TẠP CHÍ KHOA HỌC & CÔNG NGHỆ ĐẠI HỌC DUY TÂN
DTU Journal of Science and Technology 02(45) (2021) 103-108

TRƯỜNG ĐẠI HỌC DUY TÂN
DUYTAN UNIVERSITY

Feasibility rule based differential evolution algorithm developed in visual C# for solving constrained optimization problems

Thuật toán tiến hóa vi phân dựa trên các quy luật khả thi phát triển với ngôn ngữ C# để giải các bài toán tối ưu hóa có ràng buộc

Hoang Nhat Duc[a,b*]
Hoàng Nhật Đức[a,b*]

[a]*Institute of Research and Development, Duy Tan University, Da Nang, 550000, Vietnam*
[a]*Viện Nghiên cứu và Phát triển Công nghệ Cao, Đại học Duy Tân, Đà Nẵng, Việt Nam*
[b]*Faculty of Civil Engineering, Duy Tan University, Da Nang, 550000, Vietnam*
[b]*Khoa Xây dựng, Trường Đại học Duy Tân, Đà Nẵng, Việt Nam*

(Ngày nhận bài: 22/03/2021, ngày phản biện xong: 26/03/2021, ngày chấp nhận đăng: 29/03/2021)

Abstract

This study develops an advanced tool for tackling constrained optimization problems based on an integration of feasibility rules and differential evolution metaheuristic. This tool aims at finding a solution with the most desired objective function value and concurrently satisfies all of the problem constraints. The optimization approach, named as feasibility rule based differential evolution (FRB-DE), has been developed in Microsoft Visual Studio with C# programming language. The newly developed tool has been tested with two optimization tasks in the field of civil engineering.

Key words: Differential evolution; Constrained optimization; Feasibility rules; Metaheuristic.

Tóm tắt

Nghiên cứu này phát triển một công cụ để giải quyết các vấn đề tối ưu hóa có ràng buộc dựa trên sự tích hợp các quy tắc khả thi và thuật toán tiến hóa vi phân. Công cụ được xây dựng để tìm ra giải pháp với giá trị hàm mục tiêu tốt nhất và đồng thời thỏa mãn tất cả các ràng buộc. Phương pháp tối ưu hóa, được đặt tên là tiến hóa vi phân dựa trên các quy tắc khả thi (FRB-DE), đã được phát triển trong Microsoft Visual Studio với ngôn ngữ lập trình C #. FRB-DE đã được thử nghiệm với hai bài toán tối ưu hóa cơ bản trong lĩnh vực xây dựng dân dụng.

Từ khóa: Tiến Hóa Vi Phân; Tối Ưu Hóa Có Ràng Buộc; Quy Tắc Khả Thi; Thuật Toán Tìm Kiếm.

1. Introduction

Civil engineers frequently encounter constrained optimization problems in various design tasks e.g. structural design [1-4], schedule/resource planning [5-8] etc. Constrained optimization tasks are generally

* *Corresponding Author:* Hoang Nhat Duc; Institute of Research and Development, Duy Tan University, Da Nang, 550000, Vietnam; Faculty of Civil Engineering, Duy Tan University, Da Nang, 550000, Vietnam
Email: hoangnhatduc@duytan.edu.vn

sophisticated since the best found optimal solutions must satisfy a set of pre-specified restrictions stated in the form of mathematical equations or inequalities [9]. To handle such challenges, scholars have resorted to metaheuristic to find near optimal solutions for various engineering design tasks [10-12]. Initially, a simple method of penalty functions can be used to handle such restrictions by incorporating the constraints into the objective function [13-15]. However, selecting penalty coefficients can be problematic for this approach [16]. Therefore, it is desirable to utilize advanced approaches that feature separation of objective functions and constraints.

Deb [17] proposes an efficient constraint handling algorithm based on three feasibility rules:

1. Considering one feasible solution and one infeasible solution, the feasible solution always wins.

2. Considering two feasible solutions, the one having lower objective function value is preferred.

3. Considering two infeasible solutions, the one having smaller degree of constraint violation is considered to be better.

Accordingly, using these rules proposed by Deb [17], information regarding the feasibility of solutions is directly included in the selection phase of metaheuristic algorithms. Moreover, this advanced method also eliminates the need of specifying penalty coefficients. Therefore, metaheuristic algorithms coupled with feasibility rules often result in good optimization performances. With such motivations, this study develops a computer program used for coping with constrained optimization problems. This program integrates the differential evolution metaheuristic [18] and the aforementioned feasibility rules proposed by Deb [17]. The feasibility rule based differential evolution (FRB-DE) has been developed with Visual C#.NET to facilitate its implementation. The capability of the newly developed tool has been verified with two basic constrained optimization problems.

