

FRI-2G.302-1-CSN-01

OPTIMIZATION OF MODELS FOR TRAFFIC PREDICTION IN MARKOV CHAINS¹

Prof. Mihail Iliev, DcS

Telecommunications Department,
“Angel Kanchev” University of Ruse
Tel.: 082-888 673
E-mail: miliev@uni-ruse.bg

Assoc. Prof. Ivelina Balabanova, PhD

Department of Communications Equipment and Technologies,
Technical University of Gabrovo
Phone: 0896 640 473
E-mail: ivstoeva@abv.bg

Georgi Georgiev, PhD

Department of Communications Equipment and Technologies,
Technical University of Gabrovo
Phone: 0877 522 029
E-mail: givanow@abv.bg

Assoc. Prof. Boyan Karapenev, PhD

Department of Communications Equipment and Technologies,
Technical University of Gabrovo
Phone: 0877 283 982
E-mail: bkarapenev@tugab.bg

***Abstract:** The paper presents the results in optimization of the Average Arrival Rate and Average Service Time factors in investigation of the teletraffic system $M/M/1/k$. The optimization processes with respect to models of predictable variations in Arrival and Exit times for processed system calls are realized. The polynomial predictive models on the basis of design of experiment and regression diagnostics of the data were obtained. Interior-point and Genetic algorithms about the target functions in defined functional limitations by graphical user interface Optimtool in MATLAB are applied. Based on the nonlinear functional minimization performance, the optimums about the considered factors correspond to the minimums of the desired parameters of the incoming and outgoing traffics have been reached. The optimization procedures are illustrated by graphical diagrams, source code of the applications and sets of generated variables according to the target functions.*

***Keywords:** Markov Chain, Regression Models, Optimization, Interior-point Algorithm, Genetic Algorithm.*

ВЪВЕДЕНИЕ

Решаването на проблеми при оптимизация на Марковски модели като средство за имитиране на процеси в различни сфери на техниката е важна задача предвид широката им приложимост. Според (Blanchet, J., Gallego, G., & Goyal, V., 2016) некоректният избор на модели може да бъде причина за крайно неоптимални решения. При оптимизационни проблеми при вериги на Марков предимно се използват градиентни статически оценки на основата на Kiefer Wolfowitz алгоритъм, Simultaneous Perturbation Stochastic Approximation (Krishnamurthy, V., 2015, Ermon, S., Gomes, C., Sabharwal, A., & Selman, B., 2014), Cutting Plane метод (Buist, E., 2008).

¹ Докладът е представен с оригинално заглавие на български език: ОПТИМИЗИРАНЕ НА МОДЕЛИ ЗА ПРОГНОЗИРАНЕ НА ТРАФИКА ПРИ ВЕРИГИ НА МАРКОВ

Съществуват проучвания, в които се извършва обратният процес, а именно веригите на Марков се използват като инструмент за оптимизация. В (Nazemi E.,2018) са представени процеси при формиране на контролни времеви интервали чрез използване на еднакви Марковски модели при обработка на постъпваща информация в системи за намиране на възможни дефекти при поддръжка на оборудване.

Важен аспект в комуникационните системи се свързва с възможността за оптимизиране на трафика и постигане по-добро качество на обслужване. В доклада се разглежда приложението на Interior-point и Genetic алгоритми при оптимизация на фактори, участващи в регресионни модели за прогнозиране на входящи и изходящи трафични, при верига на Марков M/M/1/k. Изследванията са проведени в среда MATLAB, използвайки GUI Optintool.

ИЗЛОЖЕНИЕ

Регресионни модели за прогнозиране на входящия и изходящия трафик при верига на Марков M/M/1/k и дефиниране на целевите функции за оптимизация

Извършени са предварителни процедури по имитационно моделиране на телетрафична система M/M/1/k посредством продукта Java Modeling Tool. Веригата е изследвана на основата на планиране на експеримента при управляеми фактори:

- x_1 – средна скорост на постъпване на потребителски заявки [cust./s];
- x_2 – средно време на обслужване на повиквания [s];
- x_3 – места за чакане в опашната [cust.];

и отклици на обекта на изследване:

- y_1 – време на постъпване на повиквания в системата [s],
- y_2 – време на освобождаване на обслужени повиквания [s],

Експериментите са проведени при нива на вариране на управляемите фактори, където кодиращи стойности „-1“ съответстват на натурални стойности 0.25 [cust./s], 0.5 [s] и 7 [cust.] на x_1 , x_2 и x_3 ; „0“ заменят 0.5 [cust./s], 1.0 [s] и 17 [cust.] при x_1 , x_2 и x_3 ; „+1“ се присвояват на 0.75 [cust./s], 1.5 [s] и 27 [cust.] за x_1 , x_2 и x_3 .

