Differential Evolution with ε constrained handling method developed in Excel VBA for solving optimization problems in civil engineering

Thuật toán tiến hóa vi phân sử dụng phương pháp ε phát triển trong Excel VBA để giải các bài toán tối ưu hóa có điều kiện ràng buộc trong ngành xây dựng

Hoàng Nhật Đức[1] và **Nguyễn Huy Thành**[2]

[1]Viện Nghiên Cứu Phát Triển Công Nghệ Cao, Đại học Duy Tân, Đà Nẵng
Institute of Research and Development, Duy Tan University
*Email:*hoangnhatduc@dtu.edu.vn

[2]Công Ty Quản Lý Cầu Đường Đà Nẵng, Đà Nẵng
Da Nang Road and Bridge Management Company, Da Nang City
Email: huythanh307@gmail.com

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 ε rules proposed by Takahama et al. [1]. To facilitate the application of the optimization model, a DE Solver, named as Epsilon-DE, has been developed in Microsoft Excel VBA platform. Experimental outcomes with several basic constrained design problems prove that the Epsilon-DE developed in this study can be a useful tool for solving constrained optimization problems.

Key words: Constrained Handling, Differential Evolution; ε Rules; Stochastic Search.

Tóm tắt

Từ khóa:

1. Introduction

Constrained optimization tasks, especially nonlinear and complex optimization ones, where objective functions are minimized or maximized under certain constraints, are very crucial and ubiquitously appear in the field of civil engineering. Civil engineers have to resort to capable metaheuristic algorithms to tackle a variety of complex decision making tasks including structural optimization [2,3], schedule optimization [4-7], resource utilization [8-10], etc. Notably, a constrained optimization task is typically more difficult than an unconstrained one; the reason is that the process of finding optimal solutions must be performed by metaheuristic algorithms within the feasible domains [11,12].

A constrained optimization task can be stated generally as follows [13,14]:

Min. $f(x)$:f(x_1, x_2, x_d,…,x_D), d = 1,2,…,D (1)

Subjected to:

$g_q(x_1, x_2, x_d,\ldots,x_D) \leq 0$, d = 1,2,…,$D$, q = 1,2,…,M (2)

$h_r(x_1, x_2, x_d,\ldots,x_D) = 0$, d = 1,2,…,$D$, r = 1,2,…,N (3)

$x_d^L \leq x_d \leq x_d^U$ (4)

where, f(x_1, x_2,…,x_d) represents the objective function. x_1, x_2,…,x_d denotes a set of decision variables. $g_q(x_1, x_2,\ldots,x_d)$ and $h_r(x_1, x_2,\ldots,x_d)$ are inequality and equality constraints, respectively. x_d^L and x_d^U denote lower and upper boundaries of x_d, respectively. D is the number of decision variables. M and N represent the numbers of inequality and equality constraints, respectively.

The conventional penalty function is often utilized for dealing with constrained optimization problems by converting them to unconstrained ones [14-17]. Nhat-Duc, Cong-Hai [18] developed a Differential Evolution (DE) based constrained optimization solver using the penalty functions. The penalty function approaches are simple and therefore easy to utilize. Nevertheless, this method cannot satisfactory handle complex constraints and requires a proper setting of the penalty factors [17]. To overcome such disadvantage of the conventional penalty function, Deb [15] proposes an feasibility rules based constraint handling method; this method has been integrated with the Differential Evolution and constructed as an Add-In used in Microsoft Excel by [19]. In this study, we aim at developing another Microsoft Excel Add-In that employs the DE algorithm and the ε constraint-handling method proposed by Takahama et al. [1]. The newly developed Excel Add-In has been tested with a simplified retaining wall design problem.

2. Research Methodology

2.1 Differential Evolution (DE)

Given that the problem at hand is to minimize an objective function *f(X)*, where the number of decision variables is D, the DE [20,21] algorithm for unconstrained optimization consists of three main steps: initialization, mutation, crossover, and selection. The searching process of the DE algorithm is repeated until a stopping condition is met. Usually, the algorithm terminates when the generation counters reach the maximum number generations (G_{max}). The four steps of the DE are shortly described as follows:

(i) Initialization***:*** This step randomly generates a set of PS value of D-dimensional vectors $X_{i,g}$ where i = 1, 2, …, PS and g is the generation counter.

