

## **ДВУВЕРИЖНА ПОПУЛАЦИЯ В ЕВОЛЮЦИОННИ АЛГОРИТМИ ЗА НЕЛИНЕЙНИ ЗАДАЧИ**

**Гергана Матеева<sup>1</sup>, Делян Керемедчиев<sup>1</sup>,  
Калин Копанов<sup>1</sup>, Велизар Върбанов<sup>1</sup>, Тодор Балабанов<sup>2</sup>**  
[gergana.mateeva@iict.bas.bg](mailto:gergana.mateeva@iict.bas.bg), [delyan.keremedchiev@iict.bas.bg](mailto:delyan.keremedchiev@iict.bas.bg), [kalin.kopanov@iict.bas.bg](mailto:kalin.kopanov@iict.bas.bg),  
[velizar.varbanov@iict.bas.bg](mailto:velizar.varbanov@iict.bas.bg), [todor.balabanov@vtu.bg](mailto:todor.balabanov@vtu.bg)

*1 Българска академия на науките  
Институт по информационни и комуникационни технологии  
ул. "акад. Георги Бончев", блок 2, гр. София 1113  
РЕПУБЛИКА БЪЛГАРИЯ*

*2 Висше транспортно училище "Тодор Каблешков"  
Факултет "Комуникации и електрообзавеждане в транспорта"  
ул. "Гео Милев" 158, корпус 2, гр. София 1574  
РЕПУБЛИКА БЪЛГАРИЯ*

**Ключови думи:** *двуверижни генетични алгоритми, нелинейна оптимизация, електронни таблици*

**Резюме:** *Солвърът в LibreOffice Calc се основава на хибриден алгоритъм за евристична глобална оптимизация. Хромозомното кодиране в класическата диференциална еволюция е много по-близка абстракция до структурата на РНК, отколкото до структурата на ДНК. Това изследване предлага двойноверижна (по-подобна на ДНК) организация на хромозомите в модела на електронната таблица LibreOffice Calc. Предложеният модел е валидиран с помощта на две от най-известните оценъчни функции - Rosenbrock и Styblinski-Tang.*

### **ВЪВЕДЕНИЕ**

Основната концепция зад диференциалната еволюция е да се наподобява процесът на естествен подбор и възпроизводство, за да се генерира популация от потенциални решения на оптимизационните задачи. Изчислението започва с произволна популация от кандидат-решения и впоследствие се използват оператори, като селекция, кръстосване и мутация [1], за да се генерират нови поколения от кандидат-решения. По същия начин, оптимизирането на рояка от частици черпи вдъхновение от социалното поведение на птици и риби, първоначално предложено от Джеймс Кенеди и Ръсел Еберхарт[2]. И двата алгоритъма се използват много ефективно в хибридна реализация за глобална оптимизация.

### **ОПЕРАЦИЯ ЗА СЕЛЕКЦИЯ**

В етапа на подбор е по-вероятно да бъдат селектирани решения, демонстриращи превъзходство за дадената задача. Селекцията представлява основен компонент при

популационните алгоритми, така че най-силните индивиди от популацията се избират да служат като родители за следващото поколение [3]. Операцията за селекция е основна при вземането на решение кои индивиди ще допринесат с генетичен материал за следващото поколение, което го прави критичен фактор, влияещ върху производителността и ефективността на популационните алгоритми. Съществуват различни методи за подбор, всеки от които има различни предимства и недостатъци. Операцията за селекция е от решаващо значение, защото позволява на популационните алгоритми да идентифицират и запазят най-добрите решения, като постепенно подобряват цялостната жизненост на популацията с течение на времето. Избирайки най-подходящите индивиди като родители за следващото поколение, алгоритъмът може да се доближи до оптимално решение по-бързо и по-ефективно, отколкото ако трябваше да избира родители на случаен принцип. От съществено значение да се постигне баланс между натиска на селекцията и генетичното разнообразие, за да се гарантира, че алгоритъмът плавно се сближава към оптимално решение, като се избягват не оптимални резултати.

