

УНИВЕРЗИТЕТ У НОВОМ САДУ
ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА

**ПЛАНИРАЊЕ РАЗВОЈА
ДИСТРИБУТИВНИХ МРЕЖА
КОРИШЋЕЊЕМ УНАПРЕЂЕНОГ
ХЕУРИСТИЧКОГ ПРИСТУПА**

ДОКТОРСКА ДИСЕРТАЦИЈА

Ментор
Проф. др Драган Поповић

Кандидат
мр Војин Ђ. Керлета

Нови Сад
Новембар, 2014. година



КЉУЧНА ДОКУМЕНТАЦИЈСКА ИНФОРМАЦИЈА

Редни број, РБР:		
Идентификациони број, ИБР:		
Тип документације, ТД:	Монографска публикација	
Тип записа, ТЗ:	Штампана грађа	
Врста рада, ВР:	Докторска дисертација	
Аутор, АУ:	Војин Ђ. Керлета	
Ментор, МН:	Професор Драган Поповић	
Наслов рада, НР:	Планирање развоја дистрибутивних мрежа коришћењем унапређеног хеуристичког приступа	
Језик публикације, ЈП:	Српски	
Језик извода, ЈИ:	Српски / Енглески	
Земља публикавања, ЗП:	Србија	
Уже географско подручје, УГП:	Војводина	
Година, ГО:	2014	
Издавач, ИЗ:	Ауторски репринт	
Место и адреса, МА:	Нови Сад	
Физички опис рада, ФО: (поглавља/страна/ цитата/табела/слика/графика/прилога)	6 поглавља / 107 страна / 180 цитата / 11 табеле / 40 слика / 0 графика / 3 прилога	
Научна област, НО:	Електротехника	
Научна дисциплина, НД:	Електроенергетика	
Предметна одредница/Кључне речи, ПО:	дистрибутивна мрежа, једно-етапно планирање, метахеуристике, симулирано каљење, мешовито-целобројно програмирање, хибридизација	
УДК		
Чува се, ЧУ:	Библиотека ФТН, Нови Сад	
Важна напомена, ВН:		
Извод, ИЗ:	<p>У раду је представљен нови хибридни алгоритам симулираног каљења (<i>SA</i>) и мешовитог целобројног линеарног програмирања (<i>MILP</i>) за статичко планирање радијалних дистрибутивних мрежа са дистрибутивним генераторима. Овде је развијен један нови статички алгоритам који уважава: инвестиционе трошкове, трошкове губитака, трошкове прекида напајања потрошача услед кварова на гранама и дистрибутивним генераторима, као и трошкове губитака производње дистрибутивних генератора услед кварова на гранама.</p> <p>Проблем планирања развоја дистрибутивних мрежа је најпре моделован као проблем целобројног мешовитог целобројног линеарног програмирања (<i>MILP</i>) са циљем минимизације наведених трошкова. Да би се смањила комплексност проблема планирања, предложена је декомпозиција проблема на низ мањих подпроблема (локалних мрежа) које се решавају <i>MILP</i> моделом. Поступак декомпозиције и решавања итеративно је вођен и контролисан са предложеним <i>SA</i> алгоритмом који укључује механизам интензификације и диверзификације како би се постигло крајње решење.</p> <p>Применом алгоритма на реалним мрежама очекује се да нови хеуристички алгоритам генерише квалитетније планове развоја од хеуристичких алгоритама и алгоритама заснованих на вештачкој интелигенцији који су до сада развијени. Решења добијена применом развијеног алгоритма ће бити упоређена са правим глобалним оптимумом, и на основу тога ће се дефинисати његов квалитет.</p>	
Датум прихватања теме, ДП:	21.10.2010	
Датум одбране, ДО:		
Чланови комисије, КО:	Председник: др Владимир Стрезоски	
	Члан: др Жељко Поповић, др Андрија Сарић, др Драган Тасић	Потпис ментора
	Члан, ментор: др Драган Поповић	



KEY WORDS DOCUMENTATION

Accession number, ANO :	
Identification number, INO :	
Document type, DT :	Monograph publication
Type of record, TR :	Textual printed material
Contents code, CC :	Phd thesis
Author, AU :	Vojin Dj. Kerleta
Mentor, MN :	Professor Dragan Popović
Title, TI :	Development planning of distribution networks using an advanced heuristic approaches
Language of text, LT :	Serbian
Language of abstract, LA :	Serbian / English
Country of publication, CP :	Serbia
Locality of publication, LP :	Vojvodina
Publication year, PY :	2014
Publisher, PB :	Author's reprint
Publication place, PP :	Novi Sad
Physical description, PD : (chapters/pages/ref./tables/pictures/graphs/appendixes)	6 chapters / 107 pages / 180 ref / 11 tables / 40 pictures / 0 graphs / 3 appendix
Scientific field, SF :	Electrical engineering
Scientific discipline, SD :	Power systems
Subject/Key words, S/KW :	distribution network, single-period planning, metaheuristics, simulated annealing, mixed-integer programming, hybridization
UC	
Holding data, HD :	Library of Faculty of technical sciences, Novi Sad
Note, N :	
Abstract, AB :	<p>In this paper, we present a new hybrid algorithm of simulated annealing and mixed integer linear programming for static planning radial distribution networks with distribution generators. It was developed a new static algorithm that takes into account: investment costs, losses, costs a power of consumers due to faults on the branches and distribution generators, as well as the cost of loss of production of distribution of generators due to faults on the branches.</p> <p>The problem of planning the development of the distribution network is first modeled as a mixed integer problem of integer linear programming with the goal of minimizing those costs. To reduce the complexity of the planning problem, the proposed decomposition problem in a number of smaller sub-problems (local network) which are dealt model. The process of decomposition and solving iterativno is managed and controlled with the proposed algorithm, which includes a mechanism of intensification and diversification to achieve a final solution.</p> <p>By applying the algorithm on real networks, it is expected that new heuristic algorithm generates better plans for the development of heuristic algorithms and algorithms based on artificial intelligence that have been developed. Solutions obtained using the developed heuristic algorithm will be compared with the real global optimum, and on that basis will also define their quality.</p>
Accepted by the Scientific Board on, ASB :	21.10.2010
Defended on, DE :	
Defended Board, DB :	
President:	Phd Vladimir Strezoski
Member:	Phd Željko Popović, Phd Andrija Sarić, Phd Dragan Tasić
Member, Mentor:	Phd Dragan Popović
	Menthor's sign

САДРЖАЈ

1. УВОД	3
1.1. ПРОЦЕС ПЛАНИРАЊА РАЗВОЈА ДИСТРИБУТИВНИХ МРЕЖА	4
1.2. ЦИЉ И СТРУКТУРА ДИСЕРТАЦИЈЕ	5
2. ПРЕГЛЕД АЛГОРИТАМА И МОДЕЛА ЗА ПЛАНИРАЊЕ РАЗВОЈА ДИСТРИБУТИВНИХ МРЕЖА	8
2.1. КАТЕГОРИЗАЦИЈА АЛГОРИТАМА И МОДЕЛА ЗА ПЛАНИРАЊЕ РАЗВОЈА ДИСТРИБУТИВНИХ МРЕЖА	9
2.2. ДЕТЕРМИНИСТИЧКИ АЛГОРИТМИ И МОДЕЛИ ЗА РЕШАВАЊЕ СТАТИЧКОГ ПРОБЛЕМА ПЛАНИРАЊА ДИСТРИБУТИВНИХ МРЕЖА	10
2.2.1 Математички засновани оптимизациони модели	10
2.2.2 Хеуристички засновани оптимизациони модели	12
2.2.3 Метакхеуристички засновани оптимизациони модели	13
2.2.3.1 Алгоритам симулираног каљења	14
2.2.3.2 Алгоритам роја честица	14
2.2.3.3 Генетски алгоритам	16
2.2.3.4 Мрављи алгоритам	19
2.2.3.5 Табу претраживање	21
2.2.3.6 Резиме	23
3. ХИБРИДНИ АЛГОРИТАМ СИМУЛИРАНОГ КАЉЕЊА И МЕШОВИТОГ-ЦЕЛОБРОЈНОГ МАТЕМАТИЧКОГ ПРОГРАМИРАЊА	24
3.1. ФОРМУЛАЦИЈА ПРОБЛЕМА	24
3.2. ПРИСТУП РЕШАВАЊУ ПРОБЛЕМА	25
3.2.1. Модел мешовитог-целобројног линеарног програмирања	25
3.2.2. Алгоритам симулираног каљења	29
3.2.3. Модел линеаризације трошкова губитака	33
4. ПРИМЕРИ ПРИМЕНЕ	35
4.1. ПРИМЕР МРЕЖЕ I	35
4.2. ПРИМЕР МРЕЖЕ II	40
4.3. ПРИМЕР МРЕЖЕ III	41
4.4. ПРИМЕР МРЕЖЕ IV	42
5. ЗАКЉУЧАК	45
ПРИЛОГ 1. МЕТАХЕУРИСТИКЕ У ПЛАНИРАЊУ РАЗВОЈА ДИСТРИБУТИВНИХ МРЕЖА	46
П1.1. ОПШТЕ О МЕТАХЕУРИСТИКАМА	46
П1.1.1. Математичка оптимизација	46
П1.1.2. Хеуристике	48
П1.1.3. Метакхеуристике	49

<i>П1.2. НАЈЧЕШЋЕ КОРИШТЕНЕ МЕТАХЕУРИСТИКЕ У ПЛАНИРАЊУ РАЗВОЈА ДИСТРИБУТИВНИХ МРЕЖА</i>	54
<i>П1.2.1. Алгоритам роја честица</i>	54
<i>П1.2.2. Генетски алгоритам</i>	60
<i>П1.2.3. Мрављи алгоритам</i>	69
<i>П1.2.4. Алгоритам табу претраге</i>	75
<i>П1.2.5. Алгоритам симулираног каљења</i>	82
ПРИЛОГ 2. ХЕУРИСТИЧКЕ ТЕХНИКЕ У ПРЕДЛОЖЕНОМ АЛГОРИТМУ	94
<i>П2.1. ТЕХНИКА ИЗМЕНЕ ГРАНА</i>	94
<i>П2.2. КОНЦЕПТ ЛОКАЛНЕ МРЕЖА</i>	95
ПРИЛОГ 3. СОФТВЕРСКА ПОДРШКА ПРЕДЛОЖЕНОГ АЛГОРИТМА	97
<i>П3.1. УВОД</i>	97
<i>П3.2. ПРОГРАМСКИ ПАКЕТ <i>MATLAB</i></i>	97
<i>П3.3. ПРОГРАМСКИ ПАКЕТ <i>TOMLAB</i></i>	98
<i>П3.4. ПРОГРАМСКИ ПАКЕТ <i>MICROSOFT EXCEL</i></i>	99
6. ЛИТЕРАТУРА	100

1. УВОД

Електроенергетски систем (ЕЕС) је један од најсложенијих система које је створило људско друштво. ЕЕС укључује стотине хиљада компоненти: генератора, трансформатора, далековода, контролне и заштитне опреме итд. Изградња електроенергетских система и њихово функционисање, као и одржавање захтевају огромна новчана средства која се мере у милијардама долара. Услови рада ЕЕС-а континуирано варирају у времену - појављују се нови купци и објекти електроенергетског система, цене расту, мења се законодавство и сл. Поред тога, стално се мењају временски услови (нпр. температуре и брзине ветра) који значајно утичу на рад система. Скупи објекти и елементи имају ограничен век трајања (неколико деценија). Све наведено мотивише потребу да се процењују услови који могу настати у прилично далекој будућности. Истовремено, изградња неких објеката захтева значајне инвестиције и одређено време изградње. Стога, грешке у планирању доводе до погрешних одлука које се не могу исправити и које брзо могу да доведу до значајних финансијских губитака.

Значај електричне енергије за националне економије, високи инвестициони трошкови и потенцијални велики губитци у случају грешке у планирању, подстичу развој великог броја робустних и флексибилних метода и алата за решавање проблема планирања, који би требало да дају квалитетна решења односно квалитетне сценарије развоја. Проблем планирања развоја дистрибутивних мрежа се разматра и унапређује већ више од 50 година. Порастом значаја електричне енергије за националне економије расли су и напори да се нађе план односно оптимална стратегија развоја дистрибутивних мрежа. Ови напори су били условљени пре свега чињеницом да се у индустријски развијеним земљама скоро половина инвестиција у електроенергетском систему трошила у дистрибутивним мрежама [1]. Поред тога, последњих година се јавља нови светски тренд промена у електроенергетском сектору који је познат под називом "паметне мреже" (*smart grid*), а који има знатан утицај на рад електроенергетских дистрибутивних система [2]-[7].

Циљеви нове електроенергетске политике постају трендови минимизације штетних гасова, минимизација губитака, економичност, поузданост, квалитет и доступност снабдевања електричном енергијом. "Паметне мреже" у циљу обезбеђивања ових циљева, применом модерних и информационих и комуникационих технологија, интегришу понашање како потрошача, тако и генератора (централних и дистрибутивних) у систему. Нови концепт између осталог подразумева употребу, као и анализу утицаја дистрибуиране производње у систему, односно дистрибутивних генератора у дистрибутивној мрежи. Утицај дистрибутивних генератора се не манифестује само на губитке, него и на будућа унапређења у мрежи, односно на инвестиционе трошкове изградње и/или појачања елемената у мрежи. Из наведених разлога понашање и утицај дистрибутивних генератора на ЕЕС добија на значају, што подразумева и потребу уважавања дистрибутивних генератора у процесу планирања развоја дистрибутивне мреже.

1.1. ПРОЦЕС ПЛАНИРАЊА РАЗВОЈА ДИСТРИБУТИВНИХ МРЕЖА

Проблем планирања развоја дистрибутивних мрежа се најједноставније може формулисати као проблем доношења одлуке у вези три фундаментална питања:

- **шта** треба унапредити (појачати и/или изградити и/или заменити) у разматраној дистрибутивној мрежи,
- **где** у дистрибутивној мрежи треба урадити унапређења, и
- **када** та унапређења треба урадити да би се испунили жељени (дефинисани) циљеви уз задовољавање скупа ограничења.

Полазиште планирања унапређења дистрибутивне мреже је постојећа мрежа која је под утицајем спољних фактора. Једном када се идентификује да су дефинисани циљеви (критеријуми) током неког разматраног периода на било који начин неадекватни, време је да се започне процес планирања.

Процес, односно глобални алгоритам планирања развоја дистрибутивних мрежа, се може представити на следећи начин [8]:

- корак 1. Идентификација проблема: Експлицитно дефинисати опсег и границе проблема.
- корак 2. Одређивање циљеве: Шта су циљеви које треба остварити ? Шта треба минимизирати?
- корак 3. Идентификација алтернатива : Које су опције на располагању?
- корак 4. Процена (вредновање) алтернатива: Проценити све опције на основу дефинисаних критеријума.
- корак 5. Избор најбоље алтернативе: Изабрати опције које најбоље задовољавају циљеве у вези са проблемом.

У првом кораку се врши идентификација проблема, односно дефинише се проблем који треба решити (нпр. одређивање оптималне локације напојних трансформаторских станица, одређивање оптималне локације, типа и пресека система или појединачних водова, и сл.).

У другом кораку се врши одређивања циљева односно дефинише оно шта се жели постићи решавањем датог проблема. Ови циљеви могу бити разноврсни, али је то најчешће минимизација (новчаних) трошкова развоја дистрибутивне мреже. Најчешће се разматрају следећи трошкови [1]:

- трошкови изградње и одржавања елемената дистрибутивне мреже (водова, напојних трансформаторских станица, итд.),
- трошкови губитака снаге и енергије на елементима дистрибутивне мреже,
- трошкови услед прекида напајања (нпр. трошкови неиспоручене електричне енергије),
- трошкови везани за заштиту животне средине и сл.

Кораци од три до пет се најчешће решавају једновремено неком од оптимизационих процедура, која би требало да даје (суб)оптимално решење. Да би се то могло извести потребно је извршити математичку формулацију проблема који се решава, односно дефинисати математички модел који описује реалан проблем. Како такав математички модел тешко може бити тачан у целости, потешко је извршити одређене апроксимације. Од суштинске је важности да такав модел може да генерише сва допустива (остварљива) решења проблема

планирања (све алтернативе/опције), да сва решења (опције) вреднују на исти начин, као и да коришћене апроксимације и упрошћавања не утичу на избор најбољег решења.

Проблем планирања по својој природи припада групи мешовитих-целобројних (0-1) проблема, из разлога што се одлуке (**шта**, **где** и **када**) описују целобројним (бинарним) вредностима, док се величине као што су напони чворова и токови струја/снага описују континуалним вредностима. Ова група проблема чак и када је реч о линеарним мешовитим целобројним проблемима, припада класи НП-тешких проблема. То значи да се проблем планирања реалне дистрибутивне мреже не може решити у разумном времену, односно не може се наћи глобално оптимално решење егзактним алгоритмом у разумном времену [9]-[11]. Наиме, проблем планирања реалне дистрибутивне мреже представља проблем велике димензионалности и рачунске сложености, који се математичким алгоритмима оптимизације не може решити у разумном времену. Проблем планирања, као што је већ речено, има и временску димензију (**када** нешто унапредити), па се може формулисати као вишеетапни (динамички) проблем. Такође, неизвесност везана за неке информације у мрежи, нпр. за величину будућих оптерећења у потрошачким чворовима, чини да проблем планирања постане још комплекснији, јер то има за последицу постојања већег броја могућих будућности, односно будућих нивоа оптерећења у мрежи која је предмет планирања.

Из наведених разлога, постоји реална потреба за смањивањем комплексности проблема који се решава. Најчешће средство за то је да се сложени реални вишеетапни проблем декомпонује на више једноставнијих (статичких) подпроблема. Наиме, статички алгоритми решавају проблем развоја дистрибутивних мрежа у само једној етапи. Решавањем више оваквих статичких проблема, неким од декомпозиционих приступа (о којима ће у прегледу литературе бити више речи), генерише се решење за комплетан проблем велике димензионалности. Из тих разлога, на значају добијају алати који ефикасно решавају статичке проблеме. Такође, ови алати су присутни и код проблема планирања дистрибутивних мрежа који уважавају неизвесност [12]. Све то је и један од мотива да управо тема ове дисертације буде конституисање једног новог статичког алгоритма који решава проблем планирања дистрибутивних мрежа. Даље, и статички проблеми планирања дистрибутивних мрежа реалних (већих) димензија, припадају групи НП тешких проблема. Због тога се и за решавање статичког проблема планирања користе одговарајући декомпозициони приступи који имају за циљ смањивање рачунске комплексности проблема с једне, и генерисање квалитетних решења (блиских глобалном оптимуму) с друге стране.

1.2. ЦИЉ И СТРУКТУРА ДИСЕРТАЦИЈЕ

У овој дисертацији је предложен нови хибридни алгоритам симулираног каљења (*SA*) и мешовитог-целобројног линеарног програмирања (*MILP*) за статичко планирање радијалних дистрибутивних мрежа са дистрибутивним генераторима. Предложени алгоритам уважава:

- инвестиционе трошкове,
- трошкове губитака,
- трошкове прекида напајања потрошача услед кварова на гранама и дистрибутивним генераторима, као и

- трошкове губитака производње дистрибутивних генератора услед кварова на гранама.

Проблем планирања развоја радијалних дистрибутивних мрежа је моделован као проблем мешовитог-целобројног линеарног програмирања (*MILP*) са циљем минимизације инвестиционих трошкова, трошкова губитака, трошкова прекида напајања потрошача услед кварова на гранама и кварова на дистрибутивним генераторима, као и трошкова губитака производње дистрибутивних генератора услед кварова на гранама. Да би се смањила комплексност проблема планирања, предложена је декомпозиција проблема на низ мањих подпроблема (локалних мрежа) које се решавају *MILP* моделом. Поступак декомпозиције и решавања итеративно је вођен и контролисан предложеним *SA* алгоритмом који укључује механизам интензификације и диверзификације како би се постигло најбоље решење.

Применом предложеног алгоритма на планирање развоја радијалних дистрибутивних мрежа у присуству дистрибутивних генератора је показано да нови приступ генерише квалитетније планове развоја дистрибутивних мрежа од хеуристичких алгоритама и алгоритама заснованих на вештачкој интелигенцији који су до сада развијени. Такође је показано да предложени приступ може дати иста решења, као и модел заснован на математичкој оптимизацији (*MILP* модел).

Детаљан преглед досадашњих приступа (алгоритама и модела) за решавање проблема планирање развоја дистрибутивних мрежа је дат у поглављу 2.

У поглављу 3 је описан хибридни алгоритам симулираног каљења и математички модел мешовитог-целобројног линеарног програмирања за планирање развоја радијалних дистрибутивних мрежа у присуству дистрибутивних генератора, уз уважавање инвестиционих трошкова, трошкова губитака, трошкова прекида напајања потрошача услед кварова на гранама и дистрибутивним генераторима, као и трошкова губитака производње дистрибутивних генератора услед кварова на гранама. Такође је описан модел линеаризације трошкова губитака који се користио у предложеном алгоритму.

У поглављу 4 су дати примери примене предложеног хибридног алгоритма, као и нумерички резултати који показују да предложени приступ даје квалитетнија решења од до сада предложених модела заснованих на метахеуристикама, као и решења једнаког квалитета као и математички засновани оптимизациони модели.

Закључци и правци будућег рада и истраживања у овој области су дати у поглављу 5, а списак коришћене литературе у поглављу 6.

У прилогу 1. је дато генерално излагање о метахеуристикама код планирања дистрибутивних мрежа. Најпре је представљена општа дефиниција и подела оптимизационих метода. Затим су на општи начин представљене познате метахеуристике са припадајућим особинама и параметрима, и то: генетски алгоритми, мрављи алгоритми, алгоритми роја честица и алгоритми табу претраге. Такође, приказани су и неки од примера примене наведених метахеуристика код решавања проблема планирања дистрибутивних мрежа. На крају овог прилога, детаљно је описан алгоритам симулираног каљења у свом општем облику.

У прилогу 2. су описане хеуристичке технике које су кориштене у предложеном хибридном алгоритму, и то: техника измене грана, као и концепт локалне мреже.

На крају дисертације, у прилогу 3. је укратко описана коришћена софтверска подршка предложеног хибридног алгоритма, коју чине програмски пакети: *MATLAB*, *TOMLAB* и *MICROSOFT EXCEL*.

2. ПРЕГЛЕД АЛГОРИТАМА И МОДЕЛА ЗА ПЛАНИРАЊЕ РАЗВОЈА ДИСТРИБУТИВНИХ МРЕЖА

Сваки алгоритам који решава оптимизациони проблем, међу којима је и проблем планирања развоја дистрибутивних мрежа, има своју сложеност која се описује са два важна ресурса за решање проблема, а то су: време и простор. Сложеност алгоритма је обично нека функција која повезује величину проблема који се решава са потребним бројем корака извршавања алгоритма да дати проблем реши (временска сложеност), или потребном количином меморијских локација (просторна сложеност). За проблеме из класе НП-тешких проблема не постоји алгоритам који решава проблем (налази глобални оптимум) у полиномалном времену, односно не постоји алгоритам код кога се зависност између времена потребног за решавање проблема и величине проблема може описати полиномом одређеног степена [9]-[11]. Ова зависност је најчешће експоненцијална облика c^n , где је c константа већа од 1, а n представља величину проблема. Уколико се користи велика O нотација ова зависност се може описати како $O(c^n)$ [9]-[11],[15]. Из [15] је преузета табела 2.1. која показује различите облике комплексности проблема који се решавају. Види се да порастом димензије проблема расте и време рада алгоритма, али на различит начин у зависности од проблема који се решава, односно његове комплексности (експоненцијалне или полиномалне). Такође се уочава "временска експлозија" код проблема експоненцијалне комплексности.

И поред значајног напретка рачунске технике алгоритми који врше претрагу комплетног простора решења, односно који гарантују налажење глобалног оптимума, нису у стању да у реалном времену реше огроман број оптимизационих проблема из класе НП-тешких проблема, што значи да се за такве проблеме морају тражити неке другачије методе за решавање.

Табела 2.1. Време рада алгоритама различите комплексности у функцији величине проблема [15]

Комплексност	$n=10$	$n=20$	$n=30$	$n=40$	$n=50$
$O(n)$	0,00001 s	0,00002 s	0,00003 s	0,00004 s	0,00005 s
$O(n^2)$	0,0001 s	0,0004 s	0,0009 s	0,0016 s	0,0025 s
$O(n^5)$	0,1 s	0,32 s	24,3 s	1,7 min	5,2 min
$O(2^n)$	0,001 s	1 s	17,9 min	12,7 dana	35,7 god
$O(3^n)$	0,059 s	58 min	6,5 god.	385500 god	2x1 010

Управо из наведених разлога се за решавање проблема из класе НП-тешких проблема прибегава хеуристичним алгоритмима, који не гарантују налажење глобалног оптималног решења, али у већини случајева дају решења која су блиска оптималном.

Одлуке које се приликом планирања доносе (**шта** проширити, **где** проширити и **када** проширити), се представљају као бинарне променљиве (0-1 променљиве), док се континуалне величине (струје, напони, снаге) представљају континуалним

променљивима. Из тог разлога проблем планирања развоја дистрибутивних мрежа припада скупу мешовитих (0-1) целобројних проблема. Ови проблеми, чак и када је реч о линеарним мешовитим целобројним проблемима, припадају класи НП-тешких проблема [10]. За решавање овако комплексног проблема развијени су различити хеуристички и математички модели и приступи, чији преглед, као и предлог новог алгоритма за планирање развоја дистрибутивних мрежа, је дат у наставку.

2.1 КАТЕГОРИЗАЦИЈА АЛГОРИТАМА И МОДЕЛА ЗА ПЛАНИРАЊЕ РАЗВОЈА ДИСТРИБУТИВНИХ МРЕЖА

Алгоритми који решавају проблем планирања развоја дистрибутивних мрежа могу се класификовати по више критеријума, али се издвајају два основна, а то су:

- да ли уважавају постојање неизвесности (најчешће у погледу будућих оптерећења у потрошачким чворовима у мрежи), или подразумевају да постоји само једна будућност (детерминистички модели) за коју се решава проблем планирања.
- да ли разматрају проблем планирања развоја дистрибутивне мреже у једној етапи (статички модели), или уважавају више етапа у будућности (динамички модели).

Алгоритми који не уважавају постојање неизвесности називају се детерминистички алгоритми, и они решавају проблем планирања развоја дистрибутивне мреже тако што узимају у обзир само "сигурну" једну могућу будућност за коју решавају проблем планирања у складу са дефинисаним циљевима. С друге стране алгоритми који уважавају постојање неизвесности (најчешће будућих оптерећења у потрошачким чворовима) раде са више могућих будућности што доста отежава решавање проблема планирања. Алгоритми који уважавају постојање неизвесности моделују проблем планирања ближе реалности, јер ниво неизвесности нарочито будућих оптерећења у потрошачким чворовима, реално постоји, и исти расте уколико се планирањем иде даље у будућност [1],[24]-[26]. Овакви проблеми планирања дистрибутивних мрежа код којих се уважава неизвесност, могу се свести на већи број детерминистичких проблема као што је предложено у [12]. Све то даје на значају изради алата који ефикасно решавају детерминистички проблем, а што је у овој тези и урађено.

По броју етапа планирања развоја дистрибутивне мреже алгоритми се деле на: статичке и динамичке. Статички алгоритми решавају проблем развоја дистрибутивних мрежа у једној етапи. Динамички алгоритми решавају проблем планирања развоја дистрибутивне мреже у више етапа, што представља природан начин развоја дистрибутивне мреже и омогућава доношење одлука када нешто треба унапредити. Ови алгоритми дају конзистентније и економичније планове развоја дистрибутивних мрежа од статичких алгоритама. Динамички (вишеетапни) алгоритми су комплекснији и математички сложенији од статичких алгоритама, пре свега због постојања јаким интеракција (међузависности) између појединачних етапа. Декомпозициони приступи који се најчешће користе су засновани на концепту попуњавања унапред (*forward fill – in*) [8],[21] и попуњавања уназад (*backward pull – out*) [27]-[31].

Овим је такође истакнут значај дефинисања ефикасног алгоритма за решавање статичког детерминистичког проблема планирања, што је и основни циљ ове тезе.

У наставку је дат преглед детерминистичких алгоритама и модела који решавају проблем једноетапног планирања развоја дистрибутивних мрежа.

2.2. ДЕТЕРМИНИСТИЧКИ АЛГОРИТМИ И МОДЕЛИ ЗА РЕШАВАЊЕ СТАТИЧКОГ ПРОБЛЕМА ПЛАНИРАЊА РАЗВОЈА ДИСТРИБУТИВНИХ МРЕЖА

Као што је већ поменуто, једноетапни или статички модели решавају проблем планирања развоја дистрибутивних мрежа у једној етапи, односно доносе одлуку типа шта и где за само једну етапу (хоризонтну годину). При томе ови модели узимају у обзир прогнозирана оптерећења у потрошачким чворовима дистрибутивне мреже која ће се појавити у тој етапи (хоризонтној години), не уважавајући њихову потенцијалну неизвесност по питању износа наведених оптерећења у будућности.

Наведени статички модели се могу поделити у три велике групе:

1. математички засновани оптимизациони модели [32]-[55],
2. хеуристички засновани оптимизациони модели [56]-[65]
3. метахеуристички засновани оптимизациони модели [27],[28],[66]-[108].

2.2.1 Математички засновани оптимизациони модели

У овој групи се налазе модели који су засновани на линеарном програмирању [32]-[37], мешовитом-целобројном линеарном програмирању [38]-[49] и динамичком програмирању [50]-[55].

Међу првим математички заснованим оптимизационим моделима су модели засновани на линеарном програмирању. Они су се појавили још 60-их и 70-их година прошлог века када је започела шира примена оптимизационих техника у планирању развоја дистрибутивних мрежа. Овакви модели су радили на основу линеаризације свих трошкова у мрежи: фиксних (трошкови одржавања и инвестициони трошкови) и варијабилних (трошкови губитака). Линеаризација у овим моделима је вршена према концепту предложеном у [8]. Предложена линеаризација успоставља линеарну веза између токова снага (струја) и трошкова губитака, као и фиксних трошкова у дистрибутивној мрежи. Из тих разлога овакви линеарни модели су у стању да решавају проблеме реалних димензија без великих захтева по питању времена и рачунарских перформанси. Са друге стране, мана модела заснованих на линеарном програмирању се огледа у томе што се линеаризацијом трошкова уноси велика непрецизност у опису проблема који се решава, што за последицу има и "лош" квалитет добијених резултата. Такође, линеарни модели нису у стању да експлицитно уваже ограничење радијалности и напонска ограничења. Из тих разлога резултати добијени линеарним моделима захтевају додатну обраду како би се наведена ограничења уважила. Ове пост-оптимизационе процедуре су често комплексније и изискују више рачунарског времена од самог главног оптимизационог модела [8].

Као што је већ спомињано, проблем планирања развоја дистрибутивних мрежа припада групи мешовитих-целобројних (0,1) проблема. Дискретну природу проблема планирања чине одлуке (градити или не градити, појачати или непојачати) које се математички представљају бинарним променљивим. Са друге стране величине као што су струје и напони представљају континуалне величине па се тако и у моделима математички репрезентују. Из ових разлога се веома често користе модели мешовитог-целобројног линеарног програмирања (*MILP*) за решавање проблема планирања дистрибутивних мрежа [38]-[49]. За решавање

целобројних проблема користи се техника грана и граница (*branch and bound*). Како је реч о техници која врши претрагу свих допустивих решења, она гарантује да су добијени резултати глобални оптимум проблема који се решава. Такође се поред ове технике користе и друге технике као што су гранање и одсецање (*branch and cut*), и гранање и цене (*branch and price*) и сл. које могу да решавају проблеме планирања развоја дистрибутивних мрежа малих и средњих димензија, у разумном временском периоду [109],[110]. Поједини *MILP* модели не уважавају експлицитно ограничења радијалности, [23],[38]-[42], као ни напонска ограничења [23],[38]-[49]. У *MILP* моделима се губици третирају као линеарни, тако што се врши линеаризација квадратне зависности трошкова губитака од струје у гранама дистрибутивне мреже [8]. У највећем броју случајева линеаризација се врши тако да не повећава број целобројних променљивих, а тиме ни комплексност модела, тако што се квадратна зависност трошкова губитака од струје грана, "грубо" линеаризовала само са једним сегментом. У [42] је предложена линеаризација са више линеарних сегмената, уз увођење додатних целобројних променљивих, што повећава комплексност проблема. Оваква апроксимација тачније описује губитке у односу на "грубу" апроксимацију само са једним линеарним сегментом. Такође, постоји и велики број статичких модела [38]-[45] који не уважавају могућност реконфигурације дистрибутивне мреже, односно прерасподелу оптерећења у дистрибутивној мрежи.

Проблем планирања развоја дистрибутивних мрежа спада у проблеме који се могу формулисати у терминима динамичког програмирања [111]-[113]. Да би то било могуће потребно је дефинисати фазне променљиве, променљиве стања и променљиве одлука, тако да буде задовољен Белманов принцип оптималности [113]. На тај начин је могуће дати проблем поделити (декомпоновати) у више фаза, решавати га секвенцијално, и у свакој од фаза донети одлуке. У [50]-[55] је предложен низ модела динамичког програмирања за решавање проблема планирања који решавају статички проблем планирања развоја дистрибутивних мрежа. У њима је коришћен модел оптималне алокације ресурса формулисан као проблем динамичког програмирања [111]-[113], и то за одређивање пресека на радијалном воду [50], као и за одређивање оптималне конфигурације и пресека система водова [52]. Слични модели су предложени у [51],[53] за планирање развоја руралне, неупетљане дистрибутивне мреже. Модели засновани на динамичком програмирању због веома великог броја стања које треба да претраже и процене, нису способни да решавају проблем планирања дистрибутивних мрежа, средње и велике димензије, у разумном времену [1]. Због тога се користе квазидинамички модели [54],[55], где се предлаже приступ поделе једног великог проблема (мреже) на више мањих подпроблема (подмрежа - локалних мрежа). Квазидинамички алгоритам се користи за дефинисање одговарајућег броја секвенци (низа) подпроблема које је потребно решити. Сваки од генерисаних подпроблема се решава применом алгоритма мешовитог-целобројног линеарног програмирања. На тај начин се добија на ефикасности алгоритма, али губи на ефективности, односно добијена решења су "задовољавајуће" добра, али не представљају глобални оптимум. То је последица чињенице да се кроз декомпозицију проблема на подмреже губи способност претраге комплетног простора потенцијалних решења, те су добијена решења само "задовољавајуће" добра.

За све наведене моделе засноване на математичкој оптимизацији за решавање статичког проблема планирања важи да исти моделују само трошкове инвестиција

и трошкове губитака, док трошкови услед прекида (односно трошкови услед неиспоручене електричне енергије) нису моделовани. С друге стране то су једини модели који гарантују добијање глобалног оптималног решења. Основна слабост оваквих модела за решавање проблема планирања је у томе што су веома рачунарски захтевни и комплексни па нису у стању да решавају проблеме реалних димензија. Управо из свих ових разлога развијени су нови хеуристички и метахеуристички алгоритми за решавање проблема планирања који превладавају наведене слабости, и који имају способност да решавају мреже реалних димензија дајући задовољавајућа решења у разумном времену.

2.2.2 Хеуристички засновани оптимизациони модели

Велики развој хеуристичких метода је кренуо последњих 40 година развојем рачунарства и комбинаторне оптимизације. Основна идеја коришћења хеуристика у проблему планирања је у томе да се раздвоје променљиве одлука (нпр. стања расклопних уређаја у мрежи) од променљивих стања (нпр. струје грана и напони у потрошачким чворовима). Почетно решење од ког се полази је допустиво решење (нпр. радијална конфигурација мреже) које је дефинисано променљивима одлуке (уклопно стање расклопних уређаја у мрежи). Затим се за овакво почетно решење израчунавају променљиве стања (нпр. струје грана и напони у потрошачким чворовима). При томе се користи неки алгоритам претраге да би се нашла оптимална конфигурација. Један од таквих популарних алгоритама је заснован на унапређењу решења кроз локалну претрагу [15]. То је итеративна процедура која полазећи од почетног допустивог решења врши итеративно унапређивање истог, примењујући низ локалних модификација. Кроз сваку итерацију алгоритам тежи да локалном модификацијом пронађе боље допустиво решење од текућег. Ново допустиво решење се назива још и суседно решење, јер се само незнатно разликује од претходног решења. Ова процедура се понавља све док алгоритам више не може да пронађе боље суседно решење од текућег, односно када су сва суседна решења лошија од текућег. То значи да је алгоритам пронашао локални оптимум.

Такође, једна често коришћена техника локалног претраживања у оквиру хеуристичких алгоритама је техника измене грана (*branch – exchange*) [56]-[64]. Овај алгоритам у општем случају се може свести на следеће кораке:

- полази се од једног допустивог почетног решења (радијална структура мреже),
- затвара се један расклопни уређај тако да се формира петља у мрежи,
- отвара се један расклопни уређај из петље како би се добила радијална структура,
- уколико оваква модификација унапређује текуће решење онда ново решење постаје текуће, а уколико не ново решење се напушта,
- ови се кораци извршавају све док циљна функција не може више да се унапреди.

Један алгоритам технике измене грана кориштен у литератури детаљно је описан у прилогу у поглављу П2.

Такође, у решавању проблема статичког планирања користи се и техника минималних струја [64], која омогућује добијање почетног допустивог решења које се затим техником измене грана унапређује генерисањем суседних решења [65].

Основна предност хеуристичких метода се огледа у следећем:

- могу да решавају проблеме великих димензија уз мале рачунарске захтеве,
- могу да решавају проблеме у којима постоје нелинеарности или појава више критеријума.

У слабости хеурстичког приступа спада немогућност широке претраге проблема због "заглављивања" у локалном оптимуму, па то за последицу има добијање само "задовољавајућих" решења (локални оптимум). Из тог разлога постоји велика зависност добијених резултата од почетног решења.

Како би се ове слабости избегле прибегло се коришћењу нових алгоритама који имају способност деградације текућег решења са којим оперишу, што омогућава избегавање локалног оптимума. Из тог разлога ови алгоритми користе неку врсту стохастичког претраживања и називају се метахеуристички алгоритми.

2.2.3 Метахеуристички засновани оптимизациони модели

Метахеуристике по броју решење над којим раде, делимо на метахеуристике које раде са популацијом решења (популационе или P -метахеуристике), и на метахеуристике које раде са једним решењем (S -метахеуристике). У P -метахеуристике спадају генетски алгоритми, алгоритми роја честица, мрављи алгоритми итд., док у S -метахеуристике спадају алгоритам симулираног каљења, алгоритам табу претраживања итд. [15]. Популационе метахеуристике раде са популацијама јединки, односно скупом решења, и више су истраживачки оријентисане. Наиме, популационе метахеуристике омогућавају добру диверзификацију простора претраге. Насупрот њима, метахеуристике које генеришу једно решење су оријентисане више локалној претрази, односно интензификацији простора претраге. Тако популационе метахеуристике као резултат дају скуп решења па су погодне за решавање вишекритеријумских оптимизационих проблема [15]. То су проблеми код којих се захтева задовољење више међусобно конфликтних критеријума (циљева), и ови проблеми су рачунски далеко сложенији и комплекснији од проблема једнокритеријумске оптимизације [15],[114],[115]. Оптимално решења у вишекритеријумској оптимизацији најчешће није једно, него постоји више оптималних решење који чини скуп недоминантних решења односно Парето скуп [114]-[116]. Парето оптимална решења су она решења која не могу бити побољшана по неком критеријуму, а да притом не буду погоршана по неком другом. Смисао и циљ употребе популационих метахеуристика код решавања вишекритеријумске оптимизације, јесте да се генерише Парето оптималан скуп решења, односно скуп недоминантних решења за дати проблем. Критеријуми код проблема вишекритеријумске оптимизације код проблема планирања су најчешће трошкови инвестиција и одржавања, трошкови губитака, као и трошкови услед прекида [1],[29],[72],[79],[81],[83],[92],[93],[97]. Вредновање добијених решења се врши помоћу неке од метода за доношење одлука [114]-[116]. Често, да би се смањила рачунска комплексност проблема планирања, вишекритеријумски проблем се своди на једнокритеријумски проблем оптимизације на тај начин што се сви трошкови свде на укупан трошак који се изражава новчаним јединицама. Тако проблем оптимизације планирања развоја дистрибутивних мрежа постаје једнокритеријумски проблем минимизације укупних трошкова [8],[38]-[65].

