



Универзитет „Св. Кирил и Методиј“ во Скопје
**ФАКУЛТЕТ ЗА ИНФОРМАТИЧКИ НАУКИ И
КОМПЈУТЕРСКО ИНЖЕНЕРСТВО**

Оптимизација на производство на Електрична Енергија

Благој Атанасовски

2018-01-16

1 Претставување на проблемот

Да претпоставиме дека на територијата на една земја, постојат N населени места и M електрани. Секое од населените места во даден момент има потрошувачка/потреба од $Potrosuvacka[i]$ MWh електрична енергија каде $0 < i \leq N$. Секоја од електраните има можност во даден момент да произведе максимум $MaxProiz[j]$ MWh електрична енергија (термо електрана на јаглен) каде $0 < j \leq M$. Електраните може да го контролираат производството на енергија од 0% до 100%. За секоја од електраните имаме $MaxCena[j]$ што ја претставува цената да работи електраната во полн капацитет, цената е линеарно зависна од капацитетот со кој работи електраната со што на капацитет 0% цената е 0. Населено место е поврзано со далековод со барем една од електраните, максимум со сите.

За секој од далеководите помеѓу населено место i и електрана j познат е коефициент на загуба, $Zaguba[i][j]$ кој го дава процентот на изгубена моќност, при што $0 \leq Zaguba[i][j] < 100$, и доколку $Zaguba[i][j] = 0$ тогаш нема далековод кој ги поврзува i и j . Органичувања кои важат се:

- Електраните под полн капацитет можат да ги задоволат потребите на населените места
- Поставеноста на далеководите гарантира дека секогаш ќе може да се испорача енергија до сите населени места

Целта е, за дадени:

- Моќност на електричните центри
- Потрошувачка на населните места
- Должината на далеководите кои ги поврзуваат електраните со населените места
- Загубата на секој од далеководите

Да се најде капацитетот со кој треба да работат електраните, така што цената за производство и транспорт на електричната енергија ќе биде што е можно понизок.

2 Решение

Зададениот проблем е од оптимизациска природа. Со тоа што целта е да се минимизира цената за производство на електрична енергија при што цената ја дефинираме како

$$cena = \sum_{i=1}^M OdbranKapacitet[i] \cdot MaxCena[i]$$

каде $OdbranKapacitet[j]$ е капацитетот со кој треба да работи електраната j во решението. Ако дефинираме $Isporaka[i][j]$ = Процентот од енергијата генерирана во електраната j испорачана до населено место i , тогаш важат ограничувањата:

$$\forall i, 1 \leq i \leq N,$$

$$\sum_{j=1}^M Isporaka[i][j] \cdot OdbranKapacitet[j] \cdot MaxProiz[j] \cdot Zaguba[i][j] \geq Pot[i]$$

$$\forall j, 1 \leq j \leq M,$$

$$\sum_{i=1}^N Isporaka[i][j] \cdot OdbranKapacitet[j] \cdot MaxProiz[j] \cdot Zaguba[i][j] \leq MaxProiz[j]$$

$$0 \leq Isporaka[i][j] \leq 100$$

3 Методологија за оценување на решенијата

Да се најде brute-force решение со испробување на сите комбинации претставува проблем со $O(100^{n \cdot n})$ комплексност, бидејќи за секоја вредност од матрицата $Isporaka$ има 100 вредности.

За лесна проверка, ќе бидат земени:

1. Тривијални примери

- (a) 1 електрана со n населени места, сите се поврзани, еднаква загуба
- (b) 1 населено место со m електрани, сите се поврзани, еднаква загуба
- (c) 1 населено место со m електрани, некои се неповрзани, еднаква загуба

2. Едноставни примери

- (a) 2 населени места, 2 електрани, сите се поврзани меѓу себе, иста загуба
- (b) 3 населени места, 3 електрани, сите се поврзани, различна загуба
- (c) мали m и n , постојат неповрзани населени места со електрани, различна загуба

Дополнително ќе бидат следени влијанието на:

- ратата на мутација
- ратата на вркстување
- големината на популацијата

врз пронајдениот минимум и брзината на конвергенција.