Направена е диагностика на резултатите с помощта на регресионен анализ като съобразно нивата на коефициентите на определеност и значимостта на тези на регресия е селектиран Симетричен композиционен план на експеримента. Получени са модели за прогнозиране на потенциалните изменения на търсените времена. Факторът x_3 е оценен като незначим и изключен при съставяне на уравненията за телетрафично прогнозиране.

```

Editor - C:\Users\Iva\Documents\MATLAB\markov1.m
markov1.m  markov2.m  ogran.m  +
1  function f = markov1(x)
2  -  f = 199.554 - (134.007*x(1)) - (9.733*x(2)) + (14.552*x(1)*x(2)) + (65.441*x(1)^2);
    
```

a)

```

Editor - C:\Users\Iva\Documents\MATLAB\markov2.m
markov1.m  markov2.m  ogranicheniq.m  gacode_markov1.m  +
1  function f = markov2(x)
2  -  f = 200.366 - (134.117*x(1)) - (9.433*x(2)) + (14.383*x(1)*x(2)) + (65.599*x(1)^2);
    
```

б)

Фиг. 1. Целеви функции за минимизацията относно времената на а) постъпване и б) освобождаване на обслужени заявки в и от системата

```

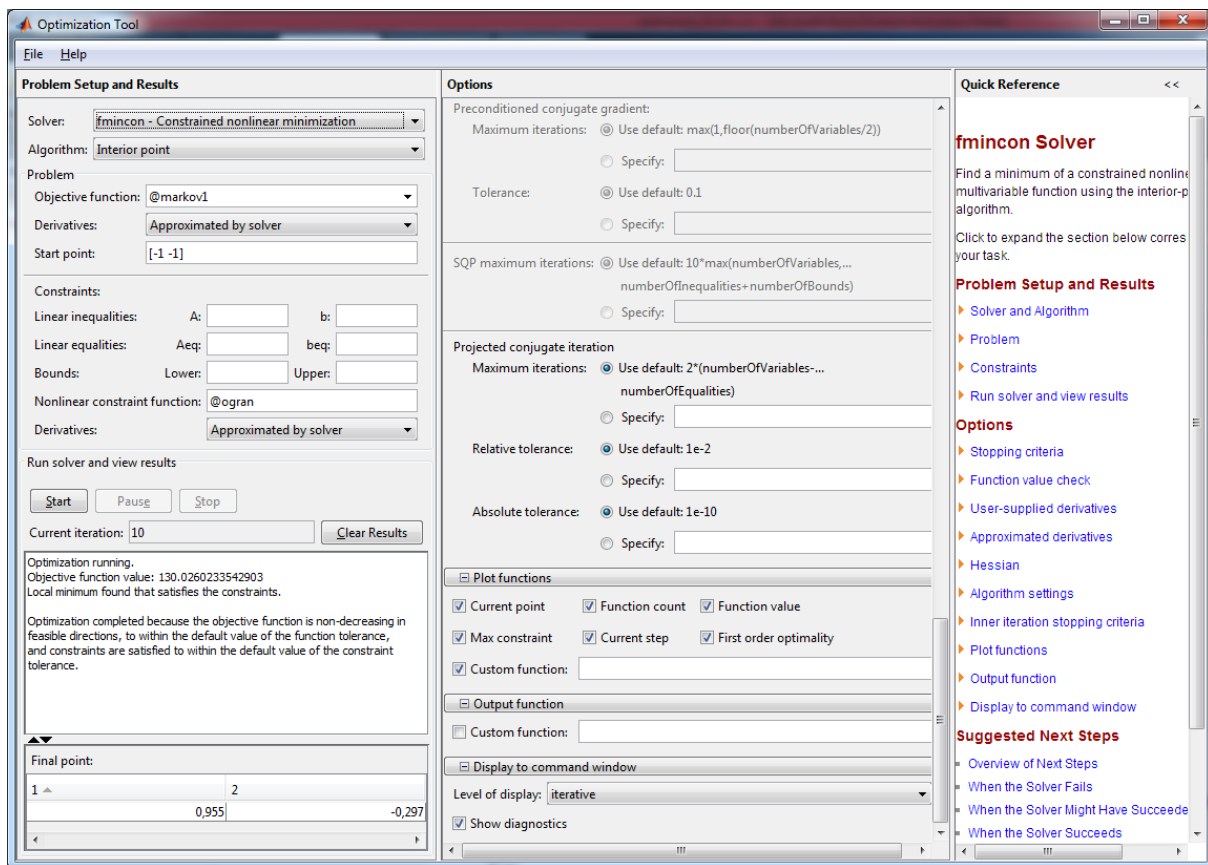
Editor - C:\Users\Iva\Documents\MATLAB\ogranicheniq.m
markov1.m  markov2.m  ogranicheniq.m
1 function [c,ceq] = ogranicheniq(x)
2     c = x(1)^2 + x(2)^2 - 1;
3     ceq = [ ];
    
```