Предност примене метахеуристика у области планирања дистрибутивних мрежа превасходно се огледа у следећем:

- могућност превазилажења проблема "заглављивања" у локалном оптимуму, што омогућује добијање квалитетнијих решења него код примене хеуристика,
- могућност решавања планирања реалних (великих) дистрибутивних мрежа што не могу математички засновани оптимизациони модели,
- решавање проблема се може постићи у разумном времену.

Основна слабост се огледа у томе што метахеуристике не гарантују добијање глобалног оптимума, тако да квалитет добијених решења није изврстан.

Такође, како би се унапредили алгоритми за решавање проблема планирања, у последње време се прибегава хибридизацији метахеуристика и математичке оптимизације [75].

Као метахеуристички оптимизациони модели у области планирања развоја дистрибутивних мрежа најчешће се користе:

- симулирано каљење (*Simulated annealing – SA*) [64]-[75],[99],
- табу претраживање (*Tabu search – TS*) [76]-[82],[117],
- генетски алгоритам (*Genetic algorithm – GA*) [1],[83]-[98],
- алгоритам роја честица, (*Particle swarm optimization – PSO*) [118]-[120], и
- мрављи алгоритам (*Ant colony optimization algorithms – ACO*) [100]-[108].

Детаљан опис набројаних метахеуристика које су конкретно примењене на проблем планирања је дат у прилогу у поглављу П1, док се у наставку текста о поменутих метахеуристикама наводи само оно што је најбитније.

2.2.3.1. Алгоритам симулираног каљења

Симулирано каљење је техника претраге која је инспирисана металуршким каљењем челика. Опис алгоритма симулираног каљења који је предложен у овој тези за решавање статичког проблема планирања радијалних дистрибутивних мрежа је дат у поглављу 4, а теоријске основе и детаљан опис параметара алгоритма су приказани у прилогу овога рада, у поглављу П2.5.

2.2.3.2. Алгоритам роја честица

Алгоритам роја честица (*Particle swarm optimization – PSO*) је настао 1995. године, када је покушано да се на рачунару симулира понашање и кретање јата птица. Овај алгоритам је стохастички популационо базирана метахеуристика инспирисана понашањем роја птица или јата риба [121]. Наиме, алгоритам роја честица имитира друштвено понашање природних организама као што су јато птица или риба када траже места са довољно хране. У ствари, у овим ројевима, координирано понашање користи локалну појаву кретања без дефинисане централне контроле. Тако је кретање јединке у роју је с једне стране узроковано сопственим искуством јединке, а с друге стране искуством блиских суседа чиме се моделује социјална интеракција. Понашање роја може бити моделовано са неколико једноставних правила. Али, чак и ако су правила понашања сваке јединке (честице) једноставна, понашање роја може бити компликовано. Проблем понашања роја је у ствари проблем доношења одлуке за сваку од јединки. У [122] је испитиван процес доношења одлуке код људи, при чему је развијен концепт

индивидуалног учења и културне трансмисије. Према тим истраживањима, човек користи две главне врсте информација у процесу доношења одлуке.

- прва врста је његово сопствено искуство; то јест, кад појединац зна последице својих ранијих одлука,
- друга врста су искуства других људи, то јест, кад појединац зна последице туђих одлука.

На тај начин свака индивидуа доноси одлуку користећи или/и своја, или/и искуства других људи око себе. Такође, у [123] су примењена следећа три једноставна правила кретања јединке у роју, што је аналогно начину претраживању простора решења оптимизационог проблема који се решава:

- корак даље од најближе јединке (избегавање колизије са суседом),
- иди према дестинацији (циљу),
- иди ка центру роја.

Понашање сваке од јединки (честице) унутар роја може бити моделовано са овим правилима. Иницијализација овог алгорита се врши популацијом случајних решења (честица). Горе наведена правила понашања честица, које се крећу по мултидимензионалном простору претраге, своде се на три вектора брзине:

- вектор претходне брзине - даје меру инерције односно величине простора претраге (да ли ће се вршити глобално или локално претраживање решења),
- вектор брзине најбољег сопственог решења - даје меру утицаја сопственог искуства,
- вектор брзине најбољег групног решења - даје меру социјалног утицаја групе на честицу.

На овај начин алгоритам роја честица упражњава како глобалну, тако и локалну претрагу. Поступак се спроводи док се не испуни неки од критеријума заустављања, а што је најчешће достигнути број итерације. Као излаз након рада алгоритма добија се најбоље решење. По свему наведеном, алгоритам роја честица има доста сличности са генетским алгоритмима, с тим што код њега нема еволуцијских оператора мутације и укрштања. Овај алгоритам има велику примену нарочито код решавања нелинеарних континуалних оптимизационих проблема, комбинаторних оптимизационих проблема и мешовитих-целобројних оптимизационих проблема. Псеудо код наведеног алгоритма је дат на слици 2.1. Детаљан опис, параметри и особине алгоритма роја честица дати су у прилогу овог рада, у поглављу П1.2.

Алгоритам роја честица има неколико варијација, укључујући интеграцију са механизмом селекције и хибридизације за руковање и дискретним и континуалним променљивима. У недостатке овог алгоритма спада пре свега проблем одабира параметара, који значајно утичу на квалитет претраге и добијање квалитетних решења. Сам проблем одабира параметара нема јединствено решење, него директно зависи од природе проблема који се *PSO* алгоритмом решава. Такође, код примене *PSO* алгоритма лако може доћи само до парцијалне оптимизације, што утиче на квалитет решења. У прилогу, у поглављу П1.2.1 дат је пример примене једног алгоритма роја честица на проблем планирања дистрибутивних мрежа [119].

АЛГОРИТАМ РОЈА ЧЕСТИЦА - псеудо код

Почетак: /*Иницијализација*/

$k = 0;$

$s^k = s_i^0;$ /*Генерисање почетне тачке претраге за целу популацију*/

$v^k = v_i^0;$ /*Генерисање почетне брзине претраге за целу популацију*/

Понављати /*Глобална итерација, k */

Оценити $f(k)$; /*Израчунати вредност функције циља сваке јединке (честице) у текућој популацији k */

If $f(k) > pbest(k)$ **Then** $pbest(k + 1) = f(k)$ /*Прихватити функцију циља $f(k)$ честице као $pbest(k + 1)$ за ту честицу*/

If $pbest(k) > gbest(k)$ **Then** $gbest(k + 1) = pbest(k)$ /*Прихватити $pbest(k)$ као $gbest(k + 1)$ */

$v_i^{k+1} = wv_i^k + c_1rand_1 \times (pbest_i - s_i^k) + c_2rand_2 \times (gbest_i - s_i^k);$ /* Ажурирај брзину за све честице*/

$s_i^{k+1} = s_i^k + v_i^{k+1};$ /* Ажурирај позицију за све честице*/

$k = k + 1;$

Док Критеријум заустављања /*нпр. $k > k_{max}$ */

Излаз: Најбоље нађено решење

Слика 2.1. Псеудо код алгоритма роја честица

2.2.3.3. Генетски алгоритам

Генетски алгоритми (*Genetic algorithms – GA*) спадају у групу популационих алгоритама. Они су настали 70-их година прошлог века и представљају стохастички метод претраживања који је инспирисан биолошком еволуцијом. Генетски алгоритми имају инспирацију у природним процесима односно у процесу еволуције која је развијена у Дарвиновој теорији о постанку врста. Главне предности генетских алгоритама, које их чине подобнима за решавање широког скупа проблема из реалног живота су:

- прилагодљивост,
- ефикасност у решавању комплексних проблема у којима је главни циљ брзо наћи добро, али не нужно и оптимално решење.

Традиционалне технике оптимизације претрагу започињу са једним кандидатом при чему се применом статичке хеуристике итеративно тражи оптимално решење. С друге стране, код генетских алгоритама иницијално се користи популација кандидата за претрагу више области у простору решења. Еволуционо рачунање омогућава прецизно моделовање проблема оптимизације, мада не гарантује добијање глобалног оптимума. Још једна предност коришћења еволутивно прорачунских техника је да нема потребе да се експлицитно дефинише функција циља. Штавише, када је функција циља доступна, она не мора да буде

диференцијабилна. Генетски алгоритми се најчешће примењују за решавање проблема комбинаторне оптимизације.

Код генетских алгоритама јединке представљају апроксимације решења проблема који се решава. Јединке се кодирају као хромозоми, тако да су хромозоми кодирана представа јединки. Хромозом може бити низ битова, реалан или природан број, низ бројева односно било каква структура података која описује својства једне јединке. Хромозом се састоји од више гена. Ген је кодирана променљива проблема који се решава. Свакој јединки (решењу) се придружује функција оцене квалитета (*fitness*) јединке (назива се још и функција циља или прилагођености) која је у општем случају еквивалентна функцији која се оптимизује. Уколико нека јединка има бољу оцену квалитета, тиме има и већу вероватноћу преживљавања и укрштања. Тако се за сваки декодирани хромозом израчунава функција циља која представља меру добротe датог решења. Функција циља је најважнији део генетских алгоритама зато што је кључна за селекцију која са укрштањем доводи до нових јединки. Веома је важно да функција циља добро одржава проблем који се решава. Треба напоменути да се све операције код генетских алгоритама врше над хромозомима, односно кодираним јединкама, изузев функције циља која се врши над јединкама односно декодираним хромозомима. Генетски алгоритам започиње дефинисањем почетне популације. Популација представља скуп не јединки већ њихових кодираних престава - хромозома. Она се може генерисати насумично, или се користе решења добијена неком другом оптимизационом методом. Над хромозомима се примењују следећи оператори:

- оператор селекције - којом се врши одабир хромозома из текуће популације (најчешће избором добрих јединки, јер се претпоставља да ће добре јединке имати и добро потомство),
- оператор укрштања - којим се врши укрштање селекованих хромозома како би се добило потомство (деца),
- оператор мутације - којим се врши насумична мутација потомства како би се повећала разноврсност генетског материјала.

Селекцијом се одабирају оне јединке које имају добру оцену квалитета, док се оне "лоше" одбацују. На овај начин се обезбеђује да "добре" јединке добијају већу шансу за преживљавање и укрштање од оних "лошијих". Укрштањем се особине родитеља преносе на децу. Добијено потомство (деца) сада улазе у постојећу популацију, а из популације се избацују "лоше" јединке. На овај начин, у општем случају, из итерације у итерацију се поправља просечан квалитет популације. Поступак се спроводи док се не испуни неки од критеријума заустављања, а што може бити достигнути број итерације или сличност јединки у популацији што је знак да је алгоритам конвергирао. Након рада алгоритма као излаз се добија најбоље решење. Како би се повећала разноврсност генетског материјала уводи се и оператор мутације који омогућује бољу претрагу простора решења оптимизације. Генетски алгоритми се због прилично опште природе могу користити за решавање широког броја проблема. Псеудо код наведеног алгоритма је дат на слици 2.2. Детаљан опис оператора код генетских алгоритама, као и њихове особине и параметри дати су у прилогу овог рада, у поглављу П1.2.2.

Генетски алгоритми су се показали као врло добри у брзом идентификовању добрих подручја за истраживање (глобална претрага), док су показали мањкавост код поправљања решења која су близу оптималном. Потребно је нагласити да

класични генетски алгоритми, који раде са класичним операторима (мутације и укрштања) не представљају погодан алат за решавање проблема планирања развоја дистрибутивних мрежа [89],[93]. Разлог томе лежи у чињеници да добијена решења класичним генетским алгоритмима, у највећем броју случајева нису допустива (нпр. нису задовољена оперативна ограничења напона или токова снага, није задовољена радијалност и сл.). Како би се ово превазишло класични генетски оператори се комбинују са операторима локалног претраживања, чиме се избегава генерисање недопустивих решења. Један такав приступ и примена генетских алгоритама код планирања дистрибутивних мрежа [93] је детаљније дат и описан у прилогу, у поглављу П1.2.2. Такође, наведени недостатак генетских алгоритама је иницирао појаву нових техника где се класични генетски оператори комбинују са неким другим техникама, односно техникама локалног претраживања или неким другим математичким алатима у зависности од карактеристика проблема који се решава [89], [93], [124]. Овакав приступ се назива хибридизација и у последње време је све више заступљен у литератури [125]. Основни циљ хибридизације јесте да се користе позитивне особине различитих техника и алгоритама који чине хибридни алгоритам. То су најчешће комбинације два алгоритма, где први алгоритам има изражену способност за глобалну, а други за локалну претрагу. Код хибридизације са генетским алгоритмом, генетски алгоритми су задужени за глобалну претрагу, док се за локалну претрагу користе технике које користе систематичнији приступ у близини доброг решења. На тај начин се добија нови хибридни алгоритам, који треба да има изражене обе ове особине. У сваком случају, генетски алгоритми су у последње време нашли широку примену у решавању великог броја проблема комбинаторне оптимизације.

ГЕНЕТСКИ АЛГОРИТАМ - псеудо код

Почетак: /*Иницијализација*/

$k = 0$

$P(k) = P(0)$ /*Генерисати на случај иницијалну популацију*/

Понављати /*Глобална итерација, k */

Оценити $P(k)$ /*Израчунати вредност добротe сваке јединке (хромозома) у текућој популацији $P(k)$ */

$P'(k)$ = Селекција ($P(k)$) /*Изабрати скуп јединки (родитеља)*/

$P'(k)$ = Репродукција $P'(k)$ /*Применити операторе укрштања и мутације за добијање потомака*/

Оценити $P'(k)$ /*Израчунати вредност добротe сваке јединке (хромозома) у $P'(k)$ */

$P(k + 1) =$ Замена ($P(k)$, $P'(k)$) /*На основу шеме замене одредити које јединке преживљавају*/

$k = k + 1$

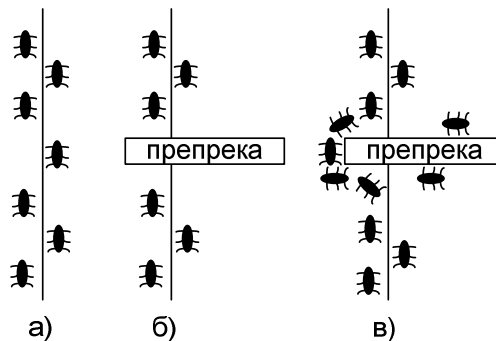
Док Критеријум заустављања /*нпр. $k > k_{max}$ нема унапређења објективне функције*/

Издаз: Најбоље нађено решење (јединка) или најбољи скуп (популација) решења

Слика 2.2. Псеудо код генетског алгоритма

2.2.3.4. Мрављи алгоритам

Као што је познато мрави су способни да пронађу најкраћу путању од извора хране до мравињака, без коришћења визуелних знакова. Они су способни да се прилагоде променама у окружењу, на пример, да нађу најкраћи пут, када постојећи стари пут није више проходан, због нове препреке. Према [126], реални мрави, када су у потрази за храном, у почетку истражују простор око мравињака на случајан начин. Чим неки мрав пронађе извор хране, он односи нешто од хране што је пронашао у мравињак. Док се се враћа, мрав оставља траг феромона на земљи. Траг феромона остављен на земљи послужиће и другим мравима као водич да дођу до извора хране. Сваки мрав ће са већом вероватноћом пратити правац са богатијим трагом феромона, него онај са слабијим трагом феромона. Тако, индиректна комуникација између мрава преко руте феромона омогућава им да пронађу најкраћу путању између мравињака и извора хране. Слика 2.3. јасно илуструје овај феномен који се може објаснити на следећи начин: на слици 2.3.а) је приказано када је путања између мравињака и извора хране чиста. На почетку имамо мраве који носе храну назад у мравињак. Предпостављајући да се сви мрави крећу приближно истом брзином, најкраћи пут ће у просеку посетити већина мрава. Стога ће се феромони на најкраћој путањи акумулирати брже него на осталим путањама. После краћег периода, разлика у количини феромона ће на путањама бити довољно видљива неком новом мраву који улази у систем. Од тог тренутка, новопридошли мрави ће жељено ићи дуж најкраће путање, што је уствари права линија која повезује мравињак са извором хране. Када се појави препрека на већ успостављеном путу, као што је приказано на слици 2.3.б), рута феромона се прекида. Они мрави који су управо испред препреке не могу да наставе да прате руту феромона, те ће стога одлучити да ли да иду лево или десно, насумично, пошто не знају који је најбољи избор. На слици 2.3.в), они мрави који су неким случајем изабрали краћи пут око препреке, пре ће успоставити прекинуту руту феромона, у поређењу са онима који су изабрали дужу путању, јер је више мрава који иду краћом путањом током истог временског периода. Отуда ће краћи пут имати већи садржај феромона, а заузврат ће већи број мрава да изабере баш ту краћу путању. Због ове позитивне повратне спреге врло брзо ће сви мрави изабрати краћу путању.



Слика 2.3. Понашање реалних мрава: а) мрави иду најкраћом путањом, б) на путањи се појављује препрека, в) мрави бирају најкраћу путању [124]

Мрављи алгоритам има више различитих изведби, али овде ће бити описана његова најједноставнија изведба за коју је дат и псеудо код на слици 2.4. Иницијализација креће са формирањем почетне популације мрава, где сваки мрав понаособ представља једно потенцијално решење. Мрави се крећу, бирајући

путању, са вероватноћом која зависи од количине феромонског трага на изабраној путањи. Када сви мрави дођу до циља врши се процена решења за сваког мрва, а затим (најчешће за најбоље мраве) и ажурирање феромонских трагова. Након тога се врши испаравање феромона са свих путања како би се спречила прерана конвергенција и омогућила диверзификација претраге. Ова процедура се циклички понавља док се не достигне неки максимални број понављања.

Мрављи алгоритми спадају у групу популационих алгоритама инспирисаних природом, који су у стању да решавају проблеме који нису строго формализовани. Предности мрављих алгоритама су [127]:

- брзо проналажење добрих решења,
- ефикасно решава проблем трговачког путника и сличних проблема,
- може се користити у динамичним апликацијама (добро се прилагођава променама).

У мане мрављих алгоритама спадају следећа својства:

- постоје делови са случајним бирањем путање,
- истраживање је често експериментално,
- време проналаска решења је неизвесно (али је загарантовано).

Примери примене мрављих алгоритама код планирања дистрибутивних мрежа су дати у [99]-[108]. У прилогу, у поглављу П1.2.3., детаљније је описан један пример примене мрављег алгорита из литературе [100].

ЈЕДНОСТАВАН МРАВЉИ АЛГОРИТАМ - псеудо код

Почетак: /*Иницијализација*/

$k = 0;$

$s^k = s_i^0;$ /*Генерисање почетног решења за сваког мрва (за целу популацију)*/

Понављати /*Глобална итерација, k */

Оценити $f(k)$; /*Израчунати вредност функције циља за сваког мрва у текућој популацији k */

$M'(k)$ =Селекција ($M(k)$) /*Изабрати n мрва од укупно m мрва у текућој популацији k */

$\tau_{k+1} = \tau_k + \Delta\tau_k$ /*Ажурирање феромонских трагова за све путање којима су прошли n мрва у текућој популацији k */

$\tau_{k+1} = \tau_k \cdot (1 - \rho)$ /*Испаравање феромонских трагова за све путање у текућој популацији k */

$s^{k+1} = f(s^k, p^k)$ /*Генерисање новог решења за сваког мрва (за целу популацију) на основу вероватноће одабира путање у текућој популацији k */

$k = k + 1;$

Док Критеријум заустављања /*нпр. $k > k_{max}$, нема унапређења критеријумске функције*/

Излаз: Најбоље нађено решење

Слика 2.4. Псеудо код једноставног мрављег алгорита

2.2.3.5. Табу претраживање

Табу претраживање (*TS*) је алгоритам који за разлику од популационих алгоритама ради са једним решењем и припада класи локалних претрага. Табу претрага се састоји из метахеуристичких процедура, које се користе за управљање хеуристичким алгоритмима, који врше локалну претрагу. Метахеуристички алгоритми се састоје од напредних стратегија, које допуштају експлоатацију простора претраге, омогућајући да се избегну замке да се за најбоље решење изабере локални оптимум. Као што се дешава са другим комбинаторним приступима, алгоритам табу претраге обавља велики број прелаза у простору претраге како би нашао оптимално решење, или решења блиска оптималном. Име табу је повезано са чињеницом да алгоритам, како би се избегле поновне претраге одређених области које су већ претражене, претвара те области у табу или их забрањује. То значи да за изванредан временски период претраживања, неће бити разматрано испитивање алтернатива које садрже тачке решења које припадају подручју које је проглашено за табу. Сам назив табу претраге је повезан са појмом табуа који долази из језика насељеника Полинезије, који означава ствари које се не могу додиривати зато што су "свете" односно које имају сакрални карактер. С друге стране, асоцијација са табуом се односи на чињеницу да се у реалном свету табу мења кроз време зашто је потребна одговарајућа "друштвена меморија", што код ове врсте претраге има кључни карактер, јер дефинише промену табу статуса током претраге [128].

Табу претрага је развијена из концепта који се користи у вештачкој интелигенцији. За разлику од других комбинаторних приступа, као што су генетских алгоритама и симулирано каљење, његово порекло није у вези са биолошким или физичким процесима оптимизације [129]-[132]. Алгоритам табу претраге се појавио у раним 1980-им годинама и од тада се примењује са успехом у бројним сложеним проблемима у науци и инжењерству. Табу претраживање је итеративни поступак претраге који крећући се од датог почетног решења покушава исто да унапреди у свакој итерацији. Поредом га са симулираним каљењем и генетским алгоритмима, табу претрага претражује простор решења на агресивнији начин. Алгоритми табу претраге крећу са почетним решењем-конфигурацијом (или са више решења када је претрага паралелна), које постаје текуће (текућа конфигурација). Код сваке итерације алгоритама, суседна структура се дефинише за текућу конфигурацију, а кретање се врши према најбољој конфигурацији у овој суседној структури (тј. код проблема минимизације, алгоритам се креће према конфигурацији која представља најмањи трошак). Наравно, проверавају се само суседне структуре које највише обећавају, односно које су атрактивне. Такође, приликом претраге дозвољен је прелазак на конфигурацију са већим коштањем, што даје алгоритму могућност да изађе из тачке локалног минимума. Суштинска особина алгоритма табу претраге је директно искључивање из претраге оних алтернатива које су привремено означена као забрањене - табу. Основна функција табуа, односно забране избора појединих решења, јесте у томе да се спречи појава циклуса односно претрага већ посећених решења. Због тога је коришћење меморије од кључног значаја за ове алгоритме, како би се пратили статуси могућих алтернатива односно табу ограничења. Алгоритам табу претраге приликом претраге, чува релевантне информације у краткорочној и дугорочној меморији, које се потом користе за преусмеравање претраге и измену алгоритма локалне претраге, што алгоритам табу претраге чини метахеуристиком. Период забране односно број итерација важење табуа се назива табу мандат

(*Tabu tenure*). Уколико претрага доспе до локалног оптимума, омогућава се избор лошијих суседних решења како би се процес претраге наставио. Код примене табуа неопходна је меморија која чува податке о садржају и мандату табуа током претраге. Сам табу, уколико се представља као решење које се забрањује, заузима велику количину меморије, па се из тих разлога уместо целих решења забрањују потези који воде до њих. Како ће се представљати табуи зависи од карактеристика проблема и начина репрезентације решења. Други механизми табу претраге су интензификација и диверзификација. Механизам интензификације омогућује алгоритму да подробније истражи атрактивне регионе (делове простора претраживања) што може довести до оптимума. Механизам диверзификације са друге стране, претрагу усмерава ка непосећеним регионима, што је некада важно да би се избегли локални оптимуми. Овде претраживачки простор треба дефинисати као скуп свих могућих решења која могу бити посећена током претраге. Док је околина (суседна структура) скуп решења која се добијају применом локалне трансформације на текуће решење. Локалне трансформације називају се још и потезима.

Основни алгоритам табу претраге је композиција локалне претраге и адаптивне меморије. Коришћење адаптивне меморије је супростављено алгоритмима који су засновани на немеморијском концепту, а који су инспирисани метафорама из физике и биологије. Значај претраге код табу претраживања, а тиме и сврха исте, било да се ради код примене детерминистичких или проблема са вероватноћом, произилази из претпоставке да лош избор даје више информација од случајно доброг избора [133]. Наиме, код коришћења меморије, лош избор може да произведе корисне информације о томе како је потребно мењати стратегију претраге. Псеудо код наведеног алгоритма је дат на слици 2.5. Детаљнији опис технике табу претраживања код планирања дистрибутивних мрежа [117] је дат у прилогу овог рада, у поглављу П1.2.4.

ТАБУ ПРЕТРАГА - псеудо код

Почетак: /*Иницијализација*/

$k = 1;$

Генерисање почетног допустивог решења, $s = s_0;$

Генерисање иницијалне табу листе, $T(s, k);$

Генерисање иницијалних аспирационих услова $A(s, k);$

Понављати /*Глобална итерација, k^* */

Одреди најбоље суседно (блиско) решење s' за које важи $s' \in \{T(s, k)/A(s, k)\}$

$s = s';$ /*Прихватити s' као текуће решење*/ Генерисати ново $T(s, k)$ и $A(s, k);$

$k = k + 1;$

Док Критеријум заустављања /*нпр. $k > k_{max}$, нема унапређења објективне функције*/

Излаз: Најбоље нађено решење

Слика 2.5. Псеудо код алгоритма табу претраге

Главна предност алгоритма табу претраживања је у томе што иста има способност да побегне из локалних оптималних решења прихватањем "лошијих" потеза, то јест способност да се претрага усмери тако, да погоршава тренутну вредност функције циља. Из тих разлога алгоритам табу претраживања се успешно примењује у многим проблемима оптимизације. Такође, овај алгоритам се разликује од других техника по употреби меморије, што је од кључног значаја за успешну реализацију табу претраживања. У поређењу са методама локалне претрага и симулираног каљења, разне компоненте табу претраживања морају бити прецизно дефинисане и прилагођене проблему који се решава. Мана алгоритма јесте у томе да решења у великој мери зависе од дефинисања ових параметара.

2.2.3.6. Резиме

У циљу превазилажења напред поменутих недостатака до сада предложених приступа за планирање дистрибутивних мрежа, у овој дисертацији је предложен хибридни алгоритам симулираног каљења и мешовитог-целобројног линеарног програмирања за решавање статичког проблема планирања радијалних дистрибутивних мрежа у присуству дистрибутивних генератора. У предложеном алгоритму је извршена хибридизација метахеуристике симулираног каљења, математичког мешовитог-целобројног програмирања, као и хеуристичких техника измене грана и локалних мрежа. Ово је учињено у циљу да се искористе најбоље особине од наведених алгоритама и техника које учествују у хибридизацији, како би се унапредио глобални предложени алгоритам за решавање проблема планирања дистрибутивних мрежа.

Проблем планирања развоја дистрибутивних мрежа је моделован као проблем мешовитог-целобројног линеарног програмирања (*MILP*) са циљем минимизације инвестиционих трошкова, трошкова губитака, трошкова прекида напајања потрошача услед кварова на гранама и дистрибутивним генераторима, као и трошкова губитака производње дистрибутивних генератора услед кварова на гранама. Као што је већ истакнуто, проблем планирања припада групи НП-тешких проблема, што значи да се проблем планирања реалне дистрибутивне мреже не може решити, односно не може се наћи глобално оптимално решење егзактним алгоритмом у разумном времену [9]-[11]. То је и показано на тест мрежи (пример II, у поглављу 4.2.) где примена *MILP* модела није могла да пронађе решење у оквиру 24 часа. Да би се смањила комплексност проблема планирања, предложена је декомпозиција оригиналног проблема (оригиналне мреже) на низ мањих подпроблема (локалних мрежа), које се ефикасно могу решавати применом *MILP* модела. Поступак декомпозиције и решавања итеративно је вођен и контролисан предложеним алгоритмом симулираног каљења који укључује механизам интензификације и диверзификације како би се добило најбоље решење.

Верификација ефикасности предложеног хибридног хеуристичког алгоритма урађена је на више примера реалних дистрибутивних мрежа упоређивањем са правим глобалним оптимумом.

3. ХИБРИДНИ АЛГОРИТАМ СИМУЛИРАНОГ КАЉЕЊА И МЕШОВИТОГ-ЦЕЛОБРОЈНОГ ПРОГРАМИРАЊА

3.1. ФОРМУЛАЦИЈА ПРОБЛЕМА

Циљ статичког планирања развоја дистрибутивних мрежа са дистрибутивним генераторима може се дефинисати на следећи начин: дефинисати скуп мрежних унапређења (појачање и/или изградње) који задовољава прогнозиране захтеве потрошње и производње дистрибутивних генератора, и скуп ограничења (термичка и напонска ограничења, као и ограничења радијалности) тако да се испуни циљ планирања (циљ доносиоца одлуке) за разматрани период планирања. Циљеви планирања се уобичајено исказују у новчаним износивама [8], па стога циљ доносиоца одлуке постаје минимизација укупних новчаних трошкова. Објективна функција за статичко планирање може се представити на следећи начин:

$$OF = C_{inv} + \sum_{i=1}^{TP} \frac{1}{(1+d)^i} * (C_{loss} + C_{rel}^{load} + C_{rel}^{DG}) \quad (3.1)$$

где је: C_{inv} инвестициони трошак, C_{loss} трошак губитака на годишњем нивоу, C_{rel}^{load} трошак прекида напајања услед отказа грана и дистрибутивних генератора на годишњем нивоу, C_{rel}^{DG} трошак губитака производње дистрибутивних генератора услед отказа грана на годишњем нивоу, TP разматрани период планирања, и d дисконтна стопа. Треба запазити да статичко планирање предпоставља да су годишњи трошкови у (3.1) исти у свакој години разматраног периода планирања (TP).

Инвестициони трошкови (C_{inv}) се појављују у иницијалној години разматраног периода планирања и садрже трошкове изградње нових елемената и/или трошкове појачања постојећих елемената у мрежи. Линеарна апроксимација трошкова губитака која је коришћена у овој дисертацији, омогућује прецизан прорачун губитака у $MILP$ моделу, без укључивања додатних целобројних променљивих које су биле неопходне у претходно коришћеним апроксимацијама, као у [42],[134]. Предложена апроксимација је представљена у поглављу 3.2.3. Трошкови прекида напајања (C_{rel}^{load}) садрже трошкове неиспоручене енергије услед отказа грана и генератора, и исти могу да се разликују за различите категорије (типове) потрошача. У радијалним мрежама не постоје алтернативни правци напајања и прекид гране прекида испоруку енергије за све потрошаче који се са те гране напајају. Како острвски рад генератора није дозвољен прописима у многим земљама (укључујући и Србију), ови потрошачи ће бити без напајања све док се квар не поправи. Шта више, кварови на генераторима у радијалним мрежама могу узроковати да део потрошача (оптерећења) мора да буде искључен како би се спречило преоптерећење грана. Ови потрошачи ће бити без напајања све док се квар на генератору не поправи. Укупни трошак узрокован испадом грана ће бити увећан трошком губитака производње дистрибутивних генератора (C_{rel}^{DG}), јер је генератор ван погона за сваки испад гране која га повезује са напојним чвором.

Ограничења као што су напон у чворовима и струје кроз гране, требало би да се одржавају у оквиру стандардних граница, а мрежа треба да буде у радијалном режиму без острвског рада генератора. Како су у дерегулисаним енергетским системима генератори најчешће у власништву инвеститора, претпоставља се да је њихова локација и снага унапред позната.

Горе описани проблем планирања најпре је моделован као *MILP* оптимизациони проблем што је презентовано у поглављу 3.2. Декомпозициони приступ базиран на *SA* техници, предложеном *MILP* моделу, техници измене грана и концепту локалне мреже, је приказан у поглављу 3.3.

3.2. ПРИСТУП РЕШАВАЊУ ПРОБЛЕМА

3.2.1. Модел мешовитог-целобројног линеарног програмирања

Овде је представљен *MILP* модел за решавање статичког проблема планирања радијалних дистрибутивних мрежа са дистрибутивним генераторима:

а) Објективна функција

$$\begin{aligned}
 \min z = & \sum_{a \in M_E} \sum_{b \in SS_a} c_{a,b}^{inv} * L_a * (w_{a,b} + w_{a,b}^{\wedge}) + \sum_{a \in M_F} \sum_{b \in SS_a} c_{a,b}^{inv} * L_a * (w_{a,b} + w_{a,b}^{\wedge}) + \\
 & + \sum_{i=1}^{TP} \frac{1}{(1+d)^i} * \left(\left(\sum_{a \in (M_E \cup M_F)} \sum_{b \in SS_a} L_a * \left(\sum_{ns=1}^{ns_{a,b}} (x_{a,b,ns} + x_{a,b,ns}^{\wedge}) \right) * KLoss_{a,b,ns} + \right. \right. \\
 & + \sum_{a \in (M_E \cup M_F)} \sum_{b \in SS_a} (x_{a,b}^f + x_{a,b}^{\wedge}) * K_{ENS} * \lambda_a * d_a * c^{rel\ base} + \\
 & + \sum_{n \in N_{TOT}} (1 - y_{n,g}) * K_{ENS} * \lambda_g * d_g * D_n * c_n^{rel} + \\
 & \left. \left. + \sum_{m \in N_{DG}} \sum_{a \in (M_E \cup M_F)} \sum_{b \in SS_a} (x_{a,b,m}^{DG} + x_{a,b,m}^{\wedge}) * K_{DG} * DG_m * \lambda_a * d_a * c_m^{DG} \right) \quad (3.2)
 \end{aligned}$$

Први члан објективне функције (3.2) описује трошкове унапређења постојећих грана, док други члан описује трошкове изградње нових грана. Трећи члан описује сведену садашњу вредност трошкова губитака. Овај трошак је моделован линеаризацијом описаном у поглављу 3.2.3. Четврти члан заједно са релацијама (3.12)-(3.13) описује сведену садашњу вредност трошкова неиспоручене електричне енергије услед отказа грана. Сведена садашња вредност трошкова неиспоручене електричне енергије услед отказа генератора дата је као пети члан у (3.2) уз релације (3.14)-(3.18). Трошкови губитка производње дистрибутивних генератора услед отказа грана описани су као шести члан у (3.2) и релацијама (3.20)-(3.23).

б) Баланс оптерећења

$$\sum_{a \in T_n} \sum_{b \in SS_a} (x_{a,b} + x_{a,b}^{\wedge}) = D_n - DG_m, \quad n \in N_{TOT}, \quad m \in N_{DG} \quad (3.3)$$

Релација (3.3) обезбеђује да је први Кирхофов закон задовољен за све чворове у мрежи. Могућност двостраног тока струје кроз гране омогућена је увођењем виртуалних грана између чворова у мрежи. Променљиве које описују ове гране имају ознаку (\wedge).

в) Ограничења капацитета

$$x_{a,b} = \sum_{ns=1}^{ns_{a,b}} x_{a,b,ns}, \quad x_{a,b}^{\wedge} = \sum_{ns=1}^{ns_{a,b}} x_{a,b,ns}^{\wedge} \quad (3.4)$$

$$x_{a,b,ns} - w_{a,b} * (Xseg_{a,b,ns} - Xseg_{a,b,(ns-1)}) \leq 1 \quad (3.5)$$

$$x_{a,b,ns}^{\wedge} - w_{a,b}^{\wedge} * (Xseg_{a,b,ns} - Xseg_{a,b,(ns-1)}) \leq 1 \quad (3.6)$$

$$Xseg_{a,b,0} = 0 \quad (3.7)$$

$$Xseg_{a,b,0} = X_{a,b}^{max} \quad (3.8)$$

$$(a \in \{M_E \cup M_F\}), (b \in \underline{SS}_a), (ns \in \{1, \dots, ns_{a,b}\})$$

Ограничења (3.4)-(3,8) обезбеђују да струје у свакој грани не прекорачују термичке капацитете.

г) Напонска ограничења

$$V_l \geq V_k + Z_{a,b} * \sum_{ns=1}^{ns_{a,b}} (x_{a,b,ns} + x'_{a,b,ns}) + \{\sum_{b \in \underline{SS}_a} (w_{a,b} + w'_{a,b}) - 1\} * V^{min} \\ (l, k \in N_{TOT}), (a \in (F_l \cap T_k)), (b \in \underline{SS}_a) \quad (3.9)$$

$$V_l \leq V_k + Z_{a,b} * \sum_{ns=1}^{ns_{a,b}} (x_{a,b,ns} + x'_{a,b,ns}) - \{\sum_{b \in \underline{SS}_a} (w_{a,b} + w'_{a,b}) - 1\} * V^{min} \\ (l, k \in N_{TOT}), (a \in (F_l \cap T_k)), (b \in \underline{SS}_a) \quad (3.10)$$

$$V^{max} \geq V_l \geq V^{min}, (l \in N_{TOT}) \quad (3.11)$$

Релације (3.9)-(3.11) гарантују да напони имају прихватљиве вредности у сваком чвору у мрежи.

д') Баланс оптерећења фиктивне мреже са фиктивним струјама без DG :

$$\sum_{a \in T_n} \sum_{b \in \underline{SS}_a} (x_{a,b}^f + x'_{a,b}) - \sum_{a \in F_n} \sum_{b \in \underline{SS}_a} (x_{a,b}^f + x'_{a,b}) = Krel_n * D_n, (n \in N_{TOT}) \quad (3.12)$$

Фиктивне струје у (3.12) се користе за израчунавање неиспоручене енергије у мрежи услед отказа грана која је описана као четврти члан у (3.2). Треба запазити да дистрибутивни генератори нису укључени у (3.12) због тога што острвски рад дистрибутивних генератора није дозвољен. Коефицијент $Krel_n$ представља релативну јединицу коштања неиспоручене енергије за чвор n , с обзиром на најмању цену коштања неиспоручене енергије ($c^{rel\ base}$) у разматраној мрежи, која се дефинише за потрошаче са најмањим приоритетом напајања. На тај начин је омогућено разматрање оптерећења са различитом ценом коштања неиспоручене енергије.

д'') Ограничење фиктивних капацитета

$$x_{a,b}^f - (w_{a,b} + w'_{a,b}) * X_{max}^f \leq 0, x'_{a,b} - (w_{a,b} + w'_{a,b}) * X_{max}^f \leq 0 \\ (a \in \{M_E \cup M_F\}), (b \in \underline{SS}_a) \quad (3.13)$$

Релација (3.13) обезбеђују да ће само гране које егзистирају у реалној мрежи бити разматране и у фиктивној мрежи. Реални капацитети грана нису од интереса у рачунању неиспоручене енергије услед отказа грана, али је битна околност да ли грана постоји или не.