3.1 1.a

Во примерот 1.a цел на оптимизација е распределбата на електрична енергија од една електрана, кон n населени места. Сите населени места се поврзани со електраната и процентот на загуба на енергија е еднаков за сите.

Влезни аргументи:

- $m = 1$
- $n = 10$
- $Potrosuvacka = [10 \ 20 \ 50 \ 100 \ 30 \ 40 \ 100 \ 5 \ 10 \ 20]$
- $MaxProiz = [500]$
- $MaxCena = [1000]$
- $\forall i 1 \leq i \leq n, \forall j 1 \leq j \leq m, Zaguba[i][j] = 10$

Очекуван излез:

- $PredlogDistribucija = [3 \ 5 \ 12 \ 23 \ 7 \ 9 \ 23 \ 2 \ 3 \ 5]$
- $OdbranKapacitet = [92]$
- $cena = 0.92 \cdot 1000 = 920$

3.2 1.b

Во примерот 1.b цел на оптимизација е распределбата на електрична енергија до 1 населено место а на располагање се m електрани. Сите населени места се поврзани со електраната и процентот на загуба на енергија е еднаков за сите.

Влезни аргументи:

- $m = 5$
- $n = 1$
- $Potrosuvacka = [100]$
- $MaxProiz = [20 \ 30 \ 10 \ 50 \ 100]$
- $MaxCena = [10 \ 20 \ 5 \ 10 \ 120]$
- $\forall i 1 \leq i \leq n, \forall j 1 \leq j \leq m, Zaguba[i][j] = 10$

Очекуван излез:

- $PredlogDistribucija = [100 \ 98 \ 98 \ 100 \ 1]$
- $OdbranKapacitet = [100 \ 98 \ 98 \ 100 \ 1]$
- $cena = 10 + 20 + 5 + 10 + 0.01 \cdot 120 = 46.2$

3.3 1.c

Во примерот 1.c цел на оптимизација е распределбата на електрична енергија до 1 населено место а на располагање се m електрани. Некои од електраните не се поврзани со електраната и процентот на загуба е еднаков за сите.

Влезни аргументи:

- $n = 1$
- $m = 5$
- $Potrosuvacka = [100]$
- $MaxProiz = [20 \ 30 \ 10 \ 50 \ 100]$
- $MaxCena = [100 \ 5 \ 50 \ 10 \ 120]$
- $Zaguba[1][1] = 10, Zaguba[1][3] = 10, Zaguba[1][5] = 10$
- $Zaguba[1][2] = 0, Zaguba[1][4] = 0$ т.е. нема довод

Очекуван излез:

- $PredlogDistribucija = [10 \ 0 \ 100 \ 0 \ 100]$
- $OdbranKapacitet = [10 \ 0 \ 100 \ 0 \ 100]$
- $cena = 0.1 \cdot 100 + 0 + 50 + 0 + 100 = 234$

3.4 2.a

Во примерот 2.a цел на оптимизација е распределба на електрична енергија до 2 населени места од 2 електрани, и двете електрани се поврзани до 2те населени места и имаат иста загуба.

Влезни аргументи:

- $n = 2$
- $m = 2$
- $Potrosuvacka = [150 \ 200]$
- $MaxProiz = [250 \ 200]$
- $MaxCena = [120 \ 50]$
- $Zaguba = \begin{bmatrix} 10 & 10 \\ 10 & 10 \end{bmatrix}$

Очекуван излез:

- $PredlogDistribucija = \begin{bmatrix} 0 & 84 \\ 69 & 26 \end{bmatrix}$
- $OdbranKapacitet = [69 \ 100]$
- $cena = 0.69 * 120 + 1 * 50 = 132.8$

3.5 2.b

Во примерот 2.b цел на оптимизација е распределба на електрична енергија до 4 населени места од 4 електрани, секое од населените места има довод од само 2 електрани.

