FR-DE Excel Solver: Differential Evolution with Deb's feasibility rules for solving constrained optimization problems in civil engineering

Sử dụng thuật toán tiến hóa vi phân kết hợp các quy tắc của Deb để giải các bài toán tối ưu hóa có điều kiện ràng buộc trong ngành xây dựng

Hoàng Nhật Đức

Viện nghiên cứu phát triển công nghệ cao, Đại học Duy Tân
Institute of Research and Development, Duy Tan University
Email tác giả: hoangnhatduc@dtu.edu.vn

Abtract
Constrained optimization is an important task in civil engineering. The objective of this task is to determine a solution with the most desired objective function value that guarantees the satisfactions of constraints. The Differential Evolution (DE) is a powerful evolutionary algorithm for solving global optimization tasks. Our research develops an optimization model based on the DE and feasibility rules proposed by Deb [1]. To facilitate the application of the optimization model, a DE Solver, named as FR-DE, has been developed in Microsoft Excel VBA platform. Experimental outcomes with several basic constrained design problems prove that the FR-DE developed in this study can be a useful tool for solving constrained optimization problems.
Key words: Constrained optimization, Differential Evolution; Feasibility Rules; Evolutionary Algorithm.

Tóm tắt
Tối ưu hóa có ràng buộc là một nhiệm vụ quan trọng trong xây dựng dân dụng. Mục tiêu của nhiệm vụ này là xác định một giải pháp có giá trị hàm mục tiêu tối ưu, đồng thời đảm bảo sự thỏa mãn của các ràng buộc. Thuật toán tiến hóa vi phân (DE) là một thuật toán tiến hóa mạnh để giải quyết các nhiệm vụ tối ưu hóa toàn cục. Nghiên cứu của chúng tôi phát triển một mô hình tối ưu hóa dựa trên và các luật về tính khả thi của giải pháp do Deb [1] đề xuất. Để tăng cường việc áp dụng mô hình tối ưu hóa, một chương trình tính toán DE Solver, được đặt tên là FR-DE, đã được phát triển trong nền tảng VBA của Microsoft Excel. Kết quả thử nghiệm với một số bài toán cơ bản chứng minh rằng FR-DE được phát triển trong nghiên cứu này có thể là một công cụ hữu ích để giải quyết các vấn đề tối ưu hóa bị ràng buộc.
Từ khóa: Tối ưu hóa có điều kiện ràng buộc, Tiến hóa vi phân; Quy tắc khả thi; Thuật toán tiến hóa.

1. Introduction

Constrained optimization problems are ubiquitous in civil engineering. Civil engineers have to resolve complex decision making tasks regarding to project financial planning [2,3], project schedule optimization [4,5], time-cost trade-off [6], production planning [7], resource utilization optimization [8-10], and various structural optimization problems [11-13]. A constrained optimization problem is typically more challenging than an unconstrained one. It is because the task of finding optimal solutions must be performed concurrently with identifying feasible domains of solutions.

A general form of a constrained optimization problem can be presented as follows [14,15]:

Find min. of $f(x)$:

$$f(x_1, x_2, x_d,\dots,x_D),\ d = 1,2,\dots,D \tag{1}$$

Subjected to:

$$g_q(x_1, x_2, x_d,\dots,x_D) \le 0,\ \mathrm{d} = 1,2,\dots,D,\ q = 1,2,\dots,M \tag{2}$$

$$h_r(x_1, x_2, \mathbf{x_d},\dots,\mathbf{x_D}) = 0,\ \mathrm{d} = 1,2,\dots,D,\ r = 1,2,\dots,N \tag{3}$$

$$x_d^L \le x_d \le x_d^U \tag{4}$$

where, $f(x_1, x_2,\dots,x_d)$ is the objective function. $x_1, x_2,\dots,x_d$ are called decision variables. $g_q(x_1, x_2,\dots,x_d)$ and $h_r(x_1, x_2,\dots,x_d)$ are inequality and equality constraints. x_d^L, x_d^U are lower and upper boundaries of x_d. D denotes the number of decision variables. M and N are the numbers of inequality and equality constraints.