ђ) Неиспоручена енергија услед отказа дистрибутивних генератора

ђ') Баланс оптерећења фиктивне мреже када су дистрибутивни генератори ван погона услед отказа

$$\sum_{a \in T_n} \sum_{b \in \underline{SS}_a} (x_{a,b,g}^f + x'_{a,b,g}) - \sum_{a \in F_n} \sum_{b \in \underline{SS}_a} (x_{a,b,g}^f + x'_{a,b,g}) = y_{n,g} * D_n - DG_m, \\ n \in N_{TOT}, m \in (N_{DG} \setminus g), g \in N_{DG} \quad (3.14)$$

ђ'') Ограничење капацитета фиктивне мреже

$$x_{a,b,g}^f - (w_{a,b} + w'_{a,b}) * X_{a,b}^{max} \leq 0, x'_{a,b,g} - (w_{a,b} + w'_{a,b}) * X_{a,b}^{max} \leq 0 \quad (3.15)$$

ђ''') Ограничење напона фиктивне меже

$$V_l^f \geq V_k^f + Z_{a,b} * (x_{a,b,g}^f + x_{a,b,g}'^f) + \{\sum_{b \in \underline{SS}_a} (w_{a,b} + w_{a,b}') - 1\} * V^{min} \quad (3.16)$$

$$V_l^f \leq V_k^f + Z_{a,b} * (x_{a,b,g}^f + x_{a,b,g}'^f) - \{\sum_{b \in \underline{SS}_a} (w_{a,b} + w_{a,b}') - 1\} * V^{min}$$

$$(l, k \in N_{TOT}), (a \in (F_l \cap T_k)), (b \in \underline{SS}_a) \quad (3.17)$$

$$V^{max} \geq V_l^f \geq V^{min}, (l \in N_{TOT}) \quad (3.18)$$

Релације (3.14)-(3.18) дефинишу стање у мрежи када су дистрибутивни генератори ван погона. Број фиктивних мрежа дефинисан је у (3.14)-(3.18) и једнак је броју дистрибутивних генератора у мрежи. Променљива $y_{n,g}$ у (3.14) дефинише проценат терета који треба искључити у чвору да би се спречило нарушавање термичких и/или напонских ограничења када је дистрибутивни генератор ван погона. Износ искљученог оптерећења дефинише трошак неиспоручене енергије услед отказа дистрибутивних генератора који је описан у петом члану објективне функције (3.2).

е) Ограничење радијалности

$$\sum_{a \in (M_E \cup M_F)} \sum_{b \in \underline{SS}_a} (w_{a,b} + w_{a,b}') \leq n_{TOT} - n_S \quad (3.19)$$

$$\sum_{a \in T_n} \sum_{b \in \underline{SS}_a} (x_{a,b,m}^{DG} + x_{a,b,m}'^{DG}) - \sum_{a \in F_n} \sum_{b \in \underline{SS}_a} (x_{a,b,m}^{DG} + x_{a,b,m}'^{DG}) = KDG_n,$$

$$(n \in N_{TOT}), (m \in N_{DG}) \quad (3.20)$$

$$KDG_m = 1, \forall m \in N_{DG} \quad (3.21)$$

$$KDG_m = 0, \forall m \notin N_{DG} \quad (3.22)$$

$$x_{a,b,m}^{DG} - (w_{a,b} + w_{a,b}') \leq 0, x_{a,b,m}'^{DG} - (w_{a,b} + w_{a,b}') \leq 0$$

$$(a \in \{M_E \cup M_F\}), (b \in \underline{SS}_a), m \in N_{DG} \quad (3.23)$$

Како дистрибутивни генератори могу напајати поједине потрошаче независно од тога да ли постоји веза са напојном трансформаторском станицом, односно са ЕЕС, у делу мреже се може успоставити острвски рад. У том случају радијални режим рада не може бити гарантован ограничењем баланса токова снага (3.3) и ограничењем радијалности (3.19). Стога су додата нова ограничења (3.20)-(3.23) која обезбеђују да генератори нису изоловани од напојног чвора, односно напојне трансформаторске станице [135]. Треба запазити да релације (3.20)-(3.23) обезбеђују да ће сваки дистрибутивни генератор бити повезан са напојним чвором. На бази ових релација гране које повезују дистрибутивне генераторе са чвором извора су идентификоване и користе се за израчунавање трошкова губитака производње дистрибутивних генератора описаних као шести члан у (3.2).

Ознаке коришћене у (3.2)-(3.23) имају следеће значење:

$w_{a,b}, w_{a,b}'$ – бинарна променљива $(w_{a,b}, w_{a,b}') \in \{0,1\}$ која узима вредност 1 ако је постојећа грана a оптерећена, или је унапређена, или је изграђена нова грана, у супротном променљива узима вредност 0,

$x_{a,b}, x_{a,b}'$ - струја гране a пресека b , у једном или другом смеру, респективно

$x_{a,b,ns}, x_{a,b,ns}'$ – фиктивна струја кроз сегмент ns гране a пресека b , у једном или другом смеру, респективно

$x_{a,b}^f, \hat{x}_{a,b}^f$ - фиктивна струја гране a пресека b у једном или другом смеру, респективно, која би текла кроз разматрану грану да у мрежи не постоје дистрибутивни генератори,

$x_{a,b,g}^f, x_{a,b,m}^{DG}, \hat{x}_{a,b,g}^f, \hat{x}_{a,b,m}^{DG}$ - фиктивна струја гране a пресека b у једном или другом смеру, респективно

V_k, V_l - напони чвора k и чвора l , респективно

V_k^f, V_l^f - фиктивни напони чвора k и чвора l , респективно

KDG_n - фиктивни терет за сваки дистрибутивни генератор који се може напојити из једног од напојних чвора

$Kloss_{a,b,ns}$ - коефицијент губитака дефинисан у поглављу 3.2.3.,

$c_{a,b}^{inv}$ - јединична цена (по km) за изградњу/појачање гране a на пресек b ,

$c^{rel base}$ - јединична цена неиспоручене енергије за потрошаче са најмањим приоритетом,

c_n^{rel} - јединична цена неиспоручене енергије за чвор n ,

c_m^{DG} - цена производних трошкова дистрибутивног генератора у чвору m ,

L_a - дужина гране a ,

SS_a – скуп пресека на који се грана a може изградити или унапредити (појачати) у мрежи

M_E, M_F - скуп опостојећих грана и скуп будућих нових грана у мрежи, респективно,

$\underline{SS}_a = \{SS_a \cup b_{aE}\}$,

b_{aE} – постојећи пресек гране a ($a \in M_E$),

D_n - прогнозирано оптерећење у хоризонтној години у чвору n ,

DG_m - прогнозирана производња генератора у хоризонтној години у чвору m ,

N_{TOT} - скуп чворова у мрежи,

N_{DG} - скуп чворова са дистрибутивним генераторима у мрежи,

n_{DG} – број дистрибутивних генератора у мрежи,

n_{tot} – укупан број чворова у мрежи (укључујући и напојну трафостаницу)

n_s – број напојних трафостаница у мрежи

F_n, T_n – скуп свих грана за које је чвор n изворни чвор, и скуп грана за које је чвор n терминални чвор, респективно,

$X_{a,b}^{max}$ - термички капацитет гране a пресека b ,

V^{min}, V^{max} - минимална и максимална граница напона у мрежи респективно,

X_{max}^f - велика позитивна константа,

$Z_{a,b} = (r_{a,b} * \sin\varphi + x_{a,b} * \cos\varphi) * L_a$ [136],

$KENS = \sqrt{3} * \alpha * V_r * \cos\varphi, KDG = \sqrt{3} * V_r * \cos\varphi_{DG}$,

$\cos\varphi, \cos\varphi_{DG}$ - фактор снаге терета (потрошње) и дистрибутивних генератора, респективно,

α - фактор оптерећења,

V_r – номинални напон мреже,

$\lambda_{a,b}, \lambda_g$ - интензитет отказа гране a пресека b , односно отказа дистрибутивног генератора DG_g , респективно

d_a, d_g - време поправке квара на грани a , односно дистрибутивном генератору DG_g , респективно.

Као софтверска подршка за извршавање прорачуна на основу предложеног *MILP* модела, кориштени су програмски пакети *MATLAB* и *TOMLAB* што је детаљније описано у прилогу, у поглављу ПЗ.

3.2.2. Алгоритам симулираног каљења

У првом кораку алгоритма симулираног каљења генерише се иницијално решење на следећи начин. Разматрана мрежа, која садржи постојеће гране и чворове, могућа појачања, као и могуће нове будуће гране и чворове, се декомпонује у низ локалних мрежа (подпроблема). Свака локална мрежа садржи скуп суседних фидера. На пример, мрежа садржи један фидер и њему прве суседне фидере (који се могу потенцијално повезати са њима преко нових повезних или постојећих грана). Могући начини креирања локалних мрежа су презентовани у [137],[138], и дати су у прилогу, поглавље П2.2. Сваки подпроблем је решаван одвојено применом предложеног *MILP* модела (3.2)-(3.23). На тај начин се добија иницијално решење ($f(x^0)$) оригиналног проблема. Локалне мреже које постоје у иницијалном решењу се чувају за даљу употребу као и скуп свих могућих повезних грана (расклопних уређаја на тим гранама) између ових локалних мрежа ($NOLN_{(k=0)}$). Иницијално решење се даље модификује корак по корак на следећи начин:

1) Дефинисање иницијалне (почетне) температуре:

$$T_{max} = T_1 = \frac{\mu}{-\ln\phi} * f(x^0) \quad (3.24)$$

где је ϕ % "лоших" потеза (потези који воде до решења са већим трошком од текућег), који су μ % лошији од почетног решења $f(x^0)$, а T_{max} добијена иницијална почетна температура [15]. Поставља се индекс $k = 1$ за *SA* процедуру.

2) Механизам интензификације. Изабрати случајно један *NOLN* расклопни уређај (повезни расклопни уређај) из скупа $NOLN_{(k-1)}$. Ако је скуп $NOLN_{(k-1)}$ исцрпљен (празан) ићи на корак 7. Треба истаћи да *NOLN* расклопни уређаји повезују не само две суседне локалне мреже, него такође и два суседна фидера који припадају овим мрежама.

3) Применити технику измене грана за суседне фидере затварањем отвореног прекидача и отварањем њему суседног прекидача, тако да сви чворови остају повезани са напојном трафо станицом, а ограничење радијалности је задовољено. Ако не постоји изводљива операција измене грана ићи на корак 2. Треба истаћи да се конфигурација од две разматране (суседне) локалне мреже

мења током операције измене грана, и на тај начин се формира нова суседна структура.

4) Применити *MILP* модел (3.2)-(3.23) за сваку модификовану локалну мрежу, полазећи од иницијалног решења и тако да се добија ново суседно решење. Иницијално решење локалне мреже садржи све гране и чворове који већ постоје у разматраној локалној мрежи, као и сва расположива појачања и могуће нове гране и чворове.

5) Дефинисати: $f(S_j)$ трошак суседног решења, $f(S_i)$ трошак текућег решења, и разлику између ових трошкова ΔE . Поступак прихватања у k -тој итерацији може се свести на следеће:

- ако је $\Delta E > 0$, онда суседно решење постаје текуће решење ($S_j = S_i$); шта више, ако је трошак суседног решења мањи од текућег најбољег решења, онда суседно решење постаје најбоље решење; нове локалне мреже се чувају и могуће повезне гране (*NOLN* расклопни уређаји) између суседних локалних мрежа постају део скупа $NOLN_k$.
- ако је $\Delta E < 0$ онда важи следеће:
 - ако је $e^{-\Delta E/T_k} >$ случајног броја $[0, 1]$, онда суседно решење постаје текуће решење; нове локалне мреже се памте и могуће повезне гране (*NOLN* расклопни уређаји) између суседних локалних мрежа постају део скупа $NOLN_k$. Ово представља способност алгоритма за прихватање и лошијег суседног решења, али са неком вероватноћом која опада са падом температуре T_k .
 - ако је $e^{-\Delta E/T_k} <$ случајног броја $[0, 1]$, онда се суседно решење не прихвата.

6) Ако је ново решење добијено на температури T_k иде се на корак 7, у противном се иде на корак 8.

7) Генерише се нова температура $T_{k+1} = \alpha * T_k$, где је α константа која је у интервалу (0.5-0.99) [15]. Ако температура достигне доњу вредност ићи на корак 17. У противном поставити $k = k + 1$, и ићи на корак 2.

8) Механизам диверзификације. Дефинисати R да буде број ресета температуре (корак диверзификације). Поставити $R = R + 1$.

9) Ако је $R + 1 > R_{max}$, где је R_{max} максимални број корака диверзификације ићи на корак 17.

10) Нека је FDR_R разлика скупа свих фидера у мрежи и скупа фидера који садржи све фидере око којих је формирана локална мрежа у текућем решењу то јест који представља центар локалне мреже.

11) Случајно изабрати фидер из скупа FDR_R и направити локалну мрежу која садржи изабрани фидер и одређени број њему суседних фидера (суседни фидери првог реда, суседни фидери другог реда итд.) [137],[138]. Уклонити све фидере из скупа FDR_R који су постали део направљене локалне мреже.

12) Извршити *MILP* модел (3.2)-(3.23) за добијену локалну мрежу, полазећи од његовог иницијалног стања.

13) Поновити кораке 11. и 12. за све фидере из скупа FDR_R . Запазити да је

изведеним корацима 11. и 12. крајњи проблем (мрежа) декомпонован у низ локалних мрежа, које су различите од локалних мрежа које постоје у текућем решењу, па је тако постигнуто велико суседно решење разматраног проблема.

14) Нека је T_r температура ресета алгоритма симулираног каљења, а T_b температура за коју је пронађено најбоље решење. Поставља се $T_r = \max \{T_{max}/2, T_b\}$, $k = k + 1$, $T_{k+1} = T_r$.

15) Извршити процедуру за прихватање добијеног великог суседног решења као што је описано у кораку 5.

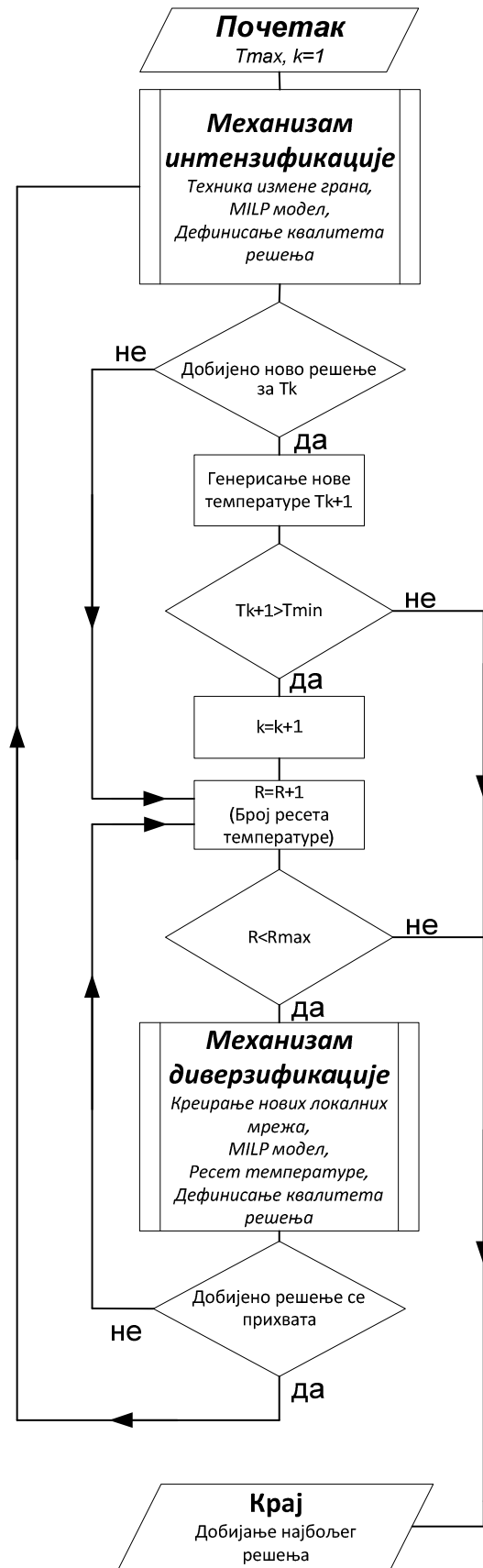
16) Ако се решење прихвата ићи на корак 2. У супротном ићи на корак 8.

17) Завршетак алгоритма симулираног каљења и добијање најбољег решења.

Треба нагласити да је горе описана имплементација механизма интензификације, која омогућава искоришћавање постојећих решења, а посебно механизам диверзификације, који омогућава претраживања до сада неистражених простора за претрагу, од суштинског значаја за побољшање ефикасности алгоритма симулираног каљења [15]. Осим тога, предложена немонотона шема редукције, која смањује температуру након сваке итерације са повременим порастом температуре (виша температура ресета), после појаве суседних структура, без прихватања било ког решења, показала је да даје боље перформансе од стандардног приступа [139]. Овде такође треба нагласити да предложени декомпозициони приступ омогућава, укључивање комплекснијих и потенцијално тачнијих модела за решавање генерисаних подпроблема, на пример нелинеарних оптимизационих модела [135].

Као софтверска подршка код аквизиције података добијених од стране *MILP* прорачуна, и прорачуна за потребе алгоритма симулираног каљења, кориштен је *MICROSOFT EXCEL*, што је детаљније описано у прилогу, у поглављу П3.4.

Дијаграм тока предложеног хибридног алгоритма дат је на слици 3.1.



Слика 3.1. Дијаграм тока предложеног хибридног алгоритма

3.2.3. Модел линеаризације трошкова губитака

Линеаризација трошкова губитака за постојећу грану a , пресека b , дата је на слици 3.2. За сваки линеарни сегмент се дефинише коефицијент губитака $KLoss_{a,b,ns}$ на следећи начин:

$$KLoss_{a,b,ns} = \frac{\Delta c_{a,b,ns}^{loss}}{\Delta Xseg_{a,b,ns}} = \frac{c_{a,b,ns}^{loss} - c_{a,b,ns-1}^{loss}}{Xseg_{a,b,ns} - Xseg_{a,b,ns-1}}, ns = 1, \dots, ns_{a,b} \quad (3.25)$$

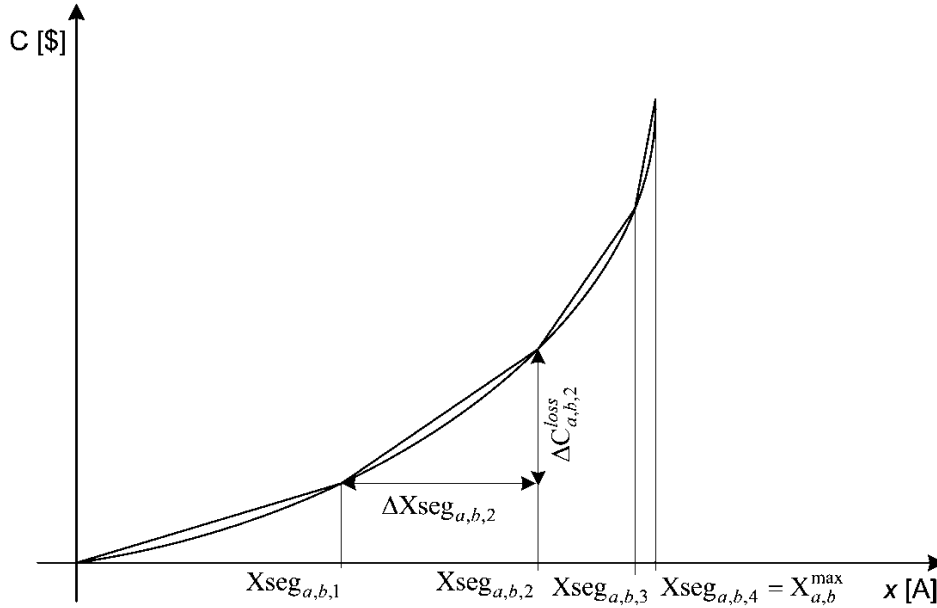
$$Xseg_{a,b,0} = 0, Xseg_{a,b,ns_{a,b}} = X_{a,b}^{max} \quad (3.26)$$

где је:

$Xseg_{a,b,ns}$ - ток струје који се дефинише у крајњој тачки линеарног сегмента ns за грану a , пресека b (видети слику 3.2.),

$ns_{a,b}$ - број сегмента који се користи за линеаризацију трошкова губитака за грану a , пресека b , и

$c_{a,b,ns}^{loss}$ - цена трошкова губитака ($US\$/km$) за грану a , пресека b , када кроз исту тече струја $Xseg_{a,b,ns}$.



Слика 3.2. Линеаризација трошкова губитака

Линеаризација трошкова губитака приказана на слици 3.2. увек може да се изврши тако да важи следеће:

$$KLoss_{a,b,1} < KLoss_{a,b,2} < \dots < KLoss_{a,b,ns_{a,b}} \quad (3.27)$$

Ако је услов (3.27) је испуњен, трошкови губитака за грану a , пресека b , у процесу оптимизације (минимизирање укупних трошкова) може да се врши линеаризација користећи било који број ($ns_{a,b}$) линеарних сегмената са дефинисањем само једна (бинарне) променљиве за разматрану грану ($w_{a,b}$), како је наведено у (3.4)-(3.8). То подразумева да предложени начин линеаризације не повећава број бинарних променљивих у моделу, то јест не повећава значајно сложеност *MILP* модела, док омогућава прецизно моделовање трошкова губитака.

На пример, разматра се грана a , пресека b , за коју се врши линеаризација са 2 сегмента ($ns_{a,b} = 2$). Нека је $Xseg_{a,b,1} = 2$ и $Xseg_{a,b,2} = X_{a,b}^{max} = 5$. Такође нека је $KLoss_{a,b,1} = 1$ и $KLoss_{a,b,2} = 2$, што задовољава (3.27). Нека је $w_{a,b} = 1, w'_{a,b} = 0$, струја која тече кроз грану је $x_{a,b} = 3$, и претпоставља се да сви могући токови могу узети само целобројне вредности. Сада, узимајући у обзир релације (3.4)-(3.8), могуће су следеће комбинације токова струја: $(x_{a,b,1} = 0; x_{a,b,2} = 3)$, $(x_{a,b,1} = 1; x_{a,b,2} = 2)$ и $(x_{a,b,1} = 2; x_{a,b,2} = 1)$. Комбинација токова која задовољава релације (3.4)-(3.6) $(x_{a,b,1} = 2; x_{a,b,2} = 1)$, је једина која обезбеђује минимум губитака по објективној функцији (3.1), тако да ће ова комбинација токова бити изабрана у процесу оптимизације. Треба напоменути да је ова комбинација једина који даје стварне трошкове губитака (видети слику 3.2.).

У овој дисертацији линеаризација губитака је примењена приликом уважавања трошкова губитака у моделу мешовитог-целобројног програмирања.

4. ПРИМЕРИ ПРИМЕНЕ

4.1. ПРИМЕР МРЕЖЕ I

Предложени приступ је кориштен за налажење решења са минималном укупном вредности трошкова за $20kV$ тест мрежу дату на слици 4.1. Тест мрежа се састоји од једног напојног чвора (напојне трафо станице), 27 постојећих (већ изграђених) грана (пуна линија) и 29 могућих нових грана (испрекидана линија). У напојном чвору постоје три трансформатора $110/20 kV/kV$ ($TRF1&2&3$, грана 0-1) са капацитетом од $33MVA$ по трансформатору. Такође се претпоставља да постоје четири величине (пресека) са којима је могуће изградити, односно на које је могуће појачати неку грану у мрежи. У Табели 4.1 су приказане физичке карактеристике и јединични трошкови изградње за сваку од величина (пресека). Трошкови појачања (замене постојећег проводника проводником већег капацитета) су дати у Табели 4.2. На слици 4.1 су капацитети сваке постојеће гране у иницијалној мрежи дати у MVA док су дужине свих грана дате у километрима. Претпостављено је да у свакој грани постоји расклопни уређај. Постојећи чворови потрошње су означени са празним кружићем, док су будући чворови означени са пуним. Прогнозирана оптерећења у хоризонтној години у сваком од чворова дати су у MVA на слици 4.1. Шест дистрибутивних генератора $SG1 = 4MVA$, $SG2 = 2,5MVA$, $SG3 = 5MVA$, $SG4 = 3MVA$, $SG5 = 4,5MVA$, $SG6 = 3,5MVA$, везани су за чворове 14, 22, 25, 28, 30 и 33, респективно. Линеаризација трошкова губитака рађена је са 21 сегментом. Трошкови губитака односно коефицијент губитака $Kloss_{a,b,ns}$ је рачунат на основу следећих параметара: јединична цена губитака је $0.1 US\$/kWh$, фактор губитака (*loss factor*) је 0.396, фактор снаге потрошача је 0.95, дисконтна рата је 0.1, а период планирања износи 30 година. Једнична цена неиспоручене енергије за потрошаче у чворовима (2)–(32) је $2 US\$/kWh$, а за потрошаче у чворовима (33)–(43) је $5 US\$/kWh$. Стопа отказа за све гране износи $0.1(kvar/km god)$, а време поправке 3 часа. Стопа отказа дистрибутивних генератора износи $0.25 kvar/god$ а време поправке 50 часова, трошак губитака производње за све дистрибутивне генераторе износи $0.1 US\$/kWh$, а фактор снаге за дистрибутивне генераторе је 1. Напон напојног чвора (1) је $20.5 kV$, горња граница напона износи $21 kV$, а доња граница $19 kV$. За MILP прорачуне кориштен је програмски оптимизациони пакет TOMLAB (CPLEX) [137] који је детаљније описан у прилогу, поглавље П3.3.

Разматрају се три случаја да би се анализирао утицај дистрибутивних генератора на планирање развоја дистрибутивних мрежа. У случају 1 минимум укупних трошкова проблема планирања се проналази узимајући у обзир присуство дистрибутивних генератора у мрежи. У случају 2 минимум укупних трошкова се проналази не узимајући у обзир неиспоручену енергију услед отказа дистрибутивних генератора, односно откази дистрибутивних генератора нису разматрани. У случају 3 проблем планирања је решаван без присуства дистрибутивних генератора.

У случају 1 иницијално решење је постигнуто декомпозицијом тест мреже, са слике 4.1, на три подпроблема (локалне мреже) тако да свака локална мрежа садржи по три суседна фидера. Проблем планирања за сваку од локалних мрежа је решаван применом MILP модела (3.2)–(3.23) како би се постигло иницијално

решење. Цена овако добијеног иницијалног решења износи 5.179.890 \$. Параметри SA алгоритма су $\phi = 220\%$ и $\mu = 25\%$, почетна температура на основу (24) је $T_{max} = 800$. SA алгоритам се завршава када температура достиже 0.01 ($T_{min} = 0.01$) или када број корака диверзификације постаје већи од 2 ($R_{max} = 2$). Константа α из (3.2) је изабрана да износи 0.6. Најбоље решење у предложеном SA алгоритму је пронађено после 13 итерација. Трошак најбољег решења за овај случај, као и за остале разматране случајеве, је дат у табели 4.3. Најбоље решење за случај 1 приказано је на слици 4.2, где су унапређења (појачања и нова изградња) представљени дебелим линијама упоредо са њиховим капацитетима изражених у MVA. Исти је проблем је решаван применом MILP модела за целу мрежу која је приказана на слици 4.1. У овом случају укупна цена решења је добијена и износи 4.782.400 US\$, што је исти резултат добијен и предложеним приступом. Наиме, предложени приступ даје исто решење као и MILP модел.

Табела 4.1 - Подаци о гранама

Капацитет [MVA]	Цена изградње [US\$x103/km]	r [Ω /km]	x [Ω /km]
5	60	0.383	0.23
8	80	0.265	0.22
10	100	0.191	0.2
14	140	0.123	0.19

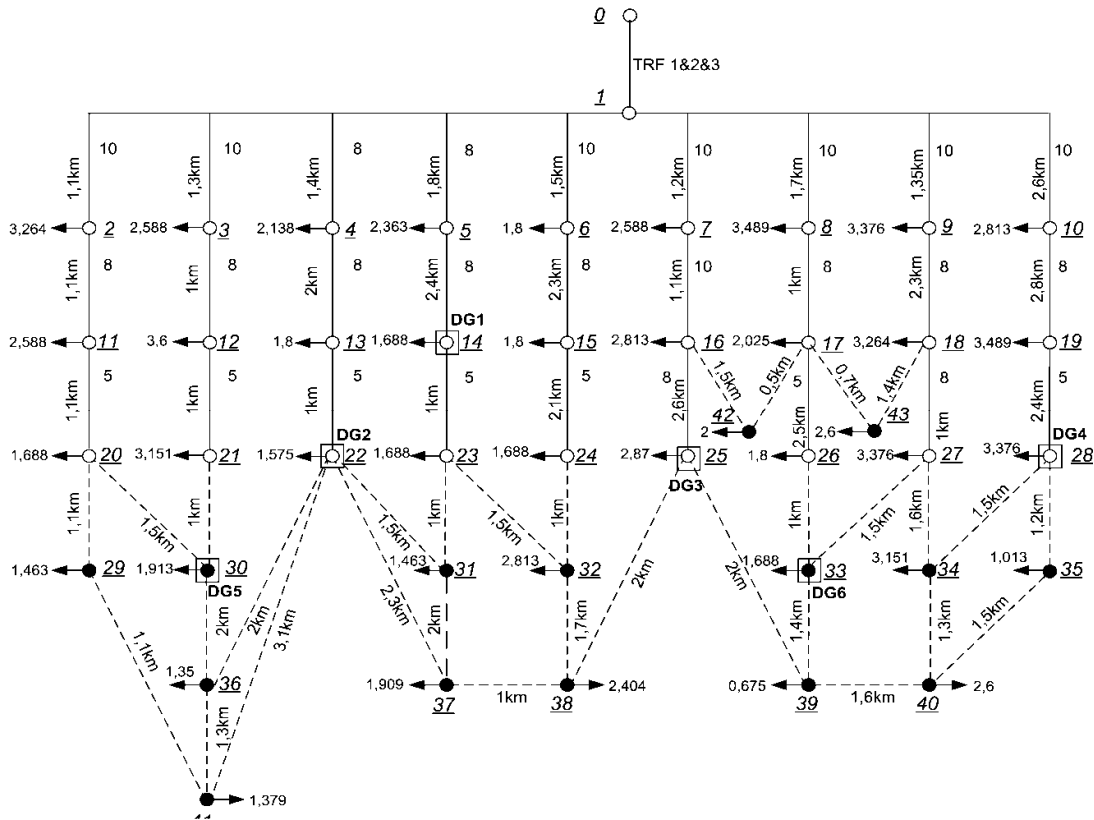
Табела 4.2 - Цене појачања [US\$ x 103/km]

На \ Од	8	10	14
5	72	91	120
8	-	85	105
10	-	-	90

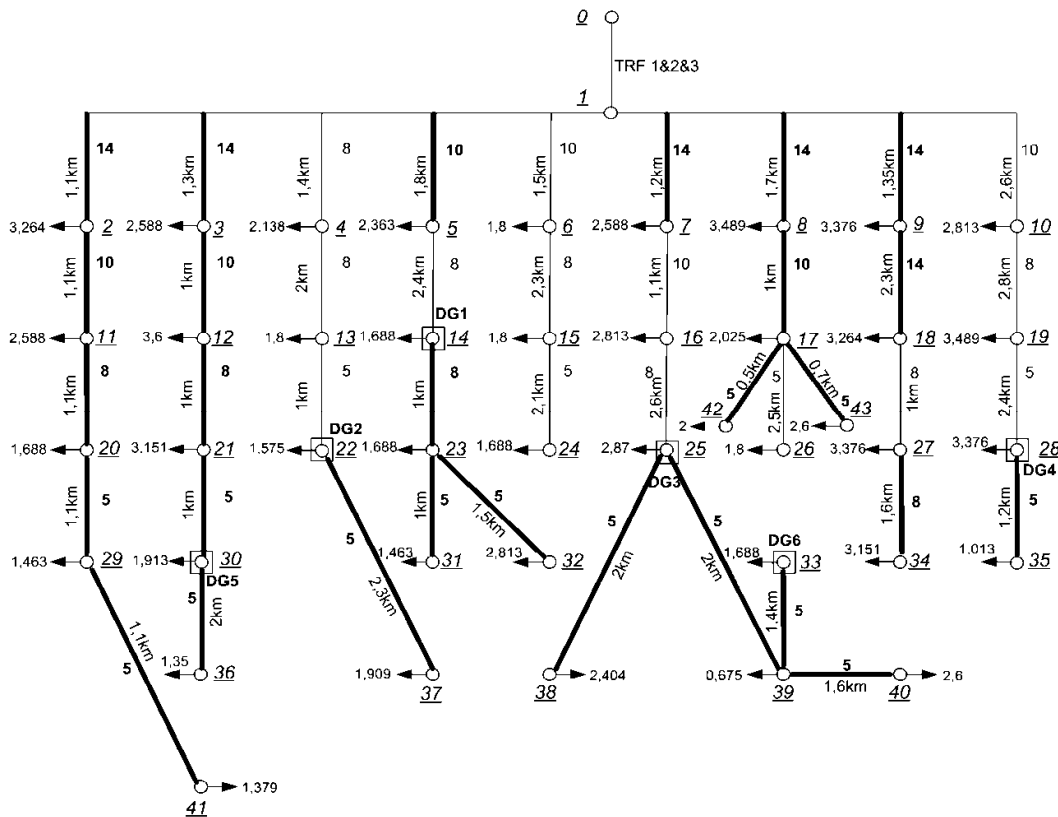
Табела 4.3 – Укупни трошкови у мрежи [US\$ x 103]

Случај	Инвестиц. трошкови	Трошкови губитака	Трошкови прекида (услед отказа грана)	Трошкови прекида (услед отказа дистрибутивног генератора)	Трошкови производних губитака дистрибутивног генератора	Укупан трошак
1	1515.30	1331.40	1900.60	0	35.10	4782.40
2	1326.80	1421.50	1907.60	1008.40	33.02	5697.32
3	1796.30	2207.9	1893.70	-	-	5897.90

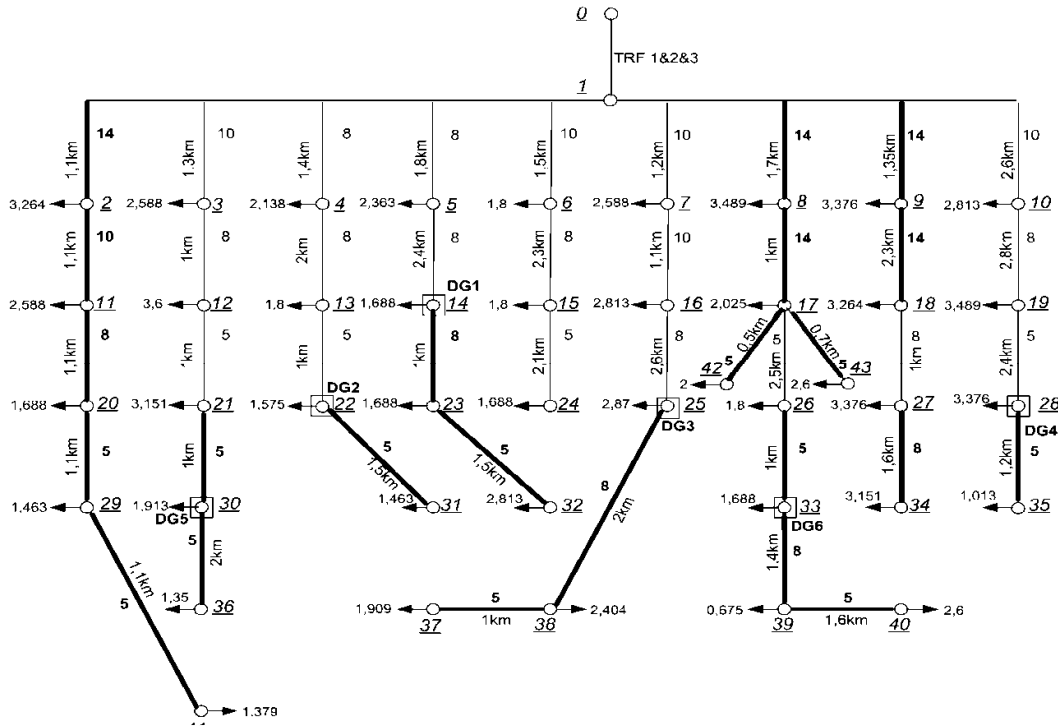
У случају 2 присуство дистрибутивних генератора је узето у обзир, али без разматрања трошкова неиспоручене енергије услед отказа генератора. Применом предложеног приступа на исти начин као и у првом случају, пронађено је најбоље решење које је дато у табели 4.3. Ово решење је приказано на слици 4.3, где су унапређења (појачања и нова изградња) представљени дебелим линијама упоредо са њиховим капацитетима изражених у MVA. Треба запазити да су инвестициони трошкови мањи, а да су трошкови губитака већи него у случају 1. Трошкови неиспоручене енергије услед отказа дистрибутивних генератора су већи него у случају 1, што исто важи и за укупне трошкове добијеног решења. Овај резултат указује на неопходност узимања у обзир отказа генератора у одређивању најбољег плана развоја радијалних дистрибутивних мрежа.



