

**БЪРЗОДЕЙСТВИЕ НА LIBRE CALC ПРИ ОПТИМИЗАЦИЯ НА  
МНОГОМЕРНИ НЕЛИНЕЙНИ ФУНКЦИИ****Тодор Балабанов**

*Българска академия на науките  
Институт по информационни и комуникационни технологии  
ул. "акад. Георги Бончев", блок 2, гр. София 1113, България  
todor.balabanov@iict.bas.bg*

**LIBRE CALC PERFORMANCE IN OPTIMIZATION OF  
MULTIDIMENSIONAL NON-LINEAR FUNCTIONS****Todor Balabanov**

*Bulgarian Academy of Sciences  
Institute of Information and Communication Technologies  
acad. Georgi Bonchev Str., block 2, 1113 Sofia, Bulgaria  
todor.balabanov@iict.bas.bg*

**Abstract**

*There are various optimization problems in real-life practice. When it comes to linear optimization, high efficient exact numerical methods are developed. The actual optimization challenge comes with nonlinear optimization problems. There are infinite nonlinearities, but even the simplest is problematic enough. This study's well-known nonlinear benchmark functions (Rastrigin, Sphere, Rosenbrock, Styblinski–Tang) are optimized with LibreOffice Calc's nonlinear solver. Even though there are attempts at Libre's nonlinear solver to be rewritten purely in C++, the current implementation is Java based hybrid algorithm of Differential Evolution and Particle Swarm Optimization. This study aims to estimate the performance of the currently used nonlinear solver. The study results would be a valuable point of comparison when the newly developed solver comes into life.*

**Keywords:** differential evolution, particle swarm optimization, non-linear optimization, LibreOffice Calc.

**ВЪВЕДЕНИЕ**

Съществуват множество видове оптимизационни задачи, в реалната практика. За линейните оптимизационни задачи (линейна целева функция и линейни ограничения) са разработени множество ефективни, точни числени методи [1]. Сериозните трудности възникват, когато оптимизационните задачи съдържат някаква форма на нелинейност [2]. Съществуват безкрайно много възможности за участие на нелинейни функции в нелинейните оптимизационни задачи, но дори най-опростената нелиней-

ност (примерно втора степен) създава достатъчно трудности.

За нуждите на настоящото научно изследване са избрани четири много популярни тестови функции - Rastrigin, Sphere, Rosenbrock, Styblinski–Tang [3]. Тези четири функции се характеризират с това, че дават възможност да се изчисляват при голяма стойност за измеренията (оси в многомерно пространство). Търсенето на оптимални и субоптимални стойности се извършва с LibreOffice Calc Nonlinear Solver [4].

По настояще, съществува средно-срочен план на екипа, разработващ солвъра, той да бъде пренаписан изцяло на програмния език C++ [5]. Въпреки тези усилия, тази работа все още не е завършена и в актуалните версии на офис пакета се използва солвър на Java, който изпълнява хибриден алгоритъм за глобална евристична оптимизация. В състава на хибридният алгоритъм влизат еволюция на разликите (Differential Evolution) и рояк от частици (Particle Swarm Optimization) [6].

Целта на настоящото научно изследване е да установи бързодействието на текущо използвания нелинеен солвър в LibreOffice Calc. Извършените експерименти и получените резултати биха били от изключителна ползност, когато бъде завършен C++ варианта на солвъра и бъде необходимо да се потвърди неговата ефективност.

Структурата на доклада е както следва: Част първа е въведение; Част втора представя хибридният алгоритъм, заложен в LibreOffice Calc Nonlinear Solver; Част трета е посветена на проведените експерименти и някои от получените резултати; Част четвърта съдържа заключение и препоръки за бъдещи изследвания.

## **ХИБРИДЕН АЛГОРИТЪМ ЗА ГЛОБАЛНА ОПТИМИЗАЦИЯ**

В офис пакета LibreOffice Calc са дадени възможности за използването на няколко нелинейни оптимизационни алгоритъма. От тях DEPSO солвърът се е оказал през годините, като най-ефективен за решаването на нелинейни оптимизационни задачи. DEPSO хибридният алгоритъм използва в съчетание два мета евристични алгоритъма за глобална оптимизация – еволюция на разликите и рояк от частици [7]. И двата алгоритъма са популационни, като еволюцията на разликите спада към групата на генетичните алгоритми. Активирането на всеки от двата алгоритъма става на случаен принцип, с предварително зададена вероятност. По-подробно, двата алгоритъма се използват с шансове 50 на 50.