For dealing with constrained optimization, penalty functions are commonly employed [15,1,16,17]. Nhat-Duc, Cong-Hai [13] developed a DE based constrained optimization solver using the penalty functions. The methods based on the penalty functions are easy to construct and implement; they can deliver acceptable performances in various case studies [5,7]. However, this type of constraint handling also suffers from certain drawbacks including the difficulty in selecting penalty coefficients [17].

To resolve the drawbacks of the penalty function methods, Deb [1] put forward an efficient constraint handling algorithm based on feasibility rules. The advantage of Deb's feasibility rules is that they can be directly integrated into the selection phase of evolutionary algorithms. Therefore, evolutionary algorithms used with these feasibility rules are free from the difficulties of selecting penalty coefficients as presented in the conventional penalty function approaches.

The spreadsheets in Microsoft Excel represent a useful tool for civil engineering calculation. However, the number of open Excel solvers that are capable of solving constrained optimization is still very limited. This study developed a DE Solver named as FR-DE to ease the application of the optimization model in civil engineering and other disciplines. The newly developed program has been tested with several basic constrained optimization problems.

2. Research Methodology

2.1 Differential Evolution (DE)

Given that the problem at hand is to minimize a cost function *f(X)*, where the number of decision variables is D, each stages of DE [18,19] can be described in detail. The whole process of the DE algorithm is repeated in G_{max} number of generations.

(i) Initialization*:* In this step, the search process begins by randomly generating NP number of D-dimensional parameter vectors $X_{i,g}$ where i = 1, 2, …, NP and g represents the current generation. Herein, NP does not change during the optimization process. Thus, these individuals can be generated in the following manner [8]:

$$X_{i,0} = LB + rand[0,1] \times (UB - LB) \quad (5)$$

where $X_{i,0}$ denotes the decision variable i at the first generation. *rand[0,1]* is a uniformly distributed random number between 0 and 1. *LB* and *UB* represent two vectors of lower bound and upper bound for the decision variables.

(ii) Mutation*:* In this step, a vector in the current population (or parent) called a target vector is selected. The terms "parent" and "target vector" are used interchangeably. For each target vector, a mutant vector is created in the following way:

$$V_{i,g+1} = X_{r1,g} + F(X_{r2,g} - X_{r3,g}) \quad (6)$$

where *r1*, *r2*, and *r3* denote three random indexes lying between 1 and *NP*. These 3 randomly chosen integers are selected to be different from the index *i of* the target vector. *F* represents the mutation scale factor, which controls the amplification of the differential variation between $X_{r2,g}$ and $X_{r3,g}$. $V_{i,g+1}$ denotes the newly created mutant vector.

(iii) Crossover*:* This step is to diversify the current population by exchanging components of target vector and mutant vector. A new vector, named as trial vector, is created. The trial vector is also called the offspring. The trial vector is generated as follows:

$$U_{j,i,g+1} = \begin{cases} V_{j,i,g+1}, & if \ rand_j \le Cr \ or \ j = rnb(i) \\ X_{j,i,g}, & if \ rand_j > Cr \ and \ j \ne rnb(i) \end{cases} \quad (7)$$

where $U_{j,i,g+1}$ denotes the trial vector. j is the index of element for any vector. $rand_j$ denotes a uniform random number lying between 0 and 1. *Cr* represents the crossover probability, which is needed to be determined by users. *rnb(i)* is a randomly chosen index of $\{1,2,...,NP\}$ which guarantees that at least one parameter from the mutant vector ($V_{j,i,g+1}$) must be copied to the trial vector ($U_{j,i,g+1}$).