Фиг. 2. Функция с ограничения при нелинейна оптимизация

Моделите са въведени като m-file скриптовни целеви функции за минимизация на фиг. 1. Също така е зададени функционални ограничения, непозволяващи на факторите да достигнат максимални стойности в хода на оптимизационните процедури (фиг. 2).

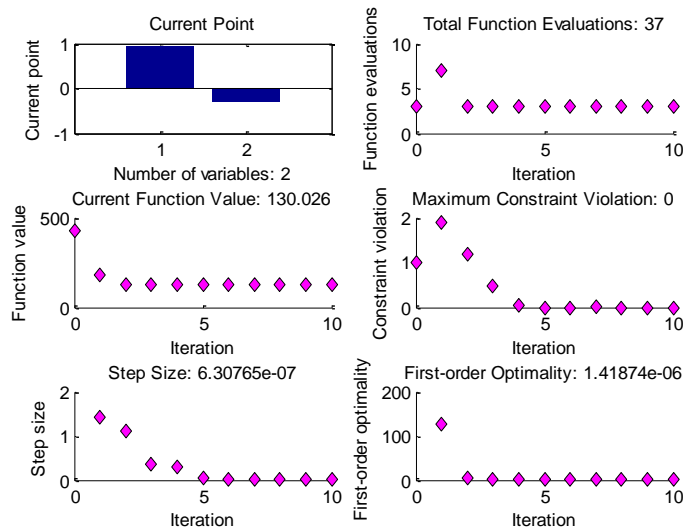
Минимизация на модела за прогнозиране на телетрафичното време на постъпване на повиквания посредством Interior-point и Genetic алгоритми

Съществуват два начина за алгоритмично изпълнение, съответно от командния прозорец на MATLAB и чрез графичен потребителски интерфейс Optimtool. Конкретно за решаване на задачата е използван вторият подход. Изглед на инструмента при търсене на минимум на въведената целева функция е показан на фиг. 3.



Фиг. 3. Изглед на графичен потребителски интерфейс Optimtool при минимизация с Interior-point алгоритъм относно времето на постъпване на заявки в системата

Фигура 4 представя резултатите от проведената оптимизация, даващи възможност за итеративно проследяване на процеса. В хода на оптимизация за извършени 37 оценки на целевата функция за период от 10 итерации. Достигнат е минимум на телетрафично време 130.0260 [s], при който е налице нулево максимално константно отклонение (индикация на коректност на минимизация).