Еволюция на разликите е алгоритъм за глобална оптимизация [8], който разчита на опити за подобряване на вече налични кан-

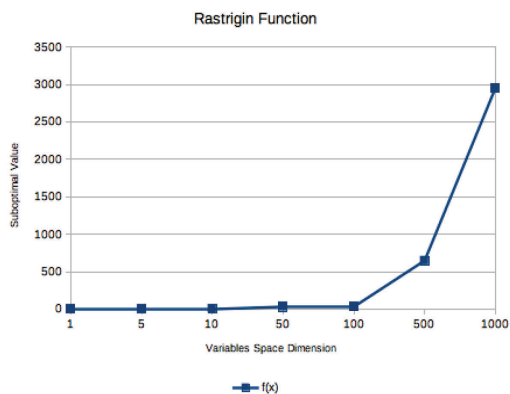
дидат решения. При подбора за подобряване на кандидат решенията се разчита на числена оценка, до колко тези решения са приемливи. Алгоритъмът не разчита на градиент и позволява използване в задачи с изключително висока размерност на пространството на променливите. Тъй като алгоритъмът е стохастичен, то той не гарантира достигането до глобалния оптимум. Характерно за еволюция на разликите е, че намира най-ефективно приложение в непрекъснати пространства. Оптимизационният процес се осъществява с множество на кандидат-решенията, наречено популация. От тези решения се избират родители и се създава потомство. В популацията остават тези индивиди, които имат по-добра оценка, получена чрез пресмятане на целевата функция. Рекомбинацията на родителите най-често се случва с помощта на операции за кръстосване и мутация. Фундаментална разлика при еволюция на разликите от генетичните алгоритми е, че мутацията се случва по всички елементи на съответния вектор, а не само на отделен елемент.

Роякът от частици е алгоритъм за глобална оптимизация [9], който също разчита на множество от кандидат-решения. Решенията се онагледяват като частици, движещи се из пространството на променливите. Движението се описва с текуща позиция и скорост. Движението на частиците силно се влияе от известното локално, най-добро решение и от известното глобално, най-добро решение. И двете (локално, глобално) решения се обновяват, ако в процеса на оптимизация бъдат открити по-добри от тях. За да се приложи оптимизация с рояка от частици, не е необходимо предварително знание за задачата и алгоритъмът може да се стартира на принципа на „черната кутия“. За рояка от частици, както и за еволюцията на разликите, е характерно, че може да се използва при задачи с много висока степен на измеренията. Също не се нуждае от градиент.

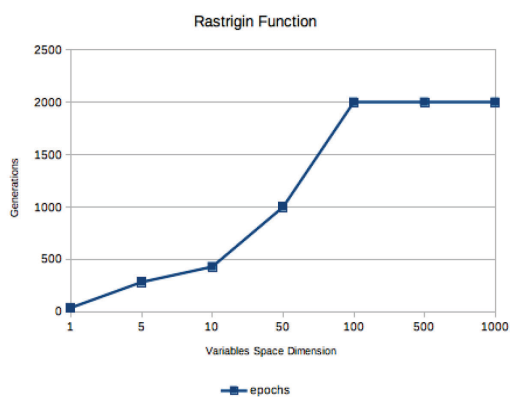
И двата алгоритъма, като имплементация в LibreOffice Calc, имат серия конфигурационни параметри, които в настоящото научно изследване се използват с подразбиращите им се стойности.

## ЕКСПЕРИМЕНТИ И РЕЗУЛТАТИ

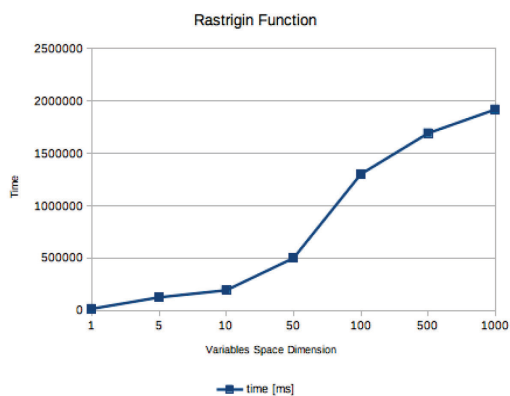
Всички експерименти са извършени на настолна компютърна конфигурация Intel Core i5 2.3GHz, 1 CPU 2 ядра, 8GB RAM, операционна система macOS High Sierra 10.13.6 и офис пакет LibreOffice 7.2.7.2.



