TẠP CHÍ KHOA HỌC & CÔNG NGHỆ ĐẠI HỌC DUY TÂN
DTU Journal of Science and Technology 4(47) (2021) 14-19

TRƯỜNG ĐẠI HỌC DUY TÂN
DUYTAN UNIVERSITY

Evolutionary algorithm with feasibility rule based constraint-handling method for intelligent production scheduling

Sử dụng thuật toán tiến hóa với quy tắc khả thi cho việc lập tiến độ sản xuất thông minh

Hoang Nhat Duc[a,b*], Nguyen Quoc Lam[a,b]
Hoàng Nhật Đức[a,b*], Nguyễn Quốc Lâm[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, Trường Đại Học Duy Tân, Đà Nẵng*
[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: 23/4/2021, ngày phản biện xong: 26/4/2021, ngày chấp nhận đăng: 20/8/2021)

Abstract

Manufacturers constantly have to encounter fluctuated product demands. Accordingly, it is required to establish a production schedule that meets the future demands and concurrently has a low level of daily labor/materials' deviation. This study constructs an evolutionary algorithm for achieving this task. The Differential Evolution (DE) algorithm and the feasibility rule based method for constraint handling are integrated to develop this evolutionary algorithm. The proposed approach has been verified by two cases of production scheduling.

Keywords: Differential Evolution; Constrained Handling; Evolutionary Algorithm; Production Scheduling.

Tóm tắt

Với nhu cầu sản phẩm biến động theo thời gian, các nhà sản xuất cần thiết lập một kế hoạch sản xuất đáp ứng nhu cầu của sản phẩm và đồng thời có mức độ biến động về lao động hoặc nguyên vật liệu thấp. Nghiên cứu của chúng tôi xây dựng một phương pháp tối ưu hóa dựa trên thuật toán tiến hóa để ứng dụng cho việc lập kế hoạch sản xuất.Thuật toán Tiến hóa vi phân (DE) và phương pháp dựa trên quy tắc khả thi để xử lý ràng buộc được tích hợp để phát triển thuật toán tiến hóa này. Tính hiệu quả của phương pháp được đề xuất được minh chứng bởi hai bài toán kế hoạch sản xuất phẩm.

Từ khóa: Tiến hóa vi phân; xử lý ràng buộc; thuật toán tiến hóa; lập kế hoạch sản xuất.

1. Introduction

In real-world circumstances, manufacturers constantly have to deal with fluctuated demands. Therefore, they need to establish a production schedule that meets the required demands and concurrently reduces large discrepancies in daily labor/materials' requirements. It is because production schedules with large fluctuations are very costly to implement and often necessitate overtime cost in high production periods as well as result in idle resources in low production periods [1]. Therefore, it is necessary to optimize the production schedule

*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

to meet all required demands as well as to obtain a relatively stable production quantity over time. This task can be formulated as a constrained optimization problem [2-6]. Therefore, in this study, we apply the Differential Evolution (DE) algorithm [7, 8] integrated with an advanced feasibility rule based method for constraint handling [9] to solve the aforementioned constrained optimization problem.

The production scheduling problem can be mathematically formulated as follows [1] [2]:

$$\text{Min. } f = \sum_{t=1}^{P} X_t^2 + \sum_{t=1}^{P} S_t \quad (1)$$

s.t.

$$S_t \geq 0 \ \forall t \quad (2)$$

$$S_t^{Allow} - S_t^{Max} \geq 0 \quad (3)$$

$$X_t, S_t \text{ are integers} \quad (4)$$

S_t is computed as follows:

$$S_t = S_{t-1} + X_t - D_t \quad (5)$$

where P is the number of time periods to be considered; D_t is the demanded number of units in t. S_t denotes the number of units in storage in t. X_t is the number of units produced in t.

As mentioned earlier, this work aims at establishing an evolutionary algorithm based approach for optimizing the production scheduling task. The DE, as a powerful evolutionary algorithm, is chosen in this study to achieve the stated research objective. The feasibility rule based constraint-handling (FRBCH) method is applied [9] to obtain solutions satisfying all of the constraints. The DE algorithm integrated with the FRBCH approach has been coded with Visual C#.NET in Microsoft Visual Studio by the authors.