2. Feasibility Rule Based (FRB) Differential Evolution (DE)

Given that the problem of interest is to minimize a cost function *f(X)*, where the number of decision variables is *D*, the optimization process of DE can be separated into three main steps:

(i) Mutation*:* In this step, a vector in the current population (or parent) called a target vector is selected. For each parent, a mutant vector is created as follows [18]:

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

where *r1*, *r2*, and *r3* are three random indexes lying between 1 and *NP*. *r1*, *r2*, and *r3* are selected to be different from the index *i of* the target vector. *F* is the mutation scale factor. $V_{i,g+1}$ represents a mutant vector. *NP* is the number of searching agents.

(ii) Crossover*:* A new vector, named as trial vector, is created as follows:

$$U_{j,i,g+1} = \begin{cases} V_{j,i,g+1}, & if \;\; rand_j \le Cr \;\; or \;\; j = rnb(i) \\ X_{j,i,g}, & if \;\; rand_j > Cr \;\; and \;\; j \ne rnb(i) \end{cases} \quad (2)$$

where $U_{j,i,g+1}$ is a trial vector. *j* denotes the index of element for any vector. $rand_j$ is a uniform random number lying between 0 and 1. *Cr* denotes the crossover probability. *rnb(i)* denotes a randomly chosen index in $\{1,2,...,NP\}$.

(iii) Selection: The selection phase is used to compare the fitness of the trial vector and the target vector. This phase is described as follows:

$$X_{i,g+1} = \begin{cases} U_{i,g} \text{ if } f(U_{i,g}) \leq f(X_{i,g}) \\ X_{i,g} \text{ if } f(U_{i,g}) > f(X_{i,g}) \end{cases} \quad (3)$$

Based on the aforementioned feasibility rules proposed by Deb [17], the formulation of the objective function in the DE metaheuristic is revised in the following manner:

$$F(X) = \begin{cases} F(X) \text{ if } g_j(x) \geq 0 \ \forall j \\ f_{\max} + \sum_{j=1}^{m} g_j(x) \end{cases} \quad (4)$$

where f_{max} denotes the objective function value of the worst feasible candidate.

Based on the DE metaheuristic and the set of feasibility rules, this study has developed the FRB-DE tool used for constrained optimization. This tool has been coded in with Visual C#.NET programming language within the Microsoft Visual Studio integrated development environment. Fig. 1 demonstrates the function interface implementing FRB-DE. The C# delegate type is used to define general functional forms of the objective function and constraints (refer to **Fig. 2**). Moreover, a C# class is used to store information of an optimization problem (refer to **Fig. 3**.).

```
public static List<double[,]> Optimize(GeneralObjFun ObjectiveFunction,
    GeneralObjFunWithConstraints ObjFunWithConstraints,
    GeneralConstraintFun ConstraintFunction, GeneralLB_Fun LB_Function,
    GeneralUB_Fun UB_Function, GeneralCheckContraintViolation CheckContraintViolation,
    GeneralConstraintViolationDegree ConstraintViolationDegree,
    int PopSize, int MaxGeneration)
{

    Stopwatch sw; sw = Stopwatch.StartNew();
    double Fmean = 0.5; double Fstd = 0.15; double CR = 0.8;
    double eps = 0.01;
    var rand = new Random();
```

Fig. 1 The function interface implementing FRB-DE

```
// Delegate functions: Objective function, Constraints, LB, and UB. ...
public delegate double GeneralObjFun(double[,] X);

// Cost Function with contraints
public delegate double GeneralObjFunWithConstraints(double[,] X, double fmax);

// Return constraint values
public delegate double[,] GeneralConstraintFun(double[,] X);

// Return lower boundaries of decision variables
public delegate double[] GeneralLB_Fun();

// Return upper boundaries of decision variables
public delegate double[] GeneralUB_Fun();

// Return constraint violation status
public delegate bool GeneralCheckContraintViolation(double[,] X);

// Return amount of constraint violation
public delegate double[,] GeneralConstraintViolationDegree(double[,] X);
```