(iv) Selection*:* The trial vector is compared to the target vector in this step. If the trial vector has a lower objective function value than its parent, the trial vector replaces the position of the target vector. The selection operator can be described as follows:

$$X_{i,g+1} = \begin{cases} U_{i,g} \ if \ f(U_{i,g}) \le f(X_{i,g}) \\ X_{i,g} \ if \ f(U_{i,g}) > f(X_{i,g}) \end{cases} \quad (8)$$

2.2 Feasibility Rule Based Constraint Handling Method

Deb [1] proposes the following feasibility rules for coping with constrained optimization problems:

1. Among one feasible solution and one infeasible solution, the feasible solution always wins.
2. Among two feasible solutions, the one having lower objective function value is preferred.
3. Among two infeasible solutions, the one having smaller degree of constraint violation is considered to be better.

Based on the aforementioned feasibility rules, the formulation of the objective function is revised as follows:

$$F(X) = \begin{cases} F(X) \ if \ g_j(x) \geq 0 \ \forall j \\ f_{\max} + \sum_{j=1}^{m} g_j(x) \end{cases} \quad (9)$$

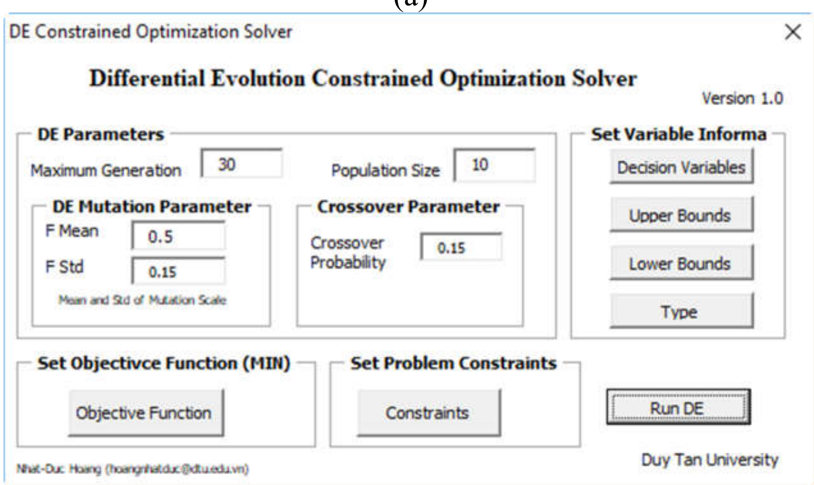
where f_{max} is the objective function value of the worst solution in the set of feasible ones.

3. Applications of the FR-DE-Excel Solver

The FR-DE-Excel Solver tool has been developed in Visual Basic for Applications (VBA) and can be accessed from the Excel ribbon entitled "***Open_DE_Constr_Solver***" (see **Fig. 1a**). The graphical user interface of the program is shown in **Fig. 1b**. The user needs to define the decision variables, upper bounds, lower bounds, type (real, integer, or binary), constraints, and the objective function of the problem. It is noted that all of the constraints must be described in the following form:

$$G(x) \geq 0 \quad (10)$$

(a)



(b)

Fig 1. FR-DE-Excel Solver: (a) Open the solver from Excel Ribbon and (b) Graphical user interface of the program

To illustrate the application of the newly developed tool, the following simple constrained optimization problem is used:

Min. $f(x,y) = (x-0.8)^2 + (y-0.3)^2$

S.t. $g_1(x,y) = 1 - [(x-0.2)^2 + (y-0.5)^2]/0.16 \quad \geq \quad 0$

$g_2(x,y) = [(x+0.5)^2 + (y-0.5)^2]/0.81 - 1 \quad \geq \quad 0$

	A	B	C	D	E
1		Optimal Solution	Lower Bounds	Upper Bounds	Type
2	X1 =	0.59	0.00	1.00	1.00
3	X2 =	0.45	0.00	1.00	1.00
4	F =	0.07			
5					
6		LHS		RHS	
7	Constraint 1	0.03	≥	0.00	
8	Constraint 2	0.47	≥	0.00	
9					
10		Note for variable type			
11	0	Binary	LB = 0, UB = 1		
12	1	Real			
13	2	Integer			

Fig 2. Optimization result of FR-DE-Excel Solver (Application 1)