2. Evolutionary Algorithm based intelligent production scheduling

The DE algorithm, proposed in [7], is a highly effective evolutionary method for dealing with unconstrained optimization problems. The structure of DE is relatively simple and a computer implementing DE can be quickly established with few lines of computer code. The DE structure can be broken down into four steps: (i) population initialization, (ii) mutation, (iii) crossover, and (iv) selection [5, 10-12]. Nevertheless, the DE metaheuristic is not readily applicable for constrained optimization tasks. Therefore, it is required to utilize a constraint handling method coupled with the DE algorithm [13-15]. In this study, to construct an evolutionary method used for intelligent production scheduling, the FRBCH method put forward by Deb [9] has been integrated into the structure of DE. With the FRBCH method, the DE's objective function is revised as follows:

$$F(X) = \begin{cases} F(X) \ \ if \ g_j(x) \geq 0 \ \forall j \\ f_{\max} + \sum_{j=1}^{m} g_j(x) \end{cases} \quad (6)$$

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

The FRBCH-DE integration used for intelligent production scheduling has been developed in Microsoft Visual Studio Visual with C#.NET programming language. **Fig. 1** demonstrates the interface of proposed computer program. **Fig. 2** shows the intelligent production scheduling problem coded as a class in Visual C#. **Fig. 3** demonstrates the revised objective function calculation.

```
public static void Run(int MaxIter, int PopSize, char Disk, double [] Input_Dt,
    double MaxXtAllow, double MaxStAllow)
{
    //----------------------------------------------------------------
    //Problem definition
    string ProblemName = "COP_ProductionScheduling";
    string AlgorithmName = "FeasibilityRuleBased_CHDE";
    var TestProb = new COP_ProductionScheduling();
    TestProb.Set_Dt(Input_Dt);
    TestProb.Set_MaxXtAllow(MaxXtAllow);
    TestProb.Set_MaxStAllow(MaxStAllow);
    var ObjectiveFunction = new GeneralObjFun(TestProb.ComputeObjFun);
    var ObjFunWithConstraints = new GeneralObjFunWithConstraints(TestProb.ComputeObjFunWithConstraint);
    var ConstraintFunction = new GeneralConstraintFun(TestProb.ComputeConstraints);
    var LB_Function = new GeneralLB_Fun(TestProb.Get_LB);
    var UB_Function = new GeneralUB_Fun(TestProb.Get_UB);
    var CheckContraintViolation = new GeneralCheckContraintViolation(TestProb.CheckContraintViolation);
    var ConstraintViolationDegree =
        new GeneralConstraintViolationDegree(TestProb.ComputeConstraintViolationDegree);
```

Fig. 1. Interface of the FRBCH-DE based intelligent production scheduling

```
class COP_ProductionScheduling
{
    private double[] Dt;
    1 reference
    public void Set_Dt(double [] X)...
    private double MaxStAllow;
    1 reference
    public void Set_MaxStAllow(double Y)...
    private double MaxXtAllow;
    1 reference
    public void Set_MaxXtAllow(double Z)...
    4 references
    public double ComputeObjFun(double[,] X)...
    6 references
    public double[,] ComputeConstraints(double[,] X)...
    3 references
    public bool CheckContraintViolation(double[,] x)...
    3 references
    public double[,] ComputeConstraintViolationDegree(double[,] x)...
    3 references
    public double[] Get_LB()...
    3 references
    public double[] Get_UB()...
    3 references
    public double ComputeObjFunWithConstraint(double[,] x, double fmax)...
    0 references
    public void ProblemAnalysis()...
    2 references
    public void ProblemAnalysis(double [,] Xmat, string SaveLoc)...
}
```

Fig. 2. Production scheduling class definition

```
public double ComputeObjFun(double[,] X)
{
    var Xv = MyMatrix.ConvertRowMatrixToVector(X);
    var Xt = MyMatrix.RoundingDoubleVector(Xv);
    int L = Dt.Length;
    var St = new double[L];
    for (int i = 0; i < L; i++)
    {
        if (i == 0)
        {
            St[i] = 0 + Xt[i] - Dt[i];
        }

        if (i >= 1)
        {
            St[i] = St[i - 1] + Xt[i] - Dt[i];
        }
    }
    double SumSt = MyMatrix.SumVectorElement(St);
    double ProdVarSum = 0;
    for (int i = 0; i < L; i++)
    {
        ProdVarSum += Xt[i] * Xt[i];
    }
    double ObjFun = ProdVarSum + SumSt;
    return ProdVarSum;
}
```