### ***ОПЕРАЦИЯ ЗА КРЪСТОСВАНЕ***

В етапа на кръстосване две или повече кандидат-решения се обединяват, за да образуват ново кандидат-решение. Операцията по кръстосване се използва за подобряване на генетичното разнообразие на популацията и предоставяне на нови решения, които могат да отговарят по-добре на разглежданата задача. Операцията за кръстосване обикновено се прилага с вероятност. Различни типове кръстосвания могат да се използват в популационните алгоритми [4]. Изборът кой тип кръстосване да се използва зависи от проблема, който се решава, и характеристиките на популацията, която еволюира. Едно предизвикателство с операцията по кръстосване е възможността за преждевременна конвергенция, при която алгоритъмът се сближава твърде бързо до не оптимално решение. За да се смекчи този риск, от съществено значение е да се балансира кръстосването с други генетични операции, като мутация, и внимателно да се контролира скоростта на кръстосване, за да се гарантира, че алгоритъмът поддържа достатъчно генетично разнообразие по време на еволюцията.

### ***ОПЕРАЦИЯ ЗА МУТАЦИЯ***

В етапа на мутация се прави произволна промяна на кандидат-решение, за да се въведе разнообразие в популацията. Това е критичен компонент при популационните алгоритми, който въвежда малки, случайни промени в генетичния материал на индивида в популацията. Операцията за мутация се използва за поддържане на генетичното разнообразие в популацията от индивиди и за предотвратяване на блокиране на генетичния алгоритъм в локални оптимуми, което може да бъде неоптимално решение на задачата, която се решава. Операцията за мутация обикновено се прилага с вероятност (степен на мутация) [5]. Тази степен определя вероятността даден ген в даден индивид да претърпи мутация. Обикновено процентът на мутация е настроен на ниска стойност, 1% или 5%, за да се гарантира, че само малка част от популацията е мутирала във всеки даден момент. Различни видове операции за мутация могат да бъдат използвани в популационните алгоритми. Изборът кой тип мутация да се използва зависи от решаваната задача и характеристиките на еволюиращата популация. Като цяло мутацията въвежда нов генетичен материал, който може да е от полза за решението на задачата. Прекомерната мутация обаче може да доведе до загуба на годност в популацията. Следователно е от решаващо значение да се балансира мутацията с други генетични оператори, като кръстосване, за да се гарантира, че популационният алгоритъм доближава ефективно към оптимално решение. С течение

на времето популацията се развива към по-добри решения, тъй като алгоритъмът многократно прилага тези оператори, за да генерира нови поколения (кандидат-решения). Алгоритми, базирани на популация, са използвани за широк кръг от проблеми, от оптимизационни проблеми в инженерството до компютърни опоненти в игри и машинно обучение.

### ***ГЕНЕТИКАТА В ПРИРОДАТА***

Рибонуклеиновата киселина е молекула, която играе решаваща роля в преноса на генетична информация в живите организми. Подобно на начина, по който РНК служи като генетичен код, предаващ информация от ДНК към рибозомите, повечето популационни алгоритми използват списък кандидат-решения като форма на генетичен код, предавайки информация за потенциални решения на задачата. Подобно на РНК, някои популационни алгоритми използват селекция, кръстосване и мутация, за да генерират ново поколение от кандидат-решения. След това тези решения се оценяват въз основа на тяхната пригодност или способност за решение на задачата. Еволюцията на тези популации с течение на времето отразява процеса на еволюция на генетичната информация в живите организми, където най-подходящите решения оцеляват и се разпространяват, докато по-малко годните решения изчезват. Евристиката в популационните алгоритми осигурява отличителна симулация на процесите, представящи еволюцията в реалния свят.

### ***АЛГОРИТМИЧНО ПОДОБРЕНИЕ***

В това научно изследване единичните хромозоми се сдвояват, за да имитират структурата на ДНК. Всяко кандидат-решение има свой допълващ аналог и заедно те образуват допълваща се двойка. Останалата част на доклада е организирана по следния начин: Вторият раздел представя концептуалната идея; третият раздел описва практическата реализация като модел на LibreOffice Calc; четвъртият раздел разглежда експерименталната част и постигнатите резултати; последният раздел завършва и предлага по-нататъшни изследвания.