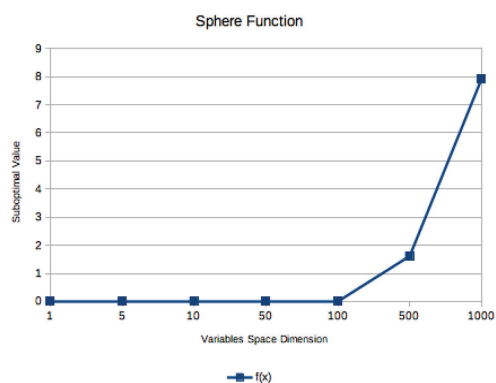
Фиг. 1. Постигнати субоптимални стойности при функцията Rastrigin



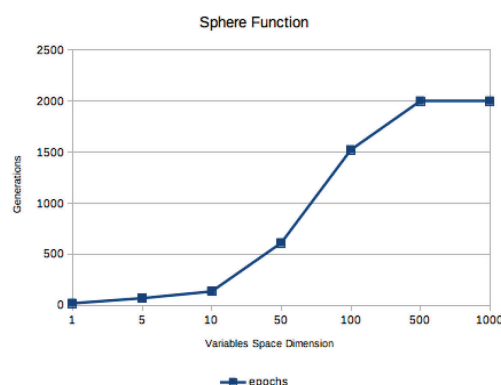
Фиг. 2. Брой поколения при функцията Rastrigin



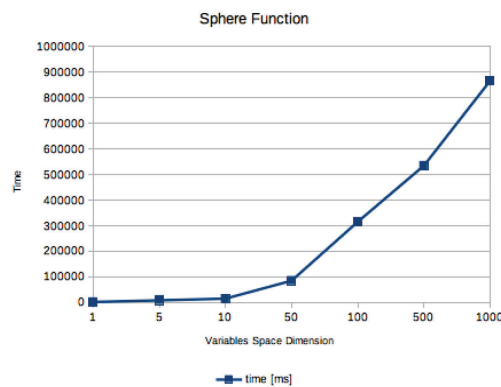
Фиг. 3. Изчислително време при функцията Rastrigin



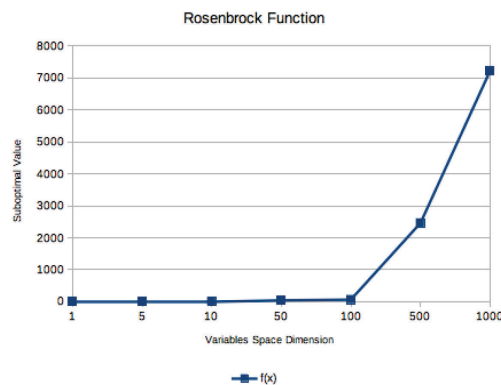
Фиг. 4. Постигнати субоптимални стойности при функцията Sphere



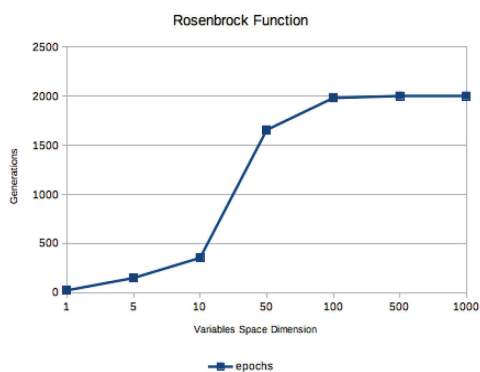
Фиг. 5. Брой поколения при функцията Sphere