(ii) Mutation: A target vector is selected. For each target vector, a mutant vector is created as follows:

$$V_{i,g+1} = X_{r1,g} + F(X_{r2,g} - X_{r3,g}) \tag{5}$$

where *r1*, *r2*, and *r3* are 3 random indexes ranging from 1 and *PS*. *F* is the mutation scale factor which is often selected as a fixed number (e.g. 0.5) or can be generated from a Gaussian distribution [22].

(iii) Crossover*:* A trial vector is created as follows:

$$U_{j,i,g+1} = \begin{cases} V_{j,i,g+1}, & if\ rand_j \le Cr\ or\ j = rnb(i) \\ X_{j,i,g}, & if\ rand_j > Cr\ and\ j \ne rnb(i) \end{cases} \tag{6}$$

where $U_{j,i,g+1}$ denotes the trial vector. j denotes the index of element for any vector. $rand_j$ represents a uniform random number of [0, 1]. *Cr* denotes the crossover probability which is often selected as a constant number (e.g. 0.8). *rnb(i)* denotes a randomly chosen index of $\{1,2,...,NP\}$.

(iv) Selection: The trial vector is compared to the target vector in this step according to the following rule:

$$X_{i,g+1} = \begin{cases} U_{i,g}\ if\ f(U_{i,g}) \le f(X_{i,g}) \\ X_{i,g}\ if\ f(U_{i,g}) > f(X_{i,g}) \end{cases} \tag{7}$$

2.2 The ε Constraint Handling Method

The ε constraint-handling method has been proposed by Takahama et al. [1]. Using this method, the constraint violation degree is defined either as the maximum of all constraints or the sum of all constraints as follows:

$$\phi(x) = \max\{\max_j \{0, \mathrm{g}_j(\mathrm{x})\}, \max_j |h_j(\mathrm{x})|\} \tag{8}$$

$$\phi(x) = \sum_j \|\max_j \{0, \mathrm{g}_j(\mathrm{x})\|^p + \sum_j \|\max_j |h_j(\mathrm{x})\|^p \tag{9}$$

where p denotes a positive integer.

Based on such definition of the constraint violation, the selection operation of the employed metaheuristic is revised as follows:

$$(f_1,\phi_1) <_\varepsilon (f_2,\phi_2) = \begin{cases} f_1 < f_2\ if\ \phi_1,\phi_2 \le \varepsilon \\ f_1 < f_2\ if\ \phi_1 = \phi_2 \\ \phi_1 < \phi_2, otherwise \end{cases} \tag{10}$$

3. The ε Constraint Handling DE (CHDE) Excel Solver Applications

The ε CHDE Excel Solver tool has been developed in Visual Basic for Applications (VBA). The graphical user interface of the Excel Solver is displayed in Fig. 1. The tool requires the decision variables, upper bounds, lower bounds, type (real, integer, or binary), constraints, and the objective function of the problem as input information. Notably, all of the constraints must be described in the following template: $G(x) \ge 0$ (11)