Влезни аргументи:

- $n = 4$
- $m = 4$
- $Potrosuvacka = [150 \ 200 \ 500 \ 300]$
- $MaxProiz = [250 \ 300 \ 1000 \ 400]$
- $MaxCena = [120 \ 50 \ 400 \ 100]$
- $Zaguba = \begin{bmatrix} 10 & 10 & 0 & 0 \\ 0 & 10 & 10 & 0 \\ 0 & 0 & 10 & 10 \\ 10 & 0 & 0 & 10 \end{bmatrix}$

Очекуван излез:

- $PredlogDistribucija = \begin{bmatrix} 0 & 56 & 0 & 0 \\ 0 & 41 & 10 & 0 \\ 0 & 0 & 50 & 14 \\ 0 & 0 & 0 & 84 \end{bmatrix}$
- $OdbranKapacitet = [0 \ 97 \ 60 \ 98]$
- $cena = 0 \cdot 120 + 0.97 \cdot 50 + 0.6 \cdot 400 + 0.98 \cdot 100 = 386.5$

3.6 2.c

Во примерот 3.c цел на оптимизација е распределба на електрична енергија до 6 населени места од 4 електрани, секое од населените места има довод од барем 2 електрани.

Влезни аргументи:

- $n = 7$
- $m = 4$
- $Potrosuvacka = [100 \ 100 \ 100 \ 100 \ 100 \ 100 \ 100]$
- $MaxProiz = [300 \ 150 \ 300 \ 150]$
- $MaxCena = [100 \ 80 \ 300 \ 150]$

$$\bullet \text{ Zaguba} = \begin{bmatrix} 0 & 10 & 10 & 10 \\ 10 & 0 & 10 & 10 \\ 10 & 10 & 0 & 10 \\ 10 & 10 & 10 & 0 \\ 0 & 0 & 5 & 5 \\ 5 & 5 & 0 & 0 \\ 10 & 10 & 10 & 10 \end{bmatrix}$$

Очекуван излез:

$$\bullet \text{ PredlogDistribucija} = \begin{bmatrix} 0 & 0 & 38 & 0 \\ 0 & 0 & 38 & 0 \\ 0 & 0 & 0 & 75 \\ 38 & 0 & 0 & 0 \\ 0 & 0 & 24 & 23 \\ 36 & 0 & 0 & 0 \\ 26 & 21 & 0 & 2 \end{bmatrix}$$

$$\bullet \text{ OdbranKapacitet} = [100 \quad 21 \quad 100 \quad 100]$$

$$\bullet \text{ cena} = 1 * 100 + 0.21 * 80 + 1 * 60 + 1 * 40 = 216.8$$

4 Имплементација на решението

Решенијата се имплементирани во програмскиот јазик Python 2.7, со користење на библиотеката PyEvolve. Оваа библиотечка вградува во себе готови имплементации на еволутивни алгоритми и дозволува нивно извршување со едноставно менување на параметрите. Во истата библиотека постојат веќе креирани едноставни репрезентации на хромозоми (1д низа, 2д низа, бинарен стринг) и оператори за нив, но овозможува и креирање на свои оператори.

4.1 Варијанта 1: Наивна имплементација

Првата имплементација претставува наивен обид ограничувањата дадени во описот на проблемот да се претстават само преку функцијата на целта, т.е. доколку некое од ограничувањата не исполнето тогаш цената се „казнува“ неколкукратно.

4.1.1 Претставување на хромозомите

Хромозомите ги претставуваме како 2 димензионална матрица:

$$\text{Predlog Distribucija}[i][j], 0 < i \leq N, 0 < j < M$$

го претставува делот од производството на електраната j (изразено во проценти) кој се доставува на населеното место i , $\forall i, \forall j, 0 \leq \text{PredlogDistribucija}[i][j] \leq$

100. Од иницијалното претставување на решението

$$\forall i, OdbranKapacitet[i] = \sum_{j=1}^N PredlogDistribucija[i][j]$$

На иницијализација вредностите на секое место во матрицата се поставуваат по случаен избор со рамномерна распределба.