Fig. 2 The use of delegate type

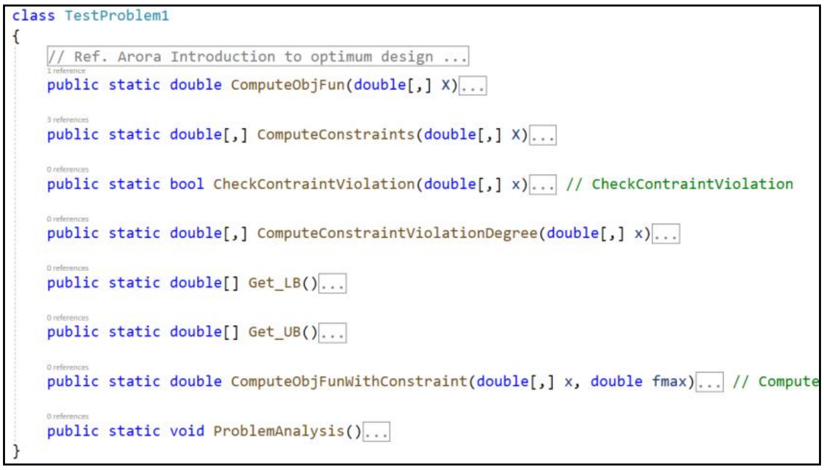
```
class TestProblem1
{
    // Ref. Arora Introduction to optimum design ...
    public static double ComputeObjFun(double[,] X)...
    public static double[,] ComputeConstraints(double[,] X)...
    public static bool CheckContraintViolation(double[,] x)... // CheckContraintViolation
    public static double[,] ComputeConstraintViolationDegree(double[,] x)...
    public static double[] Get_LB()...
    public static double[] Get_UB()...
    public static double ComputeObjFunWithConstraint(double[,] x, double fmax)... // Compute
    public static void ProblemAnalysis()...
}
```

Fig. 3 An optimization problem representation

3. Applications of the FRB-DE

In the first application, the FRB-DE is used to design the cross-section of a cantilever beam [19] shown in **Fig. 4**. There are two parameters of the beam cross-section needed to be specified: the depth of the section *w* and the thickness of the cross-section *t*. The beam is used to support a load of 20kN at its end. The beam is made of steel and its length is 1.5m. The ratio of w-to-t is less than 8 to prevent local buckling of the cross-section. It is desired to find a set of *t* and *w* resulting in a minimal cross-section area.

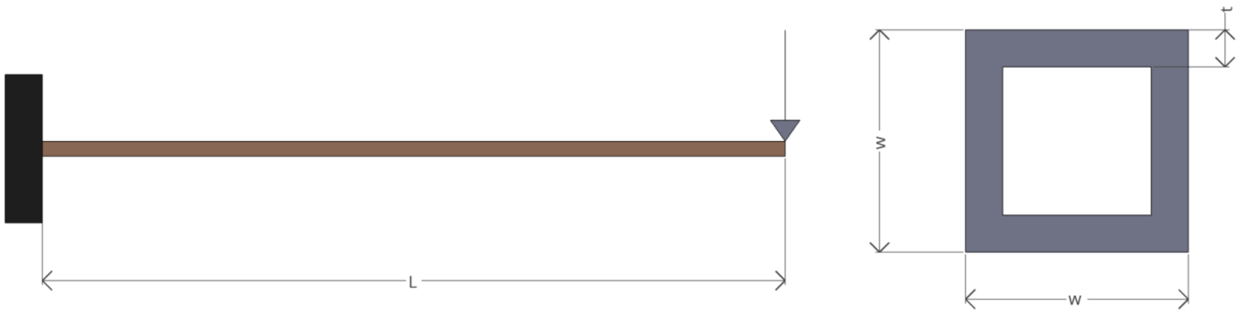

Fig. 4 Optimization problem 1

Considering the constraints on bending stress, shear stress, and vertical deflection of the free end [19, 20], the optimization task is formulated as follows:

$$\text{Min. } f = A = 4t(w-t)\,(\text{mm}^2) \quad (5)$$

s.t.

$$\sigma = Mw/(2I) \le \sigma_a$$

$$\tau = VQ/(2It) \le \tau_a$$

$$q = PL^3/(3EI) \le q_a$$

$$8 - w/t \ge 0$$

where σ, τ, and q_a are bending stress, shear stress, and vertical deflection of the free end, respectively. The variables with subscript 'a' denote the allowable values. σ_a = 165N/mm^2. τ_a = 90N/mm^2. q_a = 10mm. *I* and *Q* denote the moment of inertia and moment about the neural axis (NA) of the area above the NA, respectively [19]. E is the modulus of elasticity of steel = 21x10^{4}N/mm^2. *M* = *PL* denotes the bending moment (N.mm).

After 300 generations and with the use of 30 searching agents, the best found solution is as follows: *w* = 117.108mm and *t* = 14.6mm. With these variables, all of the four constraint values are satisfied.