### ***ДВУВЕРИЖНИ КАНДИДАТ-РЕШЕНИЯ***

Както е известно, популацията на широко използваните евристики се състои от хромозоми, всяка със своето самостоятелно съществуване в популацията. Тези хромозоми подражават на структурата на РНК, открита в биологичната сфера. Въпреки че генетичните алгоритми обикновено използват двоично кодиране, то само понякога може да бъде оптимално за някои оптимизационни или симулационни задачи. Кодирането на числа с плаваща запетая често е по-подходящо за проблеми, включващи реални числа, като функции еталони за сравнение от вида на Rastrigin, Sphere, Rosenbrock и Styblinski-Tang [6]. Използването на кодиране на числа с плаваща запетая позволява по-директно представяне на проблемната област. В това кодиране гените в хромозомата представляват стойности с плаваща запетая вместо двоични стойности, което потенциално води до по-бърза конвергенция и удовлетворяващи решения, особено при проблеми с непрекъснати променливи. Съществуват различни методи за кодиране на числа с плаваща запетая, включително кодиране на Грей. При кодиране с плаваща запетая гените на хромозомата се представят като числа с плаваща запетая. За разлика от това, гените са изобразени като двоични цифри, преобразувани в реални числа, използвайки специфична формула в кодиране на Грей. Разширяването на аналогията на генетичните алгоритми към сдвоените ДНК вериги е възможно. При този подход всяка хромозома в популацията би имала комплементарна двойка, подобна на базовите двойки в ДНК веригите, наречени двойноверижни генетични алгоритми

(DSGA) [7]. В DSGAs хромозомите се сдвояват, образувайки комплементарни двойки, третирани като единични по време на операции за селекция, кръстосване и мутация. Използването на сдвоени хромозоми в DSGAs може да осигури предимства пред традиционните популационни алгоритми, улеснявайки по-ефективно изследване на пространството за търсене и подобро боравене с ограниченията на оптимизационните задачи. Важно е да се отбележи, че DSGAs може да доведе до по-високи изчислителни разходи от традиционните алгоритми, предвид необходимостта от поддържане на допълващи се двойки и извършване на операции върху двойки, а не върху отделни хромозоми. В настоящото изложение се предлага използването на хромозоми, организирани като близнаци с плаваща запетая и стойности с обърнат знак в допълващата последователност. При този метод всяка хромозома е представена от два близнака [8], където стойностите в единия близнак имат противоположен знак, а в другия близнак са с отрицателна стойност. Този подход използва универсалността на кодирането с плаваща запетая. Едно предимство на двойните хромозоми е тяхната споделена стойност на годност, изчислена отделно за всеки близнак, смекчавайки преждевременната конвергенция до локално оптимални решения. Освен това, двойните хромозоми могат да допринесат с допълнителна информация за пространството на търсене, подпомагайки поддържането на разнообразието в популацията. Въпреки това е изключително важно да се признае, че използването на двойни хромозоми може да доведе до допълнителни изчислителни разходи поради операцията на инверсия, приложена към всяка стойност. Освен това, този подход може да е подходящ само за проблеми, които са в съответствие с кодирането с плаваща запетая.