4.1.2 Мутација

Операторот за мутација како аргумент прима еден хромозом и ратата на мутација со која е стартуван алгоритмот, потоа зависно од ратата на мутација бира колку од вредностите на матрицата на хромозомот ќе ги мутира т.е ќе бидат поставени на случајна вредност помеѓу 0 и 100.

4.1.3 Вкрстување

Операторот за вкрстувањето добива како аргумент 2 хромозоми, и за секој од елементите во матрицата на хромозомите на иста позиција постои 50% да бидат заменети меѓу двете единки.

4.1.4 Функција на цел

Функцијата на цел на влез прима еден хромозом и ја пресметува цената за производството на електричната енергија. Прво се пресметува колку изнесува цената според одбраниот капацитет на секоја од електраните

$$cena = \sum_{j=1}^M (maxCena[j] \cdot MaxProiz[j] \cdot \sum_{i=1}^N OdbranKapacitet[i][j])$$

потоа, доколку не е исполнето некое од ограничувањата се казнува цената со множење, притоа:

- Доколку електрана дистрибуира енергија до населено место до кое нема далековод, ($Zaguba[i][j] = 0$ но $OdbranKapacitet[i][j] \neq 0$) цената се множи со производот на цената на најскапата електрана и големината на прекорачувањето)
- Доколку електрана треба да произведе повеќе енергија отколку што е нејзиниот максимален капацитет ($OdbranKapacitet[i][j] > 100$), цената се множи со производот на цената на најскапата електрана и големината на прекорачувањето
- Доколку населено место добива помалку енергија од својата побарувачка

$$\forall i, \sum_{j=1}^M MaxProiz[j] \cdot OdbranKapacitet[i][j] \cdot Zaguba[i][j] < Potrosuvacka[i]$$

цената се множи со производот на цената на најскапата електрана и недостатокот на енергија

Алгоритмот за се обидува да ја минимизира цената, притоа казните на цената се земаат релативно од влезните податоци (цена на најскапа електрана). Казните се променливи и во однос на големината на грешката, прекорачување на капацитетот на една електрана за 1% се казнува пропорционално помалку во однос на прекорачување на капацитетот за 20% или 30%.

4.1.5 Резултати

Резултатите ќе ги прикажеме во табела за секоја од категоријата на решенија, споредени со очекуваните резултати. Во табелата дадена во оваа секција ги прикажуваме резултатите добиени за 10000 генерации со 1000 единки во популацијата. Притоа алгоритмот работеше со рата на мутација 0.5 и рата на вкрстување 0.5. Потоа ќе ги дискутираме добиените резултати и ќе наведеме како влијание имала промената на разни параметри на алгоритмот.

Проблем	Добиено решение	Очекувано решение
1.a	920	920
1.b	46.9	46.2
1.c	176	160
2.a	141.2	132.8
2.b	456	386.5
2.c	N/A	216.8

Table 1: Споредба на добиени резултати со очекуваните, 1000 единки, во 10000 генерации

Од табела 1 може да се забележи дека за најтривијалните проблеми, наивно то генетско решение дава добри резултати. Но со усложнување на проблемот, воведувањето на повеќе електрани кои дистрибуираат кон повеќе населени места со променливи загуби, веќе резултираат со значителни отстапувања од очекуваните решенија. Во примерите 2.b и 2.c очекуваните решенија исто така не се оптимални, туку се добиени на алчен пристап и служат само за споредба. Но иако такви, првата верзија на генетскиот алгоритам не дава решенија блиски до нив. За најсложениот пример во 2.c генетскиот алгоритам не успеа да даде решение кое не ги прекршува ни едно од дадените ограничувања (дури и по 15000 генерации со 1000 единки).