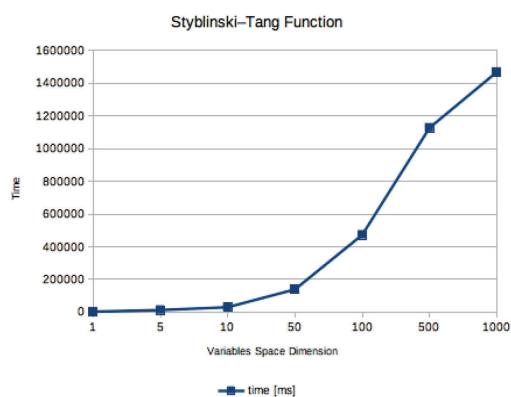
Фиг. 6. Изчислително време при функцията Sphere



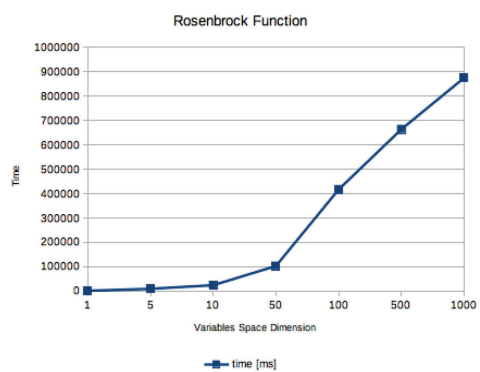
Фиг. 7. Постигнати субоптимални стойности при функцията Rosenbrock



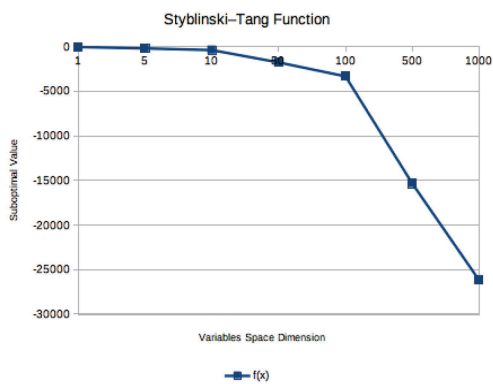
**Фиг. 8.** Брой поколения при функцията *Rosenbrock*



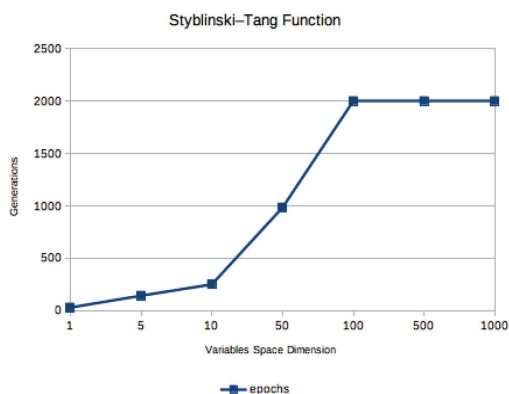
**Фиг. 12.** Изчислително време при функцията *Styblinski-Tang*



**Фиг. 9.** Изчислително време при функцията *Rosenbrock*



**Фиг. 10.** Постигнати субоптимални стойности при функцията *Styblinski-Tang*



**Фиг. 11.** Брой поколения при функцията *Styblinski-Tang*

За всяка от четирите функции са извършени независими експерименти. Всяка от четирите функции е използвана при следните измерения: 1, 5, 10, 50, 100, 500 и 1000. За над 1000 мерно пространство, работата на LibreOffice Calc става толкова забавена, че не позволява получаването на резултати в приемливо изчислително време на използваната настолна компютърна конфигурация.

И при четирите функции, до към 100 мерно пространство, солвърът намира близки до глобалния оптимум решения (Фиг. 1,4,7,10). Този резултат показва, че с рязкото нарастване на измеренията, достигането до глобалния оптимум се затруднява. За три от функциите, до към 100 мерно пространство, солвърът изпада в стагнация и не използва предварително заложените 2000 оптимизационни цикъла (Фиг. 2,8,11). Изключение прави само сферичната функция, при която се приключва със стагнация до към 500 измерения (Фиг. 5). И при четирите функции, изчислителното време (ограничено за максимум 2000 оптимизационни цикъла) стремглаво нараства, в право-пропорционална зависимост от нарастването на измеренията в многомерното пространство (Фиг. 3, 6, 9, 12).

