-3-319-59860-4, 53-72.
- [53] Fidanova, S., Luquq, G., Roeva, O., Paprzycki, M., Gepner, P.: Hybrid Ant Colony Optimization Algorithm for Workforce Planning. *FedCSIS'2018, IEEE Xplorer, ACSIS Vol. 15*, 2018, 233-236.
- [54] Fidanova S., Roeva O.. InterCriteria Analyzis of Differen Variants of ACO algorithm for Wireless Sensor Network Positioning. *Lecture Notes in Computer Science*, 11189, Springer, 2019, 88-96.
- [55] Fidanova S., Luque G., Roeva O., Ganzha M., Ant Colony Optimization Algorithm for Workforce Planing: Influence of the Evaporation Parameter, In *Proceedings of FedCSIS, IEEEExplorer, ACSIS 18*, 2019, 177-181.
- [56] Fidanova S., Roeva O., Luque G., Ant Colony Optimization Algorithm for Workforce Planing: Influence of the Algorithm Parameters, *Studies in Computational Intelligence* 793, 2019.
- [57] Fidanova S.. Hybrid Ant Colony Optimization Algorithm for Multiple Knapsack Problem. *5th IEEE International Conference on Recent Advances and Innovations in Engineering (ICRAIE)*, IEEE, 2020, DOI:10.1109/ICRAIE51050.2020.9358351, 1-5
- [58] Fidanova S., Roeva O.. Multi-Objective ACO Algorithm for WSN Layout: InterCriteria Analisys. *Lecture Notice in Computer Science*, 11958, Springer, 2020, ISBN:978-3-030-410315, 474-481.
- [59] Fidanova S., Roeva O., Ganzha M., Ant Colony Algorithm for Fuzzy Transport Modelling, In *proc of FedCSIS, ACSIS Vol. 21, IEEEExplorer*, 2020, 237-240.

- [60] Fidanova S., Roeva O., Luque G., Paprzycki M., InterCriteria Analysis of Hybrid Ant Colony Optimization Algorithm for Workforce Planing, *Studies in Computational Intelligence* 838, 2020, 61-82.
- [61] Fidanova S., Luque G., New Local Search Procedure for Workforce Planning Problem, *Cybernetics and Information Technologies*, Vol. 20(6), 2020, 40-48.
- [62] Fidanova S., Atanassov K.. ACO with Intuitionistic Fuzzy Pheromone Updating Applied on Multiple Knapsack Problem. *Mathematics*, 9, 13, MDPI, 2021, ISSN:2227-7390, DOI:10.3390/math9131456, 1-7.
- [63] Fidanova S., Roeva O., Ganzha M.. InterCriteria Analyzis of Hybrid Ant Colony Optimization Algorithm for Multiple Knapsack Problem. *Annals of Computer Science and Information Systems*, 25, IEEE, 2021, ISBN:978-83-959183-6-0, ISSN:2300-5963, DOI:10.15439/2021F22, 173-180
- [64] Fidanova S.. *Ant Colony Optimization and Applications*. *Studies in Computational Intelligence*, 947, Springer, 2021, ISBN:978-3-030-67380-2, DOI:<https://doi.org/10.1007/978-3-030-67380-2>, 142
- [65] Fidanova S., Roeva O., Ganzha M., Ant Colony Algorithm for Fuzzy Transport Modelling: InterCriteriaAnalysis, *Studies in Computational Intelligence* 986, 2022, 123-138.
- [66] Fidanova S., Roeva O., Influence of ACO Evaporation Parameter for Unstructured Workforce Planning Problem, *Large Scale Scientific Computing*, LNCS 13127, 2022, 234-241.
- [67] J. G. JIN, J. ZHAO, D. H. LEE, *A Column Generation Based Approach for the Train Network Design Optimization Problem*, *J. of Transportation Research*, **50**(1), 2013, 1 - 17.
- [68] Glover, F., Kochenberger, G., Laguna, M., Wubben, T.: Selection and assignment of a skilled workforce to meet job requirements in a fixed planning period. In:MAEB'04, 636–641. (2004)
- [69] Gonzalez, R. C., Woods, R. E.: *Digital Image Processing*. Prentice-Hall, Inc. (2002)