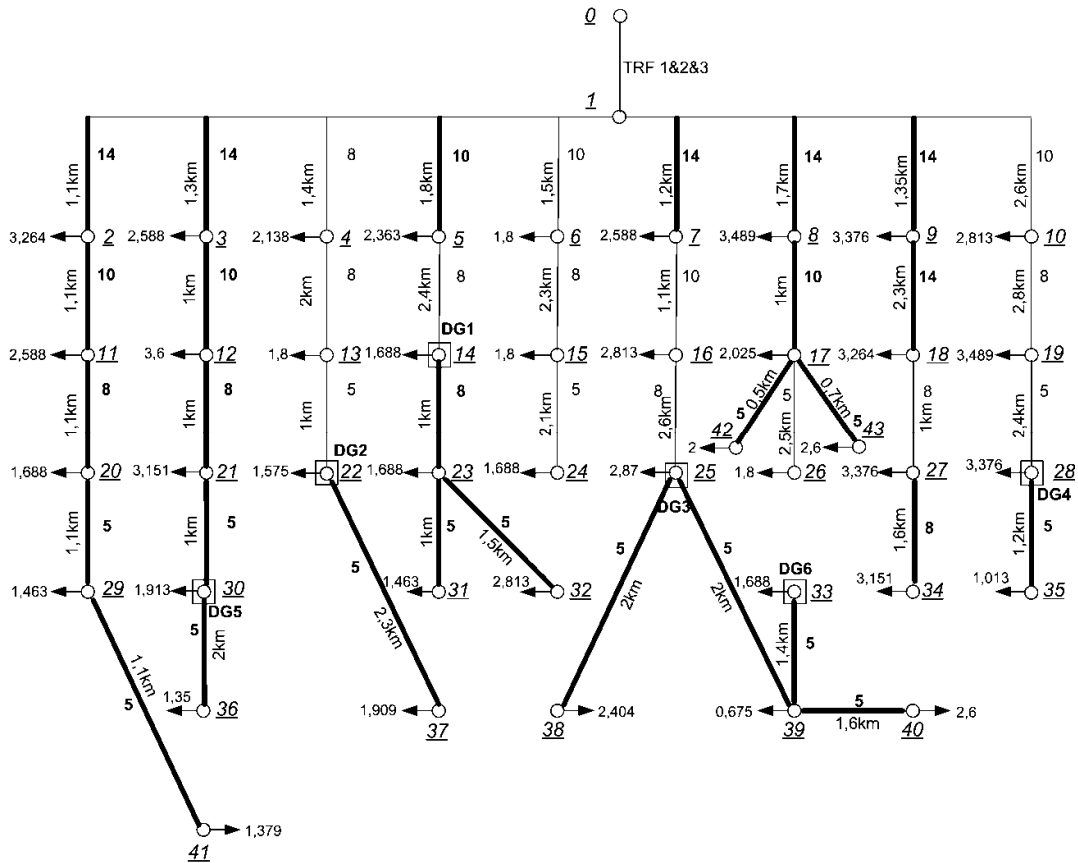
Слика 4.1. Тест мрежа за Пример 1



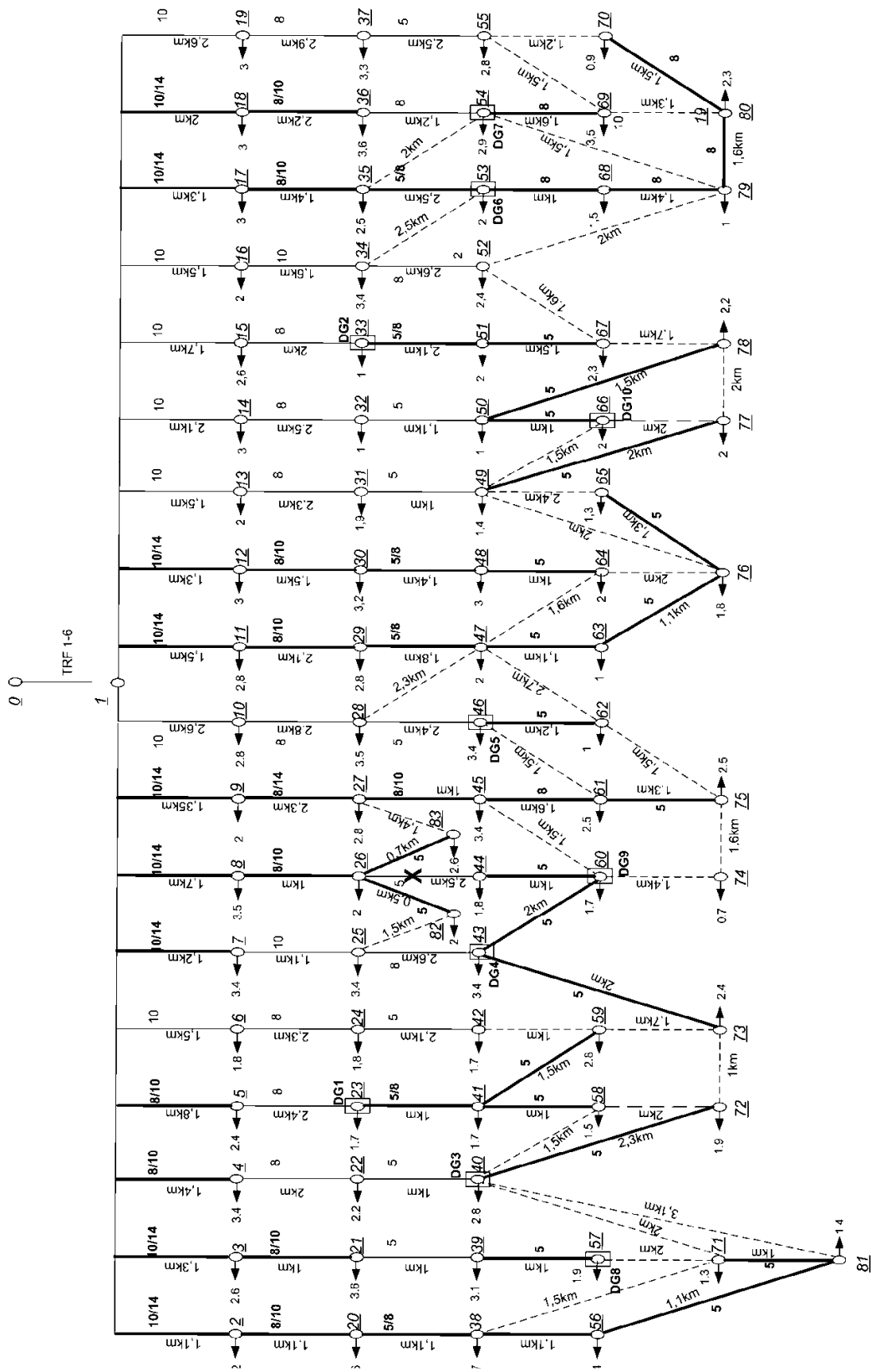
Слика 4.2. Оптимално решење за Случај 1



Слика 4.3. Оптимално решење за Случај 2



Слика 4.4. Оптимално решење за Случај 3



Слика 4.5. План развоја дистрибутивне мреже за Пример II

У случају 3 не узима се у обзир присуство дистрибутивних генератора. Користећи предложени приступ на исти начин као и у претходним случајевима, нађено је најбоље решење чија је цена дата у табели 4.3. Ово решење је представљено на слици 4.4. Треба запазити да су овде инвестициони трошкови и трошкови губитака већи него у случају 1. Трошкови прекида који представљају само трошкове неиспоручене енергије услед отказа грана су незнатно мањи него у случају 1, док су укупни трошкови решења значајно већи од укупних трошкова добијених у случају 1. Овај резултат указује на позитиван утицај дистрибутивних генератора у радијалним мрежама, односно показује да присуство дистрибутивних генератора може да смањи укупне трошкове развоја радијалних дистрибутивних мрежа, иако острвски рад генератора није дозвољен. Треба нагласити да су горе поменути проблеми (случај 2 и случај 3) решени применом *MILP* модела за целу дистрибутивну мрежу, а добијена решења су опет иста као и решења добијена предложеним приступом. Ефикасност предложеног хибридног алгоритма може се приписати комбинацији *MILP* и алгоритма симулираног каљења (*SA*) - у даљем тексту *SA – MILP* приступ. Употребом *MILP* модела, заједно са концептом локалних мрежа и технике измене грана, генерише се низ суседних решења високог квалитета у сваком кораку алгоритма симулираног каљења (*SA*), који дефинише процес претраге, тако да су најперспективнији региони у простору претраге темељно експлоатисани, у ограниченом броју корака.

4.2. ПРИМЕР МРЕЖЕ II

Дата је 20 kV дистрибутивна мрежа приказана на слици 4.5. Ова мрежа представља проширену мрежу која је разматрана у случају 1. Дужине (*km*) свих грана и прогнозирана оптерећења (*MVA*) у свим чворовима дати су на слици 4.5. Десет дистрибутивних генератора $SG1 = 4 MVA$, $SG2 = 4 MVA$, $SG3 = 2,5 MVA$, $SG4 = 5 MVA$, $SG5 = 3 MVA$, $SG6 = 4,5 MVA$, $SG7 = 5 MVA$, $SG8 = 4,5 MVA$, $SG9 = 3 MVA$, $SG10 = 4 MVA$, су повезана за чворове 23, 33, 40, 43, 46, 53, 54, 57, 60 и 66, респективно. У напојној станици додата су још три трансформатора са капацитетом од 33 *MVA* по трансформатору. Јединична цена неиспоручене енергије за потрошаче у чворовима **(60)-(62)**, **(71)-(75)** и **(81)-(83)** је 5 *US\$/kWh*, а за све остале потрошаче је 2 *US\$/kWh*. Сви остали параметри су исти као и у случају 1. Ова мрежа садржи 84 чвора и 117 грана. На слици 4.5 је представљен план развоја радијалне дистрибутивне мреже, где су подебљаним линијама означени нова изградња и појачања, пуним линијама означене постојеће гране, а испрекиданим линијама предложене (потенцијално) нове гране. Ознака типа (број1/број2) на слици 4.5, поред подебљаних линија, показује капацитет постојеће гране (број1) и капацитет појачане гране (број2). Минимална укупна вредност трошкова за ову мрежу је дата у табели 4.4. Ово решење не може да буде пронађено коришћењем предложеног *MILP* приступа у "разумном" времену. Наиме, предложени *MILP* модел не може да пронађе решење за ову мрежу у оквиру 24 часа, што се сматра "разумним" периодом времена. С друге стране, предложени декомпозициони приступ (*SA – MILP*) решење за овај проблем проналази за 35652 секунди, односно за 9 часова и 52 минута.

Табела 4.4 – Укупни трошкови у мрежи за Пример II [US\$ x 103]

Инвест. трошкови	Трошкови губитака	Трошкови прекида (услед отказа грана)	Трошкови прекида (услед отказа дист. генератора)	Трошкови производних губитака дист. генератора	Укупан трошак
2926.80	3052.10	3170,80	0	59.02	9208.70

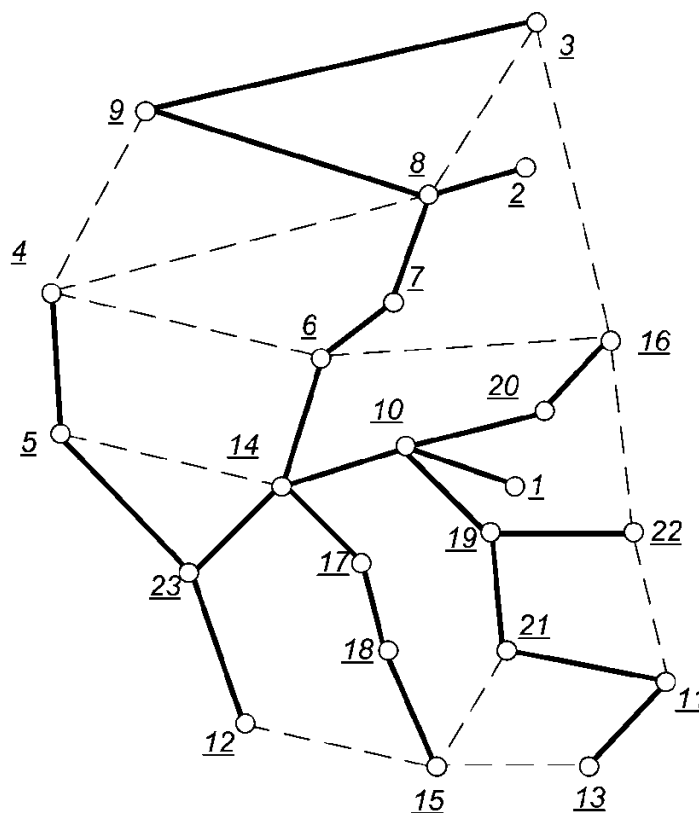
4.3. ПРИМЕР МРЕЖЕ III

Пример мреже III представља проблем планирања диситрибутивне мреже изложен у [100], са циљем минимизације инвестиционих трошкова и трошкова губитака. Ту се проблем планирања диситрибутивне мреже решавао мрављим алгоритмом, што је детаљније описано у прилогу, поглавље П2.3. Такође је у [100] показано да примена мрављеог алгоритма даје квалитетнија решења од резултата добијених применом генетског алгоритма [162]. Подаци за овај проблем су расположиви у [70],[100]. Разматра се 34,5kV диситрибутивна мрежа напајана са трансформатором снаге 10MVA, која снабдева индустријско нафтно подручје са 21 чвором потрошње, без диситрибутивних генератора. За овај пример максимално прихватљиви пад напона је 3%, фактор снаге је 0.9, цена трошкова губитака је 0.05 US\$/kWh, фактор губитака је 0.35, интересна стопа је 0.1, а период планирања је 20 година. План развоја ове радијалне диситрибутивне мреже је дат на слици 4.6, где испрекидане линије означавају предложене (потенцијално) нове гране, а пуне линије изграђене гране. Оптимално решење је добијено коришћењем MILP модела (3.2)-(3.23), на одговарајући начин прилагођеног да одговара наведеном циљу оптимизације. Исто решење је добијено коришћењем предложеним SA – MILP приступом. Ово решење је једнако решењу приказаном у [100], које је добијено коришћењем алгоритма система мравље колоне (Ant colony system algorithm – ACSA). Исто решење се добија такође у [70] коришћењем чистог алгоритма симулираног каљења (SA). Треба истаћи да оба ова приступа (ACSA и чист SA) не користе апроксимације код прорачуна токова снага који се изводе у оптимизационом алгоритму. Табела 4.5 показује сумарно добијене резултате. Може се видети да се трошкови губитака добијени MILP моделом разликују за 1,16% од оних добијених у [70],[100]. Оптимално решење је постигнуто коришћењем 21 сегмента код линеаризације трошкова губитака. Даље повећање броја сегмената доводи до занемарљивог побољшања тачности решења. С друге стране, смањење броја сегмената води до смањења потребног времена за решавања, али и тачности. За 10 сегмената време за решавање проблема MILP моделом се своди на 1.1 секунду, док се разлика у добијеним решењима трошкова губитака повећава на 6.1%. За мање од 7 сегмената тачност опада за више од 15%, па оптимално решење не може да буде пронађено. Због тога се наведени 21 сегмент користи за линеаризацију трошкова губитака у свим другим случајевима који су разматрани у овој тези. Треба истаћи да је овај проблем планирања такође разматран у [135], где се за решавање проблема користи мешовито-целобројно нелинеарно програмирање, и у [141] где се за решавање проблема користи мешовити-целобројни конични модел са полиедарном релаксацијом. У оба случаја се добијају исти резултати који су показани у табели 4.5. Треба истаћи да сви ови приступи не разматрају присуство

дистрибутивних генератора, односно не уважавају трошкове неиспоручене енергије услед отказа дистрибутивних генератора, као и трошкове губитака производње истих услед отказа грана. Уважавање наведених трошкова може имати значајан утицај на планирање развоја мрежа, као што је приказано у примерима анализираним у случају 1. Поред тога, осим у [70] сви горе поменути приступи нису уважавали трошкове неиспоручене енергије услед отказа грана, док приступи [70],[100] нису узимали у обзир ни могућност постојања више различитих пресека (капацитета) грана у дистрибутивној мрежи.

Табела 4.5 – Укупни трошкови у мрежи за Пример III [US\$]

Решења	Инвестиције	Губици	Укупно
Оптimalно	151892	21250	173142
Оптimalно [100] / [70]	151892/151892	21007/21021	172899/172913



Слика 4.6. План развоја дистрибутивне мреже за Пример III

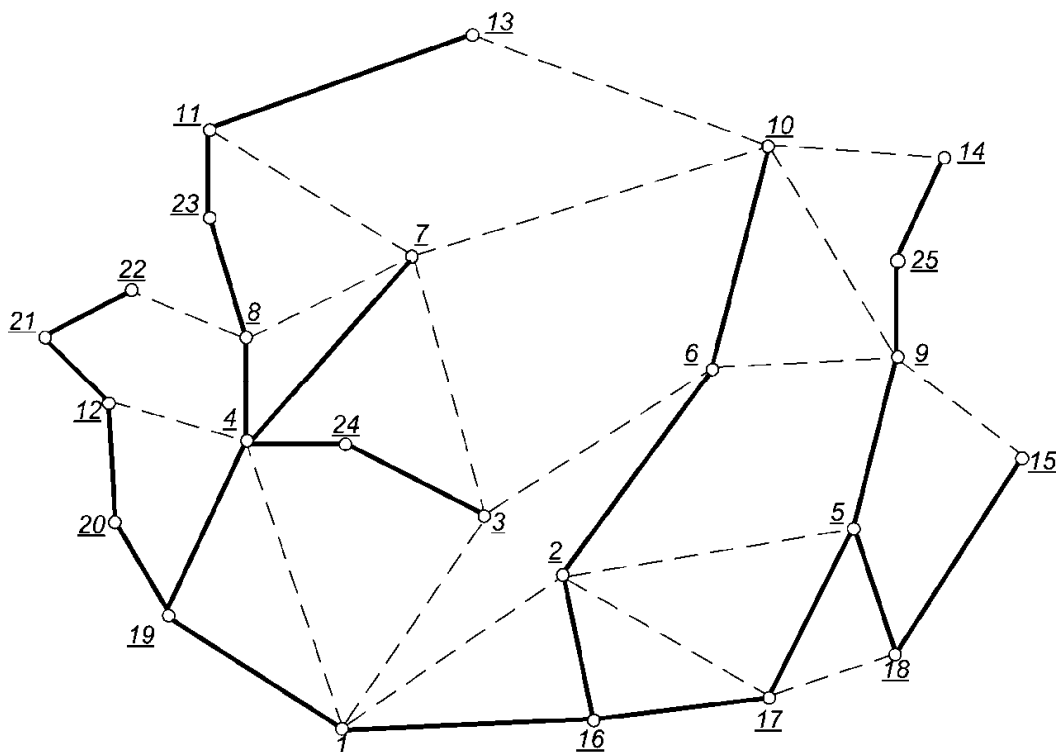
4.4. ПРИМЕР МРЕЖЕ IV

У примеру мреже IV се разматра проблем планирања мреже са 24 чвора, без дистрибутивних генератора, решавана у [70], коришћењем чистог SA алгоритма, са циљем да се добије решења са минималном укупном садашњом вредности трошка (инвестициони трошак, трошак губитака и трошак неиспоручене енергије услед отказа грана). Подаци за ову мрежу су дати у [70]. План развоја дистрибутивне мреже је дат на слици 4.7, где испрекидане линије означавају предложене (потенцијално) нове гране, а пуне линије изграђене гране. Ово решење је добијено коришћењем предложеног MILP модела (3.2)-(3.23), на

одговарајући начин прилагођеног да одговара горе наведеном циљу оптимизације. За линеаризацију трошкова губитака се користи 21 сегмент и постиже се иста тачност прорачуна губитака као и код примера мреже III. Применом предложеног *SA – MILP* приступа добија се исто решење. Ово решење је једнако решењу приказаном у [70]. Треба поново истаћи да *SA* приступ предложен у [70] не узима у обзир више пресека проводника, као ни присуство дистрибутивних генератора у мрежи, што има значајан утицај на избор завршног плана развоја дистрибутивних мрежа. Штавише, стандардни *SA* приступ предложен у [70], за разлику од *SA* приступа предложеног у овој дисертацији, нема механизам диверзификације, па се на тај начин смањује способност овог алгорита да избегне локални минимум и да пронађе глобално оптимално решење [15].

Табела 4.6 – Карактеристике *MILP* и *SA – MILP* алгорита

		<i>MILP</i>	<i>SA – MILP</i>	
Пример I	Случај 1	16244	3229	CPU време (s)
	Случај 2	216	364	
	Случај 3	239	387	
Пример II	-----	35652		
Пример III	2.3	12.4		
Пример IV	3.6	16.4		



Слика 4.7. План развоја дистрибутивне мреже за Пример IV

Прорачунска (CPU) времена за добијање најбољег решења разматраних проблема приказана су у табели 4.6. Може се видети да је за мање комплексне проблеме (случај 2, случај 3, пример мреже III и пример мреже IV) *MILP* алгорита

бржи од $SA - MILP$ приступа. Међутим, у случају 1, прорачунско време $MILP$ модела је значајно повећано. Ово повећање је резултат уважавања трошкова прекида услед отказа генератора, који значајно повећавају рачунарску комплексност проблема, као што се може видети из релација (3.14)-(3.18). У овом случају $SA - MILP$ приступ налази најбоље решење брже од $MILP$ алгоритма. Шта више, у примеру II, који је комплекснији од случаја 1, $MILP$ модел не може да нађе решење у "разумном" времену, док $SA - MILP$ приступ то може.

5. ЗАКЉУЧАК

У овој дисертацији је предложен нови приступ којим се унапређују досадашњи модели и алгоритми за статичко планирање развоја радијалних дистрибутивних мрежа реалних димензија у присуству дистрибутивних генератора. Предложени приступ уважава инвестиционе трошкове, трошкове губитака, трошкове прекида напајања потрошача услед кварова на гранама и дистрибутивним генераторима, као и трошкове губитака производње дистрибутивних генератора услед кварова на гранама.

Предложени алгоритам је комбинација, односно хибрид алгоритама симулираног каљења и мешовитог-целобројног линеарног програмирања (*MILP*), који решава статички проблем планирања. Да би се омогућило решавање реалних проблема планирања развоја радијалних дистрибутивних мрежа предвиђен је нови декомпозициони приступ који укључује концепт локалних мрежа за декомпоновање оригиналног проблема (оригиналне мреже) у низ подпроблема (локалних мрежа), који се решавају предложеним *MILP* моделом. Процес решавања је итеративно вођен алгоритмом симулираног каљења. Алгоритам симулираног каљења (*SA*) укључује механизме интензификације и диверзификације упоредо са немонотоним шемом редукције температуре, која обезбеђује да су најперспективнији региони у простору претраге темељно експлоатисани, како би се нашло најбоље, односно оптимално решење.

Верификација ефикасности предложеног хибридног хеуристичког алгоритма извршена је на више примера упоређивањем са правим глобалним оптимумом. Добијени нумерички резултати показују да предложени хеуристички алгоритам генерише квалитетније планове развоја дистрибутивних мрежа од хеуристичких и метахеуристичких алгоритама који су до сада развијени. Тиме је показано да предложени приступ има потенцијал да буде ефикасан алат за генерисање планова развоја са минималним укупним трошком за различите величине радијалних дистрибутивних мрежа са дистрибутивним генераторима. Резултати такође указују на неопходност уважавања отказа генератора у циљу добијања најбољег плана развоја радијалних дистрибутивних мрежа.

Правци даљег рада и истраживања у овој области планирања развоја дистрибутивних мрежа, треба да обухвате:

- планирање развоја потенцијално упетљаних дистрибутивних мрежа (урбане мреже), као и
- уважавање могућности изолованог (острвског) рада дистрибутивних генератора, односно процену евентуалних позитивних ефеката оваквог режима рада на укупне трошкове развоја дистрибутивне мреже.

ПРИЛОГ 1. МЕТАХЕУРИСТИКЕ У ПЛАНИРАЊУ РАЗВОЈА ДИСТРИБУТИВНИХ МРЕЖА

1.1. ОПШТЕ О МЕТАХЕУРИСТИКАМА

1.1.1. МАТЕМАТИЧКА ОПТИМИЗАЦИЈА

Оптимизација у општем смислу означава поступак проналаска најбољег (оптималног) решења неког проблема, под одређеним условима. У математици оптимизација проучава проблеме налажења екстремних вредности функције дефинисане на неком скупу.

Једнокритеријумска оптимизација може се представити:

Дата је функција $f: X \rightarrow Y$, ($Y \subseteq R$)

$$\min / \max y = f(x), x \in X \quad (\text{П1.1})$$

уз задовољавање ограничења:

$$g_l(x) \geq 0, l = 1, 2 \dots L \quad (\text{П1.2})$$

$$h_k(x) = 0, k = 1, 2 \dots K \quad (\text{П1.3})$$

$$x_i^l \leq x_i \leq x_i^u, i = 1, 2 \dots n \quad (\text{П1.4})$$

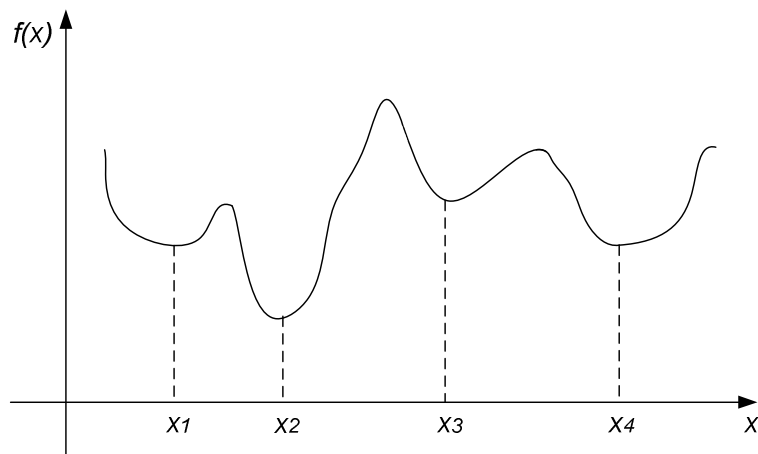
Решење x по правилу представља n -торку $x = (x_1, x_2 \dots x_n)$. Елементи решења x_j где је $j = 1, 2, 3 \dots n$ се зову променљиве одлуке. Решења x могу математички бити ограничена $(g_l(x), h_k(x), x_i^l, x_i^u)$, па свако x које задовољава ова ограничења се назива допустиво решење. Сва допустива решења формирају допустиви скуп X , који се назива простор решења или простор претраге. Простор решења односно претраге може бити:

- континуалан, када решење поприма било коју вредност из подскупа реалних бројева, или
- дискретан, када су решења одређене дискретне вредности из скупа реалних бројева (обично се ради о скупу целих бројева, о скупу природних бројева и нуле, или о бинарном скупу $\{0, 1\}$). Проблеми који имају дискретан и коначан простор решења односно претраге, називају се још комбинаторни проблеми.

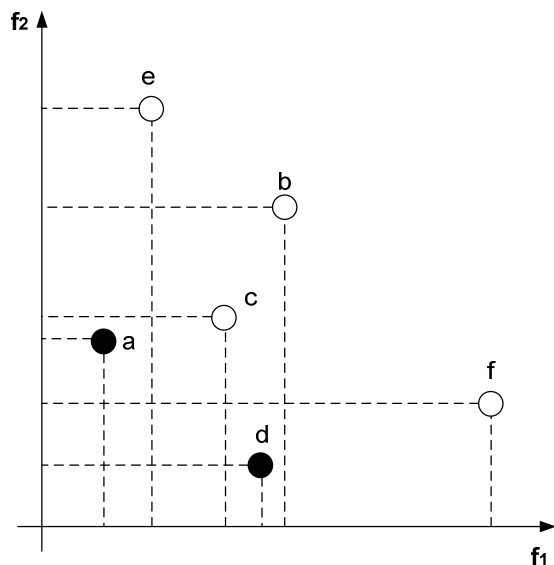
Квалитет решења x се вреднује критеријумском функцијом или функцијом циља која врши пресликавање $f: X \rightarrow Y$ ($Y \subseteq R$) и означава се $y = f(x)$. Скуп Y се назива простор вредности функције циља. С обзиром да ли се ради о минимизацији или максимизацији критеријумске функције, односно функције циља, дефинише се оптимално решење, односно глобални оптимум $x^* = (x_1^*, x_2^* \dots x_n^*)$ за који важи:

- код минимизације функције циља: $f(x^*) \leq f(x)$ за $\forall x \in X$,
- код максимизације функције циља: $f(x^*) \geq f(x)$ за $\forall x \in X$.

Код једнокритеријумске оптимизације се минимизација/максимизација функције циља врши по једном критеријуму, па се свако решење може преликати у број који означава квалитет тог решења. На слици је приказана једнокритеријумска функција циља, где код проблема минимизације, тачке x_1 , x_3 и x_4 представљају локалне минимуме, а тачка x_2 глобални минимум.



Слика П1.1. Једнокритеријумска оптимизација



Слика П1.2. Парето доминација: Решења a доминира над решењима e , c и b . Такође решење d доминира над решењима b и f . Тако решења a и d припадају Парето скупу (недоминантних решења) јер нису доминирани од ниједног другог решења

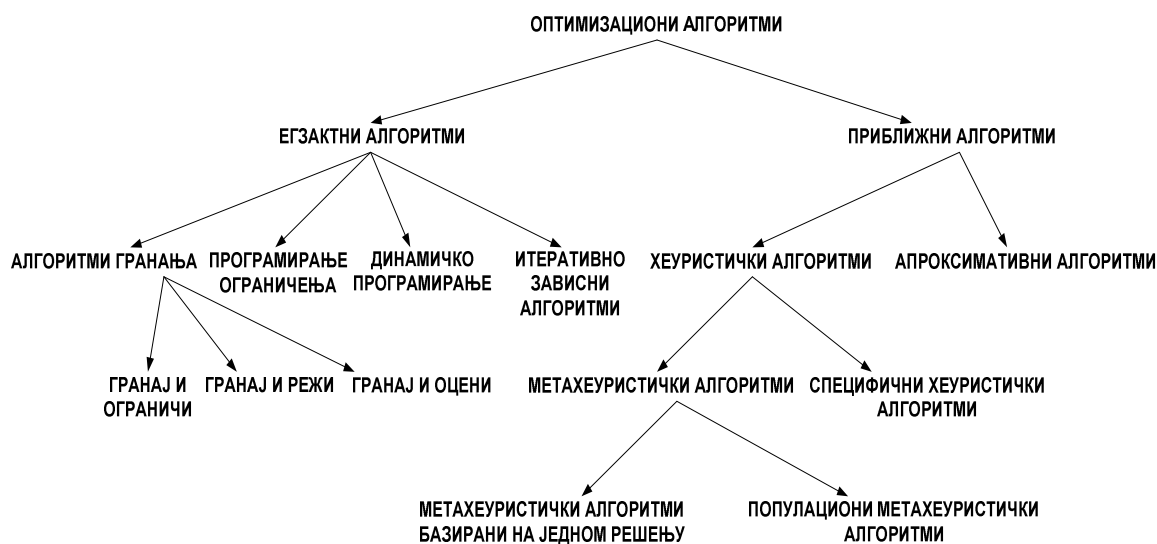
С друге стране, имамо и вишекритеријумску оптимизацију, где се минимизација/максимизација функције циља $f: X \rightarrow Y$ ($Y \subseteq R^m$) врши по више критеријума m , односно врши се једновремено оптимизовање више једнокритеријумских проблема који су међусобно зависни. У овом случају се решења преликавају у вектор бројева, где сваки елемент тог вектора представља преликовано решење по једном критеријуму. Овако добијена решења код вишекритеријумске оптимизације у општем случају нису упоредива [142].

На пример, представимо скуп решења вишекритеријумског оптимизационог проблема минимизације за две функције циља $f_1(x)$ и $f_2(x)$ као на слици П1.2. Каже се да вектор решења $a \in X$ доминира над вектором решења $b \in X$, ако за свако $i = 1, 2 \dots n$ важи $f_i(a) < f_i(b)$. Решење које није доминирано ниједним другим решења назива се Парето оптимално решење, а скуп таквих решења Парето скуп или скуп недоминатних решења. Битно је истаћи да Парето оптимална решења не могу бити побољшана по једном критеријуму, а да то нема последицу погоршања по другом критеријуму.

Оптимизационе методе, по квалитету решења које дају, могу се поделити у две велике групе:

- прецизне методе, које гарантују налажење оптималног решења,
- приближне методе, које дају задовољавајућа решења, али не гарантују да ће то решење бити оптимално.

Једна таква подела дата у [15] је приказана на слици П1.3. Овде ће се пажња посветити метахеуристичким методама са нарочитим освртом на њихове особине и значај.



Слика П1.3. Подела оптимизационих алгоритама [15]

1.1.2. ХЕУРИСТИКЕ

Главна особина и снага хеуристичких алгоритама је у томе што, користећи одређена сазнања о проблему који се решава, значајно смањују простор претраге, и претрагу упућују у атрактивне обећавајуће просторе решења. Ова околност чине хеуристике веома ефикасним средством за решавање оптимизационих проблема, тако што се претраживање смањивањем простора претраге, знатно убрзава, а усмеравањем претраге се обезбеђују задовољавајућа решења. На тај начин се постижу две битне ствари а то су:

- квалитет решења који се добија радом алгоритама - ефикасност алгоритама,
- потребно време за рад алгоритама - ефикасност алгоритама.

Битно је нагласити да хеуристички алгоритама у општем случају дају задовољавајућа решења у реалном времену, али не гарантују да су решења оптимална. Хеуристике се углавном користе за решавање слабо структурираних проблема, као и НП-тешких проблема за које не постоје егзактни алгоритама

полиномалне сложености, а по питању проблема које могу да решавају деле се на:

- Хеуристике специфичних проблема, које су намењене за решавање тачно одређених проблема,
- Метакхеуристички алгоритми, који су намењени за решавање широког скупа проблема и који ће овде бити прецизније описани.

1.1.3. МЕТАХЕУРИСТИКЕ

За разлику од егзактних метода, метакхеуристике омогућавају решавање проблема великих димензија дајући задовољавајућа решења у разумном року. При томе не постоји гаранција да ће пронађено решење бити глобално оптимално решење. Метакхеуристике добијају на значају и популарности последњих 20 година. Њихов значај је у томе што оне пружају ефикасност и ефективност код решавања широког спектра великих и сложених проблема. Нарочито, велики број проблема у инжењерству има типичне карактеристике проблема комбинаторних оптимизација великих размера. У проблемима ове врсте, број могућих решења расте експоненцијално са величином проблема. Стога, примена оптимизационе методе за проналажење оптималног решења није рачунски изводљива. Из тог разлога, метакхеуристичке технике претраге се често користе за постизање високо квалитетних решења у разумном временском року.

Као што је већ истакнуто, метакхеуристике представљају скуп правила за решавање разних оптимизационих проблема. Ова правила су најчешће заснована на некој општој идеји или на аналогiji са неким процесима у природи (биолошки, физички итд.). Такође даљим развојем, а у циљу повећања њихове ефикасности, приступа се и комбинацији више техника (хибридизација) које имају за крајњи циљ да се остваре боље перформансе, него поједини алгоритми од којих је хибрид састављен. Међу оптимизационим методама постоје оне које примењују локалну претрагу, као и оне који користе неконвексни приступ оптимизације, код којих се прихватају и решења из суседства која су гора од текућих. Најпопуларније методе које превазилазе просту локалну претрагу су генетски алгоритми (*GA*) [143-146] (као и остале еволуционе технике, као што су еволуционо програмирање, еволуционе стратегије, итд), симулирано каљење (*SA*) [147], и табу претраживање (*TS*) [131]. Такође, алгоритам роја честица (*PSO*) [121], као и мрављи алгоритми (*ACO*) [148], су технике оптимизација које су показале велики потенцијал у последње време. Примена метакхеуристика је заступљена у великом броју области као на пример [15]:

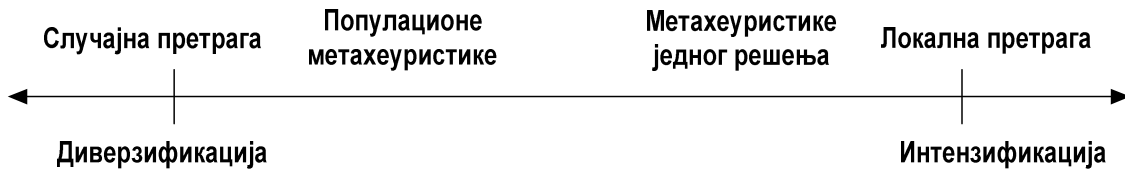
- инжењерски дизајн, оптимизација топологија и структурална оптимизација у електроници, аеродинамика, динамика флуида, телекомуникације, роботика.
- учење машина, биоинформатика и рачунарске биологије, финансије.
- систем моделовања, симулација и идентификација у хемији, физици, биологији и контрола сигнала, обрада слике.
- планирање у рутирање проблеме, планирања, производни проблеми, логистика и транспорт, управљање ланцем снабдевања итд.

Метакхеуристике се могу поделити на алгоритме који раде над једним решењем (нпр. симулирано каљење, табу претрга), и на популационе алгоритме који раде са скуповима решења (нпр. алгоритми еволуцијског рачунања).

Све метахеуристике полазе од једног (или више) почетног решења на основу којег (или којих) стварају нова решења. Како је простор претраге велики, па исцрпна претрага није могућа у разумном времену, прибегава се двома стратегијама.

- груба (глобална) претрага - истраживање (*exploration*) простора за претрагу (диверзификација), или
- фина (локална) претрага - искоришћавање (*exploitation*) најбољих пронађених решења (интензификација).

Циљ добре метахеуристике јесте постизање баланса између глобалног и локалног претраживања (*exploitation and exploration*), што може обезбедити квалитетна и брза решења проблема који метахеуристика решава. Обећавајући региони су одређени добијеним "добрим" решењима. Код интензификације се обећавајући региони подробније истражују како би се пронашла боља решења. Код диверзификације се морају посетити неистражени региони, како би се обезбедило да сви региони у простору претраге буду равномерно истражени, односно да претрага није ограничена само на делимичан број региона.



Слика П1.4. У општем случају, метахеуристике које раде са једним решењем су више оријентисани искоришћавању добрих решења (локална претрага, интензификација), док су популационе метахеуристике истраживачки оријентисане (диверзификација) [15]

Екстремни случајеви претраге у смислу истраживања, односно искоришћавања решења, као што је показано на слици П1.4, су [15]:

- случајно претраживање (код сваке итерације се генерише случајно решење у простору претраге), и
- локална претрага (кроз сваку итерацију се селекује најбоље суседно решење које унапређује (побољшава) текуће решење).

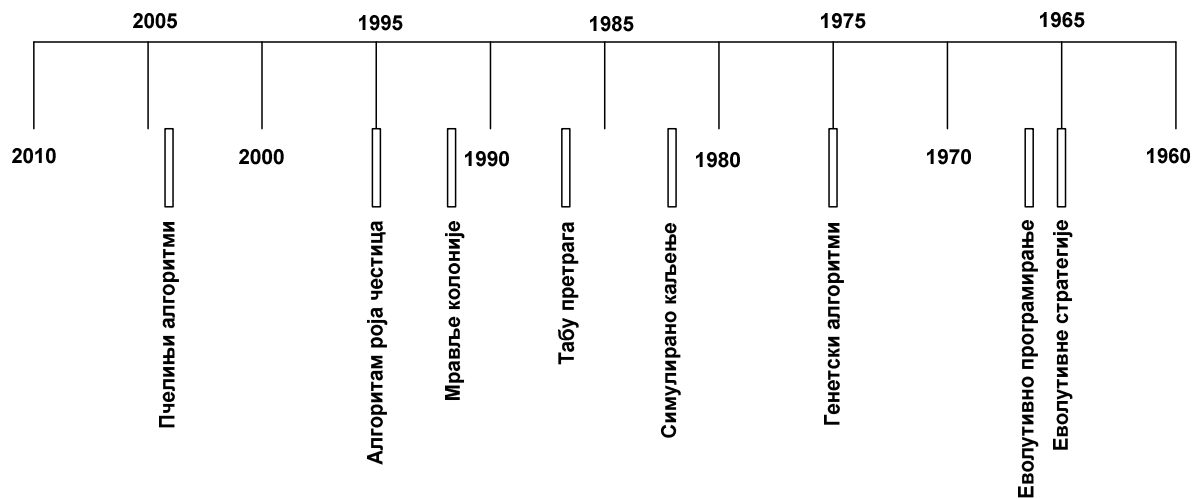
Метахеуристике могу бити подељене по више основа [15]:

- алгоритми инспирисани природом (живим светом или физичким процесима) и алгоритми који нису инспирисани природом,
- алгоритми са меморијом (који меморишу информације током претраге) и алгоритме без меморије (локална претрага, симулирано каљење...),
- детерминистички (који решавају оптимизациони проблем доношењем детерминистичких одлука. На тај начин код ових алгоритама за исто почетно решење ће се поновним пропуштањем добијати увек исто крајње решење) и стохастички алгоритми (ток претраге се одређује стохастички, тако да код ових алгоритама за исто почетно решење поновним пропуштањем се у општем случају добијају различита крајња решења,
- алгоритми који током претраге раде над једним решењем (*S* - метахеуристике, као нпр. локална претрага, симулирано каљење, табу претраживање), и на популационе алгоритме (*P*-метахеуристике) који током претраживања раде са скуповима решења (еволуцијски алгоритми,

алгоритми роја честица). Ове две групе алгоритама су комплементарне: алгоритми који раде са једним решењем су оријентисани искоришћавању добрих решења (локална претрага, интензификација). Они имају моћ да се интензивира претрага у локалним регионима. Популационо засновани алгоритми су истраживачки оријентисани. Они омогућавају бољу диверзификацију у целом простору претраге,

- итеративни алгоритми који стартују са комплетним почетним решењем (или популацијом почетних решења) које се итеративно модификује и "побољшава", и конструкциони алгоритми који стартују из "празног" решења које се кроз сваки корак претраге гради, све док се не добије комплетно решење. Већина метахеуристика припадају итеративним алгоритмима.

Хронологија појављивања неких од најчешће кориштених метахеуристика, дата је на слици П1.5.



Слика П1.5. Метахеуристике са хронологијом појављивања

У добре стране метахеуристика спадају:

- општи приступ решавања проблема,
- прилагодљивост специфичностима реалних проблема,
- лака имплементација.

У мане метахеуристика спадају:

- велики број параметара се морају подешавати експерименталним путем,
- недостатак робустности, односно мале промене параметара често доводе до великих разлика у претраживању простора решења.