Получените замервания силно се влияят от начина по който са представени функциите в LibreOffice Calc. За всяко измерение, стойност се пресмята с помощта на формула, на отделен ред в работния лист. Пресмятането на голям брой (при най-тежкия случай 1000) формули и сумирането им за получаване на единична целева клетка, води до значително забавяне на работата.

Отнетото изчислително време при по-високите измерения най-вече се дължи на механизма по който електронната таблица изчислява формулите, а не толкова на затруднение при смятането на самите тестови функции.

## ЗАКЛЮЧЕНИЕ

В настоящото научно изследване се представят резултати за постигнатата ефективност в LibreOffice Calc Nonlinear Solver. Експериментите за установяването на тази ефективност се базират на четири много известни тестови функции. Резултатите показват, че с нарастването на размерността на пространството на променливите, намаляват и възможностите на оптимизационния модул да определи глобалния оптимум. Второ много важно заключение от получените резултатите е, че ефективността на оптимизационния модул изключително много зависи от първоначалната точка, която служи за отправна точка, при стартирането на процеса по оптимизация.

Като насоки за бъдещи изследвания може да се посочат възможностите за изследване на другите нелинейни оптимизационни модули в LibreOffice Calc, както и на C++ модула, който е в текущ процес на разработване [10].

## БЛАГОДАРНОСТИ

This research is funded by Velbazhd Software LLC. It is partially supported by the Ministry of Education and Science of the Republic Bulgaria under the National Science Program “Intelligent animal husbandry”, grant agreement No. D01-62/18.03.2021/, the National Research Programme “Young scientists and postdoctoral students” approved by DCM No. 577/17.08.2018, and the Bulgarian National Science Fund by the project “Mathematical models, methods and algorithms for solving hard optimization problems to achieve high security in

communications and better economic sustainability”, KP-06-N52/7/19-11-2021.

## REFERENCE

- [1] Shu-Cherng Fang and Sarat Puthenpura. 1993. Linear optimization and extensions: theory and algorithms. Prentice-Hall, Inc., USA.
- [2] L. E. Scales. 1985. Introduction to non-linear optimization. Springer-Verlag, Berlin, Heidelberg.
- [3] Wang Peng, Mixed Particle Swarm Optimization Algorithm with Multistage Disturbances. *Frontiers in Genetic and Evolutionary Computation* (2019) 1: 1-4. <http://dx.doi.org/10.23977/fgec.2019.11001>
- [4] Dasović, B., Klanšek, U., Comparison of spreadsheet-based optimization tools applied to construction site layout problem, *IOP Conference Series: Materials Science and Engineering*, vol. 1209, no. 1, 2021. <http://dx.doi.org/10.1088/1757-899X/1209/1/012010>
- [5] Vasco Alves, 70399 - Port Solver for Nonlinear Programming to native code, 2013. [https://bugs.documentfoundation.org/show\\_bug.cgi?id=70399](https://bugs.documentfoundation.org/show_bug.cgi?id=70399), Last visited 07 Aug 2022
- [6] LibreOffice Calc DEPSO NLP Solver, <https://github.com/LibreOffice/core/blob/master/nlpsolver/ThirdParty/EvolutionarySolver/src/net/adaptivebox/deps/DEPSAgent.java>, Last visited 07 Aug 2022
- [7] M. Pant, R. Thangaraj, C. Grosan and A. Abraham, "Hybrid Differential Evolution - Particle Swarm Optimization Algorithm for Solving Global Optimization Problems," 2008 Third International Conference on Digital Information Management, 2008, pp. 18-24. <https://doi.org/10.1109/ICDIM.2008.4746766>
- [8] Price, K.V. (2013). Differential Evolution. In: Zelinka, I., Snásel, V., Abraham, A. (eds) *Handbook of Optimization*. Intelligent Systems Reference Library, vol 38. Springer, Berlin, Heidelberg. [https://doi.org/10.1007/978-3-642-30504-7\\_8](https://doi.org/10.1007/978-3-642-30504-7_8)
- [9] Poli, R., Kennedy, J. & Blackwell, T. Particle swarm optimization. *Swarm Intell* 1, 33–57 (2007). <https://doi.org/10.1007/s11721-007-0002-0>
- [10] LibreOffice Calc Native Solver, <https://github.com/LibreOffice/core/tree/master/sccomp/source/solver>, Last visited 07 Aug 2022