Дополнително беше тестирано и влијанието на промената на: ратата на мутација, ратата на вкрстување, бројот на генерации и бројот на единки. Од табелата во документот со решенија референциран во секција 6 може да се забележи дека за оваа имплементација на алгоритмот, рата на мутација под 0.4 и над 0.6 влијаеше забележително негативно на резултатите, освен во наједноставните случаи. Ратата на вкрстување имаше многу мало влијание на резултатите. За примерите од групата 1, немаше влијание бројот на единки, и

доволни беа 100 единки да дадат добар резултат, но за примерите од втората група (особено 2.b) тестовите со помалку од 1000 единки не даваа резултати кои ги задоволуваа ограничувањата. На сличен начин влијаеше и бројот на генерации, кај тривијалните примери, доволно беа и 1000 генерации, додека кај 2.c примерот ни 15000 генерации не успеаа да дадат здоволително решение.

4.2 Варијанта 2: Ограничено модифицирање на хромозомите

Во втората варијанта на алгоритмот ќе го ограничиме начинот на составување на хромозомите, со тоа што уште при иницијализацијата на единките некои од ограничувањата ќе бидат запазени. Исто така при вкрстувањето ќе биде запазено потомокот да ги задоволува истите ограничувања.

4.2.1 Претставување на хромозомите

Хромозомите ги претставуваме потполно исто како во Варијанта 1, но, во оваа варијанта, при иницијализирањето на еден хромозом, се зема во предвид матрицата на загуба. Бидејќи оваа матрица е позната на почеток на алгоритмот, и ни кажува меѓу кои населени места и електрани не треба да има никаква дистрибуција, знаеме дека на тие места во матрицата *PredlogDistribucija* треба во решението да има нули, и со тоа при иницијализација наместо случајна вредност се поставува нула. На овој начин едно од ограничувањата е задоволено од сите единки во популацијата

4.2.2 Мутација

Операторот за мутација како аргумент прима еден хромозом и ратата на мутација со која е стартуван алгоритмот, потоа зависно од ратата на мутација бира колку од вредностите на матрицата на хромозомот ќе ги мутира т.е. ќе ги постави на случајна вредност помеѓу 0 и 100. Но доколку според матрицата *Zaguba* не треба да има дистрибуција меѓу одредено населено место и електрана, тогаш вредноста во матрицата *PredlogDistribucija* на тоа место ќе си остане 0.

4.2.3 Вкрстување

Операторот за вкрстувањето добива како аргумент 2 хромозоми, и за секој од елементите во матрицата на хромозомите на иста позиција постои 50% да бидат заменети меѓу двете единки. Но, ако според матрицата *Zaguba* нема дистрибуција меѓу два елементи, тогаш на таа позиција си останува 0.

4.2.4 Функција на цел

Функцијата на цел на влез прима еден хромозом и ја пресметува цената за производството на електричната енергија. Прво се пресметува колку изнесува

цената според одбраниот капацитет на секоја од електраните

$$цена = \sum_{j=1}^M (maxCena[j] \cdot MaxProiz[j] \cdot \sum_{i=1}^N OdbranKapacitet[i][j])$$

потоа, доколку не е исполнето некое од ограничувањата се казнува цената со множење, притоа:

- Доколку електрана дистрибуира енергија до населено место до кое нема далековод, ($Zaguba[i][j] = 0$ но $OdbranKapacitet[i][j] \neq 0$) цената се множи со производот на цената на најскапата електрана и големината на прекорачувањето). Според дефиницијата на операторите, не треба никогаш ова ограничување да биде прекршено.
- Доколку електрана треба да произведе повеќе енергија отколку што е нејзиниот максимален капацитет ($OdbranKapacitet[i][j] > 100$), цената се множи со производот на цената на најскапата електрана и големината на прекорачувањето
- Доколку населено место добива помалку енергија од својата побарувачка

$$\forall i, \sum_{j=1}^M MaxProiz[j] \cdot OdbranKapacitet[i][j] \cdot Zaguba[i][j] < Potrosuvacka[i]$$