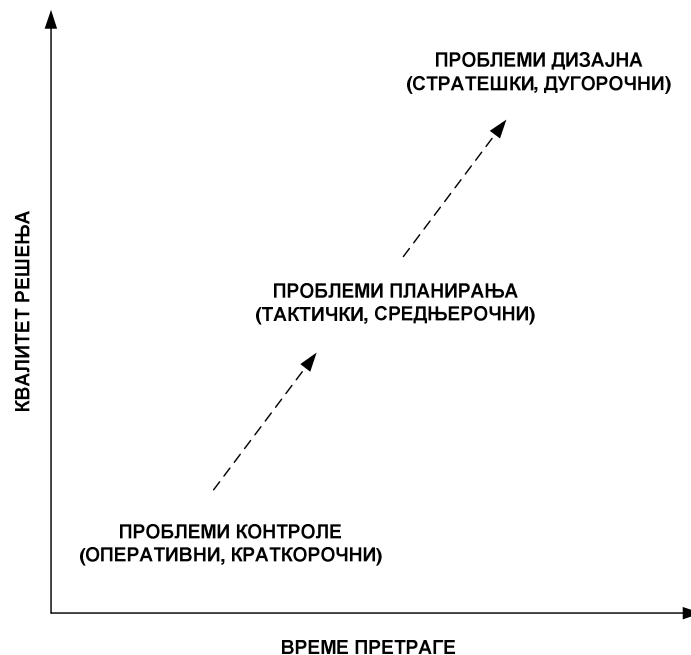
Избор метахеуристичког алгоритма

Као што је већ поменуто, међу основне перформансе метахеуристичког алгоритма спадају:

- квалитет решења које се добија радом алгоритма - ефективност алгоритма
- потребног времена за рад алгоритма - ефикасност алгоритма.

Релативни значај између две главне мере перформансе алгоритма (квалитет решења и време за претрагу), зависи од карактеристике циља проблема оптимизације који се решава. Тако се могу појавити две основне врсте проблема који се решавају [15]:

- **Проблеми дизајна:** Дизајн проблеми су углавном решавају једном. Они захтевају веома добар квалитет решења док је време на располагању за решавање истог релативно дугачко (нпр. неколико сати, дана, месеци). У овој класи проблема, се могу наћи стратешки проблеми (дугорочни проблеми), као што су планирање развоја електроенергетске мреже. Ови проблеми укључују важна финансијска улагања, и свака несавршеност ће имати дугогодишњи утицај на решење. Стога, критичан аспект је квалитет решења, а не време претраживање (слика П1.6). Ако је могуће код оваквих проблема требају да се користе егзактни алгоритми који обезбеђују глобални оптимум.
- **Проблеми контрола:** Проблеми контрола представљају другу крајност где се проблем мора често решити у реалном времену. Ова класа оперативних проблема обухвата краткорочне одлуке, као што су рутирање порука у рачунарској мрежи или управљање саобраћајем у граду. За оперативне проблеме одлучивања потребна су веома брзе хеуристике, док је квалитет решења мање битан.
- **Средњерочни проблеми:** Између ове две екстремне класе проблема, може се издвојити и средња класа проблема као што су тактички проблеми (средњерочни проблеми). У овом случају компромис између квалитета решења и времена решавања мора бити оптимизован. У општем случају, код ових проблема се не користе егзактни алгоритми.



Слика П1.6. Различите класе проблема у смислу компромиса између квалитета решења и времена решавања: дизајн (стратешки, дугорочни) проблеми, проблеми планирања (тактички, средњерочни), проблеми контроле (оперативни, краткорочни) [15]

С друге стране по теорему "нема бесплатног ручка" (енгл. *No free lunch*) не може се издвојити алгоритам који је супериорнији од осталих алгоритама по питању квалитета и брзине добијених решења [149]. Тако ако не неки алгоритам постиже врхунске резултате код једне врсте проблема, за неке друге врсте проблема тај

исти алгоритам то мора да "плати" лошијим резултатима. То значи да "нема бесплатног ручка" код оптимизације. Наиме, за одређени проблем, различити алгоритми могу давати различите резултате, али за широку скуп разноврсних проблема алгоритми се, по квалитету решења која дају, у просеку не разликују. На крају, све то упућује на суштински значај познавања природе проблема који се решава и сходно томе избора одговарајуће технике и алгоритма за његово решавање.

1.2. НАЈЧЕШЋЕ КОРИШТЕНЕ МЕТАХЕУРИСТИКЕ У ПЛАНИРАЊУ РАЗВОЈА ДИСТРИБУТИВНИХ МРЕЖА

1.2.1 АЛГОРИТАМ РОЈА ЧЕСТИЦА

Оригинални алгоритам роја честица

Алгоритам роја честица (*PSO*) је популацијски алгоритам, где популацију чини низ јединки (честица) које лете кроз вишедимензијски простор који претражују. При томе јединке свој положај мењају делом на основу властитог искуства, а делом на основу искуства блиских суседа. Сагласно горе наведеном, алгоритам роја честица је развијен симулацијом јата птица у дводимензионалном простору. Позиција сваке од птица (честице) је репрезентована са x, y координатама, а њихове брзине су изражене са v_x (брзина по x оси) и v_y (брзина по y оси). Модификација позиције честице се реализује информацијом о позицији и брзини. Нека се рој честица представи као јато птица које оптимизује одређену функцију циља. Свака птица зна своју најбољу вредност до сада ($pbest$) и своју позицију (x, y). Ове информације су аналогне сопственом искуству честице. Поврх тога, свака птица зна најбољу вредност у групи ($gbest$) између свих најбољих вредности других птица ($pbest$). Ова информација је аналогна туђем искуству, односно знању о понашању других птица. Свака птица покушава да модификује своју позицију користећи следеће информације:

- текућа позиција (x, y),
- текућа брзина (v_x, v_y),
- раздаљина између текуће позиције и $pbest$, и
- раздаљина између текуће позиције и $gbest$

Ова модификација може бити представљена концептом брзине (модификована вредност за текућу позицију). Брзина сваке птице може бити модификована следећом једначином:

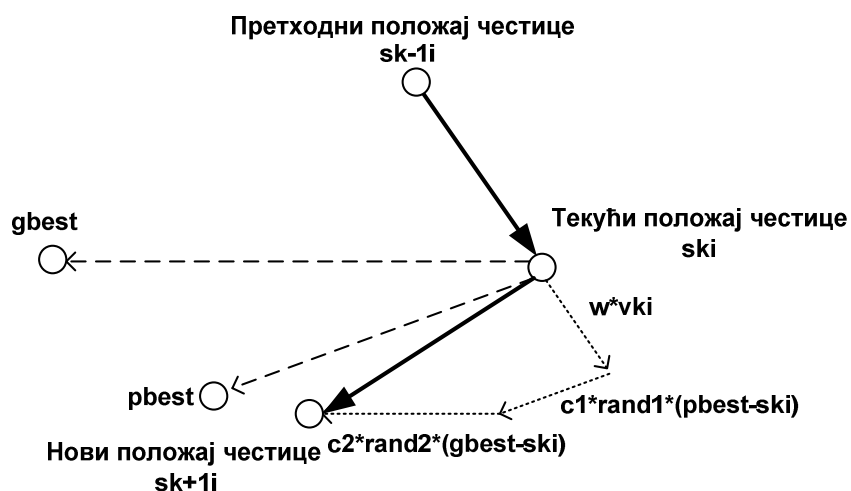
$$v_i^{k+1} = wv_i^k + c_1rand_1 \times (pbest_i - s_i^k) + c_2rand_2 \times (gbest - s_i^k) \quad (П1.5)$$

где су: v_i^k брзина птице i при итерацији k , w је пондерисана функција, c_j је пондерисани коефицијент, $rand$ је случајан број између 0 и 1, s_i^k је текућа позиција птице i при итерацији k , $pbest_i$ је $pbest$ птице i , и $gbest$ је $gbest$ групе. *PSO* који користи наведену једначину П1.5 се назива *Gbest* модел. Коефицијент w се још назива и инерцијална константа чија је вредност најчешће мања од 1. Овај коефицијент се најчешће добија из релације:

$$w = w_{max} - \frac{w_{max} - w_{min}}{iter_{max}} \times iter \quad (П1.6)$$

где су: w_{max} је иницијална вредност, w_{min} је крајња вредност, $iter_{max}$ је максимални број итерација, и $iter$ је текући број итерације. Коефицијент c_1 се назива вредност самоспознаје честице, јер пондерише утицај власитог искуства честице на резултатно кретање исте. Коефицијент c_2 се назива друштвени параметар и пондерише утицај најбољег решења групе на кретање јединке што одговара утицају туђег искуства (групе) на доношење одлуке јединке. Једначина (П1.5) садржи три члана (вектора). Први вектор у П1.5 представља скалирану стару брзину птице односно инерцију саме честице. Други вектор представља кретање према најбољем решењу које је пронашла сама јединка, одговарајуће скалираном. Трећи вектор представља помак према најбољем решењу групе, такођер одговарајуће скалираном. Крајња резултатна брзина јединке је сума сва

три наведена деловања, тако да се јединка наставља кретати по инерцији, при чему се вектор брзине јединке модификује сходно свом и групном најбољем решењу. Без другог и трећег члана, јединка ће наставити "летети" у истом правцу све док не дође до границе простора претраге. Наиме, у том случају покушавају се истражити нови простори претраге решења, стога први члан одговара диверзификацији у процедури претраге. С друге стране, без првог члана, брзина "летења" јединке је дефинисана само користећи његову текућу позицију као и његову и групну најбољу позицију у прошлости. Наиме, јединка ће покушати да конвергира до његовог $pbest$ и/или групног $gbest$, што одговара интензификацији у процедури претраге. Уобичајено је да су w_{max} и w_{min} подешени на 0,9 и 0,4. Тако је на почетку процедуре претраге диверзификација "тешко" пондерисана, док је интензификација "тешко" пондерисана тек на крају процедуре, слично као и код симулираног каљења (SA). Алгоритам роја честица који користи једначине (П1.5) и (П1.6) назива се инерцијално пондерисани приступ (IWA).



Слика П1.7. Графички приказ помераја честице

Текућа позиција (тачка претраживања у простору решења) се модификује следећом једначином:

$$s_i^{k+1} = s_i^k + v_i^{k+1} \quad (П1.7)$$

Општи дијаграм тока $PSO - IWA$ може бити описан на следећи начин:

- корак 1. Генерисање почетних услова за сваку честицу. Почетно тачка претраге (s_i^0) и брзина (v_i^0) за сваку честицу се уобичајено генерише случајно у оквиру дозвољених граница. Текућа тачка претраге се подешава на $pbest$ за сваку честицу. Најбоље оцењена вредност $pbest$ је подешена на $gbest$, и број честица са најбољом вредношћу се памти.
- корак 2. Провера тачака претраге за сваку честицу. Рачуна се вредност функције циља за сваку честицу. Ако је добијена вредност боља од текућег $pbest$ честице, $pbest$ узима вредност текуће вредности функције циља. Ако је најбоља вредност $pbest$ боља од текућег $gbest$, онда $gbest$ узима вредност најбољег $pbest$ и број честица са најбољом вредношћу се памти.
- корак 3. Модификација за сваку тачку претраге. Текућа тачка претраге за сваку честицу се мења користећи (П1.5), (П1.6) и (П1.7).

- корак 4. Проверити услов престанка алгоритма. Ако број текуће итерације достиже дефинисани максимални број итерација онда се алгоритам прекида. У супротном иди на корак 2.

Карактеристике алгоритма роја честица се могу свести на следеће:

- Алгоритам користи неколико тачака претраге и ове тачке постепено се приближавају оптимуму користећи свој *pbest* и *gbest*.
- Први члан у П1.5 одговара диверзификацији и процедури претраге простора решења. Други и трећи члан одговарају интензификацији у процедури претраге простора решења. Тако алгоритам роја честица на ефикасан начин прави баланс коришћења диверзификације и интензификације у процедури претраге.
- Наведени пример који је описан користи само дводимензионални простор претраге и решења (x, y осу). Међутим, ова метода може се лако применити на n -димензионе проблеме. На тај начин алгоритам роја честица може ефикасно решавати проблеме са континуалним променљивима у n -димензионом простору решења.

Пример решавања проблема планирања дистрибутивне мреже алгоритмом роја честица [119]

Увод

Овде је дат пример примене алгоритма роја честице [119] који решава проблем планирања развоја дистрибутивних мрежа. Дизајн дистрибутивне мреже је комплексан, нелинеарни проблем комбинаторне природе. Овај проблем се састоји у проналажењу оптималне конфигурације за мрежу, укључујући и топологију (конфигурација веза - уклопно стање) и стање проводника (капацитет сваког проводника у одређеном уклопном стању), са ограничењима везаним за техничке спецификације, као што су потрошачки захтеви (баланс оптерећења) и ограничења капацитета дистрибутивних водова [150].

Циљ планирања дистрибутивног система је оптимизација укупних трошкова електроенергетске компаније да опслужује потрошачке захтеве - оптерећења који обично расту са временом. Према томе, планирање је проблем оптимизација за одређивање оптималних вредности за [120]:

- трансформаторске локације и величине,
- рутирање фидера и величину проводника за сваку грану,
- додавање капацитета постојећих трафостаница,
- замену проводника код постојећих напојних грана.

Алгоритам роја честица је техника еволуционе оптимизације која може да реши већину проблема решаваних и са генетским алгоритмима, али са мање рачунских захтева [151].

Формулација проблема

Дистрибутивни систем може бити представљен графом у којој свака могућа веза између чворова представља променљиву. То је мултигрански граф, где више врста проводника може да се користи за повезивање чворова. Два главна фактора морају се разматрати у оптимизацији дистрибутивног система:

- минимизација губитака енергије,
- минимизација улагања у нове објекте и дистрибутивних водова,
- максимизација поузданости система.

Поврх тога, нека ограничења се морају узети у обзир:

- капацитет водова,
- напонски ниво оптерећења у чворовима,
- графикон повезивања (сваки чвор у стаблу мора бити повезан на корен графикона, директно или индиректно),
- радијалност мреже (мрежа је дрво, или потпуно повезан граф без петљи).

Проблем је формулисан са две функције циља. Прва функција циља односи се на губитке енергије. У овој функцију фигуришу трошкови одржавања и трошкови губитака као и фиксни трошкови. Друга функција циља који се користи у овом раду је везан за поузданост система и смањује укупну неиспоручену енергија [120]. У овој функцију фигурирају стопе отказа и трајања поправке сваке напојне гране. Минимизација овог циља максимизује поузданост мрежа.

Главни извор неизвесности која утиче на пројектовање мреже је ниво оптерећења у сваком чвору у будућности. Овде се неизвесност моделују са сценаријима који су генерисани према Гаусовој расподели вероватноће.

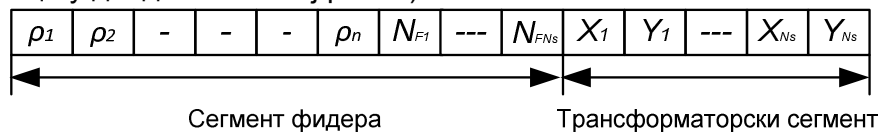
Кодирање честица

Шема кодирања честица кодира топологију мреже, односно, трансформаторске станице и путање фидера. Користе се две врсте шема кодирања у комбинаторним проблеме оптимизације:

- директно кодирање представља директно решење, и
- индиректно кодирање које кодира неке водеће информације о решењу

Сви претходни радови [83],[152],[153] користећи еволутивне алгоритме за проблем планирања користили су директно кодирање. Највећа слабост оваког приступа је у томе што се као решења могу добити нерадијалне структуре односно упетљане мреже. Како би се ово избегло користи се хибридни приступ директног и индиректног кодирања, где честице садрже два сегмента (слика П1.8):

- Сегмент фидера. Састоји се од подсегмента „пристрасности“ чвора вредности ρ [$\rho \in (-1,1)$], односно, индиректна информација која декодирањем даје мрежну топологију, и други подсегмент садржи број фидера (N_{Fi} за трансформаторске подстанице i) за сваку подстаницу, што је директна информација.
- Сегмент трафостаница. Она садржи директне информације о локација сваке трансформаторске станице (X_i и Y_i , координате трансформаторских станица у дводимензионој равни).



Слика П1.8. Предложена шема кодирања

Шема декодирања

У овом приступу који се зове трошковно-пристрасно декодирање, чворови су узастопно изабрани и додати на терминални чвор растућом путањом на основу минималне вредности производа трошкова гране и вредности „пристрасности“ чвора. Трошкови гране се узимају као удаљености (D) између његових почетних и крајњих чворова. По предложеној шеми, добија се радијална мрежа и генеришу се бочне гране у односу на главне гране. Осим тога, у реалној дистрибутивној мрежи, разумно је да се ограничи повезивања чвора са само неколико суседних чворова. За ову сврху, формирана је матрица повезивања M . За систем са n чворова, то је матрица димензије $n \times n$. Ако је дозвољено веза између чворова i и j , $M(i, j) = 1$, иначе $M(i, j) = 0$. Могућа повезаност чвора постоји са фиксним бројем својих најближих суседних чворова.

Псеудо код предложеног алгоритма је дат на слици П1.9.

Руковање ограничењима

- Енергетски биланс ограничење се аутоматски задовољава коришћењем токова снага.
- Ако је прекршено ограничење напона, решење се "кажњава" тако да његов утицај на друге честице (решења) постаје веома слаб. Казнени фактор се израчунава као производ максималног одступања напона чвора од номиналне вредности и веома високог целог броја, и додаје се у функцији циља тог специфичног решења за ову сврху.
- Радијално ограничење је увек задовољено предложеном шемом декодирања.

Кориштени параметри

Кориштени параметри за примену предложеног приступа решавања проблема планирања су дати у табели П1.1:

Табела П1.1. Параметри предложеног алгоритма

Параметри	Изабране вредности
Максимални број итерација	200
Величина популације	50
Константе учења	$c_1=c_2=2$
Максимални и минимални инерциони коефицијент	0.9 , 0.1

Закључак

Описани алгоритам роја честица решава статичке једнокритеријумске и вишекритеријумске проблеме планирања развоја дистрибутивних мрежа. Такође, овакав алгоритам може да третира и проблеме планирања са неизвесношћу оптерећења у чворовима дистрибутивне мреже, при чему се ова неизвесност моделује Гаусовом расподелом.

ПРИМЕР АЛГОРИТМА РОЈА ЧЕСТИЦА ИЗ ЛИТЕРАТУРЕ - псеудо код

Почетак: /*Иницијализација - подешавање величине популације и максималног броја итерација k_{max} */

$k = 0$;

$s^k = s_i^0$; /*Генерисање почетне тачке претраге за целу популацију по предложеној шеми кодирања*/

$v^k = v_i^0$; /*Генерисање почетне брзине претраге за целу популацију по предложеној шеми кодирања*/

Декодирање свих честица; /*по предложеној шеми декодирања*/

Оценити $f(k)$; /*Израчунати вредност функције циља сваке честице у текућој популацији k */

Понављати /*Глобална итерација, k */

$v_i^{k+1} = wv_i^k + c_1rand_1 \times (pbest_i - s_i^k) + c_2rand_2 \times (gbest_i - s_i^k)$; /* Ажурирај брзину за све честице*/

$s_i^{k+1} = s_i^k + v_i^{k+1}$; /* Ажурирај позицију за све честице*/

Декодирање свих честица; /*по предложеној шеми декодирања*/

Оценити $f(k)$; /*Израчунати вредност функције циља сваке честице у текућој популацији k */

If $f(k) > pbest(k)$ **Then** $pbest(k + 1) = f(k)$ /*Прихватити функцију циља $f(k)$ честице као $pbest(k + 1)$ за ту честицу*/

If $pbest(k) > gbest(k)$ **Then** $gbest(k + 1) = pbest(k)$ и памти k -ту честицу /*Прихватити $pbest(k)$ као $gbest(k + 1)$ и сачувај k -ту честицу као најбољу*/

$k = k + 1$;

Док Критеријум заустављања /* $k > k_{max}$ */

Излаз: Најбоље нађено решење

Слика П1.9. Псеудо код алгоритма роја честица из литературе [119]

1.2.2. ГЕНЕТСКИ АЛГОРИТАМ

Оператори

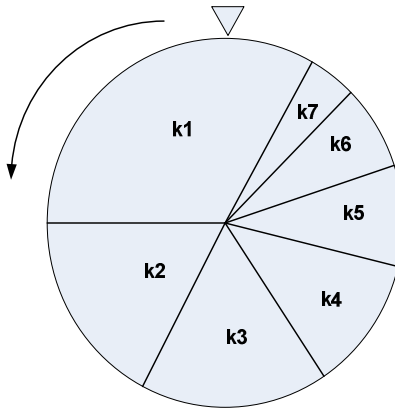
Селекција

Оператор селекције је поступак којим се по неком критеријуму одабиру јединке из популације (најчешће у сврху размножавања). Циљ селекције је да се одабиром јединки из популације и њиховим размножавањем пренесу добре особине на следећу генерацију. Ово се најчешће постиже одабиром најбољих јединки чиме се обезбеђује да њихове особине пређу у следећу генерацију. С друге стране, мора се дати шанса и лошијим јединкама да буду одабране, како не би био изгубљен квалитетан генетски материјал који је садржан у њима. У општем случају добрим јединкама се даје већа вероватноћа преживљавања и одабира него оним лошијим. Овде ће се описати само две најчешће врсте селекције:

- Пропорционална селекција.
- Турнирске селекције.

- Пропорционална селекција

Ова селекција још је позната под називом селекција "рулет точка" (*Roulette – wheel selection*). Код ове селекције се свакој јединци придружује одређена вероватноћа одабира у зависности од њене функције циља. То се најчешће представља помоћу рулет-точка, због чега је дата селекција и добила име. Наиме, цела популација је мапирана на точку рулета, на тај начин да већи делови рулет точка припадају јединкама са већом функцијом циља (слика П1.10). Тако приликом случајног одабира односно "покретања" рулет точка, већу вероватноћу одабира имају јединке са бољом функцијом циља. Пример ових вероватноћа одабира јединки код пропорционалне селекције је дат у табели П.1.2. Рулет точак се "окреће" онолико пута колико је потребно селекувати нових јединки.



Слика П1.10. Селекција "рулет точка" [154]

Табела П1.2. Таблица вероватноће одабира за све јединке [154]

Јединка	1	2	3	4	5	6	7
Доброта јединке	10	6,66	5,66	3,66	3	2,33	1,66
Вероватноћа одабира	0,3	0,2	0,17	0,11	0,09	0,07	0,05

- *k*-турнирска селекција (енгл. *Tournament selection*)

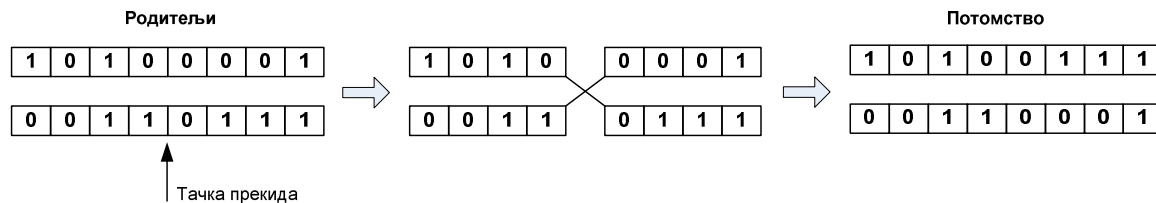
Ово је због своје једноставности једна од популарнијих врста селекције. Основна идеја ове врсте селекције је у томе да се из популације величине n јединки, случајно одабира k јединки, а потом узима она најбоља.

Укрштање (енгл. *crossover*)

Оператор укрштања разменом генетског материјала родитеља омогућава стварање нових јединки - деце. То је бинарни оператор над двома родитељима при чему настаје ново потомство - деца (једна или две јединке). При томе, деца наслеђују особине својих родитеља, што током рада алгоритма омогућава еволуцију решења. Основна идеја је у томе да уколико су родитељи добри пошто су прошли селекцију, и њихова деца вероватно требају да буду добра или боља од родитеља. Битан параметар код генерацијских алгоритама који се односи на укрштање, представља вероватноћа укрштања, која дефинише са којом вероватноћом ће се над селектованим јединкама извршити оператор укрштања. Постоји више врста укрштања и њихов краћи опис је дат овде.

- Укрштање са једном тачком прекида

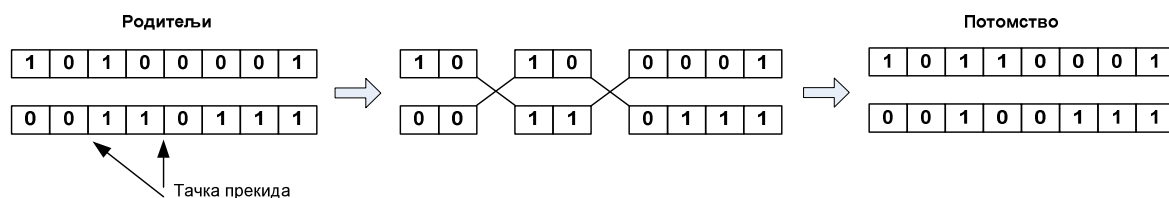
Случајно се одабере тачка прекида односно "кидања" за оба родитеља. Онда се укрштањем прекинутих делова родитеља стварају два нова решења - деца. Поступак је приказан на слици П1.11.



Слика П1.11. Укрштање са једном тачком прекида

- Укрштање са k тачака прекида

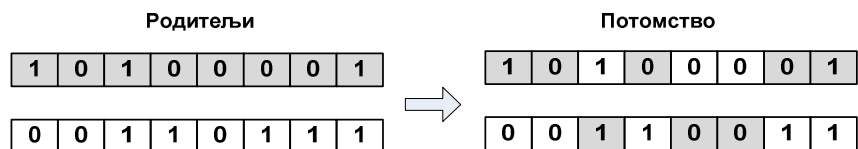
Овај је поступак сличан горе наведеном, али има општији карактер. Наиме, број k узима вредност између 1 и n (број који дефинише величину приказа јединки родитеља). На слици П1.12. је дат пример када је $k = 2$.



Слика П1.12. Укрштање са k – тачака прекида

- Униформно укрштање

Ово укрштање је укрштање са k тачака прекида који узима вредност између 1 и n , али где постоји вероватноћа да дете наследи особине родитеља $p = 0,5$. Овде се може и формирати вектор (маска), који дефинише вероватноћу преузимања генетског материјала од првог, односно другог родитеља, за сваки од гена детета. Ако при укрштању настају два детета, онда друго дете настаје на супротан начин од првог.



Слика П1.13. Униформно укрштање

Мутација

Мутација је унарни оператор који се изводи над једном јединком, а као резултат даје измењену јединку. Она као резултат има случајну промену једног или више гена једне јединке. Основни параметар мутације је вероватноћа мутације једног бита. Најједноставнија мутација је када се сваки бит мења са истом вероватноћом за сваку јединку по једном. Код бинарне репрезентације најчешће се користи проста мутација, где се за промену гена дефинише нека мала вероватноћа, а број гена који подлежу мутацији није фиксиран него се одабире случајно. Уколико је n број битова којим се кодира једна јединка, број гена који ће бити мутиран једнак је $p * n$.



Слика П1.14. Проста мутација (случајно изабрани трећи, четврти и седми ген мутирају)

Мутација је један од главних механизма да алгоритам избегне локални минимум, јер се мутацијом претражује простор решења. На пример, околност да све јединке у популацији имају на истом месту исти ген доводи до тога да ће се та вредност провлачити кроз све генерације, јер операторима селекције и укрштања та вредност неће моћи да се измени. На тај начин, уколико је реч о бинарном приказу, пола простора решења генетског алгоритма неће бит претражено. Тек применом оператора мутације могу да се избегну овакве ситуације. Тако код избора вредности вероватноће мутације имамо два екстремна случаја: ако би вероватноћа мутације била једнака јединици онда би се алгоритам претворио у случајну претрагу, у супротном овај оператор не би ни био у функцији, и тиме би се створили ризици да алгоритам не претражи велики део простора решења и "заглави" се у локалном оптимуму. Због тога је износ вероватноће мутације важан параметар генетских алгоритама. У пракси код генетских алгоритама, мутација се обично примењује са ниском вероватноћом, од 0.001 до 0.01.

Могући проблеми код рада генетског алгоритма

Проблеми који се јављају у вези функције циља могу бити следећи:

- Преурањена конвергенција - Чест проблем са генетским алгоритмима, познат као "обмана", јесте да ће гени из неколико релативно високо квалитетних (али не оптималних) јединки брзо да доминирају популацијом, чинећи да конвергирају ка локалном максимуму, или стагнирају негде у простору претраге. Када је популација конвергирала, способност генетског алгоритма да настави потрагу за бољим решењима је скоро елиминисана. Укрштање од скоро идентичних хромозома углавном производи слично потомство. Само мутација, са случајним механизмом пертурбације, преостаје као могућност да се истраже нови региони у простору претраге.
- Спора конвергенција - Ово је супротно од проблема преурањене конвергенције. После много генерација, популација је скоро конвергирала, али је ипак могуће да глобални минимум/максимум (или висококвалитетни

локални минимум/максимум) није пронађен. Просечна функција циља јединки је висока, а разлика између најбољег и просечних јединки је мала. Дакле, нема довољно варијација у вредности функције циља да се локализује минимум/максимум. Исте технике које се користе за спречавање превремене конвергенција, такође се користе и за спречавање спорог завршетка.

Као алат за оптимизацију, генетски алгоритам се суочава са проблемом ограничења оптимизационог проблема [155]. Укрштање и мутација, тј. механизам варијације генетског алгоритма, су општи оператори који не узимају у обзир допустивост региона претраге. Дакле, неизводљиво потомство применом генетског алгоритма је прилично често. Због тога је развијено више техника за руковање ограничењима када се користи генетски алгоритам. Најједноставнији алтернатива је техника одбијања у којој се неизводљиви хромозоми одбацују у целој популацији. Другачија стратегија је процедура поправке, који користи конвертор за трансформацију неизводљивих хромозома у изводљиве. Следећа могућа техника је стварање специфичних генетских оператора да сачувају изводљивост хромозома. Претходни поступци не стварају неизводљива решења, али то није увек предност. У ствари, за велике, изузетно ограничене проблеме оптимизације, ово је велики недостатак. Посебно за проблеме електроенергетског система, где су оптимална решења обично на границама изводљивих региона, горе наведене технике за руковање ограничења често доводе до лоших решења. Један могући начин за превазилажење овог недостатка је примена процедуре поправке само дела (нпр. 10%) од неизводљиве популације. Сугерише се да руковање ограничењима, код таквих типова проблема оптимизације, треба да се врши тако да се омогући претрага кроз неизводљиве регионе. Казнене функције омогућавају истраживање неизводљивих подпростора [156]. Неизводљива тачка (решење) близу оптималном решењу може да садржи много више информација о оптимуму, него допустива тачка далеко од оптимума. Са друге стране, дизајн казних функција је тежак, јер не постоји а priori информација о удаљености од оптималне тачке, и он зависи од проблема који се решава.

Пример решавања проблема планирања дистрибутивне мреже генетским алгоритмом [93]

Увод

Овде је представљен специфичан генетски алгоритам који решава вишекритеријумски проблем, а који је дат у [93].

Оптимизација проблема планирања развоја дистрибутивних мрежа мора да размотри четири главна циља.

- минимизација губитака енергије,
- минимизација улагања у нову опрему и дистрибутивне водове,
- минимизација просечног броја кварова,
- минимизација просечног времена прекида услед кварова.

Осим тога, треба уважити следећа ограничења:

- капацитете водова (грана),
- ниво напона у потрошачким чворовима,
- радијалност мреже,
- индекс квалитета и поузданости.

Овај проблем планирања има две главне карактеристике:

- то је комбинаторни проблем, са ограничењима, и
- функције циља су нелинеарне. Линеарно оптимизациони алгоритми, који су корисни за решавање проблема планирања неких мрежа постају бескорисни у овом случају. Стога, алгоритми хеуристичке претраге постају главна алтернатива за конституисање метода за оптимизацију оваквог проблема планирања.

Треба истаћи да вишекритеријумски генетски алгоритам развија целокупну популацију према скупу Парето решења, уместо да се развија једна „текућа тачка“. Тако ће у једном кораку бити генерисан цео скуп Парето решење или бар његов велики део. То значи да у случају проблема вишекритеријумске оптимизације, рачунарски напор при решавању наведеног проблема је далеко мањи код оваквог генетског алгоритма, него код детерминистичког, јер би детерминистички алгоритам морао да се покреће сваки пут за проналажење једног Парето решења [157],[158].

Класично структурирани генетски алгоритми су се показали лошим за решавање тешких комбинаторних проблема [159]. Такође у [89] је запажена неадекватност конвенционалних генетских алгоритама за овакве проблеме, те користе генетске алгоритме са конвенционалним операторима заједно са оператором локалне претраге, који се заснива на техникама математичког програмирања.

У овом примеру предложени алгоритам користи проблемску структуру планарног графа за генерисање ефикасних генетских оператора (укрштање и мутација) који омогућавају разноврсност генетског материјала у популацији и избегавају неизводљива решења.

Формулација проблема

Овде се проблем представља са две функције циља. Прва функција циља обухвата новчане трошкове и то: трошкове одржавања, фиксне трошкове, као и променљиве трошкове (пропорционалне губицима). Друга функција циља моделује трошкове отказа, односно трошкове услед неиспоручене електричне енергије. Предложена процедура минимизира обе функције циља и даје скуп њихових Парето решења.

Специфични генетски алгоритам

Специфичан пример вишекритеријумског генетског алгоритма, који се користи у [93] направљен је на бази оператора селекције и елитизма *NSGA – II* алгоритма [160]. Компоненте овог алгоритма су описани у даљем тексту.

Избор и елитизам

Оператор селекције се заснива на шеми недоминантног сортирања [24],[161], користећи своју оптимизовану верзију. Овај алгоритам чува резултате сваког доминантног поређење које је извршено у првој петљи, у којима се одређује недоминантни подскуп. Ови подаци се користе у наредним петљама за проналажење следећих скупова.

Након операције брзог недоминантног сортирања, популација постаје одвојена подскуповима F_1, F_2, \dots, F_{k-1} . Сваки подскуп садржи јединке које су доминирани само од оних који припадају претходним сетовима, тако први подскуп F_1 садржи јединке

које су недоминиране. Ово успоставља рангирање по којем су јединке из F_1 у најбољем рангираном положају, а јединке последњег F_{k-1} су у најгорем положају.

Поступак процеса додељивање окружења процењује густину суседних решења од неког појединачног решење у популацији [160]. Овај процес је повезан са стохастичком турнирском селекцијом која се обично користи у генетским алгоритмима.

Оператор стохастичке турнирске селекције генерише избор N појединаца са следећом процедуром која се понавља N пута.

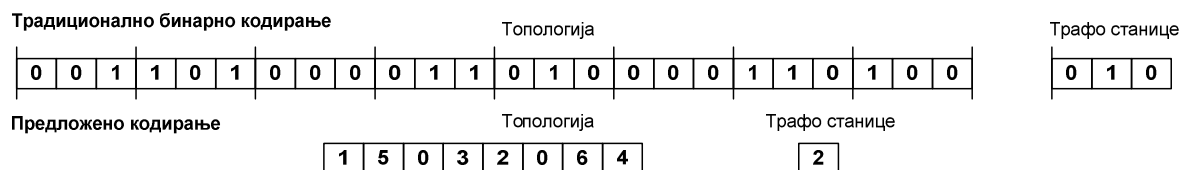
1. две јединке p_1 и p_2 су насумично одабране,
2. јединка у најбољој позицији у доминирајућем рангу (припада скупу F_i са мањим индексом i) је изабрана
3. ако p_1 и p_2 припадају истом скупу F_i , јединка са већом густином окружења (са више суседних решења) је изабрана.

NSGA – II је елитистички алгоритам по својој структури, па нису потребне додатне операције да би се увео елитизам.

Са овим операторима селекције и елитизма *GA* постаје "прави вишекритеријумски алат за оптимизацију". На тај начин читав Парето скуп може бити пронађен са таквим алгоритмом [158],[159].

Контролисано похлепно кодирање

За променљиве и мрежну структура проблема која се анализира овде, предлаже се кодирања у којој сваки део генетског слога (обично се назива ген) представља један од могућих веза између чворова. Гени могу бити представљени помоћу бинарних променљиви, као код традиционалних генетских алгоритама. Међутим, ова врста представљања генерише велике хромозоме који нису погодни за даљи рад. Из тих разлога примењено је кодирање где се користе цели бројеви за представљање различитих врста мрежних веза између чворова. На слици П1.15. је приказано поређење између бинарног и целобројног кодирања на примеру мреже. Други слог је коришћен за кодирање напојних трансформаторских станица. Кодирање је овде веома битно с обзиром на чињеницу да број проблемских променљивих (могуће везе између чворова) расте комбинаторно како број чворова у мрежи расте. Обично кодирање дозвољава све могуће везе између чворова, тако да ограничава генетски алгоритам да оперише са само врло малим мрежама, односно онемогућава рад са већим мрежама.



Слика П1.15. Целобројно кодирање у поређењу са традиционалним бинарним кодирањем

Из тих разлога се дефинише скуп дозвољених веза које се сматрају оптимизационим променљивима, и само такве везе се кодирају и појављују у простору претраге алгоритма. Ово је врста похлепног поступка у смислу да се одређени број комбинација једноставно не разматра и не вреднује. У случају

електричних мрежа, ова процедура одсликава реалност, јер у реалним случајевима директна веза сувише удаљених чворова није реална. Тако се за неки чвор разматра само скуп њему суседних чворова. Овај поступак се користи у већини референци које се баве дизајном електричних дистрибутивних мрежа, где се скуп могућих веза за сваки чвор дефинисао ручно од стране корисника. На овај начин се дефинише мањи простор претраживања од оригиналног, а самим тим олакшава претрага.

Овде се предлаже аутоматски поступак, назван контролисаним похлепним кодирањем, за успостављање скупа могућих веза. Два параметра контроле, који дефинишу обим у којем се смањује простор претраживања су:

- mxt - максималан број могућности за повезивање који се додељују сваком чвору;
- mnn - минималан број могућности повезивања да буде додељен сваком чвору.

У првом кораку, растојање између свих чворова се мери, и за сваки чвор се дефинише средња удаљеност између тог чвора и других чворова. Чворови су класификовани по својим средњим дистанцама у растућем редоследу. Чвор са минималном средњом удаљености може да се повеже са mxt најближих чворова, и чвор са максималном средњом удаљеношћу, може да се повеже само на mnn најближих суседа. Број могућности повезивања (између mnn и mxt) за остале чворове је дефинисан линеарном интерполацијом следећи операцију заокруживања. Стварни скуп могућих повезивања која су кодирана, почиње да се дефинише од чвора са најмањим средњим растојањем (који може да се повеже са mxt чворова), до чвора са максималним средњим растојањем (који може да се повеже са mnn чворова), укључујући, у сваком кораку, најближе алтернативе за повезивање, до броја могућих повезивања које су додељене сваком чвору.

На овај начин за дистрибутивну мрежу са 21 чвором који се разматра у [93], број променљивих је смањен са 208 на 62, а за дистрибутивну мрежу са 100 чворова, број променљивих је смањен са 4950 на 397.

Неки чворови који не могу бити повезани, због географских карактеристика, као што су планине, реке, и друге препреке, су искључени из кодирање.

Иницијално решење

У класичним генетским алгоритмима, почетна популација се генерише насумично. Међутим, за мрежне проблеме са графовском структуром ограничења, ова процедура није ефикасна, јер већина случајних почетних решења не представљају изводљива (допустива) решења.

Овде је предложен алгоритам одсецања (*Prune algorithm*) или инверзни Крускалов алгоритам (*Kruskal's algorithm*) [162]. То је "похлепни" алгоритам који проналази минимално разапињуће стабло за повезани тежински граф. За генерисање јединки иницијалне популације, генерише се случајна почетна мрежа (то је упетљана мрежа - није радијална), и њене гране су смештене у листи, у опадајућем редоследу трошкова, на основу растојања између чворова. Гране се бришу из мреже док се не постигне радијална конфигурација мреже. Затим се тип сваког проводника додељује насумично. Поступак почиње из различитих случајних почетних мрежа и доводи до мрежа које су довољно различите за пружање

разноврсности топологије у почетној популацији. Међутим, сва таква решења су заснована на "похлепним" операцијама. Дакле, она немају разноликост која је неопходне да генетски алгоритам изврши глобалну претрагу.