- [70] Grzybowska, K., Kovács, G.: Sustainable Supply Chain – Supporting Tools. Proceedings of the 2014 Federated Conference on Computer Science and Information Systems 2, 1321–1329. (2014)
- [71] F. S. HANSELER, N. MOLYNEAUX, M. BIERLAIRE, AND A. STATHOPOULOS, *Schedule-based estimation of pedestrian demand within a railway station*, Proceedings of the Swiss Transportation Research Conference (STRC) 14-16 May, 2014.
- [72] B. HARVEY, D. ELFORD AND C. TURNER, *Calculation of 3D Control Surveys*, Aus. Surv. 43, (1998) pp. 109–117.
- [73] Hewitt, M., Chacosky, A., Grasman, S., Thomas, B.: Integer programming techniques for solving non-linear workforce planning models with learning. European J of Operational Research **242(3)**, 942–950. (2015)
- [74] Hu, K., Zhang, X., Gen, M., Jo, J.: A new model for single machine scheduling with uncertain processing time, J. Intelligent Manufacturing **28(3)**, 717–725. (2015)
- [75] Ikonomov, N., Vassilev P., Roeva, O.: ICRAData – Software for InterCriteria Analysis. International Journal Bioautomation, **22(2)**, (2018)
- [76] Isah, O. R., Usman, A. D., Tekanyi, A. M. S.: A Hybrid Model of PSO Algorithm and Artificial Neural Network for Automatic Follicle Classification. Int. J. Bioautomation **21(1)**, 43–58. (2017)
- [77] Jain, A. K.: Fundamentals of Digital Image Processing. Prentice-Hall, Inc. (1989)
- [78] Jourdan D.B.: Wireless Sensor Network Planing with Application to UWB Localization in GPS-denied Environments, Massachusetts Institute of Technology, PhD thesis, (2000)
- [79] Hernandez H., Blum C.: Minimum Energy Broadcasting in Wireless Sensor Networks: An ant Colony Optimization Approach for a Realistic Antenna Model, J. of Applied Soft Computing, **11(8)**, 5684–5694, (2011).

- [80] Kochemberger G., McCarl G., Wymann F.: Heuristic for general inter programming, *J. of Decision Sciences* 5, (1974), 34–44.
- [81] Konstantinidis A., Yang K., Zhang Q., Zainalipour-Yazti D.: A multi-objective Evolutionary Algorithm for the deployment and Power Assignment Problem in Wireless sensor Networks, *J. of Computer networks*, **54(6)**, 960–976, (2010)
- [82] S. KORTESIS AND A. DERMANIS, *An Application of Graph theory to the Optimization of Costs in Trilateration Networks*, *Manuscr. Geod.* 12, (1987) pp. 296–308.
- [83] S. KUANG, *On Optimal Design of Leveling Networks*, *Aus. Surv.* 38, (1993) pp. 257–273.
- [84] S. KUANG, *On Optimal Design of Tree-dimensional Engineering Networks*, *Manuscr. Geod.* 18, (1993) pp. 33–45.
- [85] A. LEICK, *GPS Satellite Surveying*, 2nd. edition., Wiley, Chichester, England. (1995).
- [86] Leguizamon, G., Michalevich, Z.: *A New Version of Ant System for Subset Problems* In: *Proceedings of Int. Conf. on Evolutionary Computations*, Washington (1999).
- [87] Lessing, L., I. Dumitrescu, T. Stutzle. A Comparison between ACO Algorithms for the Set Covering Problem. – In: *ANTS Workshop, 2004*, 1–12.
- [88] Li, G., Jiang, H., He, T.: A genetic algorithm-based decomposition approach to solve an integrated equipment-workforce-service planning problem. *Omega* **50**, 1–17. (2015)
- [89] Li, R., Liu, G.: An uncertain goal programming model for machine scheduling problem. *J. Intelligent Manufacturing* **28(3)**, 689–694. 2014
- [90] Martello S., Toth P.: A mixtures of dynamic programming and branch-and-bound for the subset-sum problem, *Management Science*, 30, (1984), 765–771.
- [91] V. K. MATHUR, *How Well do we Know Pareto Optimality?* *J. of Economic Education* **22(2)**, 1991, 172 - 178.