, цената се множи со производот на цената на најскапата електрана и недостатокот на енергија

Алгоритмот за се обидува да ја минимизира цената, притоа казните на цената се земаат релативно од влезните податоци (цена на најскапа електрана). Казните се променливи и во однос на големината на грешката, прекорачување на капацитетот на една електрана за 1% се казнува пропорционално помалку во однос на прекорачување на капацитетот за 20% или 30%.

4.2.5 Резултати

Резултатите ќе ги прикажеме во табела за секоја од категоријата на решенија, споредени со очекуваните резултати. Во табелата дадена во оваа секција ги прикажуваме резултатите добиени за 10000 генерации со 1000 единки во популацијата. Притоа алгоритмот работеше со рата на мутација 0.5 и рата на вкрстување 0.5. Потоа ќе ги дискутираме добиените резултати и ќе наведеме како влијание имала промената на разни параметри на алгоритмот.

Од табела 2 може да се забележи дека значително се подобри резултатите од претходната варијанта на алгоритмот, особено за примерот 2.b и 2.c. Кај примерот 2.b имаме добиено подобро решение и од очекуваното кое беше пресметано на рака со алчен пристап на обид и грешка. Кај примерот 2.c со првата варијанта на алгоритмот не успеавме да добиеме решение кое ќе ги задоволува

Проблем	Добиено решение	Очекувано решение
1.a	920	920
1.b	46.9	46.2
1.c	176	160
2.a	141.2	132.8
2.b	385.7	386.5
2.c	241.6	216.8

Table 2: Споредба на добиени резултати со очекуваните, 1000 единици, во 10000 генерации

сите ограничувања, со ова варијанта се добива решение. Предлог дистрибуцијата за примерот 2.b е

$$PredlogDistribucija = \begin{bmatrix} 0 & 56 & 0 & 0 \\ 0 & 41 & 10 & 0 \\ 0 & 0 & 49 & 17 \\ 1 & 0 & 0 & 83 \end{bmatrix}$$

Дополнително беше тестирано и влијанието на: намалена рата на мутација, зголемена и намалена рата на вкрстување и зголемен број на генерации. Од табелата со сите резултати од тестовите референцирана во секција 6 може да се забележи дека за оваа имплементација на алгоритмот:

- користење на повеќе од 10000 итерации даде погодни резултати само за примерот 2.c, кој е најсложениот од примерите, (пронајдено беше решение со цена од 228.4 за разлика од претходното 241.6) и може да се извлече хевристика која алгоритмот ќе го конфигурира колку итерации ќе работи во зависност од должината на влезните податоци
- зголемена рата на вкрстување не даде подобри резултати
- намалена рата на вкрстување даде малку подобри резултати само за примерот 2.c

5 Заклучок

Градењето на генетски алгоритам за проблемот претставен овде се покажа како доволно добра опција. Просторот на решенија кој треба да се пребара се значително намалува, а својствата на генетскиот алгоритам овозможуваат пребарувањето да биде добро насочено. Со втората варијанта на алгоритмот се помогна тој да конвергира во решение кое ги задоволува сите органичувања. Можноста некои од ограничувањата да бидат запазени од самите генетски оператори резултираше со доста поволност за работата на алгоритмот и способноста негова да решава посложени проблеми. За посложени влезни податоци, алгоритмот работи и дава резултати, но не скалира многу добро во поглед на време на извршување.

6 Дополнителни документи

Целосната имплементација е прикачена во јавен репозиториум на веб страницата GitHub на https://github.com/atanasovskib/power_distribution_optimization.
Овде воедно се наоѓа и документацијата на проектот. Сите резултати од тестовите се наоѓаат во датотеката *power_distribution_results.xlsx*