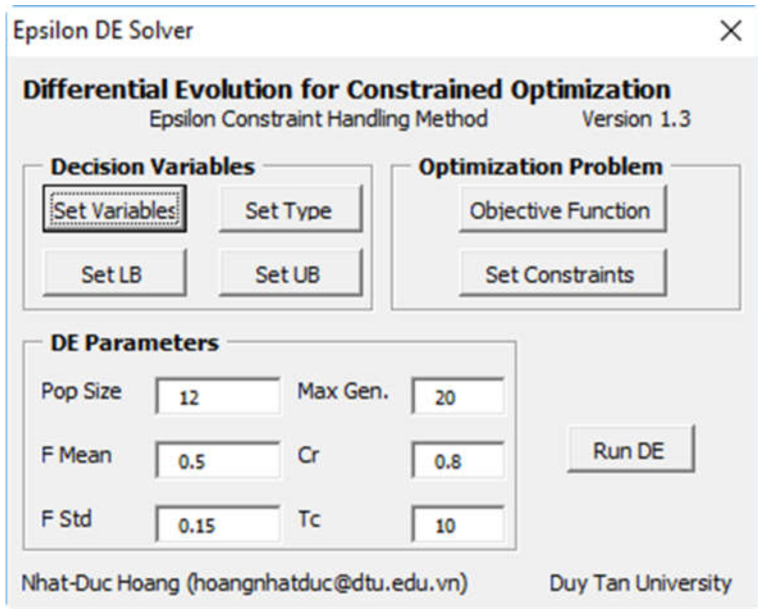

Fig 1. The ε CHDE Excel Solver

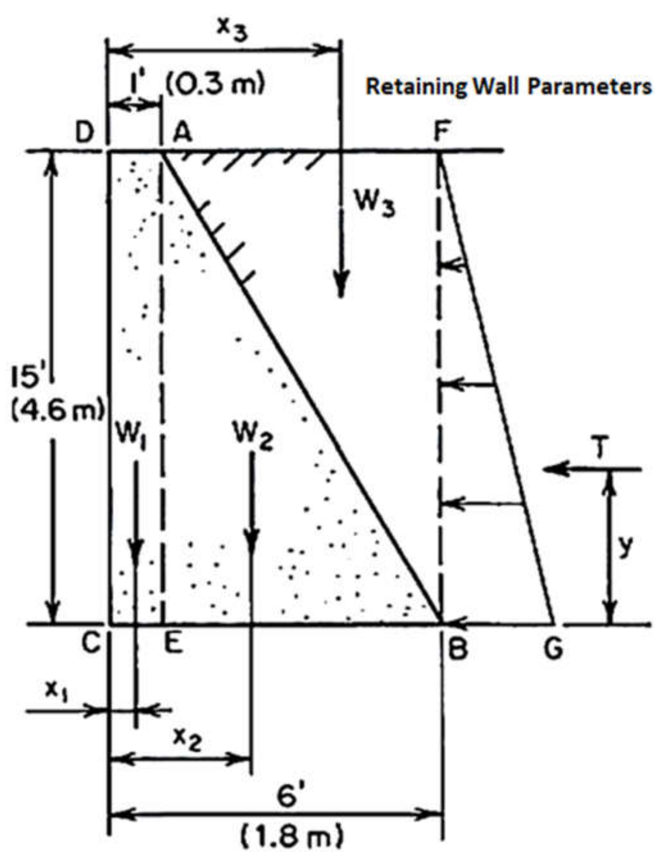

Fig 2. Illustration of the simplified retaining wall design problem (Adopted from [23])

The ε CHDE Excel Solver tool is applied to optimize the design of a simplified retaining wall [23] as illustrated in **Fig. 2**. The design variables of the problem are the lengths of the base and the top of the retaining wall. For more detail of the problem formulation, the readers are guided to the work of [23]. The optimization outcome performed by the newly developed tool is reported in **Fig. 3** with the number of population size = 12 and the maximum number of generations = 100. As can be seen from the figure, the Excel Solver based on DE and the ε rules can help to find the decision variables which result in low value of the objective function within the feasible domain.

	A	B	C	D	E	F	G
1	**Concrete Weight**	150	lb/ft3	23.56	kN/m3		
2	**Earth Weight**	100	lb/ft3	15.71	kN/m3		
3	**Coefficient of friction**	0.6					
4	**Coefficient of active earth pressure**	0.333					
5							
6	**Decision Variables**			**LB**	**UB**	**Type**	
7	EB	5.124678039	ft	3.00	10.00	1.00	
8	DA	1.002816214	ft	1.00	5.00	1.00	
9	DC	15	ft				
10	BF	15	ft				
11	AF	5.124678039	ft				
12	CE	1.002816214	ft				
13							
14	**Objective Function (MIN) F_Cost =**	**8021.599275**	lb (/1ft long)				
15							
16	W1 =	DC x DA x W_Concrete =	2256.34	lb			
17	W2 =	0.5 x Eb x AE x W_Concrete =	5765.26	lb			
18	W3 =	0.5 x AF x BF x W_Earth =	3843.51	lb			
19	**Sum_W =**		11865.11	lb			
20							
21	Distance from forces to C	W1	x1 =	CE/2 =	0.50	ft	
22		W2	x2 =	CE + EB/3 =	2.71	ft	
23		W3	x3 =	DA + 2xAF/3 =	4.42	ft	
24							
25	BG = C_ActiveEarthPress x BF * W_Earth x 1ft	=	499.5	lb/ft			
26	**Horizontal thrust T =**	0.5 x BG x BF =	3746.25	lb			
27							
28	**Maximum Frictional Force Preventing Sliding**						
29	Fm = C_Friction x Sum_W =	7119.064682	lb				
30							
31	**The moment of the Overtuning (Mo) =**	T x BG/3 =	18731.25	lb . ft			
32	**The moment of Stabilizing Forces (Ms) =**						
33	W1.x1 + W2.x2 + W3.x3 =	=	33746.71	lb . ft			
34							
35						**Constraints**	
36	**Factor of Safety Against Sliding (FSS) =**	Fm/T =	1.90	≥	1.9	1	0.00 ≥ 0
37	**Factor of Safety against Overtuning (FSO) =**	Ms/Mo =	1.80	≥	1.8	2	0.00 ≥ 0