Фиг. 4. Графични резултати от минимизация на функция „време на постъпване на заявки в системата“ с Interior-point алгоритъм

```

Editor - C:\Users\iva\Documents\MATLAB\code_markov1.m
markov1.m  markov2.m  ogran.m  code_markov1.m  +
This file can be published to a formatted document. For more information, see the publishing video or help.
1  function [x,fval,exitflag,output,lambda,grad,hessian] = code_markov1(x0)
2  %% This is an auto generated MATLAB file from Optimization Tool.
3
4  %% Start with the default options
5  options = optimoptions('fmincon');
6  %% Modify options setting
7  options = optimoptions(options,'Display','iter');
8  options = optimoptions(options,'PlotFcns',{@optimplotx @optimplotfunccount @optimplotfval});
9  options = optimoptions(options,'Diagnostics','on');
10 [x,fval,exitflag,output,lambda,grad,hessian] = ...
11 fmincon(@markov1,x0,[],[],[],[],[],[],@ogran,options);
12

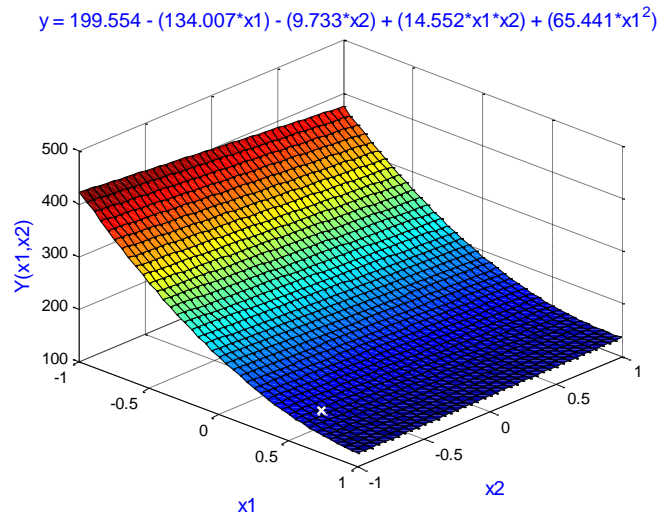
```

Фиг. 5. Програмен код при приложение на Interior-point алгоритъм при минимизация на функция „време на постъпване на заявки в системата“

Сорс кода на алгоритмично приложение е даден на фиг. 5. Получени са променливи от изпълнение на оптимизационните процедури (фиг. 6). Променливата “x” представлява достигнатия оптимум със стойности на факторите $x_1 = 0.9548$ и $x_2 = -0.2972$, при които се получава минимум на целевата функция $fval = 130.0260$. Положителната стойност на аргумента $exitflag = 1$ потвърждава коректността на направената нелинейна минимизация. Структурите $output$ и $lambda$ съдържат обща информация за оптимизацията и генерирания набор от множители на Лагранж. Матриците $grad$ и $hessian$ представляват намерените градиент и Хесиан на целевата функция, чиито достигнат екстремален минимум е показан на фиг. 7.

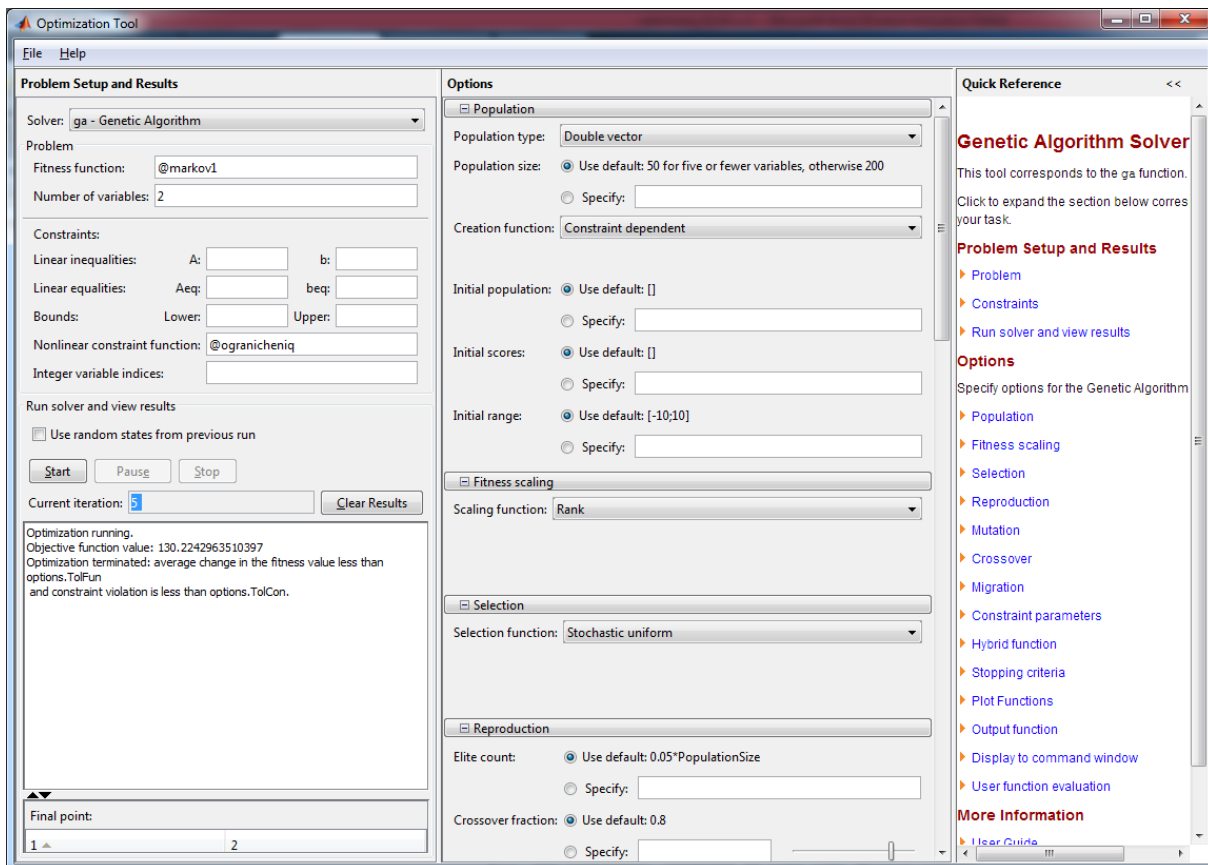
Field	Value	Min	Max
x	[0.9548 -0.2972]	-0.2972	0.9548
fval	130.0260	130.02...	130.02...
exitflag	1	1	1
output	1x1 struct		
lambda	1x1 struct		
grad	[-13.3664;4.1612]	-13.36...	4.1612
hessian	[136.8306 15.5390;15.5390 15.5360]	15.5360	136.83...