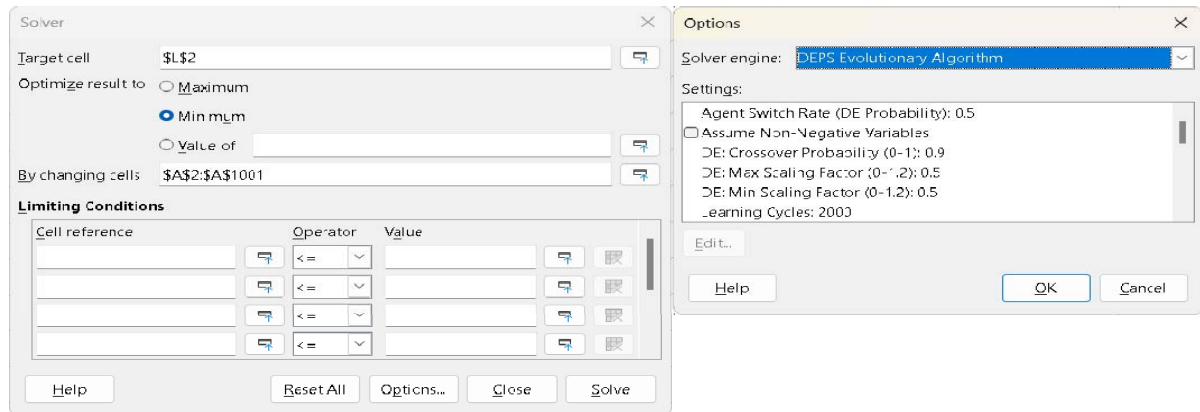
## ПРАКТИЧЕСКА РЕАЛИЗАЦИЯ

LibreOffice Calc има широко използван модул, наречен NLP Solver, за глобална оптимизация на нелинейни задачи. Налични са няколко алгоритъма, но хибридната комбинация от диференциална еволюция и оптимизация на рояк частици дава най-използваеми резултати. В табличния лист се въвежда числената задача, за която ще се търси оптимално решение. Първоначалните стойности, избрани на случаен принцип, се задават в последователни клетки, за предпочитане в колона, но не е задължително. В съседна колона се извършват изчисления за всяка стойност на функцията, която се оптимизира. За функции, които включват суми, конкретна клетка е определена за получаване на целевата стойност за оптимизиране. Предложеният модел на електронна таблица може да бъде достъпен от следния източник на информация [9].

	+X	-X	Rosenbrock +X	Rosenbrock -X	Styblinski-Tang +X	Styblinski-Tang -X	Function	f(+X)	f(-X)	Min
1	0.45805066258	-0.45805066258	2.720560031258	35.23492867886	-1.02269283101	-5.60319945680882	Rosenbrock	10229.12713752	10370.54317365	10229.12713752
2	0.365593943502	-0.365593943502	12.37212923673	39.47728483265	-0.29270847692	-3.9486479119391	Styblinski-Tang	-627.833675541	-650.044176008	-650.044176008
3	0.479630869581	-0.479630869581	0.442845043655	19.71284993027	-1.2296569322	-6.02596562800703				
4	0.18856553338	-0.18856553338	11.33710686417	17.24484509324	0.37518059827	-1.51047473552646				
5	0.3623394717	-0.3623394717	9.085447760734	32.9007504032	-0.27170388959	-3.89509860659129				
6	0.425888548746	-0.425888548746	0.584284829381	19.108898484	-0.73975506406	-4.99864055152742				
7	0.231846972265	-0.231846972265	0.975287548827	4.392940875022	0.302075951552	-2.01639377110102				
8	0.115819795073	-0.115819795073	25.97798461132	23.81988793499	0.364651317903	-0.79354663282343				
9	-0.48854404003	0.48854404003	56.38576307824	6.95168105044	-6.20455877616	-1.31911837583297				
10	-0.49732743643	0.49732743643	5.085468846323	44.24875236296	-6.38281605251	-1.40954168824708				
11	0.415960774925	-0.415960774925	22.40806874864	3.535316070998	-0.65863290051	-4.81824064976211				
12	-0.29673153718	0.29673153718	28.33642449225	12.06715231465	-2.89469863546	0.08261873634755				
13	-0.42823434591	0.42823434591	6.03876938402	3.10920233289	-5.04169627807	-0.75935281900105				
14	-0.01658824408	0.016588244079	3.16629959902	3.083902287966	-0.08734386214	0.07853857864827				
15	-0.14576756161	0.145767561611	2.430703739444	2.92687063189	-1.06835723511	0.38931838100587				
16	0.126979938563	-0.12697993856	12.11254937536	14.89734013676	0.377177196361	-0.89262218926697				
17	0.35302723437	-0.35302723437	1.164683317512	4.483633790165	-0.21338328419	-3.74365562788974				
18	0.038250594641	-0.03825059464	1.636636678995	1.839866926273	0.167845386043	-0.21466056036502				
19	0.085823925927	-0.08582392593	8.94925493749	10.15348398403	0.311321943668	-0.5469173155978				
20	0.292208458	-0.292208458	11.56596634208	4.289971143992	0.102160495096	-2.81992408490778				
21	-0.24725512669	0.247255126691	24.04298320675	12.95265790682	-2.21069969609	0.26185157082262				
22	-0.41307306185	0.413073061847	2.157100793185	9.417715505721	-4.76632060342	-0.6358998494978				
23	0.130588710689	-0.13058871069	1.035403290702	0.034733541776	0.380379790533	-0.92550731635464				
24	0.06992375283	-0.06992375283	24.76870941138	26.01415119112	0.271413370352	-0.42782415794636				
25	0.493803102603	-0.4938031026	25.28721451972	2.247391034369	-1.37298987409	-6.31102090012651				
26	0.76646010716	-0.76646010716	1.077301601104	4.242064506411	-0.22047042474	0.22426740601731				