The next application involves a preliminary planning of a multistory commercial building [19] (refer to **Fig. 5**). The construction site is a 100x100-m area. The maximum height of the building is 25 m. Moreover, the parking area outside the building is at least 25% of the total floor area. The height of a single story is 3.5 m. In addition, the cost of the project is roughly estimated to be 0.5*h* + 0.001*A* where *A* denotes the floor area. Herein, the decision variables are the two sides of the constructed area (*L*1 and *L*2) as well as the number of stories (*n*). This optimization problem is modeled as follows:

$$\text{Min.} \quad f = Cost = 0.5n3.5 + 0.001L1L2 \ (\text{mm}^2) \qquad (6)$$

$$\text{s.t.} \quad L1 \times L2 \times n - 20000 \geq 0$$

$$25 - n \times 3.5 \geq 0$$

$$(100 \times 100 - L1 \times L2) - (L1 \times L2 \times 0.25) \geq 0$$

After 300 generations, the FRB-DE found the following solution: *L*1 = 90.479m, *L*2 = 73.682m, and *n* = 3.

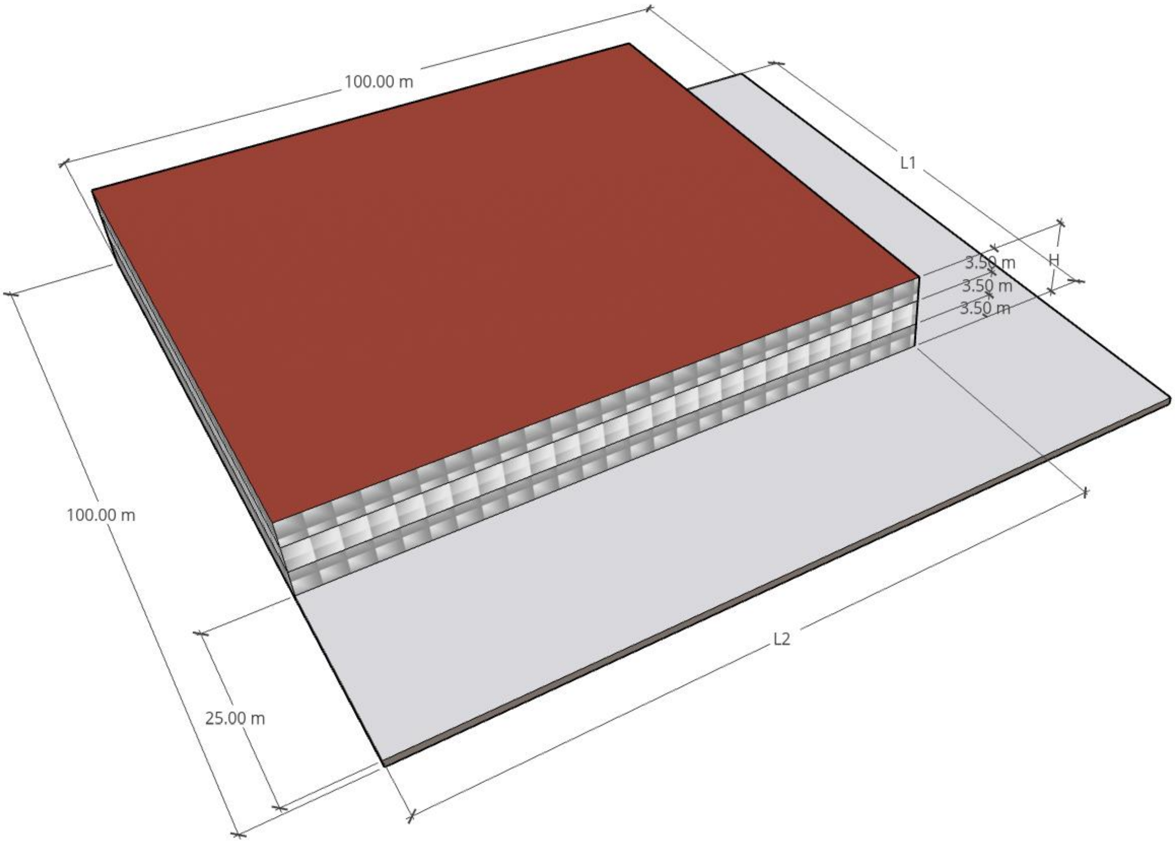

Fig. 5 Optimization problem 2

4. Concluding remarks

In this study, a FRB-DE program based on the DE metaheuristic and a set of feasibility rules is developed to deal with constrained optimization tasks. The FRB-DE is programmed in Visual C# language. Two basic applications are used to test the applicability of the newly developed tool. Based on the optimization results, the newly developed has successfully identified good sets of decision variables which satisfy all of the specified constraints. Therefore, FRB-DE can be a promising alternative to assist civil engineers in dealing with constrained optimization problems.