Фиг. 6. Променливи от хода на оптимизация при приложение на Interior-point алгоритъм относно времето на постъпване на заявки в системата



Фиг. 7. Минимизирана целева функция „време на постъпване на заявки в системата“ с Interior-point алгоритъм

Изследвана е ефективността на Генетичния алгоритъм при провеждане на константна нелинейна оптимизация.



Фиг. 8. Изглед на графичен потребителски интерфейс Optimtool при минимизация с Генетичен алгоритъм относно времето на постъпване на заявки в системата

Алгоритмите от този вид се базират на следната последователност на изпълнение:

- ✓ Създава се *случайна* начална популация;
- ✓ Генерират се последователно поредица от нови популации, като от всяка текуща се създава следващата на базата на *три основни правила*:

- *Селекция* – избор на определен брой индивиди с най-добри оценки за *родители*, т.е. избор на множество точки с най-малки стойности на целевата функция. Част от тях с най-високи оценки се наричат *елитни* и се класират директно за новата популация. Към останалите се прилага кръстосване или мутация;
- *Кръстосване* – комбиниране на двойка родители чрез размяна на *гени* за създаване на потомци. В оптимизацията размяна на една или няколко координати на две точки-родители;
- *Мутация* – произволна промяна на един или няколко гена случайно избран родител. В оптимизацията – произволна промяна на една или няколко координати на точка-родител. Точката и кои от координатите ѝ да се променят се избира на случаен принцип. Това осигурява изследване на по-голяма част от допустимата област на параметричното пространство и избягване на вероятността от „засядане” в локален минимум.

```
>> optimtool
```

Generation	f-count	Best f(x)	max constraint	Stall Generations
1	2650	130.965	0	0
2	5250	130.224	0	0
3	7850	130.224	0	1
4	10450	130.224	0	2
5	13050	130.224	0	3

Optimization terminated: average change in the fitness value less than options.TolFun and constraint violation is less than options.TolCon.

Фиг. 9. Резултати от минимизация на фитнес функцията с Генетичен алгоритъм относно времето на постъпване на заявки в системата

```

1 function [x,fval,exitflag,output,population,score] = gacode_markov1(nvars)
2 %% This is an auto generated MATLAB file from Optimization Tool.
3
4 %% Start with the default options
5 options = gaoptimset;
6 %% Modify options setting
7 options = gaoptimset(options,'Display','iter');
8 [x,fval,exitflag,output,population,score] = ...
9 ga(@markov1,nvars,[],[],[],[],[],[],@ogranicheniq,[],options);
    
```

Фиг. 10. Програмен код при приложение на Генетичен алгоритъм при минимизация на функция „време на постъпване на заявки в системата“

Field	Value	Min	Max
x	[0.9256 -0.3634]	-0.3634	0.9256
fval	130.2243	130.22...	130.22...
exitflag	1	1	1
output	1x1 struct		
population	50x2 double	-40.19...	27.1667
score	50x1 double	88.5591	5.5880...