Овај недостатак је решен коришћењем случајног генератора мрежа. Наиме, овај алгоритам генерише случајне мреже које представљају изводљива и радијална решења. Већина мрежа произведених на тај начин имају низак квалитет, по свим критеријумима циљне функције, али они дају потребну разноликост локалних конфигурација које ће се користити, у генетским операцијама, за проналажење глобално оптимизоване дистрибутивне мреже. Након експериментисања, нађено је да се равнотежа међу ефектима ове две процедуре може постићи користећи следећи састав јединки у почетној популацији:

- 40% јединки насталих коришћењем алгоритма одсецања,
- 60% јединки генерисаних случајним генератором мреже.

Оператори мутације и укрштања

Класични оператори генетског алгоритма који насумично мењају јединке нису ефикасни код проблема планирања јер, већину времена, они трансформишу изводљива решења у она неизводљива (ово је услед ограничења радијалности). На тај начин, у неколико генерација, популација почиње да садржи само неизводљива решења. Из тих разлога предложени су специфични оператори прилагођени проблему планирања, како би се избегли наведени проблеми:

- Четири оператора укрштања који одређују заједничке делове од два родитеља (мреже) и копирају ово у двоје деце (нове мреже), користећи максималне или минималне индексе (у зависности од оператора). Начин на који ће деца бити комплетирана биће другачији у сваком оператору.
- Девет оператора мутације који су подељени у две класе: четири од њих се састоји од једноставних промена у врсти грана или трансформаторских станица, који покушавају да унапреде решење. Они су названи једноставни оператори. Пет преосталих оператора се заснивају на неким хеуристикама за промену топологије мреже. Они су названи паметни оператори.
- Два оператора детерминистичке мутације су коришћени у циљу побољшања објективне функције најбоље цене мреже и најбоље поузданости мреже, са неким детерминистичким операцијама.

Ако алгоритам утврди да се мутација или укрштање морају јављати (у зависности од вероватноће укрштања и мутације), оператор се случајно одабере са дефинисаном вероватноћом за сваки оператор који се користи. То значи да се мутације које мењају топологију мреже јављају 5/9 од позиваних оператора мутације, а мутације које мењају типове грана мреже се јављају у 4/9 од позиваних оператора мутације.

Избор решења из Парето скупа

Након добијања Парето скупа решења, дизајнер кроз процес доношења одлуке мора да изврши избор најбољег решења [163]. С обзиром да процес доношења одлуке спада у проблематику метода одлучивања, исти овде неће бити даље описиван.

Закључак

У [93] је представљен вишекритеријумски приступ решавања проблема планирања. При томе је коришћен генетски алгоритам са дизајнираним специфичним операторима мутација и укрштања, као и ефикасна шема кодирања решења. Овакав приступ као решење даје Парето скуп решења која се затим неком од техника доношења одлуке одабирају, како би се пронашло најбоље решење.

ПРИМЕР ГЕНЕТСКОГ АЛГОРИТМА ИЗ ЛИТЕРАТУРЕ - псеудо код

Почетак: /*Иницијализација*/

$k = 1$

$P(k) = P(1)$ /*Генерисати иницијалну популацију алгоритмом одсецања и генератором мреже*/

$O(k)$ = Парето екстракција ($P(k)$) /*преузимање недоминантних решења из популације $P(k)$ */

$P(0) = \emptyset$ /*празан скуп*/

Понављати /*Глобална итерација, k */

$R(k) = P(k) \cup P(k - 1)$ /*генерисање помоћне популације*/

Оценити $R(k)$ /*Израчунати вредност добротe сваке јединке (хромозома) у текућој помоћној популацији $R(k)$ користећи брзо недомнатно сортирање и додељивање окружења*/

$P'(k)$ = Селекција ($R(k)$) /*Изабрати скуп јединки (родитеља) стохастичком турнирском селекцијом*/

$P'(k)$ = Репродукција $P'(k)$ /*Применити операторе укрштања за добијање потомака*/

$P'(k)$ = Мутација $P'(k)$ /*Применити операторе мутације за добијање потомака*/

$P(k + 1) = P'(k)$ /*Добијене јединке представљају нову популацију*/

$O(k + 1)$ = Парето екстракција ($O(k) \cup P(k + 1)$) /*преузимање недоминантних решења из уније парето скупа $O(k)$ и нове популације $P(k + 1)$ */

$k = k + 1$

Док Критеријум заустављања /* $k > k_{max}$ */

Издаз: Најбоље нађено решење (јединка) или најбољи скуп (популација) решења

Слика П1.16. Псеудо код генетског алгоритма из литературе [93]

1.2.3. МРАВЉИ АЛГОРИТАМ

Мрављи систем

Мрављи систем (AS) је био први алгоритам мравље колоне представљен као хеуристички метод за решавање комбинаторних проблема почетком 1990-тих година. Мрављи систем се може описати на следећи начин: након фазе иницијализације, m број мрва се поставља на њихове почетне тачке на конструкционом графу, а сви феромони на путањама се постављају да буду мале вредности (у границама од 0 до 1). У сваком кораку конструкције решења, сваки мрав постепено конструише решење додајући једну компоненту решења на парцијална решења која су до тада већ изграђена. Предпоставимо да је k -ти мрав у тачки c_i у конструкционом кораку m . Наведени k -ти мрав обавља случајан ход из тачке c_i до следеће тачке да би постепено конструисао изводљиво решење, на тај начин да је следећа тачка изабрана стохастички према вероватноћи транзиције. Вероватноћа са којом k -ти мрав у тачки i бира да се помери у тачку j , може се описати случајно-пропорционалним правилом транзиције (прелаза), који је дат као:

$$p_k(i, j) = \begin{cases} \frac{[\tau(i, j)]^\alpha \cdot [\eta(i, j)]^\beta}{\sum_{u=1}^t [\tau(i, u)]^\alpha \cdot [\eta(i, u)]^\beta}, & j, u \in N_{k, i} \\ 0, & j, u \notin N_{k, i} \end{cases} \quad (П1.8)$$

где $\tau(i, j)$ представља траг феромона придружен $l_{i, j}$, који представља везу између тачака i и j , $\eta(i, j)$ представља хеуристичку вредност која одсликава колико је пожељно додавање везе $l_{i, j}$ решењу које је још у фази израде, и ова хеуристичка вредност се одређује према оптимизационом проблему. Оно се обично поставља да буде обрнуто пропорционална са трошковима изградње везе $J_{i, j}$ који се додељују путањи $l_{i, j}$ тј. $\frac{1}{J_{i, j}}$, $N_{k, i}$ је изводљивост k -тог мрва у тачки c_i уз поштовање проблемских ограничења Ω , $N_{k, i} \subseteq N_i$ а α и β су два параметра која одређују релативни значај феромона наспрам хеуристичке вредности ($\alpha, \beta > 0$).

Једном када су сви мрави дошли до својих решења, феромонски траг се ажурира на свим путањама према правилу глобалног феромонског ажурирања, који износи:

$$\tau(i, j) \leftarrow (1 - \alpha) \cdot \tau(i, j) + \sum_{k=1}^m \Delta\tau_k(i, j) \quad (П1.9)$$

где је α параметар распада феромона у границама $0 < \alpha \leq 1$, и:

$$\Delta\tau_k(i, j) = \begin{cases} \frac{1}{J_{i, j}}, & (i, j) \in \varphi \\ 0, & (i, j) \notin \varphi \end{cases} \quad (П1.10)$$

где је φ скуп потеза (путања) које је учинио k -ти мрав.

Код хеуристичке вредности $\eta(i, j)$ се фаворизују путање које су ефикасне у смислу трошкова и које имају велики садржај феромона. Први члан у формули П1.9 представља испаравање феромона, у коме се феромони смањују на основу фактора распада. Феромони на нефаворизованим путањама испаравају експоненцијално брже. Други члан у формули П1.9 представља појачавање феромонског трага. Тако се постиже да путеви које су чешће посећивали мрави, постају пожељнији за неке будуће мраве. Мрави полажу феромоне чија је количина повезана са квалитетом решења, тако да ће решења која су ефективнија имати већу количину феромона. Правило ажурирања феромона је створено да

симулира промене у садржају феромона током два процеса: депоновања нових феромона које остављају мрави, и испаравања феромона.

Систем мравље колоне

Систем мравље колоне (ACS) је развијен после алгоритма мрављег система (AS). Систем мравље колоне може надградити мрављи систем користећи три модификације:

- правило транзиције омогућава директан начин да се успостави равнотежа између истраживања нових путева и експлоатације пронађеног решења и акумулираног знања о проблему,
- правило глобалног ажурирања феромона се примењује само на путање које припадају најбољем путу,
- док мрави праве решење, локално ажурирање се примењује за сваки конструкциони корак.

Систем мравље колоне се у основи може овако описати: m мрав се иницијално поставља на насумично изабрана темена конструкционог графа. Свакој конекцији на графу додељује феромонски траг који је мала позитивна вредност. Сваки мрав насумично на графу конструише своје решење кроз понављање додавања једне компоненте решења својим парцијалним решењима пратећи правило транзиције, које се разликује од правила транзиције коришћеног код мрављег система. Предпоставимо да је k -ти мрав у темену c_i и при конструкцији решења се стигло до корака m . K -ти мрав ће отићи у следеће теме и додаће свом решењу једну компоненту. Његово следеће теме, c_j , се може описати:

$$c_j = \begin{cases} \arg \max_{u \in N_{k,i}} \{[\tau(i,j)]^\alpha \cdot [\eta(i,j)]^\beta\}, & q \leq q_0 \\ c^*, & q \not\leq q_0 \end{cases} \quad (\text{П1.11})$$

где је q насумично додељен број $[0,1]$, а q_0 је параметар ($0 < \alpha < 1$), који дефинише релативни значај локално претраживања у односу на глобално претраживање, и c^* је случајна променљива која је добијена према вероватноћи дистрибуције датој у П1.8. Правило прелаза израчунато из П1.8. и П1.11 се назива пропорционално псеудо-случајно правило прелаза. Ово правило прелаза фаворизује кретање ка теменима која су повезана са путањама са минималним трошковима и гушћим феромонским траговима. Применом овог правила за сваки корак конструкције, мрав на темену c_i , ће изабрати теме c_j и у том тренутку се униформном расподелом генерише случајан број $q_0 \in [0,1]$. Ако је $q \leq q_0$ онда ће следеће теме бити описано изразом П1.11, што доприноси најбољем кретању, у супротном то ће бити компонента c^* добијена из П1.8. Док мрави праве своја решења, за сваки корак, трагови феромона на путањама које су посетили мрави, биће промењени применом правила локалног ажурирања као што је:

$$\tau(i,j) \leftarrow (1 - \rho) \cdot \tau(i,j) + \rho \cdot \Delta\tau(i,j) \quad (\text{П1.12})$$

где је ρ параметар ($0 < \rho < 1$), а $\Delta\tau(i,j)$ је количина феромона коју k -ти мрав ставља на путању $l_{i,j}$. Феромонско ажурирање $\Delta\tau(i,j)$ може бити постављен на вредност иницијалне феромонске путање за једноставне алгоритме система мравље колоне. Кад сви мрави комплетирају своје путање феромонски трагови ће бити ажурирани на основу правила глобалног ажурирања феромона као што је:

$$\tau(i,j) \leftarrow (1 - \alpha) \cdot \tau(i,j) + \alpha \cdot \Delta\tau_k(i,j) \quad (\text{П1.13})$$

где је α параметар распада феромона ($0 < \alpha \leq 1$), и:

$$\Delta\tau_k(i,j) = \begin{cases} (J_{gb})^{-1}, & l_{i,j} \in s^* \\ 0, & l_{i,j} \notin s^* \end{cases} \quad (\text{П1.14})$$

где је J_{gb} цена најбољег мрављег решења s^* од почетка процеса.

Пример решавања проблема планирања дистрибутивних мрежа мрављим алгоритмом [100]

Алгоритам мрављег система се успешно промењује на различите проблеме оптимизације као што је проблем трговачког путника и квадратни проблем. Мрави су чланови породице социјалних инсеката који живе у организованим колонијама. Неки од њих су у стању да пронађе пут по веома сложеним путањама. Иако су њихове индивидуалне способности учења веома ограничени, сложеност организације колоније омогућава веома ефикасну комуникацију, засновану на хемијским медијима. Током својих експедиција у потрази за изворима хране, они ослобађају хемијске секрет или феромон, како би обележили путеве који су коришћени, водећи на овај начин, нове истраживаче (мраве) ка извору хране. Најкраћи путеви ће тежити да имају већи интензитет феромонског трага и због тога ће бити пожељни од стране других истраживача (мрава).

Алгоритам система мравље колоне је побољшани алгоритам мрављег система - више робустан, бржи и са већом вероватноћом постизања глобалног оптимума. Његов учинак за решавање класичних проблема оптимизације је повољнији у поређењу са генетским алгоритмима и техникама симулираног каљења [164].

У овом примеру датом у [100], алгоритам система мравље колоне је прилагођен решавању проблема оптималног планирања дистрибутивних мрежа за дато максимално оптерећење, користећи модул токова снага. Пример је посвећен решавању статичког проблема планирања дистрибутивних мрежа, где се постојећа мрежа проширује (унапређује) за дати период планирања. Такође, дугорочна динамика оптерећења услед промена у коришћењу земљишта, до утицаја промене становништва, као и промене у расположивим технологијама, између осталих фактора, може да буде разматрано предложеним алгоритмом.

Формулација проблема

Проблем планирања може бити формулисан као проблем оптимизације на следећи начин: Минимизација функције циља која представља фиксне трошкове (унапређење грана и трафо станица) и варијабилне трошкове везане за функционисање система (губици), уз задовољење ограничења у која спадају: баланс оптерећења, ограничење капацитета, напона и радијалности.

Предложена методологија

Као што је већ поменуто, алгоритам система мравље колоне је побољшани алгоритам мрављег система. Мрави тражећи храну остављају феромон на путањама који су прошли и на тај начин врше меморисање пређеног пута и комуникацију са осталим мравима. У почетку група мрава истражује путање (простор претраге) насумично без одређеног правца. Након што проналазе храну враћају се у полазну тачку (мравињак). Како сви мрави путују истом брзином за најкраће путање је већа вероватноћа да ће бити посећене од стране више мрава, па ће те путање имати веће количине феромона. Након кратког временског

периода разлике у количини феромона по путањама ће утицати на одлуке других мрави да бирају баш такве путање.

Математичка формулација алгоритма

Алгоритам система мравље колоне користи две функције како би вршио претрагу за налажење оптималног решења:

- функција која је пропорционална количини феромона по путањама,
- хеуристичка функција представља помоћну функцију која је једнака инверзној вредности дужине поједине путање и на тај начин помаже проналажењу квалитетнијих путања.

Предложени алгоритам функционише на следећи начин: у почетку, случајни ниво феромона је депонован у свакој грани иницијалне мреже. Затим, мрави отпочињу кретање (истраживање) кроз различите гране мреже, вођени хеуристичком функцијом, као и функцијом која је пропорционална количини феромона по путањама, у складу са пробаблистичким правилом прелаза, док се сви чворови оптерећења не напоје и тако комплетира експедиција мравље колоније.

Пробаблистичко правило прелаза дефинише вероватноћу избора путање мрави из колоније да из једног чвора пређе на други изабрани чвор.

Након унапред дефинисаног броја независних експедиција или мрављих колонија, ниво феромона се ажурира.

Локално правило ревизије феромона мења количине феромона по путањама. Део феромона испарава, и претпоставља се да свака мравља колонија доприноси повећању нивоа феромона пропорционално укупним трошковима резултатне мреже.

Предложени алгоритам користи и глобално правило ревизије феромона које симулира ефекат додавања феромона од стране мравље колоније на путање, као и испаравање феромона. Феромони сваке путање се ажурирају.

Феромони играју улогу дугорочне меморије тако што се дистрибуирају у различитим путањама, омогућавајући на тај начин комуникацију између различитих колонија мрави.

Унапређење алгоритма система мравље колоне у односу на мрављи алгоритам

Алгоритам система мравље колоне обухвата три главне разлике у односу на алгоритам мрављег система:

- Ново пропорционално псеудо-случајно правило прелаза, које пондерише приоритете за истраживање нових путања, коришћењем акумулирања претходног познавања проблема, а све у циљу унапређења избора "најбољег пута". У алгоритму мравље колоније, мрав позициониран у текућем чвору, бира грану да би прешао у следећи чвор на следећи начин: Дефинише се q_0 реалан број ($0 \leq q_0 \leq 1$), који одсликава значај експлоатације решења у односу на истраживање решења. Када је мрав позициониран у чвору, генерише се случајни број q за који важи $0 \leq q \leq 1$. Ако је $q \leq q_0$, онда је изабрана најбоља грана. То значи да је експлоатација одлучујући фактор, док у супротном случају, избор трасе се врши према пробабилистичком правилу прелаза.

- Глобално правило ревизије нивоа феромона се примењује само на оне гране које припадају најбољим мрежама које су до сада пронађене. Феромони се депонује само у оним гранима које припадају најбољој мрежи. Ова промена има за циљ да претрагу учини више директном, односно да усмерава истраживање ка најбољим решењима која су до сада пронађена. Ова модификација је слична елитним стратегијама код генетских алгоритама.
- Локално правило ревизије нивоа феромона којим се ажурирају нивои феромона током процеса претраге.

Примењени алгоритам на проблем планирања дистрибутивне мреже

Примењени алгоритам система мравље колоне за планирање дистрибутивне мреже врши максимизацију функције циља и хеуристичке функције. Функција циља се дефинише као инверзна вредност укупних трошкова (трошкова одржавања и трошкова губитака). Хеуристичка функција има главну улогу у првим фазама претраге због чињенице да иста омогућава стварање добрих решења (мрежа) која имају добре напонске прилике. Међутим, како алгоритам еволуира, ниво феромона који се акумулира у гранима расте, па утицај хеуристичке функције на претрагу опада. Хеуристичка функција се дефинише као инверзна вредност дужине, дајући предност избору најкраћих путања. У конкретном примеру у хеуристичку функцију су укључене следеће променљиве: дужина, прираштај трошкова мреже и величина оптерећења на крају пута. Хеуристичка функција је дефинисана са две величине. Прва величина је производ дужине и оптерећења која се исказује у kVA_m , а друга величина је прираштај трошкова додавања нових грана у мрежи и исказује се као $\$/kVA$.

Израчунавања губитака у мрежи врши се прорачуном токова снага за максимално оптерећење мреже. Такође се уважава "временска" вредност новца за сваку од година за разматрани период планирања.

Предложени алгоритам је примењен на дистрибутивну мрежу са 21 чвором без дистрибутивних генератора. То је иста дистрибутивна мрежа као у примеру III, поглавље 4.3., која је обрађивана у овој дисертацији. Статистичка студија је изведена на основу 1000 пропуштања алгоритма, у циљу идентификовања најбољег опсега вредности за параметре хеуристичке функције и параметара који контролишу функционисање алгоритма. Овом студијом је показано да је алгоритам веома робустан, и да варијације параметара углавном утичу на карактеристике конвергенције. Као и други метахеуристички приступи, предложени алгоритам се показао као веома ефикасан на почетку претраге, постижући велики пад трошкова мреже у свега неколико итерација. Међутим, детаљније претраживање захтева већи број итерације и велико рачунско време.

Предложени алгоритам је дат у виду псеудо кода на слици П1.17.

МРАВЉИ АЛГОРИТАМ ИЗ ЛИТЕРАТУРЕ - псеудо код

Почетак: /*Иницијализација*/

$k = 0; m = 0;$

Случајно постављање феромона по свим гранама мреже

Понављати /*Глобална итерација, m */

Понављати /*Локална итерација, k */

/* Мравља колонија k шаље мраве у истраживање док се експедиција истих не комплетира (сви чворови су повезани), генерушући мрежу кандидата */

Кретање мрва из мравље колоније k је вођено псеудо случајним правилом прелаза на следећи чвор

Локално правило ревизије феромона на путањама

$k = k + 1;$

Док Критеријум заустављања /* процес се понавља за унапред дефинисан број мрављих колонија стварајући једнак број кандидата мреже за скуп експедиција */

Глобално правило ревизије феромона

$m = m + 1;$

Док Критеријум заустављања /* процес се понавља док није достигнута конвергенције или унапред дефинисан максимални број скупова експедиција */

Излаз: Најбоље нађено решење

Слика П1.17. Псеудо код мрављег алгоритма из литературе [100]

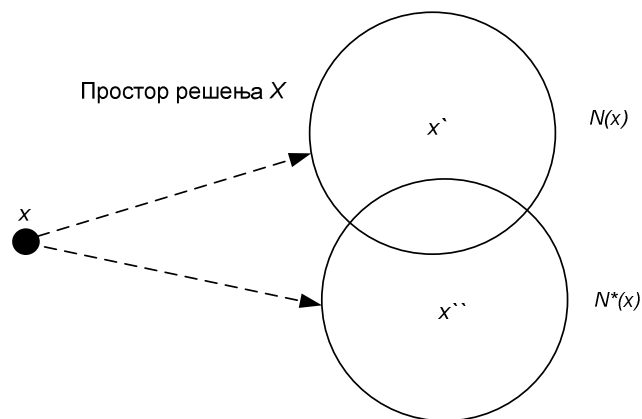
1.2.4. АЛГОРИТАМ ТАБУ ПРЕТРАГЕ

Формулација проблема

Алгоритам табу претраге (*TS*) може се формулисати на следећи начин:

$$\text{Min } f(x), \forall x \in X \quad (\text{П1.15})$$

где је x конфигурација (или променљива одлуке), $f(*)$ функција циља и X је простор претраге. Различити комбинаторни оптимizacionи проблеми могу бити представљени као минимизација функције циља и скуп алгебарских ограничења. Међутим, постоје случајеви да ограничења не могу да се лако представе као алгебарска ограничења. То је на пример случај код ограничења радијалности у дистрибутивним мрежама и проблема планирања. Овај тип ограничења који може бити сметња код неких алгоритама, лако се обрађује код алгоритма табу претраге, као и обично табу претрага не оперише са алгебарским ограничењима, конфигурације су уместо тога кодиране. Алгоритам табу претраге решава горњи проблем, прво примењујући локалну хеуристичну претрагу с обзиром на решење x , суседство од x које се дефинише као скуп свих решења $x' \in N(x)$ може бити добијено од истог механизма за прелаз примењеног на x (слика П1.18.). Услови потребни за x' као сусед од x , дефинишу структуру суседства x . Локална претрага тражи прелаз који води до конфигурације x' , представљајући највеће умањење функције циља. Понављање ове процедуре може да доведе до локалног оптималног решења.



Слика П1.18. Илустрација транзиције (прелаза) код табу претраге [124]

Алгоритам табу претраге се разликује од горе наведеног једноставног алгоритма локалне претраге по две суштинске ствари:

1. Правило прелаза доводи до решења за које је функција циља већа од текућег, а како се разматра проблем минимизације, добија се деградација решења.
2. Суседство од x , то је $N(x)$, није статично, односно може се мењати по величини као и по структури. Модификовано суседство $N^*(x)$ се дефинише на различите начине, нпр:
 - а) Користећи табу листу, која садржи атрибуте од решења која се забрањују. У том случају $N^*(x) \subset N(x)$, као што је горе наведено, ово је корисно како би се избегло кружење и обезбедио излаз из локалног оптимума.

- б) Користећи стратегију да се редукује величина суседства како би се убрзало локално претраживање. Као и у претходном случају, суседство је смањено тако да је $N^*(x) \subset N(x)$.
- в) Користећи тзв. елитне конфигурације за обављање повезивања путање (*path relinking*). У том случају није нужно да је $N^*(x) \subset N(x)$.
- г) Редифинисање $N(x)$ током процеса оптимизације. То се нормално ради да би се искористиле специфичне особине проблема.

Суседна структура

Околина, тј. суседна структура је скуп решења која се добијају применом локалне трансформације на текуће решење. Локалне трансформације називају се још и потезима. Тако свако решење x има своје суседство које чини скуп свих решења до којих се стиже једним потезом. Ова решења могу бити допустива или недопустива. Од допустивих решења од интереса су она која су атрактивна, односно која имају добру вредност функције циља. С друге стране, постоје и недопустива решења преко којих привремени прелаз, у одређеним околностима, може бити најкраћи пут према атрактивном допустивом решењу. Оваква стратегија претраге се успешно користи код проблема планирања преносне електроенергетске мреже, односно код проблема са доста локалних оптималних решења [161],[162]. Прелаз преко недопустивих решења је олакшано укључивањем пенала у функцији циља који су представљени трошком неизводљивости. Тако, ако је смањење стварних трошкова веће од додатног трошка услед неизводљивости, онда се дозвољава прелаз преко недопустивих решења.

Табела П1.3. Примери табуа [128]

Врста проблема	Табу	Значење
Бинарни проблеми	Индекс променљиве - (i)	Забрањује се промена вредности (i) са 1 на 0 и обратно
Графови	Индекс лука - (i)	Забрањује се додавање (одузимање) лука (i) графу

Критично питање у табу претраживању је потреба процене конфигурације у суседној структури. Број конфигурација у суседству може бити веома велики и квалитет ових конфигурација може да много варира. У свакој итерацији основног алгоритма табу претраге, нормална је потреба да се идентификују најбоље конфигурације у суседству тренутног решења, узимајући у обзир да ново решење нема табу атрибуте или, ако се деси да има, да критеријум аспирације буде задовољен (тј. нова конфигурација је довољно добра да се оправда релаксација табу ограничења). Један од главних стратегија табу претраживање састоји се у кретању ка најбољој конфигурацији у суседству текуће конфигурације. Обично, величине суседства су много веће од оних које могу бити оцењене од стране алгоритма, и стога се заправо истражују само најатрактивнији делови суседства. Од кључне важности је за табу претрагу постојање ефикасног средства за проналажење ових најатрактивнијих региона.

Функције и стратегије у табу претрази

Стратегија табу претраживања је алгоритам који иначе представља део општијег поступка табу претраживања. Основне функције табу претраживање су:

интензивирање, диверзификација, стратешка осцилација, елитне конфигурације и повезивање путање (*path relinking*). У даљем тексту се описује како се ове функције могу комбиновати да се изгради низ табу стратегија претраживања.

Табу претрага заснована на блиској прошлости

Меморија заснована на блиској прошлости (*Recency based memory*) је важна карактеристика табу претраживања. То је врста краткотрајне меморија која прати решење атрибута које су се промениле током најновијег кретања од стране алгоритма. Информације садржане у овој меморији омогућавају означавање одабраних атрибута недавно посећених решења као табу - активни. Ова особина избегава поновну посету решењима која су претходно већ била посећена. Ово је најосновнија врста алгоритма табу претраживања која се заснива на листи забрањених атрибута (табуа) и на критеријуму аспирације.

Основни алгоритам табу претраге

Табу претраживање алгоритам са краткорочном меморијом има следеће карактеристике:

1. То је процес од kt потеза у простору претраге, који обухвата допустива и недопустива решења. Број потеза kt се унапред дефинише или се одређује адаптивно.
2. Суседство текуће конфигурације се претражује како би се нашла конфигурација са минималним трошком. Потез је валидан уколико потез није наишао на табу, или је наишао на табу, али је критеријум аспирације испуњен. Помераји ка конфигурацијама са високом трошковима у односу на тренутну конфигурацију су дозвољени.
3. У свакој итерацији ажурира се листа табу атрибута.

Стратегија листе кандидата

Када се дефинише суседство, оно се оцењује. Уобичајено, када је суседство претерано велико или је оцењивање алтернатива дуготрајно, потребна је селекција да ограничи претрагу само на најатрактивније суседе. Зато се неком од стратегија креира листа кандидата, која памти квалитетна решења ради каснијег детаљнијег испитивања.

Табу мандат

Табу мандат (*Tabu tenure*) је број итерација табу претраживања (тј. број потеза или прелаза) при чему атрибут остаје забрањен. У пракси је могуће да неколико табу листа могу да буду истовремено одржаване, свака са другачијим мандатима. Дакле, не само појединачни атрибут него и комбинације атрибута могу бити забрањене. Врло ретко табу даје потпуну информацију о нежељеној конфигурацији. Уобичајено је да се само неки атрибути конфигурације стављају на листу табуа.

Критеријум аспирације

Типично, табу тест почиње са дефинисањем решења k_0 у околини тренутног решења k . Затим се идентификују атрибути од промењеног k у потезу од k до k_0 . Ако међу атрибутима постоје табу активни атрибути, потез још увек може бити проверен ако k_0 задовољава ниво аспирације (нпр. ако је k_0 довољно добар за оправдавање релаксације табуа).

Дугорочна меморија

Иако се значајан део литературе о табу претраживању бави алгоритмима базираним искључиво на техникама краткорочне меморије, сложеније софистициране апликације захтевају коришћење дугорочне меморије.

Постоји неколико различитих начина за коришћење дугорочне меморије, а то су:

1. реиницирање претраге из висококвалитетних конфигурација,
2. редефинисање суседства базирано на висококвалитетним решењима,
3. редефинисање функције циља за кажњавање одређених атрибута висококвалитетног решења, и
4. промена стратегије претраге засноване на знању стеченом током претходних претрага.

У пракси, обично је усвојена комбинација краткорочне и дугорочне меморије. Краткорочна меморија се обично спроводи као потпрограм за општи дугорочни алгоритам. Дугорочни алгоритам се обично заснива на три технике:

1. фреквентно заснована меморија,
2. интензификација, и
3. диверзификација.

Пример решавања проблема планирања дистрибутивних мрежа алгоритмом табу претраге [117]

У примеру који је дат у [117], проблем планирања дистрибутивних мрежа је формулисан као вишеетапни вишекритеријумски проблем мешовитог-целобројног нелинеарног програмирања. Вишекритеријумски проблеми у општем случају немају једнозначна решења, него имају скупове (Парето скуп) недоминантних решења која планер одабира неком од метода за доношење одлука.

Да би се решавао овако формулисан проблем планирања, предложен је вишекритеријумски алгоритам табу претраге. Исти се показао као добар алат за решавање великих и комплексних проблема као што је проблем планирања дистрибутивних мрежа изражен бинарним и континуланим променљивима.

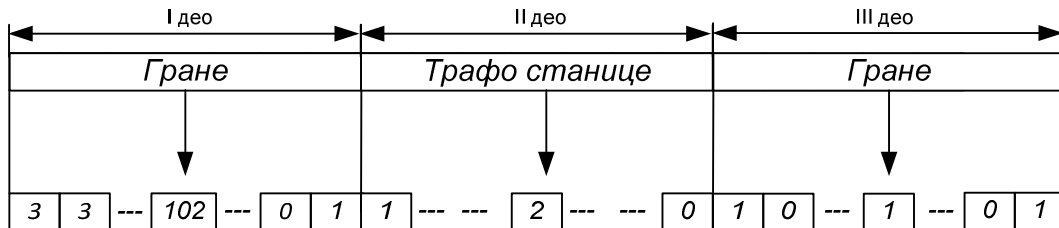
Вишекритеријумски проблем садржи две функције циља:

- прва функција циља презентује инвестиционе и фиксне трошкове (трошкови унапређања грана и трансформаторских станица, трошкови алокације прекидача, трошкови губитака) који се дефинишу за сваки од периода,
- друга функција циља презентује поузданост система, односно трошкове неиспоручене електричне енергије.

Код израчунавања неиспоручене електричне енергије узима се у обзир и могућност рестаурације "испалог" оптерећења манипулацијама у мрежи. Ограничења обухватају: баланс оптерећења чворова и трансформатора, ограничења капацитета, напонска ограничења, максимални број алокације прекидача. Такође се код исказивања трошкова узима у обзир "временска" вредност новца. Овакав модел се решава предложеним табу алгоритмом. То је итеративни поступак који генерише недоминантна решења кроз претрагу суседних решења у односу на текуће решења (мрежу). Прорачун табу алагоритмом се врши

за сваки од периода. Предложено кодирање решења (мреже) је децимално векторски и дато је на слици П1.19.

Децимални бројеви при кодирању означавају различите статусе и врсте грана, трафо станица и прекидача у мрежи. Први део вектора дефинише статусе и врсте грана, други део вектора дефинише статусе трафо станица, трећи део дефинише локације прекидача по гранама. Овако кодирана мрежа се из једног периода преноси у наредни следећи. Свака модификација решења (мреже) се реализује променом одговарајућих децималних бројева у вектору. На тај начин се генеришу суседна решења.



Слика П1.19. Предложена шема кодирања за један период

Квалитет ових добијених суседних решења је у директној вези са ефикасношћу предложеног табу алгоритма. Приликом генерисања суседних решења води се рачуна о два важна услова: повезаности грана и радијалности мреже.

Генерисање суседних решења се одвија на следећи начин:

- Реконфигурација мреже се врши техником измене грана.
- Елиминација/додавања грана које напајају потрошаче. Ако елиминисана/додата грана није повезана, радијалност и повезаност мора бити верификована.
- Измена капацитета трафо станица, измена капацитета грана.
- Елиминисање/додавање трафо станица. Елиминисање трафо станице подразумева елиминацију свих грана повезаних са истом, и додавање нових грана да изврше потребан трансфер оптерећења. Додавање трафо станице подразумева креирање грана повезаних са новом трафо станицом тако да је мрежа радијална.
- Елиминисање/додавање прекидача у мрежи.
- Промена позиције постојећих прекидача у мрежи.

У предложеном табу алгоритму поред класичне табу листе (*TL*), укључују се још две нове листе. Прва је Парето листа (*PL*) која меморише недоминантна решења генерисана током процеса претраге. Друга листа је листа кандидата (*CL*) која меморише сва друга недоминантна решења која нису меморисана у Парето листи, и која још нису генерисана до текуће итерације. Ова решења могу бити изабрана да буду нова решења (мрежа), тј. њихова суседства ће бити истраживана током претраге предложеног алгоритма.

На исти начин као код класичне табу претраге, суседна решења се генеришу из текућег решења (мреже), и сортирају се као доминантна и недоминантна решења. Једно од недоминантних решења се селекује да буде ново решење са којим се наставља претрага. То је процес интензификације (експлоатације) решења, односно претрага само у одређеном региону од целокупног простора претраге. С друге стране, процес диверзификације се примењује када недоминантна решења

нису пронађена међу текућим суседним решењима, и то на начин да се селекује ново решење (мрежа) из листе кандидата (CL). Претрага поновно започиње из овог решења да претражује нови регион, и на тај начин се избегава преурањени прекид процеса због постојања региона са локалним оптимумом.

Критеријум конвергенције у предложеном табу алгоритму је или максимални број итерација, или не постојање недоминантних решења који се могу пронаћи, и CL је празан скуп.

Предложени табу алгоритам је тестиран на дистрибутивној мрежи са 54 чвора и 21 граном. Проблем планирања је вршен за укупан период планирања од 15 година, који је подељен на три периода од по 5 година.

Добијени резултати представљају Парето скуп (недоминантних) решења. На тај начин предложени алгоритам је веома погодан за третирање вишекритеријумске природе проблема планирања дистрибутивних мрежа. Кроз добијени скуп решења планер може третирати различите сценарије развоја дистрибутивне мреже и правити компромис између (инвестиционих и оперативних) трошкова с једне, и поузданости (трошкова неиспоручене електричне енергије) с друге стране.

Предложени табу алгоритам је дат у псеудо коду на слици П1.20.

ПРИМЕР АЛГОРИТМА ТАБУ ПРЕТРАГЕ ИЗ ЛИТЕРАТУРЕ - псеудо код

Почетак: /*Иницијализација*/

$k = 1; PL = \emptyset; CL = \emptyset; TL = \emptyset;$

Генерисање почетног допустивог (радијалног) решења (мреже), $s = s_0;$

$PL = PL \cup s;$

Понављати /*Глобална итерација, k */

/*Генерисање и сортирање суседства*/

Генерисати суседство $N(s)$ од текућег решења $s;$

Сортирати из $N(s)$ доминантна и недоминантна решења;

Дефинисати скуп $N_{dod}(s)$ добијених недоминантних решења из $N(s);$

$\bar{N}(s) = \{s' | s' \in N_{dod}(s) \wedge s' \notin T(s, k)\};$

/* Проналажење кандидата */

Елеминисати из $\bar{N}(s)$ доминантна решења доминирана решењима из PL и $CL;$

Дефинисати скуп кандидата $C = \{\bar{N}(s) / \text{eleminisana dominantna rešenja}\};$

/*Ново изабрано решење*/

Ново решење се изабира случајно из $C, s_{\text{ново}} = s'_{\text{selektovano}};$

Ако је $C = \emptyset$ у текућој итерацији онда је ново решење најстарије из $CL,$

$s_{\text{ново}} = s_{\text{najstarije}}^{CL};$

/*Ажурирање листи*/

Додај TL атрибуте који "погоршавају" $s'_{\text{selektovano}};$

Ако је $C \neq \emptyset$ уклони из PL и CL решења доминирана решењима из $C,$ иначе уклони $s_{\text{najstarije}}^{CL}$ из $CL;$

Ако је $C \neq \emptyset$ додај $C - s'_{\text{selektovano}}$ скупу $CL;$

$PL = PL \cup s_{\text{ново}}; k = k + 1;$

/*Критеријум заустављања*/

Док Критеријум заустављања /*нпр. $k > k_{\text{max}}$ */

Излаз: Најбоље нађено решење

Слика П1.20. Псеудо код алгоритма табу претраге из литературе [117]

1.2.5. АЛГОРИТАМ СИМУЛИРАНОГ КАЉЕЊА

Увод

Симулирано каљење (*Simulated Annealing – SA*) је једна од најфлексибилнијих техника за решавање тешких комбинаторних проблема. Главна предност овог алгорита је да се може применити на велике проблеме, без обзира на услове диференцијабилности, континуитета и конвексности који се обично захтевају код конвенционалних метода оптимизације.

Каљење је процес термичке обраде челика који се користи у металургији. Најпре се врши загревање челика на високу температуру, а затим се врши хлађење, тако да се добију квалитетне кристалне структуре (тј. кристали чија структура формира савршену кристалну решетку) [147]. Ово се ради из разлога што се на тај начин постижу боља својства метала (тврдоћа и еластичност). Симулација каљење емулира физички процес жарења и првобитно је предложен у домену статистичке механике као средство моделовања природних процеса очвршћавања и формирања кристала. Током процеса хлађења, претпоставља се да се одржавају термички услови равнотеже. Процес хлађења се завршава када материјал достигне стање минималне енергије, који одговарају савреном кристалу. Познато је да се кристали без дефекта (тј. чврста супстанца са минимумом енергије) формирају са већом вероватноћом кад је процес хлађења успорен. Две главне карактеристике процеса симулираног каљења су: (1) механизам прелаза између стања и (2) план хлађења. Када се примењује на проблеме комбинаторне оптимизације, симулирано каљење има за циљ да пронађе оптималну конфигурацију, односно стање са минималном "енергијом".

Симулирано каљење је првобитно предложио Метрополис раних 1950-их година, као модел процеса кристализације. Тек 1980-их година започела су независна истраживања које су рађена у [167],[168], где је примећена сличност између физичког процеса каљења и неких комбинаторно оптимизационих проблема. Установљено је да постоји кореспонденција између алтернативних физичких стања материјала и простора решења код оптимизационих проблема. Такође је примећено да функција циља оптимизационог проблема кореспондира са слободном енергијом материјала. Оптимално решење је повезано са савршеним кристалом, док кристал са дефектом одговара локалном оптималном решењу [169]. Аналогија није потпуна јер у процесу каљења постоји физичка променљива температуре, која под одговарајућом контролом доводи до формирања савреног кристала. Када се симулирано каљење користи као оптимизациона техника, "температура" постаје једноставно контролни параметар који се мора правилно утврдити у циљу постизања жељених резултата.

Табела П1.4. Аналогија између металуршког и метахеуристичког каљења [15]

Каљење метала	Комбинаторна оптимизација
Могућа стања система	Прихватљива решења
Енергија система	Функција циља - трошак
Промена стања система	Прелазак у суседно решење
Температура	Контролни параметар
Замрзнуто стање	Оптимум (глобални или локални)

У табели П1.4. дате су наведене аналогије између физичког процеса каљења метала и комбинаторне оптимизације [170]. На основу наведеног види се да симулирано каљење у суштини одговара оптимизационом проблему минимизације, тако што минимум енергије система код замрзнутог стања одговара оптимуму.