Fig. 3. The revised objective function

3. Program applications

As mentioned earlier, the objective of this study is to construct a production schedule that meets the required demands and concurrently reduces large discrepancies in daily labor/materials' requirements. The reason is that production schedules with large fluctuations are very costly to execute and often result in extensive uses of resources in high production periods as well as idle resources in low production periods. Accordingly, this study employs FRBCH-DE for developing intelligent production plans given data on the future product demands. The computer program based on FRBCH-DE is demonstrated in **Fig. 4.**

In this section of the article, two case studies have been used to verify the capability of the proposed approach. The first case study involves the production planning over 7 periods. In the second case study, the number of planning periods is 12. The demanded numbers of units for the first and the second optimization problem are shown in **Table 1** and **Table 2**, respectively. The production schedules for the first and the second problems optimized by the FRBCH-DE are reported in **Fig. 5** and **Fig. 6**, respectively. It can be seen from these two figures that the proposed metaheuristic has helped to find production schedules featuring low degree of fluctuations in the numbers of produced units for both case studies.

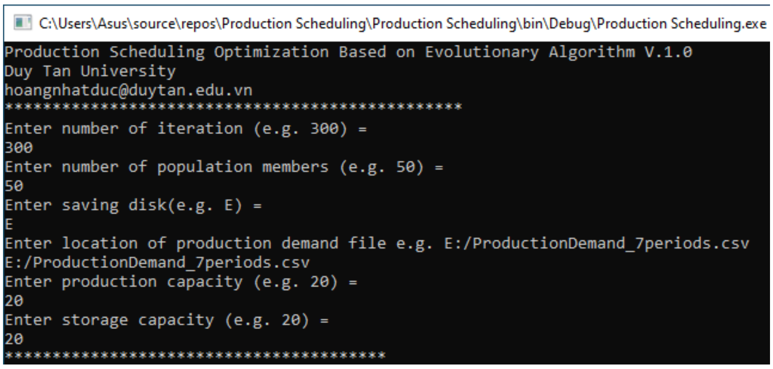

Fig. 4. FRBCH-DE computer program for developing intelligent production plans

Table 1 Production scheduling result for the case study 1

Period	*Dt*	*Xt*	*St*
1	10	12	2
2	15	13	0
3	8	11	3
4	12	10	1
5	6	10	5
6	4	10	11
7	22	11	0

Table 2 Production scheduling result for the case study 2

Period	*Dt*	*Xt*	*St*
1	10	13	3
2	15	12	0
3	8	11	3
4	12	12	3
5	6	10	7
6	4	11	14
7	22	10	2
8	12	11	1
9	8	11	4
10	15	11	0
11	7	9	2
12	11	9	0