Фиг. 11. Променливи от хода на оптимизация при приложение на Генетичен алгоритъм относно времето на постъпване на заявки в системата

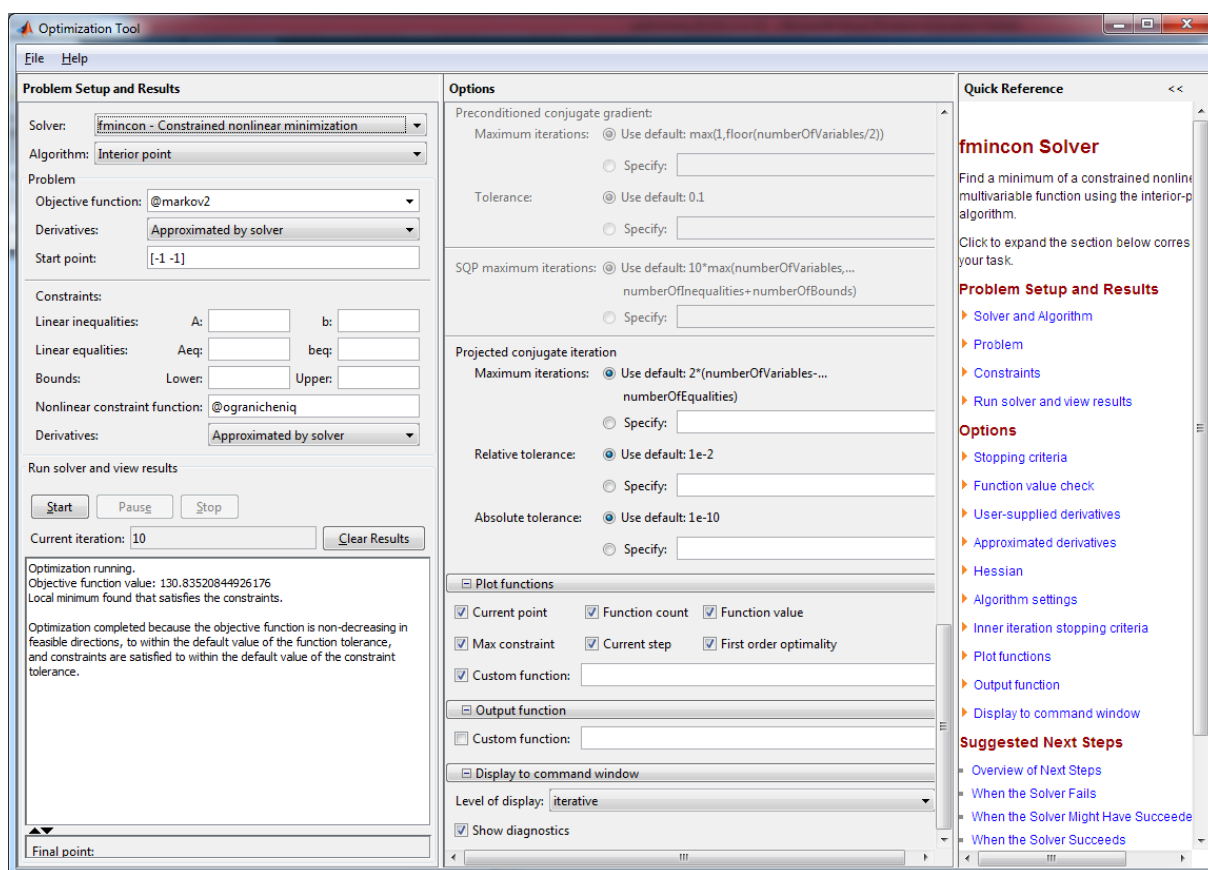
На фиг. 8 и фиг. 9 може да се наблюдава действието на Генетичния алгоритъм при търсене на най-добра оценка на анализиранията фитнес функция сред множество от точки в дадено поколение. В процеса на оптимизация са създадени пет популации като в първата са намерени 2650, а в последната 13050 решения. Получената най-добра оценка на фитнес

функцията е 130.224 [s] при пето поколение, създадено по време на трета итерация при константно отклонение.