In the next application, the FR-DE-Excel Solver is applied to optimize a simple structure optimization problem (see **Fig. 3**). A structure consisting of two bars is designed to withstand a force F. With α = 30°, the optimization problem is formulated as follows [23]:

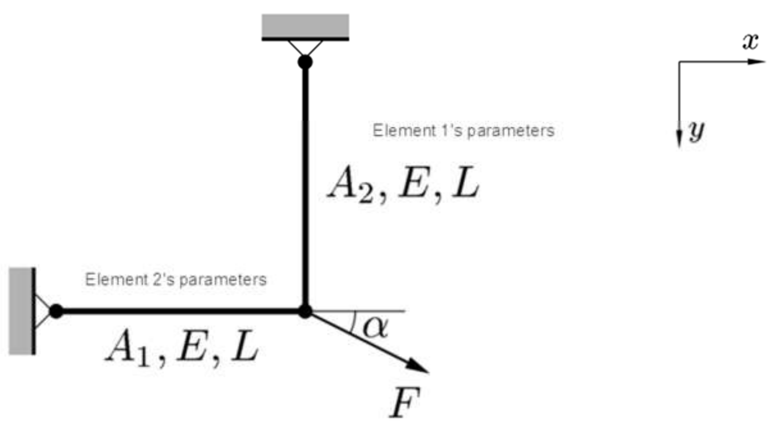
Min $(A1+A2).\gamma.L$ (11)

S.t:

$A1 \geq F\cos(\alpha)/\sigma\_allow$ (12)

$A2 \geq F\sin(\alpha)/\sigma\_allow$ (13)

where d1 và d2 are the two decision variables which are diameters of the two bars. L is the length of the bar. γ is the density of the material. σ_allow is the allowable stress. A1 and A2 are the areas of the bars.



	A	B	C	D	E	F	G	H	I	J
1	Decision Variables	d1=	6.6413	mm	A1 =	34.6410	mm^2	Gama=	0.000007850	kg/mm^3
2		d2=	5.0529	mm	A2 =	20.0527	mm^2	L =	1000	mm
3								F =	10000	N
4	Objective Function							σ_allow =	250.0000	MPa
5	Min f =	0.429345526	kg					a=	0.5236	Rad
6										
7	Lower Bound	d1	1.00	mm		Constraint 1	σ1=	F x cos(a)/A1	<=	σ_allow
8		d2	1.00	mm		Constraint 2	σ2=	F x sin(a)/A2	<=	σ_allow
9	Upper Bound	d1	30.00	mm						
10		d2	30.00	mm		Constraint 1	0.00	>=	0.0000	
11	Variable Type	d1	1.00	(real)		Constraint 2	0.66	>=	0.0000	
12		d2	1.00	(real)						
13										

Fig 3. Optimization result of FR-DE-Excel Solver (Application 2)

Using the FR-DE-Excel Solver tool with the number of members and generations of DE algorithm is 12 and 100, the optimization results of the first and second applications are provided in **Fig. 2** and **Fig. 3**. As can be seen from these two figures, the Excel Solver based on DE and Deb's feasibility rules can help to find good values of the decision variables which result in low value of the objective function and satisfy all the constraints.

4. Conclusion

In this study, a FR-DE-Excel Solver based on the DE algorithm and the feasibility rules is developed to solve constrained optimization problems. The FR-DE-Excel Solver is programmed in VBA language. The users can easily formulate the optimization problems and solve them in Excel. Two simple applications are used to demonstrate the effectiveness of the FR-DE-Excel Solver. Thus, the newly developed tool can be highly useful for engineers in dealing with complex optimization tasks.