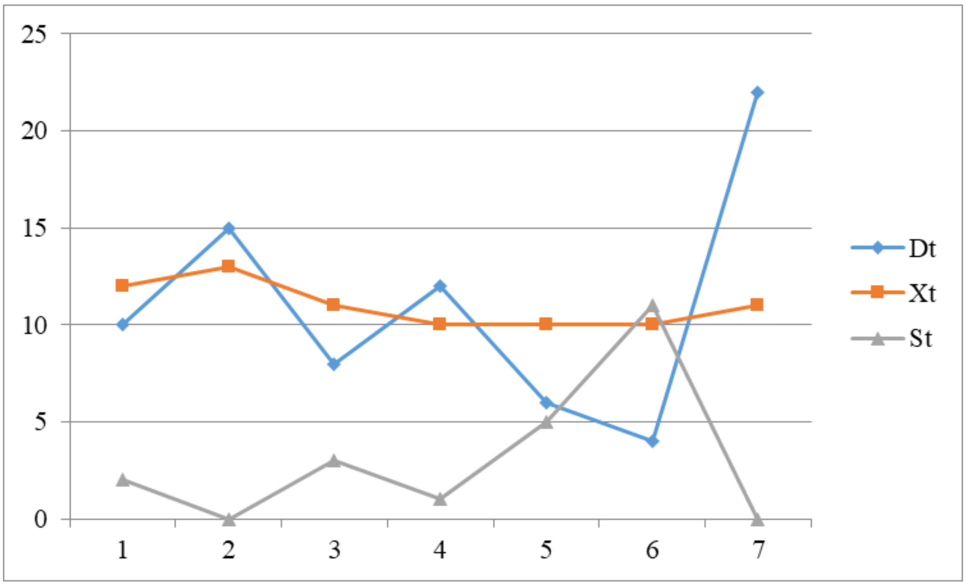

Fig. 5. Results of D_t, X_t, and S_t for the case study 1

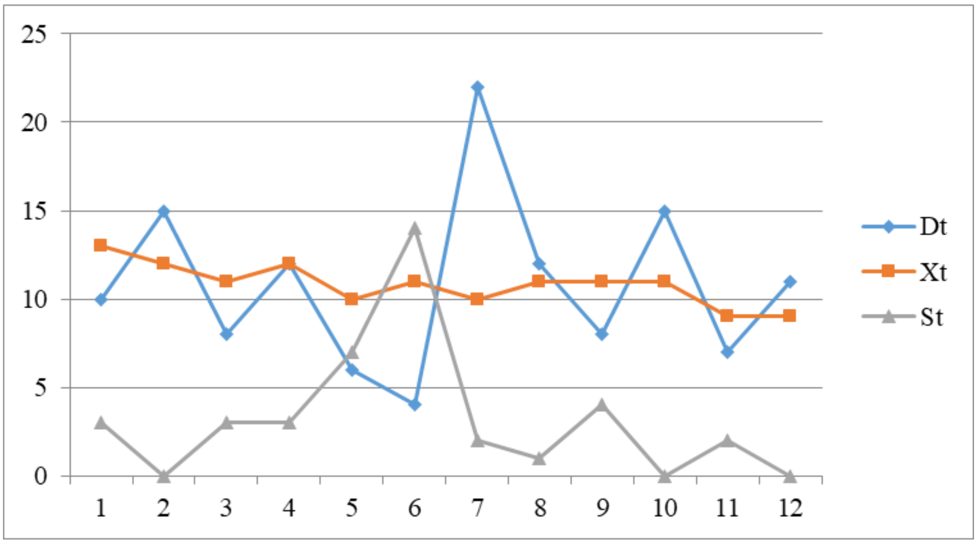

Fig. 6. Results of D_t, X_t, and S_t for the case study 2

4. Concluding remarks

This study has developed and validated a production scheduling optimization program based on the utilization of the DE evolutionary algorithm and the FRBCH method. The integrated approach, denoted as FRBCH-DE, has been developed with Visual C#.NET. Case studies involving the determination of 7 and 12 decision variables have been employed to verify the capability of FRBCH-DE. Experimental result shows that FRBCH-DE is able to find a good set of decision variables that feature a low objective function and satisfy all the required constraints.

References

[1] E. Castillo , A. J. Gonejo , P. Pedregal , R. Garciá , and N. Alguacil, *Building and Solving Mathematical Programming Models in Engineering and Science*: John Wiley & Sons, 2001.

[2] B.-T. Nguyen, N.-D. Hoang, T.-H. Vu, and L. T. Phan, "Giải bài toán tối ưu nguồn lực trong lập kế hoạch sản xuất các cấu kiện chế tạo sẵn trong dự án xây dựng với phương pháp quy hoạch phi tuyến," *In Proc. of ATCESD 2016, Da Nang, Vietnam*, 2016.

[3] J. O. McClain, L. Joseph Thomas, and E. N. Weiss, "Efficient Solutions to a Linear Programming Model for Production Scheduling With Capacity Constraints and No Initial Stock," *IIE Transactions*, vol. 21, pp. 144-152, 1989/06/01 1989.

[4] S. Rasmussen, "Production Planning in the Linear Programming Model: Linear Programming," in

Production Economics: The Basic Theory of Production Optimisation, ed Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 211-223.

[5] 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.

[6] 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.

[7] 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.

[8] K. Price, R. M. Storn, and J. A. Lampinen, *Differential Evolution - A Practical Approach to Global Optimization*: Springer-Verlag Berlin Heidelberg, 2005.

[9] 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.

[10] 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.

[11] N. Đ. Hoàng, Q. L. Nguyễn, and Q. N. Phạm, "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*, vol. 1, pp. 135–141, 2015.

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

[13] N. D. Hoang, "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)*, 2019.

[14] 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.

[15] 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.