References

[1] A. Cevik, M. H. Arslan, and M. A. Köroğlu, "Genetic-programming-based modeling of RC beam torsional strength," *KSCE Journal of Civil Engineering,* vol. 14, pp. 371-384, 2010/05/01 2010.

[2] C. C. Coello, F. S. Hernández, and F. A. Farrera, "Optimal design of reinforced concrete beams using genetic algorithms," *Expert Systems with Applications,* vol. 12, pp. 101-108, 1997/01/01/ 1997.

[3] C. A. Coello Coello, A. D. Christiansen, and F. S. Hernández, "A simple genetic algorithm for the design of reinforced concrete beams," *Engineering with Computers,* vol. 13, pp. 185-196, December 01 1997.

[4] V. Govindaraj and J. V. Ramasamy, "Optimum detailed design of reinforced concrete continuous beams using Genetic Algorithms," *Computers & Structures,* vol. 84, pp. 34-48, 2005/12/01/ 2005.

[5] M.-Y. Cheng, D.-H. Tran, and N.-D. Hoang, "Fuzzy clustering chaotic-based differential evolution for resource leveling in construction projects," *Journal of Civil Engineering and Management,* vol. 23, pp. 113-124, 2017/01/02 2017.

[6] H.-H. Tran and N.-D. Hoang, "A Novel Resource-Leveling Approach for Construction Project Based on Differential Evolution," *Journal of Construction Engineering,* vol. 2014, p. 7, 2014.

[7] N.-D. Hoang, "NIDE: A Novel Improved Differential Evolution for Construction Project Crashing Optimization," *Journal of Construction Engineering,* vol. 2014, p. 7, 2014.

[8] N.-D. Hoang, Q.-L. Nguyen, and Q.-N. Pham, "Optimizing Construction Project Labor Utilization Using Differential Evolution: A Comparative Study of Mutation Strategies," *Advances in Civil Engineering,* vol. 2015, p. 8, 2015.

[9] P. W. Christensen and A. Klarbring, *An Introduction to Structural Optimization*: Springer, 2009

[10] S. M. Nigdeli, G. Bekdaş, and X.-S. Yang, "Metaheuristic Optimization of Reinforced Concrete Footings," *KSCE Journal of Civil Engineering,* vol. 22, pp. 4555-4563, November 01 2018.

[11] T. V. Dinh, H. Nguyen, X.-L. Tran, and N.-D. Hoang, "Predicting Rainfall-Induced Soil Erosion Based on a Hybridization of Adaptive Differential Evolution and Support Vector Machine Classification," *Mathematical Problems in Engineering,* vol. 2021, p. 6647829, 2021/02/20 2021.

[12] N.-D. Hoang and Q.-L. Nguyen, "A Novel Approach for Automatic Detection of Concrete Surface Voids Using Image Texture Analysis and History-Based Adaptive Differential Evolution Optimized Support Vector Machine," *Advances in Civil Engineering,* vol. 2020, p. 4190682, 2020/07/28 2020.

[13] H. Nhat-Duc and L. Cong-Hai, "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,* vol. 03, pp. 97-102, 2019.

[14] C. A. C. Coello, "Constraint-handling techniques used with evolutionary algorithms," presented at the Proceedings of the Genetic and Evolutionary Computation Conference Companion, Kyoto, Japan, 2018.

[15] C. A. Coello Coello, "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,* vol. 191, pp. 1245-1287, 2002/01/04/ 2002.

[16] R. M. John, G. R. Robert, and B. F. David, "A Survey of Constraint Handling Techniques in Evolutionary Computation Methods," in *Evolutionary Programming IV: Proceedings of the Fourth Annual Conference on Evolutionary Programming,* ed: MITP, 1995, p. 1.

[17] K. Deb, "An efficient constraint handling method for genetic algorithms," *Computer Methods in Applied Mechanics and Engineering,* vol. 186, pp. 311-338, 2000/06/09/ 2000.

[18] R. Storn and K. Price, "Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces," *Journal of Global Optimization,* vol. 11, pp. 341-359, December 01 1997.

[19] J. S. Arora, *Introduction to Optimum Design, Fourth Edition*: Academic Press, 2016.

[20] J. M. Gere and B. J. Goodno, *Mechanics of Materials, SI Edition*: Cengage Learning, 2013.