Фиг. 1. Модел на оптимизираната задача

Както е илюстрирано на Фиг. 1, началната колона с данни съдържа X вектора, предоставен на целевата функция. Втората колона дублира същия вектор, но с отрицателни стойности. Тези два вектора съставляват допълваща се двойка. Стойностите на функцията Rosenbrock и Styblinski-Tang се изчисляват за всеки вектор. От получените стойности по-добрата се попълва в целевата клетка. Като се има предвид, че целта е минимизиране или максимизиране, по-малката или по-голямата от двете стойности се избира като стойност на годност на двойката.



Фиг. 2. Конфигуриране на нелинейния Солвър

Както е изобразено на Фиг. 2, целевата клетка съдържа изчислената стойност на функцията и само на първия близък от двойката е разрешено да се променя чрез диференциална еволюция и рояка частици. Вторият близък се генерира автоматично поради модификациите, направени на първия близък.

## ЕКСПЕРИМЕНТИ И РЕЗУЛТАТИ

Функциите на Rosenbrock и Styblinski-Tang са избрани поради асиметрията, присъстваща в отрицателния близък. Съотношението на DE към PSO, е 50 към 50%. Вероятността за кръстосване е зададена на 90% при диференциална еволюция, а коефициентът на мащабиране е 0.5. За оптимизацията на рояка от частици когнитивната константа е 1,494, а коефициентът на свиване е 0,729. Не се прилага мутация в рояка частици и социалната константа се поддържа на 1.494. Размерът на рояка е конфигуриран да има 70 индивида.

Таблица 1 Експериментални резултати

	Rosenbrock – единично		Rosenbrock – двоено		Styblinski-Tang – единично		Styblinski-Tang – двоено	
	Стойност	Време [s]	Стойност	Време [s]	Стойност	Време [s]	Стойност	Време [s]
Средна	5265.11	17.79	5131.65	17.64	-293.30	17.32	-280.76	17.30
Отклонение	749.38	0.70	613.06	0.66	61.40	0.74	45.50	0.77

Процесът на оптимизация е стартиран 30 пъти. Средните стойности и тяхното стандартно отклонение са представени в Таб. 1. Очевидно е, че двуверижната оптимизация дава по-добри резултати в рамките на същата времева рамка в сравнение с класическата оптимизация.

## ЗАКЛЮЧЕНИЕ

В заключение, подобряването на популационните алгоритми, вдъхновено от ДНК в LibreOffice Calc, може потенциално да подобри ефективността при решаване на

задачи с еволюционна оптимизация. Предложената адаптация може да се окаже особено полезна за инсталации на LibreOffice Calc с ограничени ресурси, като дава възможност да се търсят решения на сложни задачи с повишена скорост и прецизност. Като бъдещи научни изследвания може да се проучат допълнителни оценъчни функции и да се отчете производителността на предложеното подобрене в различни изчислителни платформи.

### **БЛАГОДАРНОСТИ**

This research is conducted as part of project No. KP-06-M75/3 “Research of methods and technologies for digitalization of education“, financed by the Bulgarian National Science Foundation (NSF).