Използвайки възможностите на интерфейса са генерирани изходни програмен код на алгоритмична $ga()$ реализация на фиг. 10 и набор от променливи на фиг. 11. Минимум относно телетрафичното време на постъпване клиентски заявки се достига при кодирани нива на средната скорост на постъпване на повиквания „0.9256“ и средното време за обслужване на повиквания „-0.3634“. Променливата $population$ съдържа множеството от точки в околността на минимума на целевата функция, а $score$ изчислените функционални оценки.

Минимизация на модела за прогнозиране на телетрафичното време на освобождаване на обслужени заявки чрез Interior-point и Genetic алгоритми

Фигура 12 илюстрира оптимизацията на целева функция „време за освобождаване на обслужени повиквания“ с приложение на Interior-point алгоритъм. При настройки се изисква да коректно да бъдат указани целевата и ограничаващата функции, както и начална точка с примерни координати във факторното пространство [-1 -1].



Фиг. 12. Изглед на графичен потребителски интерфейс Optintool при минимизация с Interior-point алгоритъм относно времето за освобождаване на обслужени повиквания

Минимизацията на разглежданата функция протича съгласно установените критерии в рамките на продължителност от 10 итерации. Във връзка с показаните числовите резултати, променливата F -count отразява количеството оценки на функциите при оптимизация. Стойността и за конкретния случай е 37. Достигнат е функционален минимум 130.8352 секунди.


```

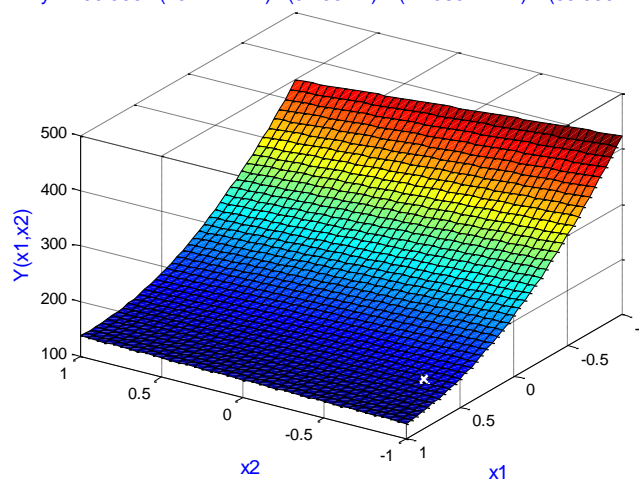
Editor - C:\Users\Iva\Documents\MATLAB\intpotntcodemarkov2.m
intpotntcodemarkov2.m x +
This file can be published to a formatted document. For more information, see the publishing video or help.
function [x,fval,exitflag,output,lambda,grad,hessian] = intpotntcodemarkov2(x0)
%% This is an auto generated MATLAB file from Optimization Tool.
%% Start with the default options
options = optimoptions('fmincon');
%% Modify options setting
options = optimoptions(options,'Display','iter');
options = optimoptions(options,'PlotFcns',{ @optimplotx @optimplotfunccount @optimplotfval});
options = optimoptions(options,'Diagnostics','on');
[x,fval,exitflag,output,lambda,grad,hessian] = ...
fmincon(@markov2,x0,[],[],[],[],[],[],@ogranicheniq,options);
    
```

Фиг. 13. Програмен код при приложение на Interior-point алгоритъм при минимизация на функция „време за освобождаване на обслужени повиквания“

Field	Value	Min	Max
x	[0.9530 -0.3030]	-0.3030	0.9530
fval	130.8352	130.83...	130.83...
exitflag	1	1	1
output	1x1 struct		
lambda	1x1 struct		
grad	[-13.4434;4.2740]	-13.44...	4.2740
hessian	[154.9196 13.7758;13.7758 13.0010]	13.0010	154.91...

Фиг. 14. Набор от променливи от хода на оптимизация при приложение на Interior-point алгоритъм относно параметър y_2

$$y = 200.366 - (134.117 \cdot x_1) - (9.433 \cdot x_2) + (14.383 \cdot x_1 \cdot x_2) + (65.559 \cdot x_1^2)$$



Фиг. 15. Минимизирана целева функция „време за освобождаване на обслужени от системата“ с Interior-point алгоритъм

Генериран е програмен код (фиг. 13), реализиращ процеса на нелинейна константна минимизация, както и са изведен набор от изходни променливи на фиг. 14. Стойностите на факторите x_1 и x_2 , удовлетворяващи решението за оптимум са 0.9530 -0.3030. Коректността на процеса е потвърдена от положителната стойност на променливата exitflag. 3D представяне на минимизираната функция е дадено на фиг. 15.