- [92] M. MILLER, H. DRAGERT, E. ENDO, J. FRAYMUELLER, C. GOLDINGER, H. KELSEY, E. HUMPHREYS, D. JOHNSON, R. MCCAFFREY, J. OLDOW, A. QAMAR AND C. RUBIN, *Precise Measurements Help Gauge Pacific Northwest's Earthquake Potential*, EOS Trans. Am. Geophys. Union 79, (1998) pp. 267–275.
- [93] Mlsna, Ph. A., Rodriguez, J. J.: Gradient and Laplacian-Type Edge Detection. Handbook of Image and Video Processing (Ed. Al Bovik), Academic Press, 415–431. (2000)
- [94] N. MOLYNEAUX, F. HANSELER, M. BIERLAIRE, *Modelling of train-induced pedestrian flows in rail-way stations*, Proceedings of the Swiss Transportation Research Conference (STRC) 14-16 May, 2014.
- [95] C. WORONIUK, M. MARINOV, *Simulation Modelling to Analyze the Current Level of Utilization of Sections Along Rail Rout*, J. of Transport Literature, textbf7(2), 2013, 235 - 252.
- [96] Molina G., Alba E., Talbi El-G: Optimal Sensor Network Layout Using Multi-Objective Metaheuristics, Universal Computer Science **14(15)**, 2549–2565, (2008)
- [97] I. NATIO, Y. HATANAKA, N. MANNOJI, R. ICHIKAWA, S. SHIMADA, T. YABUKI, H. TSJI AND T. TANAKA, *Global Positioning System Project to Improve Japanese Weather Earthkaik Prediction*, EOS Trans. Am. Geophys. Union 79, (1998) pp. 301–311.
- [98] Ning, Y., Liu, J., Yan, L.: Uncertain aggregate production planning. Soft Computing **17(4)**, 617–624. (2013)
- [99] Othman, M., Bhuiyan, N., Gouw, G.: Integrating workers' differences into workforce planning. Computers and Industrial Engineering **63(4)**, 1096–1106. (2012)
- [100] Paek J., Kothari N., Chintalapudi K., Rangwala S., Govindan R.: The Performance of a Wireless Sensor Network for Structural Health Monitoring, In Proc. of 2nd European Workshop on Wireless Sensor Networks, Istanbul, Turkey, Jan 31 – Feb 2, <https://escholarship.org/uc/item/9bf7f3n5>, (2004)

- [101] Mathur V.K.: How Well do we Know Pareto Optimality? J. of Economic Education **22(2)**, 172–178, (1991)
- [102] Parisio, A, Jones, CN.: A two-stage stochastic programming approach to employee scheduling in retail outlets with uncertain demand. Omega **53**, 97–103. (2015)
- [103] Pottie G.J., Kaiser W.J.: Embedding the Internet: Wireless Integrated Network Sensors, Communications of the ACM, **43(5)**, 51–58, (2000)
- [104] Pratt, W. K.: Digital Image Processing. Second ed., John Wiley & Sons, (1991)
- [105] *Reiman, M., M. Laumanns.* A Hybrid ACO Algorithm for the Capacitated Minimum Spanning Tree Problem. – In: First Int. Workshop on Hybrid Metaheuristics. Valencia, Spain, 2004, 1-10.
- [106] Roeva, O., Vassilev P., InterCriteria Analysis of Generation Gap Influence on Genetic Algorithms Performance, Advances in Intelligent Systems and Computing **401**, 301–313 (2016)
- [107] Roeva O., Fidanova S., Luque G., Paprzycki M., InterCriteria Analysis of ACO Performance for Workforce Planing Problem, Studiec in Computational Intelligence 795, 2019, 47-68.
- [108] Roeva O., Fidanova S.. Different InterCriteria Analysis of Variants of ACO algorithm for Wireless Sensor Network Positioning. Studies in Computational Intelligence, 838, Springer, 2020, ISBN:978-3-030-22723-4, 83-103.
- [109] Roeva O., Fidanova S., Ganzha M., InterCriteria Analysis of the Evaporation Parameter Influence on Ant Colony Optimization Algorithm: A Workforce Planing Problem, Studies in Computational Intelligence 920, 2021, 89-110.
- [110] Romer K., Mattern F.: The Design Space of Wireless Sensor Networks, IEEE Wireless Communications, **11(6)**, 54–61, (2004)
- [111] Saleh H.A.: Ants Can Successfully Design GPS Surveying Networks, GPS World **9**, 48–60 (2002)