Supplementary material

Reference

1. Deb K (2000) An efficient constraint handling method for genetic algorithms. Computer Methods in Applied Mechanics and Engineering 186 (2):311-338. doi:https://doi.org/10.1016/S0045-7825(99)00389-8

2. LAM KC, HU T, CHEUNG SO, YUEN RKK, DENG ZM (2001) Multi-project cash flow optimization: non-inferior solution through neuro-multiobjective algorithm. Engineering, Construction and Architectural Management 8 (2):130-144. doi:doi:10.1108/eb021176

3. Donkor EA, Duffey M (2013) Optimal Capital Structure and Financial Risk of Project Finance Investments: A Simulation Optimization Model With Chance Constraints. The Engineering Economist 58 (1):19-34. doi:10.1080/0013791X.2012.742948

4. Hoàng NĐ, Nguyễn QL, Phạm QN (2015) Tối ưu hóa tiến độ và chi phí cho dự án xây dựng sử dụng thuật toán tiến hóa vi phân. Tạp Chí Khoa Học và Công Nghệ, Đại Học Duy Tân 1 (14):135–141

5. Hoang N-D (2014) NIDE: A Novel Improved Differential Evolution for Construction Project Crashing Optimization. Journal of Construction Engineering 2014:7. doi:10.1155/2014/136397

6. Yang I-T (2007) Using Elitist Particle Swarm Optimization to Facilitate Bicriterion Time-Cost Trade-Off Analysis. Journal of Construction Engineering and Management 133 (7):498-505. doi:doi:10.1061/(ASCE)0733-9364(2007)133:7(498)

7. Hoang N-D, Nguyen Q-L, Pham Q-N (2015) Optimizing Construction Project Labor Utilization Using Differential Evolution: A Comparative Study of Mutation Strategies. Advances in Civil Engineering 2015:8. doi:10.1155/2015/108780

8. Tran H-H, Hoang N-D (2014) A Novel Resource-Leveling Approach for Construction Project Based on Differential Evolution. Journal of Construction Engineering 2014:7. doi:10.1155/2014/648938

9. Cheng M-Y, Tran D-H, Hoang N-D (2017) Fuzzy clustering chaotic-based differential evolution for resource leveling in construction projects. Journal of Civil Engineering and Management 23 (1):113-124. doi:10.3846/13923730.2014.982699

10. El-Rayes K, Jun DH (2009) Optimizing Resource Leveling in Construction Projects. Journal of Construction Engineering and Management 135 (11):1172-1180. doi:doi:10.1061/(ASCE)CO.1943-7862.0000097

11. Govindaraj V, Ramasamy JV (2005) Optimum detailed design of reinforced concrete continuous beams using Genetic Algorithms. Computers & Structures 84 (1):34-48. doi:https://doi.org/10.1016/j.compstruc.2005.09.001

12. Coello Coello CA, Christiansen AD, Hernández FS (1997) A simple genetic algorithm for the design of reinforced concrete beams. Engineering with Computers 13 (4):185-196. doi:10.1007/bf01200046

13. Nhat-Duc H, Cong-Hai L (2019) Sử dụng thuật toán tiến hóa vi phân cho các bài toán tối ưu hóa kết cấu với công cụ DE-Excel solver. DTU Journal of Science and Technology 03 (34):97-102

14. Reklaitis GV, Ravindran A, Ragsdell KM (1983) Engineering Optimization Methods and Applications. Wiley, New York

15. Hoàng NĐ, Vũ DT (2015) Tối ưu hóa kết cấu có điều kiện ràng buộc sử dụng thuật toán bầy đom đóm và các hàm phạt. Tạp Chí Khoa Học và Công Nghệ, Đại Học Duy Tân 2 (15):75–84

16. Kramer O (2010) A Review of Constraint-Handling Techniques for Evolution Strategies. Applied Computational Intelligence and Soft Computing 2010. doi:10.1155/2010/185063

17. John RM, Robert GR, David BF (1995) A Survey of Constraint Handling Techniques in Evolutionary Computation Methods. In: Evolutionary Programming IV: Proceedings of the Fourth Annual Conference on Evolutionary Programming. MITP, p 1

18. Price K, Storn RM, Lampinen JA (2005) Differential Evolution - A Practical Approach to Global Optimization. Springer-Verlag Berlin Heidelberg. doi:10.1007/3-540-31306-0

19. Storn R, Price K (1997) Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces. Journal of Global Optimization 11 (4):341-359. doi:10.1023/a:1008202821328