Аналогични действия са проведени с помощта на Генетичен алгоритъм, където оптимизационният процес е с продължителност от 4 итерации, представен на фиг. 16. В хода на оптимизацията, онагледен на фиг. 17, са генерирани четири поколения или множества от точки, чието количество се указва чрез променливата f-count. Относно всяко множество е намерена най-добра оценка на целевата фитнес функция, която в момента на достигане на оптимум се равнява на 131.083 секунди. Променливата max constant отразява максималното

нелинейно константно отклонение за всяко поколение, която в момента на намиране на оптимум е нула. Аргумента Stall Generations показва, че по време на нулева итерация са генерирани съответните четири полулация.

```
>> optimtool

      Generation   f-count      Best      max      Stall
      Generation   f-count      f(x)      constraint  Generations
      1             2670      133.017   4.521e-07    0
      2             5270      131.907    0            0
      3             8070      131.792    0            0
      4            14170      131.083    0            0

Optimization terminated: average change in the fitness value less than options.TolFun
and constraint violation is less than options.TolCon.
```

Фиг. 17. Резултати от минимизация на функция „време за освобождаване на обслужени повиквания“ с Генетичен алгоритъм

```
Editor - C:\Users\Iva\Documents\MATLAB\gacodemarkov2.m
gacodemarkov2.m x +
This file can be published to a formatted document. For more information, see the publishing video or help.
1 function [x,fval,exitflag,output,population,score] = gacodemarkov2(nvars)
2 %% This is an auto generated MATLAB file from Optimization Tool.
3
4 %% Start with the default options
5 options = gaoptimset;
6 %% Modify options setting
7 options = gaoptimset(options,'Display','iter');
8 [x,fval,exitflag,output,population,score] = ...
9 ga(@markov2,nvars,[],[],[],[],[],[],@ogranicheniq,[],options);
```

Фиг. 18. Програмен код при приложение на Генетичен алгоритъм при минимизация на функция „време за освобождаване на обслужени повиквания“

Field	Value	Min	Max
x	[0.9775 -0.1866]	-0.1866	0.9775
fval	131.0832	131.08...	131.08...
exitflag	1	1	1
output	1x1 struct		
population	50x2 double	-31.19...	20.7866
score	50x1 double	129.25...	2.8248...

Фиг. 19. Променливи от хода на оптимизация при приложение на Генетичен алгоритъм относно времето за освобождаване на обслужени от системата

Програмният код, илюстриращ приложението на Генетичния алгоритъм, е онагледен на фиг. 18. Съгласно достигнатия оптимум (фиг. 19), съответстващ на условието на достигане на глобален минимум, времето за постъпване на повиквания ще бъде минимизирано при нива на средната скорост на постъпване на заявки в системата и средно обслужващо време, съответно 0.9775 и -0.1866.

ИЗВОДИ

Установена е добра приложимост на използваните оптимизационни алгоритми, потвърждаваща се от получените резултати при търсене на оптимални решения по отношение на времената на входящия и изходящия трафик. Това дава основание за продължаване на работата по дефинираната проблематика при други обектни телетрафични системи и методи за оптимизация.

БЛАГОДАРНОСТИ

Тази публикация е резлизирана във връзка с Национална научна програма „Информационни и комуникационни технологии за единен цифров пазар в науката, образованието и сигурността (ICTinSES)“, финансирана от “Министерството на образованието и науката” на България“.

REFERENCES

Blanchet, J., Gallego, G., & Goyal, V. (2016). A Markov chain approximation to choice modeling. *Operations Research*, 64(4), 1-42.

Krishnamurthy, V. (2015). *Reinforcement learning: Stochastic approximation algorithms for Markov decision processes*. Cornell University, 1-56.

Ermon, S., Gomes, C., Sabharwal, A., & Selman, B. (2014). *Designing fast absorbing Markov chains*. Association for the Advancement of Artificial Intelligence, 1-7.

Buist, E. (2008). *Speeding up call center simulation and optimization by Markov chain uniformization*. Proceedings of the 40th Winter Simulation Conference, 7th-10th December 2008, Florida, 1652-1660.

Nazemi E. (2018). Development of an inspection optimization model using Semi-Markov process and delay time concept. *Industrial Engineering & Management*, 7(2), 1-11.