Алгоритам Метрополис

Првобитна идеја за алгоритам симулираног каљења је Метрополис алгоритам који моделује микроскопска понашање скупова великог броја честица у чврстој супстанци, помоћу *Monte Carlo* симулације. У материјалу појединачне честице имају различите нивое енергије, у складу са одређеном статистичком дистрибуцијом. Могуће најнижи ниво енергије, познат као фундаментални ниво, одговара стању где све честице мирују, и он се јавља при температури 0°K . За температуре изнад тог нивоа, честице ће заузети различите нивое енергије, тако да се број честица у сваком нивоу смањује како се повећава ниво енергије (односно максималан број честица се налази у фундаменталном нивоу). Дистрибуција честица у различитим нивоима варира са температуром. На пример, за $T = 0^\circ\text{K}$, све честице су у фундаменталном нивоу. Са порастом температуре, више честице се налазе у вишим енергетским нивоима, али увек као опадајућа функција нивоа енергије. Метрополис алгоритам генерише низ стања чврсте супстанце као што следи: дато је чврсто стање S_i , са енергијом E_i , следеће стање S_j генерише се механизмом прелаза (транзицијом) који се састоји од мале пертурбације у односу на првобитно стање, добијеном померањем једне од честица чврсте супстанце по избору *Monte Carlo* методом. Нека је енергија добијеног стања, која је такође пронађена пробабилистички E_j ; онда ако је разлика $\Delta E = E_j - E_i$ мања или једнака нули, ново стање S_j се прихвата. Иначе, у случају да је разлика већа од нуле, ново стање S_j се прихвата са вероватноћом:

$$P = \exp\left(\frac{E_i - E_j}{k_B T}\right) \quad (\text{П1.16})$$

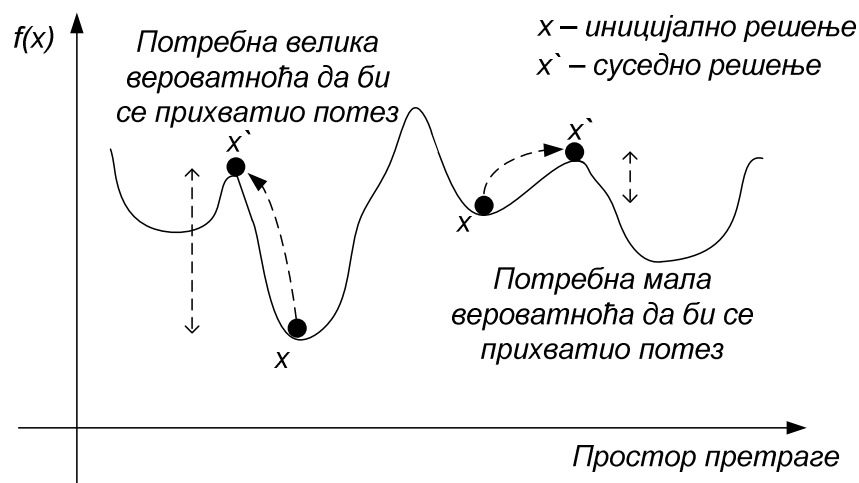
где је T температура чврсте супстанце, и k_B је Болцманова константа ($k_B = 1.3806503 \cdot 10^{-23} \frac{\text{m}^2 \text{kg}}{\text{s}^2 \text{K}}$).

Ово правило прихватања је такође познато као Метрополисов критеријум. Претпоставља се да температура има стопу варијација, тако да је постигнут термодинамичка равнотежа за тренутни температурни ниво, пре преласка на следећи ниво. Ово обично захтева велики број прелаза стања у Метрополис алгоритму.

Алгоритам Симулираног каљења

Комбинаторни проблем оптимизације који се решава алгоритмом симулираног каљења формулисан је на следећи начин: нека буде G коначан велики скуп конфигурација и v трошак придружен свакој конфигурацији G . Решење комбинаторног проблема се састоји од тражења простора конфигурација за пар (G, v) представљајући најнижу цену. Алгоритам симулираног каљења почиње са почетном конфигурацијом G_0 и почетном "температуром" $T = T_0$ и генерише секвенцу конфигурација $N = N_0$. Онда се у свакој итерацији врши "хлађење" односно смањивање температуре и процењивање потенцијалних нових конфигурација кандидата. Кандидат конфигурација је прихваћена ако је цена мања од тренутне конфигурације. Ако су трошкови конфигурације кандидата већи

од трошкова текуће конфигурације, она још увек може бити прихваћена са одређеном вероватноћом. Ова способност деградације текућег решења, односно способност да се прихватају и решења која су лошија од текућег, омогућава симулираном каљењу да побегне из локалних оптималних конфигурација. Овај процес се понавља док се не испуни неки од критеријума заустављања који ће бити касније ближе описани. Цео процес се контролише планом хлађења који одређује како се температура смањује током процеса оптимизације.



Слика П1.21. Симулирано каљења има способност да "побегне" од локалног оптимума. Виша температура даје већу вероватноћу прихватања најгорих потеза. Бољи потез је увек прихваћен [15]

Псеудо код резимира алгоритам симулираног каљења, који се састоји од два основна механизма: генерисање алтернатива и правила прихватања. T_k је контролни параметар који је аналоган са температуром у физичком процесу каљења челика, и N_k је број алтернатива које настају на k -ом нивоу температуре (ово одговара времену да систем остаје на датом нивоу температуре и треба бити довољно велики да омогући систему да достигне стање које одговара "термичкој равнотежи"). У почетку кад је T је велики, већа погоршања у функцији трошкова су дозвољена (има се већа деградација решења); како се температура смањује, алгоритам за симулирано каљење постаје "похлепнији" и само се мања погоршања прихватају (има се мала деградација решења); и коначно кад температура T тежи нули, погоршања се опште не прихватају (деградација решења је онемогућена). Из тренутног стања S_i , са трошковима $f(S_i)$, суседно решење S_j , са трошковима $f(S_j)$, се генерише механизмом транзиције. Вероватноћа прихватања новог решења се израчунава тестом на следећи начин:

$$P_t\{\text{prihvatanja } S_j\} = \begin{cases} 1, & f(S_j) \leq f(S_i) \\ \exp\left(\frac{f(S_i) - f(S_j)}{T_k}\right), & f(S_j) > f(S_i) \end{cases} \quad (\text{П1.17})$$

АЛГОРИТАМ СИМУЛИРАНОГ КАЉЕЊА - псеудо код

Улаз: Програм (распоред) хлађења: /*Почетна температура (T_{max}), Крајња температура (T_{min}), Темпо (брзина) хлађења, Број итерација на одређеној температури*/. $s = s_0$; /*Генерисање почетног допустивог решења*/

$T_k = T_{max}$; /*Почетна температура*/

$k = 1$;

Понављати /*Глобална итерација, k */

Понављати /*На одређеној температури T_k , односно у глобалној итерацији k */ Генерисати суседно (блиско) решење s' ;

$\Delta E = f(s') - f(s)$; /*Разлика циљних функција*/

If $E \leq 0$ **Then** $s = s'$; /*Прихватити s' као текуће решење*/

If $f(s') - f(s_{best}) < 0$ **Then** $s_{best} = s'$; /*Прихватити s' као најбоље решење до сада (s_{best}) /

Else Прихватити s' са вероватноћом $e^{-\Delta E/T_k}$, односно прихватити s' ако важи: $e^{-\Delta E/T_k} > rand[0,1]$;

Док Равнотежни услов /*нпр. дефинисани број итерација на свакој температури T_k */ $T_{k+1} = g(T_k)$; /*Темпо (брзина) хлађења*/

$k = k + 1$;

Док Критеријум заустављања /*нпр. $T_k < T_{min}$ */

Излаз: Најбоље нађено решење

Слика П1.22. Псеудо код алгоритма симулираног каљења

План Хлађења

План хлађења је стратегија контроле која се користи од почетка до конвергенције алгоритма симулираног каљења. План хлађења одликују следећа четири параметра:

1. Почетна температура T_0 ,
2. Завршна температура T_f (критеријум заустављања),
3. Стопа промене температуре,
4. Број прелаза Nk , при температуре T_k .

Да би алгоритам симулираног каљења био ефикасан, потребно је добро подесити параметре алгоритма чиме алгоритам прилагођавамо проблему који се решава.

Почетна температура

Међу главне параметре који се подешавају, спада план хлађења, односно почетна температура. Тако на пример, уколико је почетна температура превисока, постојаће велика вероватноћа прихватања "лоших" решења, односно алгоритам се гура у насумичну широку претрагу простора решења приликом које ће са великом вероватноћом прихватити свако решење (независно од његовог

квалитета), што је временски веома захтевно. Уколико се узме у обзир да је време за рад алгоритма ограничено, на тај начин остаје мање времена за "усмерену" претрагу квалитетних решења. С друге стране, уколико је почетна температура премала, вероватноћа прихватања "лоших" решења је у старту мала, тако алгоритам у старту креће у "усмерену" претрагу и на тај начин остаје "заробљен" у неком од атрактивних региона претраге где ће доћи до најбољег решења за тај регион, што је вероватно локални оптимум. На тај начин, велики део простора решења са квалитетним решењима од стране алгоритма неће бити ни посећен. Очито да се мора успоставити равнотежа између ова два екстремна случаја. У литератури се могу наћи разне стратегије како дефинисати почетну температуру. Тако уколико се не познаје проблем, у [171] се предлаже приступ при којем се креће са врло високом почетном температуром која се затим нагло спушта, све док је стопа прихватања лошијих решења већа од 60%. Температура при којој ова стопа падне испод 60% постаје почетна температура са којом се даље проводи класичан алгоритам симулираног каљења. У [170] се предлаже другачији приступ. Наиме, почиње се са неком ниском температуром која се постепено подиже (врши се загревање). На тај начин стопа прихватања лошијих решења све више расте. Када се оваквим загревањем достигне жељена стопа прихватања, достигнута температура се узима за почетну температуру са којом се проводи класичан алгоритам симулираног каљења. Ефикасност алгоритма у погледу квалитета коначних решења, као и број итерација ће зависити од избора ових параметара. Процедуре које се користе за израчунавање наведених параметара заснивају се на идеји о термичкој равнотежи и детаљно су описани у даљем тексту.

- **Одређивање почетне температуре T_0**

Постоји неколико начина за одређивању почетне температуре T_0 за алгоритам симулираног каљења.

Једна начин предложен у [147], полази од претпоставке да почетна температура треба да буде довољно велика да приближно сви прелази на овој температури буду прихватљиви.

$$T_0 = \frac{\overline{\Delta V}^+}{\ln\left(\frac{m_2}{m_2 X_0 - m_1(1 - X_0)}\right)} \quad (\text{П1.18})$$

Ово се може постићи генеришући број решења m_0 , од тога m_1 одговара броју потеза (решења) са опадајућим трошковима, док је m_2 број потеза (решења) са повећањем трошкова, и важи $m_0 = m_1 + m_2$. Износ $\overline{\Delta V}^+$ представља просечну вредност разлике између функције циља текућег (почетног) решења и лошијих решења (којих има m_2). Вредност X_0 кореспондира са односом прихватања нових конфигурација. У литератури се најчешће користи вредност $X_0 = 0,85$, што значи да се на почетној температури T_0 , 85 % потенцијалних решења која се тестирају према П1.17, прихвата.

Алтернативни начин одређивања T_0 предложен је у [172]:

$$T_0 = \frac{\mu}{-\ln\phi} f(x^0) \quad (\text{П1.19})$$

где се претпоставља да је ϕ % "лоших" потеза (потези који воде до решења са већим трошком од текућег), који су μ % лошији од почетног решења $f(x^0)$, а који се прихватају на почетном нивоу температуре T_0 .

Завршна температура

Уобичајено је да се смањење температуре врши док не достигне вредност нула. Међутим, ово може имати за последицу да алгоритам ради временски веома дуго, нарочито када се користи геометријски распоред хлађења (видети доле одељак: *Дефинисање плана хлађења*). У пракси, није потребно да температура достигне нулу јер како се приближава нули, шансе за прихватање лошијих потеза су скоро исте (једнако мале) као када је температура једнака нули. Дакле, зауставни критеријум може бити или погодно ниска температура или ситуација када је систем "замрзнут" на некој температури (тј. ниједан потез није прихваћен).

Стопа промене температуре

Стопа промене температуре у општем случају се може изразити као:

$$T_{k+1} = g(T_k)T_k \quad (\text{П1.20}).$$

где је $g(\cdot)$ је функција која контролише температуру.

Код одређивања стопе промене температуре битно је да се на свакој температури омогући довољан број итерација како би се систем стабилизовао на датој температури. На жалост, да би се то постигло број итерација на одређеној температури је експоненцијалан величини проблема. Како је то неизводљиво, прави се неки од избора: или велики број итерација на малом броју температура, или мали број итерација на великом броју температура, или равнотежа између ова два приступа.

Број прелаза N_k

Број потеза обављених на сваком нивоу температуре треба да буде да такав да је услов топлотне равнотеже загарантован. Стога, вредност овог параметра је тесно повезана са стопом смањења температуре. Већина алгоритама користе вредност N_k који зависи од димензије проблема (број управљачких варијабли). Два предлога који се појављују у литератури су у овде дати.

1. Статички - Константан број прелаза N_k

$$N_{k+1} = N_0 \quad (\text{П1.21})$$

где је N_0 је број тестова према П1.17. на иницијалном температурном нивоу.

2. Динамички - Променљив број прелаза N_k :

$$N_{k+1} = \rho N_k \quad (\text{П1.22})$$

где је ρ је кориснички параметар. Овај приступ са променљивим бројем прелаза је рачунски захтевнији, али у општем случају доводи до бољих резултата у односу на оне добијене са првим приступом. Суштина је да се број итерација мења динамички како алгоритам напредује. На нижим температурама важно је да се врши велики број итерација тако да се уочени атрактивни региони у потпуности истраже. На вишим температурама, број итерација може бити мањи јер се тада случајним избором врши "широка" прерага атрактивних региона у комплетном простору решења.

Дефинисање плана хлађења

Постоји више начина да се изврши смањење температуре код симулираног каљења. Све ове методе су засноване на основу чињенице да термална равнотежа треба да буде постигнута пре смањења температуре. Такође, увек

постоји међузависност између брзине смањивања температуре и квалитета добијених решења. Наиме, уколико се температура брзо смањује, алгоритам ће се извршити брзо, али добијена решења неће бити квалитетна, с друге стране, уколико се температура полако смањује, то ће имати за последицу дуже извођење алгоритма, али ће решења бити квалитетнија. Овде ће бити представљени неки од планова хлађења који се користе:

- Линеарни план хлађења

Код линеарног плана хлађења температура се за сваку итерацију смањује за фиксни износ на следећи начин:

$$T_k = T_0 - k \cdot \beta \quad (\text{П1.23})$$

где је T_0 почетна температура, k је број итерације, док је β фиксни износ за који се смањује температура по итерацији. Уколико се зна да се има укупно K итерација, и да се температура смањује на крајњу вредност T_K , може се израчунати параметар β .

$$\beta = \frac{T_0 - T_K}{K - 1} \quad (\text{П1.24})$$

- Геометријски план хлађења

Код геометријског плана хлађења температура се за сваку итерацију смањује за фиксни износ који се мења кроз итерације на следећи начин:

$$T_k = \alpha^k T_0 \quad (\text{П1.25})$$

где је T_0 почетна температура, а k започиње број итерације који започиње од 0, α је параметар који узима вредности из интервала $(0, 1)$. Уколико се зна да имамо K итерација и да се температура смањује на крајњу вредност T_K , може се израчунати параметар α .

$$\alpha = \sqrt[k-1]{\frac{T_K}{T_0}} = \left(\frac{T_K}{T_0}\right)^{\frac{1}{K-1}} \quad (\text{П1.26})$$

- Логаритамски план хлађења

Код овог плана температура у k -тој итерације се рачуна као:

$$T_k = \frac{T_0}{\log k} \quad (\text{П1.27})$$

Код овог плана је хлађење веома споро, па се из тих разлога веома ретко и користи.

- План веома спорог хлађења

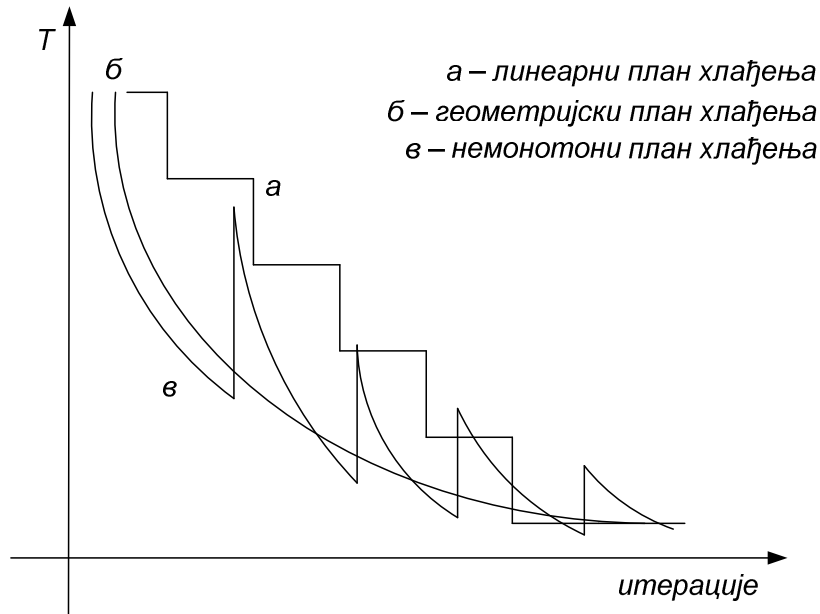
У овом случају се температура веома полако мења, што захтева веома велики број корака k . На тај начин број прелаза Nk на одређеној температури, може да се сведе да буде једнак јединици, чиме се из псеудо кода избацује унутрашња петља.

$$T_{k+1} = \frac{T_k}{1 + \beta T_k} \quad (\text{П1.28}).$$

где је T_k температура у кораку k , а β контролни параметар.

- Немонотони план хлађења

Типичан план хлађења користи монотоне температуре. У литератури [173] су предложене неке немонотоне шеме, где се температура осим што опада, може и да се повећава. Ово подстиче диверзификацију у простору претраге. За неке врсте претраживања простора решења, оптималан план хлађења је немонотон [174]. Овакав план хлађења је примењен у овој дисертацији код хибридног алгоритма за решавање проблема планирања развоја радијалних дистрибутивних мрежа, јер се показало да даје боље резултате од класичних планова хлађења који се примењују код симулираног каљења [139].



Слика П1.23. Графичка представа неких планова хлађења код симулираног каљења

- Адаптивни план хлађења

Већина планова за хлађење су статични у смислу да је план хлађења у потпуности дефинисан априори (унапред). У том случају план хлађење је "слеп" на карактеристике претраге простора решења који се решава. У адаптивном плану хлађења, стопа смањења температуре је динамична и зависи од неких информација добијених приликом саме претраге [175]. Динамичан план хлађење може да се користи тамо где је мали број итерација које се изводе на високим температурама, и велики број итерација на ниским температурама.

Критеријум заустављања

Критеријуми заустављања се међусобно много разликују како по степену сложености тако и рачунских захтева. Неке од најчешћих стратегија су дати и могу се свести на следеће:

1. Постизање коначне температуре T_f . Ово је најпопуларнији критеријум заустављања. Ова температура мора да буде ниска (нпр. , $T_{min} = 0.01$).
2. Дефинисати константан број температурних редукција (обично између 6 и 50 редукција).

3. Користити стопу побољшања функције трошкова за дефинисање критеријума заустављања. Тако уколико се после серије температурних редукција актуелно решење или трошкови најбољег решења до сада не побољшавају, претпоставља се да је постигнута конвергенција и процес се зауставља.
4. Дефинисати број "лошијих" потеза (оних који дају решење са већим трошком) које треба прихватити. Процес се зауставља кад број прихватања нових решења постаје мањи од наведене вредности.

Унапређење алгоритма симулираног каљења

Иако се алгоритам симулираног каљења показао као добар за решавање великог броја оптимизационих проблема, увек је корисно извршити нека од унапређења која би побољшале перформансе алгоритма као што су: квалитет решења, рачунарски захтеви, време рада алгоритма итд. Неке од могућих унапређења су овде описана:

Иницијализација

Алгоритам симулираног каљења уобичајено започиње са неким иницијалним решењем које се добија случајно. Међутим, утврђено је да алгоритам даје боље резултате за краће време рада када се за иницијално решење узима решење које је већ хеуристички изграђено. На пример, код проблема трговачког путника за иницијално решење би било корисно да буде изграђено коришћењем неке од хеуристика, нпр. похлепном претрагом.

Хибридизација

Хибридизација је нова техника која означава комбиновање два (или више - али обично два) алгоритма претраге заједно, са циљем да новодобијени алгоритам садржи позитивне особине алгоритама од којих је састављен. Ово је релативно нова област истраживања, али је суштина у томе да се популационо засноване стратегије претраживања (P - метахеуристике као нпр. алгоритми роја честица, генетски алгоритми итд.) користе као примарни механизам за претрагу, док се део јединки из популације добија локалном претрагом, алгоритмима који раде над једним решењем (S - метахеуристике као нпр. симулирано каљење и табу претрага). На овај начин се користи особина техника локалне претраге (симулирано каљење и табу претрага), да се под одређеним условима дозвољава деградације текућег решења, што онемогућава прерану конвергенцију претраге у локалном оптимуму.

Остали слични алгоритми

Друге сличне методе симулираног каљења су предложене у [15], као што су: праг прихватања, алгоритам великог постопа (*great deluge*), алгоритам *record to record*, и алгоритам демона (слика П1.24). Главни циљ у пројектовању ових алгоритама, инспирисаних симулираним каљењем, је да се убрза претрага без "жртвовања" квалитета решења.



Слика П1.24. Алгоритми настали из алгорита симулираног каљења [15]

Паралелни алгоритам симулираног каљења

Упркос свим својим добрим карактеристикама, алгоритам симулираног каљења изузетно је захтеван у погледу потребног рачунарског времена за извођење претрага. Из тих разлога у [147] предлаже се паралелизација рада овог алгоритма. Међутим, паралелизација симулираног каљења није једноставан задатак. Главна тешкоћа лежи у чињеници да симулирано каљење ради као Марковљев ланац, који је типичан секвенцијални ентитет. Модел Марковљевог ланца моделује процес симулираног каљења као низ тестова где је вероватноћа исхода датог теста зависи само од исхода текућег теста (тренутне конфигурације-решења или топологије), и не зависи од тестова у низу који су дошли пре текућег. Стога паралелизација треба да се обавља на такав начин, да се ова секвенцијални структура не нарушава када је алгоритам за симулирано каљење мапиран у паралелној архитектури.

Код алгоритма симулираног каљења тестирање Nk потенцијалних решења се обавља на свакој температури Tk . Свако тестирање подразумева следеће кораке:

1. Из суседства тренутне конфигурације (решења), избор конфигурације за тестирање.
2. Прорачун разлике између трошкова текућег решења и потенцијалног решења.
3. Одлука да ли потенцијално решење треба да буде прихваћено или не.
4. Замена тренутне конфигурације новом, ако је то случај.

Следеће питања треба размотрити код обављања паралелизације:

1. Прва три корака су независна и могу се извести паралелно а да не утичу на секвенцијалну природу алгоритма. Уколико би се 4-ти корак радио паралелно, то би мењало секвенцијалну природу алгоритма.
2. Прорачуни који су укључени у ова четири корака су веома зависни од проблема.

- Алгоритам поделе

Нека np означава број процесора у паралелној машини. Једноставан и ефикасан начин паралелизације за алгоритам симулираног каљења се састоји у подели посла у стварању одговарајућег ланца Маркова над NP процесора: сваки процесор обавља Nk/np тестова. Да би сачувао главну карактеристику SA алгоритма, за сваки ниво температуре, када сви процесори заврше обраду својих појединачних задатака, актуелна оптимална решења се шаљу до мастер чвора

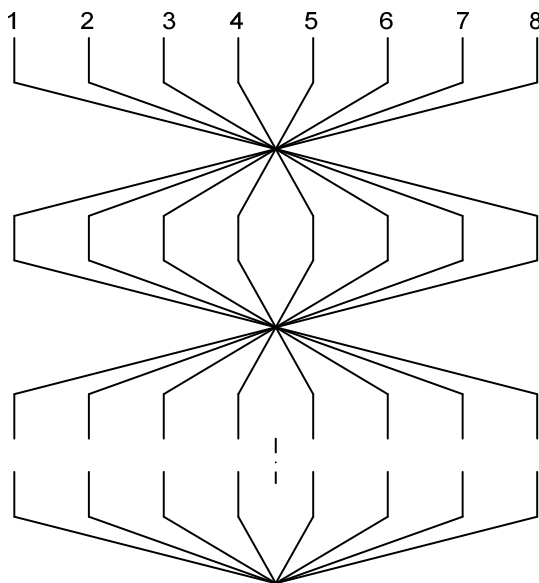
(чвор $p = 0$), који затим бира најбоље решење и прослеђује резултате свим осталим процесорима ($p = 1, \dots, np - 1$).

Захтеви за комуникацијом код овог алгоритма су релативно мали, што значи да је потенцијал за ефикасност алгоритма висока. Квалитет решења зависи од броја процесора који се користе у паралелном израчунавању. За веће бројеве процесора, број конфигурација који се обрађује у сваком процесору, може постати сувише мали, што не дозвољава систему да достигне топлотну равнотежу. Да би се тај проблем превазишао, број обављених тестова по нивоу температуре може бити повећан, што се компензује повећањем стопе хлађења.

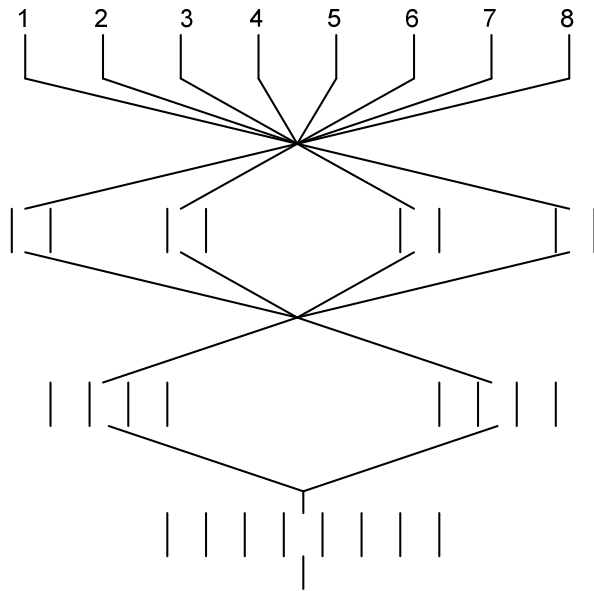
Слика П1.25. илуструје рад овог алгоритма поделе. Постоји осам процеса на слици. Вертикална колона илуструје рад сваког процеса који врши обраду решења за сваки ниво температуре. Затим се врши комуникација са мастер чвором, да би се од истог добили подаци о новом текућем решењу, које се затим поново обрађује од стране осам процесора.

- Алгоритам кластера

У овом случају, за разлику од алгоритма поделе који је горе описан, секвенцијална природа алгоритма симулираног каљења се строго поштује [147],[176]. Имамо Np процесора који тестирају Nk решења кооперативно, што значи да сви процесори увек раде са истим текућим решењем. Стога, кад један процесор прихвати актуално решење, он то јавља осталим процесорима. Овакав приступ паралелизације је мање ефикасан на високим температурама, где је учесталост прихватања нових решења релативно висока. Супротно је код ниских температура, где је стопа прихватања нових решења мала, па тада алгоритам показује најбоље перформансе.



Слика П1.25. Алгоритам поделе [124]



Слика П1.26. Хибридни алгоритам подела/кластер [124]

Добра алтернатива је коришћење хибридног алгоритма подела/кластер. Процес почиње као алгоритам поделе, а затим код нижих температура прелази на алгоритам кластера. Тачка у којој се врши прелазак са једног на други алгоритам се дефинише на основу стопе прихватања нових решења. Начин рада овог хибридног алгоритма је дат на слици П1.26.

Резиме

Алгоритам симулираног каљења је стохастички алгоритам који не користе меморију, јер не користи било какве информације прикупљене приликом претраге, и који под неким условима омогућава деградацију решења. Основни квалитет овог алгоритма је да поседује способност да побегне из локалног оптимума и тако одложи прерану конвергенцију, те се показао као веома погодна техника за решавање великог броја оптимizacionих проблема, како комбинаторних тако и континуалних, и даје добра решења, чак и у ситуацијама када се добро не познаје структура проблема који се решава.

На који начин је овај алгоритам примењен у овој дисертацији код планирања радијалних дистрибутивних мрежа, детаљно је описано је у поглављу 3. Претходни приступи примене алгоритма симулираног каљења су код генерисања решења користили углавном само механизам интензификације (искоришћавања - *exploitation*), односно механизам локалне претраге у околини полазног доброг решења. За разлику од тога у [75] је дата примена овог алгоритма који комбинује како механизам интензификације, тако и механизам диверзификације (истраживања - *exploration*), односно механизам глобалне претраге чиме се посећују остали обећавајући региони допустивог простора претраживања.

ПРИЛОГ 2. ХЕУРИСТИЧКЕ ТЕХНИКЕ У ПРЕДЛОЖЕНОМ ХИБРИДНОМ АЛГОРИТМУ

2.1. ТЕХНИКА ИЗМЕНЕ ГРАНА

Техника измене грана је широко кориштена код проблема планирања дистрибутивних мрежа. Она спада у технике локалног претраживања и представља једноставну хеуристику са којом је могуће вршити модификацију постојећих решења. Наиме, ова техника креће од постојећег решења (конфигурације) које представља уклопно стање дате дистрибутивне мреже. Даље се изменом положаја нормално отвореног (НО) расклопног уређаја и нормално затвореног (НЗ) расклопног уређаја, врши измена уклопног стања дистрибутивне мреже, а самим тим и модификација текућег решења при чему се мењају трошкови (појачања и изградње водова, губитака, поузданости итд.) који су предмет минимизације. Тако се измена грана своди на измену места НО расклопног уређаја и са њим спрегнутим НЗ расклопним уређајем. На тај начин се генеришу нова суседна решења са којима може даље да се иде у обраду одговарајућим алгоритмима. Један пример алгоритма измене грана који је дат у [177], има следећи псеудо код приказан на слици П2.1.

ТЕХНИКА ИЗМЕНА ГРАНА псеудо код

Почетак: /*Иницијализација*/

$s = s_0$; /*Генерисање почетног допустивог решења*/

$n_{NO} = 1$

Понављати /*итерација, n_{NO} */

Генерисати скуп решења $f(s')$, $s' \in n_{NZ}$; /* за сваки спрегнути НЗ расклопни уређај*/

Израчунај критеријумску функцију за сваки $f(s')$ /*Израчунај $f(*)$ за сваки спрегнути НЗ расклопни уређај*/

If постоји ли бар један за који важи $f(s') < f(s)$;

Then нађи $f(s')$ који највише унапређује критеријумску функцију

Прорачунати токове снага за новодобијену конфигурацију (уклопно стање) s' ;

$n_{NO} = n_{NO} + 1$;

Док Критеријум заустављања /* $n_{NO} = NO$, обрађени су сви НО расклопни уређаји*/

Излаз: Најбоље нађено решење (конфигурација) односно уклопно стање

Слика П2.1. Псеудо код технике измене грана

Потребно је напоменути да код овог алгоритма измене грана постоје ограничења у смислу избора смера кретања у мрежи где ће бити НО расклопни уређај.

- Место где се налази НО расклопни уређај се помера дуж одређеног правца све докле се остварује побољшање критеријумске функције, или док се не дође до другог НО расклопног уређаја.
- Није дозвољено померање НО расклопног уређаја тако да дође до престанка напајања дела мреже.

Ефикасност ове методе се огледа у томе што није потребно рачунати целокупни режим након промене статуса пара НО и НЗ расклопних уређаја, већ само за део мреже која је променила уклопно стање. Основна мана ове методе је (као и код свих метода локалног претраживања) што је добијено решење веома зависно од почетног решења.

2.1.1. Резиме

У предложеном алгоритму техника измене грана је примењена за генерисање суседних решења приликом примене механизма интензификације код симулираног каљења. Детаљнији опис примене ове технике у предложеном алгоритму је дат у поглављу 3.

2.2. КОНЦЕПТ ЛОКАЛНЕ МРЕЖЕ

Концепт локалне мреже се примењује код планирања дистрибутивних мрежа, тако што се читава мрежа дели у више локалних мрежа (фаза, подпроблема). На овај начин, димензије укупног проблема су драстично смањени што омогућава примену одговарајућих алата и алгоритама у свакој фази (у сваком подпроблему). Тако се оптимални дизајн мреже добија секвенцијалним решавањем одређеног броја дефинисаних подпроблема [137].

2.2.1. Дефинисање зоне Z_k

У првом кораку процедуре се дефинише локација и тип проблема у мрежи. Типови проблема који могу бити решени у оквиру ове процедуре су: нарушавање термичких ограничења, нарушавање напонских ограничења, напајање нових потрошача (чворова) као и комбинација наведених проблема. Почетно стање мреже за идентификовање и лоцирање горе наведених проблема у хоризонтној години је конфигурација мреже у базној (нултој) години. Део мреже у којем се манифестују неки од горе наведених проблема се назива зона Z .

Пораст оптерећење може узроковати да се ови проблеми јављају једновремено у различитим деловима мреже. У том случају може се појавити више зона Z , па ће се вршити означавање за сваку од k зона - Z_k . Дефинисање зона Z_k за горе наведене проблеме је описано на следећи начин:

- Нарушавање термичких и/или напонских ограничења: У том случају зона Z_k се састоји од фидера у којима су једно или оба ограничења нарушена.
- Нови потрошачки захтеви (нови чворови): У том случају зона Z_k се састоји од фидера који имају могућност напајања нових чворова (потрошача) изградњом нових грана.

Када се претходни случајеви појављују једновремено, зона Z_k садржи унију фидера из првог и другог случаја.

2.2.2. Дефинисање локалне мреже

Локална мрежа се дефинише за сваку зону Z_k . Свака од ових локалних мрежа састоји се од зоне Z_k и одређеног броја фидера помоћу којих је могуће реалоцирати део оптерећења зоне Z_k или напојити нове чворове. Реалоцирање оптерећења се врши затварањем нормално отворених (НО) прекидача и отварањем нормално затворених (НЗ) прекидача у мрежи. Главна идеја у одређивању величине локалне мреже је процена броја фидера потребних да

преузму део терета из зоне Zk који узрокује темичке и/или напонске проблеме у зони и/или да напоје нове чворове. За то сврху дефинише се индекс расположивости за сваки од фидера кандидата који ће бити укључен у локалну мрежу. Предложени индекс је поједностављена провера могућности фидера да преузме терет од суседних фидера, путем затварања његових НО прекидача и отварања НЗ прекидача суседних фидера, и/или могућности напајања нових терета (чворова) путем повезивања истих са фидером [138]. Дефинисање фидера који припадају локланој мрежи урађено је у итеративним циклусима. У првој итерацији, сви фидери који имају НО прекидаче према фидерима из зоне Zk и/или имају могућност да напајају нове потрошаче (чворове) у овој зони, укључени су у локалну мрежу. Затим, следећи круг фидера је сачињен од свих фидера који имају НО прекидаче према свим фидерима из претходног круга (итерације) итд. Ширење локалне мреже се наставља све док се један од следећих услова не испуни:

$$1. \sum_{i \in \alpha_F^j} I_i \geq I_c, \quad (П2.1)$$

где су:

I_i - расположиви капацитет (оптерећење) i -тог фидера

I_c - оптерећење које треба да буде алоцирано да би се решио термички и/или напонски проблем и/или напојило ново оптерећење у зони Zk

α_F^j - скуп индекса фидера који се завршавају са j -тим кругом локалне мреже, напајајући фидере из зоне Zk

2. Ширење локалне мреже (укључујући нове фидере) не може унапредити процес реалокације оптерећења и/или процес напајања нових потрошача (чворова) - (нпр. услед ограничења радијалности расположиви капацитет фидера из $(j + 1)$ круга не може бити искориштен).

Након формирања локалне мреже за сваку од зона Zk , може доћи до два случаја:

- не постоје заједнички елементи (фидери) за било које од две или више локалних мрежа,
- постоје заједнички елементи за било две или више локалних мрежа.

У првом случају, цела мрежа се може поделити на k независних локалних мрежа. Свака од њих се посматра одвојено и решавањем k независних проблема, добија се решење (сценарио планирања) целе мреже. У другом случају, примењује се процедура (псеудо)динамичког програмирања за добијање најбољег сценарија планирања за целу мрежу [137].

2.2.3. Резиме

У предложеном алгоритму концепт локалне мреже је примењен за поделу (декомпозицију) мреже приликом примене механизма диверзификације код симулираног каљења. Детаљнији опис примене овог концепта у предложеном алгоритму је дат у поглављу 3.

ПРИЛОГ 3. СОФТВЕРСКА ПОДРШКА ПРЕДЛОЖЕНОГ АЛГОРИТМА

3.1. Увод

Као софтверска подршка предложеног алгоритма коришћени су следећи програмски пакети:

- *MATLAB* и *TOMLAB* за извршавање *MILP* прорачуна код решавања локалних мрежа,
- *MICROSOFT EXCEL* код аквизиције података добијених од стране *MILP* прорачуна, и прорачуна за потребе алгоритма симулираног каљења.

3.2. Програмски пакет *MATLAB*

MATLAB је језик високог нивоа, као и интерактивно окружење за нумеричке рачунање, визуализацију и програмирање. Користећи *MATLAB* могу се реализовати и израчунати различите математичке функције, развијати алгоритми, анализирати, моделовати и вршити симулације, као и графички представљати резултати. Језик, алати и уграђене математичке функције омогућавају да се вишеструко брже приступи и долази до решења, него када се то ради са табелама или применом традиционалних програмских језика, као што су *C/C++* или *Java* [178]. *MATLAB* може имати широки спектар примене, укључујући и обраду сигнала и комуникација, графичку обраду слике, системе контроле, тестирања и мерења, рачунарске биологије итд. *MATLAB* је настао као скраћеница за „*MATrix LABoratory*“ („лабораторија за матрице“). Овај програмски пакет је направљен касних 1970-их године на Универзитету Нови Мексико и Стендфорд Универзитету у Сједињеним америчким државама, с циљем да се примењује у матричној теорији, линеарној алгебри и нумеричкој анализи. Производи га фирма *MathWorks*.

Кључне карактеристике овог програмског пакета су:

- језик високог нивоа за нумеричку анализу, визуелизацију и развој апликација,
- интерактивно окружење за итеративно истраживање, пројектовање, и решавање проблема,
- математичке функције линеарне алгебре, статистике, Фуријеова анализа, филтрирање, оптимизација, нумеричка интеграција, и решавање обичних диференцијалних једначина,
- уграђене графике за визуализацију података и алати за креирање прилагођених графикчких решења,
- алати за креирање апликација са прилагођеним графичким интерфејсом,
- функције за интеграцију *MATLAB* алгоритама са спољним апликацијама и језицима као што су *C, Java, .NET* и *Microsoft Excel*.

Овај програмски пакет има широку примену како у индустрији, тако и на универзитетима широм света. На тај начин је *MATLAB* постао стандардно наставно средство на универзитетима на курсевима из математике и различитих инжењерских и научних дисциплина. Основна језгра *MATLAB*-а чини матрични калкулатор, који може да извршава низ наредби груписаних у скрипте, односно функције. Скупови оваквих функција које су оријентисане за решавање кориснички дефинисаних проблема, чини један алат (*Toolbox*). Велика предност *MATLAB*-а је околност да је направљен као програм са отвореном архитектуром, тако да је могуће од стране корисника креирати сопствене функције и алате за посебне корисничке потребе, што *MATLAB* издваја од осталих сличних програма.