- [112] Saleh H. A., and Dare P.: Effective Heuristics for the GPS Survey Network of Malta: Simulated Annealing and Tabu Search Techniques. *Journal of Heuristics* **7**(6), 533–549 (2001)
- [113] Sinha A., Zoltner A. A. : The multiple-choice knapsack problem, *J. of Operational Research*, 27, (1979), 503–515.
- [114] Soukour, A., Devendeville, L., Lucet, C., Moukrim, A.: A Memetic algorithm for staff scheduling problem in airport security service. *Expert Systems with Applications* **40**(18), 7504–7512. (2013)
- [115] *Stutzle, T. M. Dorigo.* ACO Algorithm for the Travelling Salesman Problem, – In: K. Miettinen, M. Makela, P. Neittaanmaki, J. Periaux, (Eds), *Evolutionary Algorithms in Engineering and Computer Science*. Wiley, 1999, 163-183.
- [116] *Stutzle, T., H. H. Hoos.* MAX-MIN Ant System. – In: M. Dorigo, T. Stutzle, G. Di Caro (Eds). *Future Generation Computer Systems*. Vol 16. 2000, 889-914.
- [117] Verhulst, P.F.: A Note on the Law of Population Growth. *Correspondence Mathematiques et Physiques*, **10**, 113–121, (1938) (in French).
- [118] D. WELLS, W. LINDLOHR, B. SCHAFFRIN AND E. GRAFAREND, *GPS Design: Undifferenced Carrier Beat Phase Observations and the Fundamental Differencing Theorem*, Tech. Rep. 116, Department of Geodesy and Geomatics Engineering, University of New Brunswick, Fredericton, (1987).
- [119] Werner-Allen G., Lorinez K., Welsh M., Marcillo O., Jonson J., Ruiz M., J.Lees J.: Deploying a Wireless Sensor Nnetwork on an Active Volcano, *IEEE Internet Computing* **10**(2), 18–25, , (2006)
- [120] Wolf S., Mezz P.: Evolutionary Local Search for the Minimum Energy Broadcast Problem, in C. Cotta, J. van Hemez (eds.), *VOCOP 2008, Lecture Notes in Computer Sciences* **4972**, Springer, Germany, 61–72, (2008)
- [121] Xu Y., Heidemann J., Estrin D.: Geography Informed Energy Conservation for Ad Hoc Routing, *Proceedings of the 7th*

- ACM/IEEE Annual International Conference on Mobile Computing and Networking, Italy, 70–84, (2001)
- [122] Yang, G., Tang, W., Zhao, R.: An uncertain workforce planning problem with job satisfaction. *Int. J. Machine Learning and Cybernetics*, Springer, doi:10.1007/s13042-016-0539-6, (2016)
- [123] Yuce M.R., Ng S.W., Myo N.L., Khan J.Y., Liu W.: Wireless Body Sensor Network Using Medical Implant Band, *Medical Systems* **31(6)**, 467–474, (2007)
- [124] Zadeh, L. Fuzzy Sets. *Inf. Control.* **1968**, *12*, 94–102.
- [125] Zeng, J., Li, Y.: The Use of Adaptive Genetic Algorithm for Detecting Kiwifruit’s Variant Subculture Seedling. *Int. J. Bioautomation* **21(4)**, 349–356. (2017)
- [126] Zhang, T., S. Wang, W. Tian, Y. Zhang. ACO-VRPTWRV: A New Algorithm for the Vehicle Routing Problems with Time Windows and Re-Used Vehicles Based on Ant Colony Optimization. – In: Sixth International Conference on Intelligent Systems Design and Applications. IEEE Press, 2006, 390-395.
- [127] Zhou, P., Ye, W. Q., Wang Q.: An Improved Canny Algorithm for Edge Detection. *Journal of Computational Information Systems*, **7(5)**, 1516–1523 (2011)
- [128] Zhou, C., Tang, W., Zha,o R.: An uncertain search model for recruitment problem with enterprise performance, *J Intelligent Manufacturing*, **28(3)**, Springer, 295–704. (2014)