Fig 3. Solving the constrained optimization problem using the ε CHDE Excel Solver tool

4. Conclusion

In this study, ε CHDE Excel Solver tool relied on the DE metaheuristic and the ε constraint handling method has been developed. The ε CHDE Excel Solver is programmed in VBA environment and can directly solve optimization problems formulated in Microsoft Excel. A simplified case of retaining wall design is employed to demonstrate the effectiveness of the ε CHDE Excel Solver. Hence, the newly constructed tool can be a useful tool for engineers in dealing with optimization problems.

Supplementary material

The Excel solver can be downloaded at:
https://github.com/NhatDucHoang/Epsilon_CHDE_ExcelSolver

Reference

1. Takahama T, Sakai S, Iwane N Solving Nonlinear Constrained Optimization Problems by the ε Constrained Differential Evolution. In: 2006 IEEE International Conference on Systems, Man and Cybernetics, 8-11 Oct. 2006 2006. pp 2322-2327. doi:10.1109/ICSMC.2006.385209

2. Shaqfa M, Orbán Z (2019) Modified parameter-setting-free harmony search (PSFHS) algorithm for optimizing the design of reinforced concrete beams. Structural and Multidisciplinary Optimization. doi:10.1007/s00158-019-02252-4

3. Senouci AB, Al-Ansari MS (2009) Cost optimization of composite beams using genetic algorithms. Advances in Engineering Software 40 (11):1112-1118. doi:https://doi.org/10.1016/j.advengsoft.2009.06.001

4. Monghasemi S, Nikoo MR, Khaksar Fasaee MA, Adamowski J (2015) A novel multi criteria decision making model for optimizing time–cost–quality trade-off problems in construction projects. Expert Systems with Applications 42 (6):3089-3104. doi:https://doi.org/10.1016/j.eswa.2014.11.032

5. Rogalska M, Bożejko W, Hejducki Z (2008) Time/cost optimization using hybrid evolutionary algorithm in construction project scheduling. Automation in Construction 18 (1):24-31. doi:https://doi.org/10.1016/j.autcon.2008.04.002

6. 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

7. 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

8. 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

9. 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

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. Coello CAC (2018) Constraint-handling techniques used with evolutionary algorithms. Paper presented at the Proceedings of the Genetic and Evolutionary Computation Conference Companion, Kyoto, Japan,

12. Coello Coello CA (2002) Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art. Computer Methods in Applied Mechanics and Engineering 191 (11):1245-1287. doi:https://doi.org/10.1016/S0045-7825(01)00323-1

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

14. 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

15. 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

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. 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

19. Hoang ND (2019) FR-DE Excel Solver: Differential Evolution with Deb's feasibility rules for solving constrained optimization problems in civil engineering. DTU Journal of Science and Technology 04 (35)

20. 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

21. 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

22. Hoang N-D, Tien Bui D, Liao K-W (2016) Groutability estimation of grouting processes with cement grouts using Differential Flower Pollination Optimized Support Vector Machine. Applied Soft Computing 45:173-186. doi:https://doi.org/10.1016/j.asoc.2016.04.031

23. Hicks TG (2007) Handbook of civil engineering calculations. McGraw-Hill