3.3. Програмски пакет *TOMLAB*

Постоји много добрих алата на располагању у области нумеричке анализе, операционих истраживања и оптимизације, али због различитих језика и система, као и недостатка стандардизације, коришћење наведених алата је компликовано и споро. У процесу оптимизације често се жели да се проблем сагледа графички, као и сам процес решавања. Тако се на пример у настави желе видети детаљи алгоритама или се жели обратити посебна пажња на различите алгоритамске кораке. Такође, код неискусних корисника постоји потреба да се на што једноставнији начин реши проблем, што се може постићи коришћењем система менија или графичким корисничким интерфејсом (*GUI*). Коришћењем *GUI* или система менија могу се веома лако променити параметри који утичу на процес решење. Некад постоји и потреба да се неки практичан проблем решава веома много пута, са незнатно различитим условима при решавању. То се нарочито односи приликом израде нових алгоритама када је потребно исте тестирати на хиљаде проблема, како би се у потпуности сагледале предности и мане новог алгорита.

Све горе наведено, као и недостатак добрих алата за оптимизацију мотивисало је рани развој неких оптимизационих решавача још 1990-их година. *TOMLAB* се појавио 1997. године [179], као посебан програмски пакет у *MATLAB* окружењу, за решавање теоријских и практичних проблема оптимизације. *TOMLAB* је флексибилан, лак за употребу, робустан и поуздан алат за све врсте оптимизационих проблема. Он даје лак приступ великом скупу стандардних тест и оптимизационих проблема. Осим тога, лако је дефинисати нове проблеме у *TOMLAB* формату и решавати их користећи било који од оптимизационих решавача (*Solver*).

TOMLAB решава оптимизационе проблеме из следећих области [140]:

- мешовито-целобројно програмирање,
- квадратно и нелинеарно програмирање,
- семидефинитно програмирање са билинеарним матричним ограничењима,
- нелинеарно програмирање,
- глобална оптимизација,
- нелинеарна и целобројна ограничења,
- глобална неконвексна оптимизација,
- метода линеарног и нелинеарног најмањег квадрата,
- оптимизација без ограничења,
- линеарно и квадратно програмирање,

- апроксимација емпиријских података,
- геометријско програмирање.

У предложеном алгоритму *TOMLAB* се користи за извршавање *MILP* прорачуна код решавања локалних мрежа по моделу који је дат у поглављу 3.2.1, релације (3.2)-(3.23).

3.4. Програмски пакет *MICROSOFT EXCEL*

MICROSOFT EXCEL је програм за табеларне калкулације софтверске куће *Microsoft*. *MICROSOFT EXCEL* углавном служи за решавање проблема математичког типа помоћу таблица и поља које је могуће повезивати различитим формулама и релацијама. Поред тога, овај програм се може користити и за израду једноставнијих база података. Такође, *MICROSOFT EXCEL* има развијену визуелизацију добијених резултата, који се могу представити таблицама, сликама или графиконима [180].

Могућности овог програма су следеће:

- нумеричка обрада података, табеларна израчунавања, израда предрачуна, математичка, статистичка, графичка и друга анализа резултата,
- израда спискова, шема и других докумената потребних за организацију података,
- графичко приказивање података применом великог броја различитих дијаграма,
- коришћење екстерних података, могућност размене података са другим програмима,
- употреба слика и графичких модела итд.

У предложеном алгоритму *MICROSOFT EXCEL* је кориштен за аквизицију података добијених од стране *MILP* прорачуна, као и за прорачуне потребне за алгоритам симулираног каљења датог у поглављу 3.2.2.

6. ЛИТЕРАТУРА

1. V. Neiname, "On development planning of electricity distribution networks", *Ph.D. dissertation*, Royal Institute of Technology, Stockholm, 2001.
2. European Commission, *European Technology Platform SmartGrids: Vision and Strategy for Europe's Electricity Networks of the Future*, 2006, dostupno: <http://www.smartgrids.eu/>.
3. European Commission, *European Technology Platform SmartGrids: Strategic Research Agenda for Europe's Electricity Networks of the Future*, 2007, dostupno: <http://www.smartgrids.eu/>.
4. European Commission, *European Technology Platform SmartGrids: Strategic Deployment Document for Europe's Electricity Networks of the Future*, 2010, dostupno: <http://www.smartgrids.eu/>.
5. Electric Power Research Institute (EPRI), *Report to NIST on the Smart Grid Interoperability Standards Roadmap*, 2009, USA.
6. National Institute of Standards and Technology (NIST), *NIST Framework and Roadmap for Smart Grid Interoperability Standards: Release 1*, 2009, USA.
7. G. T. Heydt, "The Next Generation of Power Distribution Systems", *IEEE Trans. on Smart Grid*, vol. 1, no. 3, pp. 225-235, 2010.
8. H.L. Willis, *Power Distribution Planning Reference book*, Marcel Dekker, Inc., 1997.
9. M.R. Garey, D. S. Johnson, *Computers and intractability: A guide to the theory of NP-Completeness*, W.H. Freeman, New York, 1979.
10. R.K. Ahuja, T.L. Magnati, J.B. Orlin, *Network flows-Theory, Algorithms, and Applications*, Prentice-Hall, Inc., New Jersey, USA, 1993
11. Ch. H. Papadimitriou, K. Steiglitz, *Combinatorial optimization : algorithms and complexity*, Dover Publications, Inc. New York, USA, 1998
12. Ž. Popović "Planiranje razvoja distributivnih mreža u prisustvu neizvesnosti", *Doktorska disertacija, FTN*, 2011
13. V. Miranda, L. M. Proenca, "Why Risk Analysis Outperforms Probabilistic Choice as the Effective Decision Support Paradigm for Power System Planning", *IEEE Trans. on Power Systems*, vol.13, no. 2, pp. 643-648, 1998.
14. V. Miranda, L. M. Proenca, "Probabilistic Choice vs. Risk Analysis – Conflicts and Synthesis in Power System Planning", *IEEE Trans. on Power Systems*, vol. 13, no. 3, pp. 1038-1043, 1998.
15. T. El-Ghazali, *Metaheuristics: from design to implementation*, John Wiley & Sons, New Jersey, USA, 2009.
16. E. Masud, "Distribution Planning: State of the Art and Extensions to Substation Sizing", *Electric Power Systems Research*, no.1, pp. 203 – 212, 1977/78.
17. H.L. Willis and J.E.D Northcote-Green, "Comparison of Several Computerized Distribution Planning Methods." *IEEE Trans on Power Systems*, vol. 104, no.1, pp. 233-240, 1985.
18. T. Gonen and I. Ramirez-Rosado, "Review of distribution system planning models: a model for optimal multi-stage planning", *IEE Proc. Gen. Trans. Distrib.*, Vol. 133, no. 7, pp. 397-408, 1986.
19. H. K. Temraz and V. H. Quintana, "Distribution system expansion planning models: an overview", *Electric Power Systems Research*, no. 26, pp. 61-70, 1993.
20. H.L. Willis, H. Tram, M.V. Engel and L. Finley, "Selecting and applying distribution optimization methods", *IEEE Computer Applications in Power*, pp. 12-17, 1996.
21. S.K. Khator and L.C. Leung, "Power distribution planning: a review of models and issues", *IEEE Trans. on Power Systems*, vol. 12, no. 3, pp. 1151-1159, 1997.
22. M. Vaziri, K. Tomsovic, T. Gonen, "Distribution expansion problem revisited - Part I: Categorical analysis and future directions", *Proc. Int. Assoc. Science and Technology for Development*, pp. 283-290, 2000.
23. R. Sempértégui, J. Bautista, R. Griño, J.P. Cubero, "Models and procedures for electric energy distribution planning: A review", *Proceedings of 15th Triennial World Congress (IFAC)*, 2002.

24. IEEE Task Force Report, "Load Representation for Dynamic Performance Analysis," *IEEE Trans. on Power Systems*, vol. 8, no. 2, pp. 472-482, 1993.
25. P. Linares, "Multiple Criteria Decision Making and Risk Analysis as Risk Management Tools for Power Systems Planning", *IEEE Trans. on Power Systems*, vol. 17, no. 3, pp. 895-900, 2002.
26. M.E.El-Hawary, *Electric Power Applications of Fuzzy Systems*, IEEE PRESS, New York, 1998.
27. I. J. Ramirez-Rosado, J. L. Bernal-Agustin, "Genetic Algorithms Applied to the Design of Large Power Distribution Systems", *IEEE Trans. on Power Systems*, vol. 13, no. 2, pp. 696-703, 1998.
28. I. J. Ramírez-Rosado, J. L. Bernal-Agustín, "Reliability and Costs Optimization for Distribution Networks Expansion Using an Evolutionary Algorithm", *IEEE Trans. on Power Systems*, vol. 16, no. 1, pp. 111-118, 2001.
29. D. I. Sun, D.R. Farris, P.J. Cote, R.R. Shoults and M.S. Chen, "Optimal distribution substation and primary feeder planning via fixed charge network formulation", *IEEE Trans. on Power Apparatus and Systems*, vol. 101, no. 3, pp. 602-609, 1982.
30. I.J. Ramirez-Rosado, T. Gonen, "Pseudodynamic planning for expansion of power distribution systems", *IEEE Trans. on Power Systems*, vol. 6, no. 1, pp. 245-253, 1991.
31. W.H. Quintana, H.K. Temraz, K.W. Hipel, "Two-stage power system distribution planning algorithm", *IEE Proc-C*, vol. 104, no. 1, pp. 17-29, 1993
32. U.G.W. Knight, "The Logical Design of Electrical Networks Using Linear Programming Methods", *Proc. IEE*, vol. 107 A, pp. 306-316, 1960.
33. M. J. Carson, G. Cornfield, "Design of Low-Voltage Distribution Networks", *IEE Proc. Gen. Trans. Distrib.*, vol. 120, no. 5, pp. 585-592, 1973.
34. Dale M. Crawford, Stewart B. Holt, "A Mathematical Optimization Technique for Locating and Sizing Distribution Substations and Deriving their Optimal Service Areas", *IEEE Trans. on Power Apparatus and Systems*, vol. 94, no. 2, pp. 230-235, 1975.
35. M. J. Juricek, et al., "Transportation Analysis of An Electric Power Distribution System", *IEEE Proceedings*, no. A 76 052-1, pp. 1-8, 1976.
36. D.L. Wall, G.L. Thompson, J.E.D. Northcote-Green, "An optimization Model for planning Radial Distribution Networks". *IEEE Trans. on Power Apparatus and Systems*, vol. 98, no. 6, pp.1061-1067, 1979.
37. M.A. Farraga, M.M. El-Metwallya, M.S. El-Bagesb, "A new model for distribution system planning" *Electrical Power and Energy Systems*, no. 21, pp. 523-531, 1999.
38. E. Masud, "An interactive procedure for sizing and timing distribution substations using optimization techniques", *IEEE Trans. on Power Apparatus and Systems*, vol. PAS-93, no. 5, pp. 1281-1286, 1974.
39. R.N. Adams, M.A. Laughton: "Optimal planning of power networks using mixed-integer programming. Part 1: Static and time phased network synthesis", *IEE Proc. Gen. Trans. Distrib.*, no.2, pp. 139-147, 1974.
40. G.L.Thompson, D.L. Wall, "A branch and bound model for choosing optimal substation locations", *IEEE Trans. on Power Apparatus and Systems*, vol. 100, no.5, pp. 2683-2688, 1981.
41. D.I. Sun, D.R. Farris, P.J. Cote, R.R. Shoults, M.S. Chen, "Optimal distribution substation and primary feeder planning via the fixed charge network formulation", *IEEE Trans. on Power Apparatus and Systems*, 1982, vol. 101, no. 3, pp. 602-609, 1982.
42. T. Gonen, B.L. Foote, "Distribution-system planning using mixed-integer Programming", *IEE Proc. Gen. Trans. & Distrib.*, vol. 2, pp. 70-79, 1981.
43. K.S. Hindi, A. Brameller, : "Design of low-voltage distribution networks: a mathematical programming method", *IEE Proc. Gen. Trans. Distrib.*, no. 1, pp. 54-58, 1977.
44. T.H. Fawzi, K.F. Ali, S.M. El-Sobki, "A new planning model for distribution systems", *IEEE Trans. on Power Apparatus and Systems*, vol.102, no.9, pp. 3010-3017, 1983.
45. M.A. El-Kady, "Computer-aided planning of distribution substation and primary feeders", *IEEE Trans. on Power Apparatus and Systems*, vol-103, no.6, pp. 1183-1189, 1984.

46. Ž. Popovic, D.S.Popovic, M.D.Prica: "A Model for Development Planning of Distribution Networks", *Proceedings of the 36th Universities Power Engineering Conference UPEC 2001*, pp. 44-50, 2001.
47. Ž. Popović, D.S.Popović, M.D.Prica: "Prilog metodama za planiranje razvoja distributivnih mreža", *Zbornik radova sa 25. savetovanja JUKO CIGRE*, pp. 43-49, 2001.
48. S. S. Sharif, M.M.A. Salama, A. Vannelli, "Optimal Model for Future Expansion of Radial Distribution Networks Using Mixed Integer Programming", *Proceedings of the Canadian Conference on Electrical and Computer Engineering*, pp. 152-155, 1994.
49. P. C. Paiva, H. M. Khodr, J. A. Domínguez-Navarro, J. M. Yusta, A. J. Urdeneta, "Integral Planning of Primary– secondary Distribution Systems Using Mixed Integer Linear Programming", *IEEE Trans. on Power Systems*, vol. 20, no. 2, 2005.
50. M. Ponnaikko, K.S. Prakasa Rao: "Aa Approach to Optimal Distribution System Planning Through Conductor Gradation", *IEEE Trans. on Power Apparatus and Systems*, vol.101, no. 6, pp. 1735 – 1742, 1982.
51. E. Díaz-Dorado, E. Miguez, J. Cidrás: "Design of Large Rural Low-Voltage Networks Using Dynamic Programming Optimization", *IEEE Trans. on Power Systems*, vol. 16, no. 4, pp. 898-903, 2001.
52. N. G. Boulaxis, M. P. Papadopoulos: "Optimal Feeder Routing in Distribution System Planning Using Dynamic Programming Technique and GIS Facilities", *IEEE Trans. on Power Systems*, vol. 17, no. 1, pp. 242 – 247, 2002.
53. E. Díaz-Dorado, J. C. Pidre: "Optimal Planning of Unbalanced Networks Using Dynamic Programming Optimization", *IEEE Trans. on Power Systems*, vol. 19, no. 4, pp. 2077 – 2085, 2004.
54. Ž. Popovic, D. S. Popovic: "A Novel Decomposition Procedure for Distribution Network Planning", *Proceedings of the 38th Universities Power Engineering Conference UPEC 2003*, 2003..
55. Ž. Popovic, D. S. Popovic: "A Dynamic Programming Based Procedure for Distribution Network Planning", *Proceedings of the 1th regional conference on electricity distribution JUKO CIRED*, 2004.
56. S. K. Goswami, "Distribution system planning using branch exchange technique", *IEEE Trans. on Power Systems*, vol. 12, no. 2, pp. 718-723, 1997.
57. K. Nara, T. Satoh, K. Aoki, M. Kitagawa, "New approximate optimization for distribution system planning", *IEEE Trans. on Power Systems*, vol. 5, no. 1, pp. 126 – 132, 1990.
58. K. Aoki, K. Nara, T. Satoh, M. Kitagawa, K. Yamanaka, "Multi-Year expansion planning for distribution systems", *IEEE Trans. on Power Systems*, vol. 6, no. 6, pp. 952 – 958, 1991.
59. K. Nara, T. Satoh, K. H. Kuwabara, K. Aoki, M. Kitagawa "Distribution system expansion planning by multi-stage branch exchange", *IEEE Trans. on Power Systems*, vol. 7, no. 1, pp. 208 – 214, 1992.
60. K. Nara H. Kuwabara M-Kitagawa K. Ohtaka, "Algorithm for Expansion Planning in Distribution Systems Taking Faults into Consideration", *IEEE Trans. on Power Systems*, vol. 9, no. 1, pp.324-330,1994.
61. H. Kuwabara, K. Nara, "Multi-Year and Multi-State Distribution Systems Expansion Planning by Multi-Stage Branch Exchange", *IEEE Trans. on Power Delivery*, vol. 12, no. 1, pp. 457 – 463, 1997.
62. G.J.Pebonis, M.P. Palpadopoulos, "New Dynamic, Branch Exchange Method for Optimal Distribution System Planning", *IEE Proc. Gener. Transm. Distrib.*, vol. 144, no. 3, pp. 333-339, 1997.
63. E. Miguez, J. Cidrás, E. Diaz-Dorado, J. L. Garcia-Dornelas, "An improved branch-exchange algorithm for large-scale distribution network planning", *IEEE Trans. on Power Systems*, Vol. 17, No. 4, pp. 931 – 936, 2002.
64. D. Popović, D. Bekut, V. Treskanica, *Specijalizivani DMS algoritmi*, DMS Group, Novi Sad, 2004.
65. Ž. Popović, I. Janković, V. Marijanović, „Planiranje razvoja distributivnih mreža velikih dimenzija primenom rekonfiguracije mreže i tehnike izmene grana“, *Zbornik radova sa šestog savetovanja o elektrodistributivnim mrežama CIRED Srbija*, 2008.

66. C.W. Hasselfield, P. Wilson, L. Penner, M. Lau, A.M. Gole, "An Automated Method for Least Cost Distribution Planning", *IEEE Trans. on Power Delivery*, vol. 5, no. 2, pp.1188-1194, 1990.
67. V. Parada, J. A. Ferland, M. Arias, K. Daniels, "Optimization of Electrical Distribution Feeders Using Simulated Annealing", *IEEE Trans. on Power Delivery*, vol. 19, no. 3, pp.1135-1141, 2004.
68. M. Gandomkar, M. Vakilian, M. Ehsan, "A Combination of Genetic Algorithm and Simulated Annealing for Optimal DG Allocation in Distribution Networks", *Proceedings of the Canadian Conference on Electrical and Computer Engineering*, pp. 645-648, 2005.
69. A. C. N. Gonzalo, B. Julian, "Optimal Feeder Configuration in Expansion Planning using Simulated Annealing", *Proceedings of the IEEE Region 10 Conference (TENCON 2007)*, 2007.
70. J. M. Nahman, D. M. Perić, "Optimal Planning of Radial Distribution Networks by Simulated Annealing Technique", *IEEE Trans. on Power Systems*, vol. 23, no. 2, pp. 790-795, 2008.
71. Z. Tu, X. Zeng, "The Study on the Electric Network Planning Problem Based on the Ant Colony- Simulated Annealing Algorithm", *Proceedings of the Third International Conference on Genetic and Evolutionary Computing*, 2009.
72. A. I. Aly, Y. G. Hegazy, M. A. Alsharkawy, "A Simulated Annealing Algorithm for Multi-Objective Distributed Generation Planning", *IEEE Power and Energy Society General Meeting*, pp. 1-7, 2010.
73. M. Gandomkar, H. B. Toulabi, "Investigation of Simulated Annealing, Ant-Colony and Genetic Algorithms for Distribution Network Expansion Planning with Distributed Generation", *Proceedings of the 9th WSEAS Int. Conference on Instrumentation, Measurement, Circuits and Systems*, 2010.
74. A. I. Aly, Y. G. Hegazy, M. A. Alsharkawy, "A Simulated Annealing Algorithm for Multi-Objective Distributed Generation Planning", *IEEE Power and Energy Society General Meeting, 2010*.
75. Ž. Popovic, V. Kerleta, , "Expansion planning of distribution networks using simulated annealing technique", *Proceedings of the 21th conference on electricity distribution CIRED*, 2011.
76. A. Gallego, R. Romero, A. J. Monticelli, " Tabu Search Algorithm for Network Synthesis", *IEEE Trans. on Power Systems*, vol. 15, no. 2, pp. 490-495, 2000.
77. H. Mori, Y. Iimura, "Application of Parallel Tabu Search to Distribution Network Expansion Planning with Distributed Generation", *Proceedings of the IEEE Bologna PowerTech Conference*, pp.23-26, 2003.
78. H. Mori, Y. Iimura, "An improved Tabu Search Approach to Distribution Network Expansion Planning Under New Environment", *Proceedings of the International Conference on Power System Technology (POWERCON)*, pp. 981-986, 2004.
79. I. J. Ramírez-Rosado, J. A. Domínguez-Navarro, "Possibilistic Model Based on Fuzzy Sets for the Multiobjective Optimal Planning of Electric Power Distribution Networks," *IEEE Trans. on Power Systems*, vol. 19, no. 4, pp. 1801–1810, 2004.
80. H. Mori, Y. Yamada, "Two-Layered Neighborhood Tabu Search for Multiobjective Distribution Network Expansion Planning" *Proceedings of the International Symposium on Circuits and Systems (ISCAS)*, pp. 3706-3709, 2006.
81. I. J. Ramírez-Rosado, J. A. Domínguez-Navarro, "New Multiobjective Tabu Search Algorithm for Fuzzy Optimal Planning of Power Distribution Systems", *IEEE Trans. on Power Systems*, vol. 21, no. 1, pp. 224-233, 2006.
82. R. Romero, J. R. S. Mantovani, A. M. Cossi, "Planning and Projects of Secondary Electric Power Distribution Systems" *IEEE Trans. on Power Systems*, vol. 24, no. 3, pp. 1599-1608, 2009.
83. V. Miranda, J.V. Ranito and L.M. Proenca "Genetic algorithms in optimal multistage distribution network planning", *IEEE Trans. on Power Systems*, Vol. 9, No. 4, pp. 1927-1933, 1994.
84. E.-C. Yeh, S.S. Venekata, Z. Sumic, "Improved Distribution System Planning Using Computational Evolution", *IEEE Trans. on Power Systems*, vol. 11, no. 2, pp. 668-674, 1996.

85. W.M. Lin Y.S. Su, M.T. Tsay, "Genetic Algorithm for Optimal Distribution System Planning", *Proceedings of the International Conference on Power System Technology (POWERCON '98)*, pp. 241-245, 1998.
86. W.-M.Lin, C.-D.Yang, M.-T.Tsay, "Distribution system planning with evolutionary programming and a reliability cost model", *IEE Proc. Gener. Transm. Distrib.*, vol. 147, no. 6, pp. 336-341, 2000.
87. V. Neimane, "Distribution Network Planning Based on Statistical Load Modeling Applying Genetic Algorithms and Monte-Carlo Simulations", *Proceedings of the IEEE Porto Power Tech Conference*, 2001.
88. E. Díaz-Dorado, J. Cidrás, E. Míguez, "Application of Evolutionary Algorithms for the Planning of Urban Distribution Networks of Medium Voltage", *IEEE Trans. on Power Systems*, vol. 17, no. 3, pp. 879-884, 2002.
89. G. Duan, Y.Yu, "Problem-specific genetic algorithm for power transmission system planning," *Electric Power System Research*, vol. 61, pp. 41–50, 2002.
90. E. Díaz-Dorado, J. C. Pidre, E. M. García, "Planning of Large Rural Low-Voltage Networks Using Evolution Strategies", *IEEE Trans. on Power Systems*, vol. 18, no. 4, pp. 1594-1600, 2003.
91. A. M. Cossi, R. Romero, J. R. S. Mantovani, "Planning of Secondary Distribution Circuits Through Evolutionary Algorithms", *IEEE Trans. on Power Delivery*, vol. 20, no. 1, pp. 205 – 213, 2005.
92. F. Mendoza, J. L. Bernal-Agustín, J. A. Domínguez-Navarro, "NSGA and SPEA Applied to Multiobjective Design of Power Distribution Systems", *IEEE Trans. on Power Systems*, vol. 21, no. 4, pp. 1938 – 1945, 2006.
93. E. G. Carrano, L. A. E. Soares, R. H. C. Takahashi, R. R. Saldanha, O. M. Neto, "Electric Distribution Network Multiobjective Design Using a Problem-Specific Genetic Algorithm", *IEEE Trans. on Power Delivery*, vol. 21, no. 2, pp. 995-1005, 2006.
94. Y.F. Dong, J.H. Gu, N.N. Li, X.D. Hou, W.L. Yan, "Combination of Genetic Algorithm and Ant Colony Algorithm for Distribution Network Planning" *Proceedings of the Sixth International Conference on Machine Learning and Cybernetics*, pp. 999 – 1002, 2007.
95. N. Rajkumar, T. Vekara, J. T. Alander, "A Review of Genetic Algorithms in Power Engineering", *Proceedings of the 13th Finnish Artificial Intelligence Conference STeP* , pp. 15–32,2008.
96. E.G. Carrano, R.T.N. Cardoso, R.H.C. Takahashi, C.M. Fonseca, O.M. Neto, "Power distribution network expansion scheduling using dynamic programming genetic algorithm", *IET Gener. Transm. Distrib.*, vol. 2, no. 3, pp. 444–455, 2008.
97. T. Konga, H. Chenga, Z. Hua, L. Yaob, "Multiobjective planning of open-loop MV distribution networks using ComGIS network analysis and MOGA", *Electric Power Systems Research*, no. 79, pp. 390–398, 2009.
98. S. Najafi, S. H. Hosseinian, M. Abedi, A. Vahidnia, S. Abachezadeh, "A Framework for Optimal Planning in Large Distribution Networks", *IEEE Trans. on Power Systems*, vol. 24, no. 2, pp. 1019 – 1028, 2009.
99. M.T.Ponce de Leao, A. Matos, "Multicriteria Distribution Network Planning Using Simulated Annealing", *International Trans. in Operational research*, vol. 6, pp. 377-391, 1999.
100. J. F. Gómez, H. M. Khodr, P. M. De Oliveira, L. Ocque, J. M. Yusta, R. Villasana, A. J. Urdaneta, "Ant Colony System Algorithm for the Planning of Primary Distribution Circuits", *IEEE Trans. on Power Systems*, vol. 19, no. 2, pp. 996 – 1004, 2004.
101. M. G. Ippolito, E. R. Sansevenno, F. Vuinovich, "Multiobjective Ant Colony Search Algorithm For Optimal Electrical Distribution System Strategical Planning", *Proceedings of the Congress on Evolutionary Computation (CEC)*, pp. 1924-1931, 2004.
102. L. F. Ibrahim, O. Metwaly, A. Kapel, A. Ahmed, "Enhancing the Behavior of the Ant Algorithms to Solving Network Planning Problem", *Proceedings of the 2005 Third ACIS International Conference on Software Engineering Research, Management and Applications (SERA'05)*, 2005.
103. Z. Jia-mei, R. Hong-e, "A New Idea for Power Distribution Network Planning Using Ant Colony Optimization", *Proceedings of the 5th International Conference on Wireless Communications, Networking and Mobile Computing*, 2009.

104. J.M. Alvarado, E.V. Alvarado, M.A. Arévalo, S.P. Quituisaca, J.F. Gomez, P.M. De Oliveira-De Jesus, "Ant Colony Systems Application for Electric Distribution Network Planning", *Proceedings of the 15th International Conference on Intelligent System Applications to Power Systems (ISAP)*, 2009.
105. F. Yang, H. Rong, C. Jia-lin, M. Ling-he, "Multi-objective Power Network Planning Based on Improved Pareto Ant Colony Algorithm", *Proceedings of the Asia-Pacific Power and Energy Engineering Conference*, 2009.
106. Y.-q. Li, L. Wang, H.-l. Xie, Q. Xie, "Distribution Network Optimal Planning Based on Clouding Adaptive Ant Colony Algorithm", *Proceedings of the Asia-Pacific Power and Energy Engineering Conference*, 2009.
107. L. Aiguo, W. Huiiong, "Distribution Network Structure Planning Based On Ant Algorithm", *Proceedings of the 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE)*, 2010.
108. Z. Jia-mei, R. Hong-e, "Improved Ant Colony Algorithm for Power Distribution Network Planning", *Proceedings of the 2nd International Conference on Advanced Computer Control (ICACC)*, 2010.
109. K. Hoffman, M. Padberg, "Combinatorial and integer optimization", dostupno: <http://iris.gmu.edu/~khoffman/papers/newcomb1.html>
110. M. Junger, T. Liebling, D. Naddef, G. Nemhauser, W. Pulleyblank, G. Reinelt, G. Rinaldi, L. Wolsey, *50 Years of Integer Programming 1958–2008: From the Early Years to the State-of-the-Art*, Springer-Verlag, Berlin, Germany, 2010.
111. R. Bellmann, *Dynamic programming*, Princeton University Press, New Jersey, 1957.
112. G.R. Walsh, *Methods of optimization*, London, John Wiley & Sons, 1975.
113. M. A. Trick, "A tutorial on dynamic programming", dostupno: <http://mat.gsia.cmu.edu/classes/dynamic/dynamic.html>
114. V. Chankong, Y.Y. Haimes, "Multiobjective Decision Making: Theory and Methodology", *North-Holland, Amsterdam, Netherlands*, 1983.
115. Matthias Ehrgott, "Multicriteria Optimization", *Springer-Verlag*, Berlin, Germany, 2005.
116. S. Opricović, "Višekriterijumska optimizacija", *Naučna knjiga*, Beograd, 1986.
117. Benvindo Rodrigues Pereira Junior, Antonio Marcos Cossi, Javier Contreras, José Roberto Sanches Mantovani, "Multiobjective multistage distribution system planning using tabu search", *IET Generation, Transmission & Distribution*, ISSN 1751-8687, 2013
118. Meysam Shamshiri, Chin Kim Gan, Kamaruzaman Jusoff, Ihsan Jabbar Hasan, Mohd Ruddin, AbGhani and Mariana Yusoff : "Using Particle Swarm Optimization Algorithm in the Distribution System Planning", *Australian Journal of Basic and Applied Sciences*, 7(3): 85-92, 2013.
119. Rasoul Mohammadi tab, Seyed-Jalal Seyed-Shenava, Khalil Valipour, "Novel Particle Swarm Optimization approach for multi objective planning of power distribution network under load uncertainty", *International Research Journal of Applied and Basic Sciences*, Available online at www.irjabs.com, ISSN 2251-838X / Vol, 3 (12): 2408-2419, 2012.
120. S. Ganguly, N. C. Sahoo and D. Das, "Mono- and multi-objective planning of electrical distribution networks using particle swarm optimization", *Applied Soft Computing* (Published by Elsevier), Vol. 11, No. 2, pp. 2391–2405, 2011.
121. J. Kennedy and R. C. Eberhart, "Swarm Intelligence", *Morgan Kaufmann, San Francisco, CA*, 2001.
122. Boyd R, Richerson PJ. "Culture and the evolutionary process", *University of Chicago Press*; 1985.
123. Reynolds C. Flocks, herds, and schools: "A distributed behavioral model", *Computer Graphics*; 21(4):25–34, 1987.
124. K. Y. Lee, M. A. El-Sharkawi, *Modern Heuristic Optimization Techniques – Theory and Applications to Power Systems*, John Wiley & Sons, New Jersey, USA, 2008.
125. C. Blue, M. J. B. Aguilera, A. Roli, M. Sampels, *Hybrid Metaheuristics – An Emerging Approach to Optimization*, Springer-Verlag, Berlin, Germany, 2008.
126. Blum C., Dorigo M. "The hyper-cube framework for ant colony optimization", *IEEE Trans Systems Man Cybernetics B*; 34(2):1161–1172, 2004.

127. N. Stanković, Seminarski rad, "Metaheuristika – upoređivanje algoritma optimizacije mravljim kolonijama (ACO) sa algoritmom optimizacije rojevima čestica (PCO)", Matematički fakultet, Beograd, 2013.
128. M. Stanković, Master rad, "Rešavanje nekih problema kombinatorne optimizacije algoritmom tabu pretraživanja", Matematički fakultet, Beograd, 2013.
129. Glover F, Kelly JP, Laguna M. "Genetic algorithms and tabu search: Hybrids for optimization", *Computers Operations Res*; 22(1):111–134,1994,
130. Glover F, Taillard E, de Werra D. "A users guide to tabu search", *Ann Operations Res*; 41:3–28,1993.
131. Glover F, Laguna M. "Tabu Search", *Boston: Kluwer Academic Publishers*; 1997.
132. Reeves CR. "Modern heuristic techniques for combinatorial problems", *New York: McGraw-Hill*; 1995.
133. Glover F. "Tabu search fundamental and uses", *Graduate School of Business, University of Colorado*; 1995.
134. R. C. Lotero, J. Contreras, Distribution system planning with reliability, *IEEE Trans. Power Deliv.*, 26(4) 2552-2562,2011.
135. M. Lavorato, J. F. Franco, M. J. Rider, R. Romero, Imposing radiality constraints in distribution system optimization problems, *IEEE Trans. Power Syst.*, 27(1) 172-180,2012.
136. T. Gonen, *Electric power distribution system engineering*, New York, McGraw-Hill, 1986
137. Ž. N. Popović, D. S. Popović, A novel decomposition procedure for distribution network planning, *Proc.38th Univer. Power Eng. Conf.(UPEC 2003)*, Thessaloniki, Greece,2003.
138. D.S.Popović, R.M.Ćirić, A multi-objective algorithm for distribution networks restoration, *IEEE Trans. Power Deliv.*, 16 (3), 1134-1141.1999.
139. S. R. Thangiah, I. H. Osman, T. Sun, Hybrid genetic algorithm, simulated annealing and tabu search methods for vehicle routing problems with time windows, *Comput. sci. dep., Slippery Rock Univ., Tech. rep., SRU CpSc-TR-94-27* 69, 1994.
140. <http://tomlab.biz/tomlab>
141. R. A. Jabr, Polyhedral formulations and loop elimination constraints for distribution network expansion planning, *IEEE Trans. Power Syst.* 28 (2), 1888-1897,2013.
142. M. Pecić, Master rad, "Robusna rešenja u višekriterijumskoj optimizaciji", Matematički fakultet, Beograd, 2009.
143. Goldberg DE. "Genetic algorithms in search, optimization and machine learning", *Reading, MA: Addison-Wesley*; 1989.
144. Beasley D, Bull DR, Martin RR. "An overview of genetic algorithms: Part 1, fundamentals", *University Computing*; 15(2):58–69,1993.
145. Beasley D, Bull DR, Martin RR. "An overview of genetic algorithms: Part 2, research topics", *University Computing*; 15(4):170–181,1993.
146. Whitley D. "A genetic algorithm tutorial", *Statistics Computing*; 4:65–85,1994.
147. Aarts E, Korst J. "Simulated annealing and Boltzmann", New York: *John Wiley & Sons*; 1989.
148. Dorigo, M., & Stützle, T. "Ant Colony Optimization", *MIT Press, Cambridge, MA*, 2004.
149. Wolpert, D.H., Macready, W.G. "No Free Lunch Theorems for Optimization," *IEEE Transactions on Evolutionary Computation* 1, 67.1997.
150. E. G. Carrano, F. G. Guimaraes, R. H. C. Takahashi, O. M. Neto, F. Campelo, "Electric Distribution Network Expansion Under Load-Evolution Uncertainty Using an Immune System Inspired Algorithm", *IEEE Trans. on power systems*, vol.22, no. 2, pp. 851- 861, 2007.
151. Hassan R, Cohanin B, DeWeck OL, Venter G. 2005. "A comparison of particle swarm optimization and the genetic algorithm", in: *Proceedings of the First AIAA Multidisciplinary Design Optimization Specialist Conference*, pp. 18–21., Austin, Texas, 2005.
152. F. R. B. Cruz, M. G. Smith, and G. R. Mateus, "Solving to optimality the uncapacitated fixed-charge network flow problem", *Comput. Oper.Res.*, vol. 25, no. 1, pp. 67–81, 1998.
153. G. Duan and Y. Yu, "Power distribution system optimization by an algorithm for capacitated Steiner tree problems with complex-flows and arbitrary cost functions", *Elect. Power Energy Syst.*, vol. 25, pp. 515–523, 2003.
154. M. Čupić, Skripta, "Prirodom inspirirani optimizacijski algoritmi", 2012.

155. Michalewicz Z, Schoenauer M. "Evolutionary algorithms for constrained parameter optimization problems" *Evol Computat*, 4(1):1–32,1996.
156. Gen M, Cheng R. "A survey of penalty techniques in genetic algorithms", *Proc. 3rd IEEE Conf. on Evolutionary Computation. Piscataway, NJ: IEEE Press*; 804–809,1996.
157. C. M. Fonseca and P. Fleming, "An overview of evolutionary algorithms in multiobjective optimization", *Evol. Comput.*, vol. 3, no. 1, pp. 1–16, 1995.
158. C. A. C. Coello, "An updated survey of GA-based multiobjective optimization techniques", in *Proc. ACM Computing Surveys*, vol. 32, pp. 109–143,2000.
159. D. K. Smith and G. A. Walters, "An evolutionary approach for finding optimal trees in undirected networks" *Eur. J. Oper. Res.*, vol. 120, pp. 593–602, 2000.
160. K. Deb, A. Pratap, S. Agrawal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II", *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.
161. N. Srinivas and K. Deb, "Multiobjective optimization using nondominated sorting in genetic algorithms," *Evol. Comput.*, vol. 2, no. 3, pp.221–248, 1994.
162. M. S. Bazaraa and J. J. Jarvis, "Linear Programming and Network Flows",*New York: Wiley*, 1977.
163. V. Chankong and Y.Y. Haimes, "Multiobjective Decision Making: Theory and Methodology",*Amsterdam, The Netherlands: Elsevier*, 1983.
164. M. Dorigo, L.M. Gambardella, "Ant colony system: A cooperative learning approach to the traveling salesman problem", *IEEE Trans. Evol. Comput.*, vol. 1, pp. 29-41, Apr. 1997.
165. Garver LL. Transmission network estimation using linear programming. *IEEE Trans Power App Syst* 1970; PAS-89:1688–1697.
166. Gallego RA, Romero R, Monticelli A. Tabu search algorithm for network synthesis. *IEEE Trans Power Systems* 2000; 15(2):490–495.
167. Kirkpatrick S, Gelatt CD, Jr, Vecchi M. "Optimization by simulated annealing", *Science*; 220(4598):498–516,1983.
168. Cerny V. "Thermodynamical approach to the Traveling Salesman Problem: An efficient simulation algorithm", *J Optim Theory Appl*; 45(1):41–51,1985.
169. Van Laarhoven PJM, Aarts EH. "Simulated annealing: theory and applications", *Holland: Dordrecht, D. Reidel Publishing Company*; 1987.
170. K. Dowsland. "Simulated annealing" In C. Reeves, editor, "Modern Heuristic Techniques for Combinatorial Problems", pages 41_49, *USA, McGraw-Hill*,1995.
171. V. Rayward-Smith, I. Osman, C. Reeves, and G. Smith. "Modern Heuristic Search Methods", *John Wiley and Sons*, 1996.
172. Reeves CR. "Modern heuristic techniques for combinatorial problems", *New York: JohnWiley & Sons*; 1993.
173. T. C. Hu, A. B. Kahng, and C.-W. A. Tsao. "Old bachelor acceptance: A new class of nonmonotone threshold accepting methods", *ORSA Journal on Computing*, 7(4):417–425, 1995.
174. B. Hajek and G. Sasaki. "Simulated annealing: To cool or not", *Systems and Control Letters*, 12:443–447, 1989.
175. L. Ingber. "Adaptive simulated annealing", *Control and Cybernetics*, 25(1):33–54, 1996.
176. Gallego RA, Alves AB, Monticelli A, Romero R. "Parallel simulated annealing applied to long term transmission network expansion planning", *IEEE Trans Power Systems*; 12(1):181–188,1997.
177. D. Popovic, D. Bekut, V. Treskanica: "Specijalizovani DMS algoritmi", *DMS Group, Novi Sad*, 2004.
178. <http://www.mathworks.com>
179. Kenneth Holmstrom, "Practical Optimization with the TOMLAB Environment in Matlab", *Center for Mathematical Modeling Department of Mathematics and Physics, MalardalenUniversity, Sweden*, 2001.
180. <http://office.microsoft.com>