### **ЛИТЕРАТУРА**

- [1] Lambora, A.; Gupta, K.; Chopra, K. 2019. "Genetic Algorithm- A Literature Review". International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon), Faridabad, India, 380-384.
- [2] Kennedy, J.; Eberhart, R. 1995. "Particle swarm optimization". Proceedings of ICNN'95 - International Conference on Neural Networks, Perth, WA, Australia, vol.4, 1942-1948.
- [3] Miller, B. L.; Goldberg, D. E. 1996. "Genetic Algorithms, Selection Schemes, and the Varying Effects of Noise" in Evolutionary Computation, vol. 4, no. 2, pp. 113-131.
- [4] Umbarkar, A. J.; Sheth, P. D. 2015. “Crossover operators in genetic algorithms: a review”. ICTACT journal on soft computing, vol. 6, no. 1, 1083-1092.
- [5] Greenwell, R.N.; Angus, J.E.; Finck, M. 1995. "Optimal mutation probability for genetic algorithms". Mathematical and Computer Modelling, vol. 21, no. 8, 1-11.
- [6] Jamil, M.; Yang, X; Zepernick, H.J. 2013. "8 - Test Functions for Global Optimization: A Comprehensive Survey". Swarm Intelligence and Bio-Inspired Computation, Elsevier, 193-222.
- [7] Zang, W.; Zhang, W.; Wang, Z.; Jiang, D.; Liu, X. and Sun, M. 2019. "A Novel Double-Strand DNA Genetic Algorithm for Multi-Objective Optimization". in IEEE Access, vol. 7, 18821-18839.
- [8] Yang, S. 2003. "PDGA: the primal-dual genetic algorithm". IOS Press. 1-10.
- [9] Balabanov, T. 2024. "Rosenbrock and Styblinski-Tang benchmark function reproduced as double-stranded chromosomes for LibreOffice Calc NLP Solver", ResearchGate GmbH, 10.13140/RG.2.2.14156.59528

## DOUBLE-STRANDED POPULATION IN EVOLUTIONARY ALGORITHMS FOR NONLINEAR PROBLEMS

Gergana Mateeva<sup>1</sup>, Delyan Keremedchiev<sup>1</sup>,  
Kalin Kopanov<sup>1</sup>, Velizar Varbanov<sup>1</sup>, Todor Balabanov<sup>2</sup>  
[gergana.mateeva@iict.bas.bg](mailto:gergana.mateeva@iict.bas.bg), [delyan.keremedchiev@iict.bas.bg](mailto:delyan.keremedchiev@iict.bas.bg), [kalin.kopanov@iict.bas.bg](mailto:kalin.kopanov@iict.bas.bg),  
[velizar.varbanov@iict.bas.bg](mailto:velizar.varbanov@iict.bas.bg), [todor.balabanov@vtu.bg](mailto:todor.balabanov@vtu.bg)

<sup>1</sup> *Bulgarian Academy of Sciences  
Institute of Information and Communication Technologies  
acad. Georgi Bonchev Str., block 2, 1113 Sofia  
THE REPUBLIC OF BULGARIA*

<sup>2</sup> *Higher School of Transport - Todor Kableshkov  
158 Geo Milev Str., block 2, 1574 Sofia  
THE REPUBLIC OF BULGARIA*

**Key words:** *double-stranded genetic algorithms, non-linear optimization, spreadsheets*

**Abstract:** *Currently, the supported non-linear solver in LibreOffice Calc is based on a hybrid algorithm for heuristic global optimization. The hybridization is achieved with differential evolution and particle swarm optimization. Natural phenomena like evolution and swarm behavior inspire both algorithms. Such meta-heuristic algorithms are often employed to tackle complex and time-consuming problems where exact numerical methods are not applicable within a reasonable timeframe. Chromosome encoding into classical differential evolution is a much closer abstraction to the RNA structure than the DNA structure. This research proposes a double-stranded (more DNA-like) organization of the chromosomes in the LibreOffice Calc NLP Solver spreadsheet model. The proposed model is validated using two of the most well-known benchmark functions - Rosenbrock and Styblinski-Tang.*