A Modified Differential Evolution Algorithm for Resource Constrained Multi-project Scheduling Problem

Rui Yan and Weijia Li,
School of Naval Architecture and Ocean Engineering, Huazhong University of Science and Technology, Wuhan, 430074, China
Email:{yanrui1001@163.com,whlwj@vip.sina.com}

Ping Jiang* and Yinzhi Zhou,
The State Key Laboratory of Digital Manufacturing Equipment and Technology, School of Mechanical Science and Engineering, Huazhong University of Science and Technology, Wuhan, 430074, China
Email:{ jiangping@mail.hust.edu.cn, leitezhou@gmail.com}

Guoqing Wu
Wuchang Shipbuilding Industry Co. Ltd., Wuhan, 430060, China
Email:npterry@163.com

Abstract—**The resource constrained multi-project scheduling problem (RCMPSP) is an important issue in business applications. In this paper, a modified differential evolution algorithm is introduced to achieve higher computational efficiency for RCMPSP. Two parallel mutation operations are used to improve the search ability with a modified DE/rand-best/1/bin strategy. The selection operation is used to choose best individual from target vector and two trail vector. The computational results show that the modified DE algorithm performs better than several other algorithms of deterministic and heuristic nature**.

Index Terms—**differential evolution, resource constrained multi-project scheduling problem, parallel mutation operation**

I. Introduction

Project scheduling deals with the allocation of given resources and determines the start and completion times of each activities. The resource constrained project scheduling problem (RCPSP)has been the focus of research for decades, such as the works of Kolisch [1], Brucker et al [2] , and Demeulemeester and Herroelen [3].However, the multi-project [4] , a generalization of RCPSP, plays a more important role in business applications according to the survey of Payne [5] , and Lova and Tormos [6] .

Multi-project scheduling has been a research topic since the late 1960s. Pritsker [7] presents a zero-one integer programming approach for the problem. Fendley [8] introduces the modeling of a complete multi-project scheduling system, proposes methods to distribute due dates, and uses priority rules for sequencing activities. Lenstra and Kan [9] show that RCMPSP is strongly NP-hard. Hence, efficient heuristics and meta-heuristics are used for solving these problems.

Most literatures focus on heuristic methods that were proposed from 1980s to 2000s. Priority rules are one of the most popular methods. Kurtulus and Davis [10] designed 6 new priority rules and tested the performance. Lawrence and Morton [11] studied the resource price based priority rules with the objective of minimizing weighted tardiness costs. Lova et al. [12] developed a multi-criteria heuristic algorithm consisted of several algorithms to improve two criteria. Tyson and Ali [13] tested 20 priority rules with12,320 problems and provided guidance for various situations and objectives. Doreen and Armin [14] proposed a heuristic solution for RCMPSP taking into consideration of sequence and resource dependent transfer times.

In recent years, meta-heuristic algorithms, like genetic algorithm (GA), simulated annealing (SA), have been introduced to solve RCMPSP.

Kim et al. [15] proposed a hybrid genetic algorithm with a fuzzy logic controller for RCMPSP with an objective of minimizing makespan and total tardiness penalty. Goncalves et al. [16] took an approach of a solution by building parameterized active schedules based on priorities, delay times and release dates and Using GA as a search method. Chen et al. [17] proposed a hybrid of GA and SA. The authors compared the performance of GA, SA, GA-SA hybrid, and several heuristic rules. The works on heuristic algorithms also

Manuscript received November 12, 2013; revised January 25, 2014; accepted January 28, 2014.

*Corresponding author.

doi:10.4304/jcp.9.8.1922-1927

include Deng and Lin [18] , Zhao et al. [19] , and Cai and Li [20].

As shown in those literatures, GA is wildly used. However, it has been proven that some new meta-heuristic algorithms can achieve better performance than GA, such as differential evolution (DE) algorithm. In this paper, we present a novel differential evolution algorithm to solve the resource constrained multi-project scheduling problem. The rest of the paper is organized as follows. Section 2 describes the problem and its mathematical model. Section 3 introduces the classical differential evolution algorithm and proposes the improved differential evolution algorithm. Numerical experiments are reported in Section 4. Conclusion remarks are listed in Section5.

II Multi-project Scheduling Problem

According to the problem description in Tyson etc. [13] , resource constrained multi-project scheduling problem contains several parallel projects with limited supply of resources. The problem can be described as follows:

There are multi projects should be manufactured, where each project is comprised of one or more activities. Each activity needs a deterministic processing time and requires certain units of resource type during the processing time. The capacity units of resource type are constant. Each activity cannot start processing without maintaining the resource requirement. In addition, an activity I can not start until its all predecessors are completed.

The aim of problem is to find a schedule for the activities in all projects (i.e. to determine the start and completion times of these activities) that optimizes a performance measure, for example, minimizing makespan.

We give an example of RCMPSP in Figure 1. There are *L* projects in this problem. Each project consists of several activities N_l . Activities *0* and *N+1* are dummy activities, added to represent the start and end of all activities.

Let F_{il} represents the finish time of activity *i* in project *l* ; P_{il} represents the processor activity in project *l*; d_{il} means processing time of activity *i* in project *l* ; *A(t)* be the set of activities being processed at time *t*; R_k means the total number of resource k. The mathematical model of the RCMPSP can be described as:

$$\begin{aligned} &\text{minimize } \textit{makespan} \\ &s.t.\ F_{jl} \le F_{il} - d_{il},\ i \in N_l,\ j \in P_{il},\ l \in L \\ &\quad \sum_{i,l \in A(t)} r_{ilk} \le R_k,\ k \in K,\ t \ge 0 \qquad (1) \\ &\quad F_{il} \ge 0,\ i \in N_l,\ l \in L \end{aligned}$$

The 1st constraint guarantees the precedence relations between activities in the same projects; the 2ndconstraintpreventsthe resources being occupied at time t from exceeding the available capacity; the 3rdconstraintensuresthe finish times of activities are non-negative.

For RCMPSP, multiple resource allocation is a vital issue. In this paper, next time frame mentioned in Moselhi and Lorterapong [21] is used. It is a least impact algorithm. An activity can be listed in the set of processing activities only when all its predecessors are completed. The selections among activities that can start at the same time are based on priority of these activities. The details of this progress are listed in Moselhi and Lorterapong [21] .

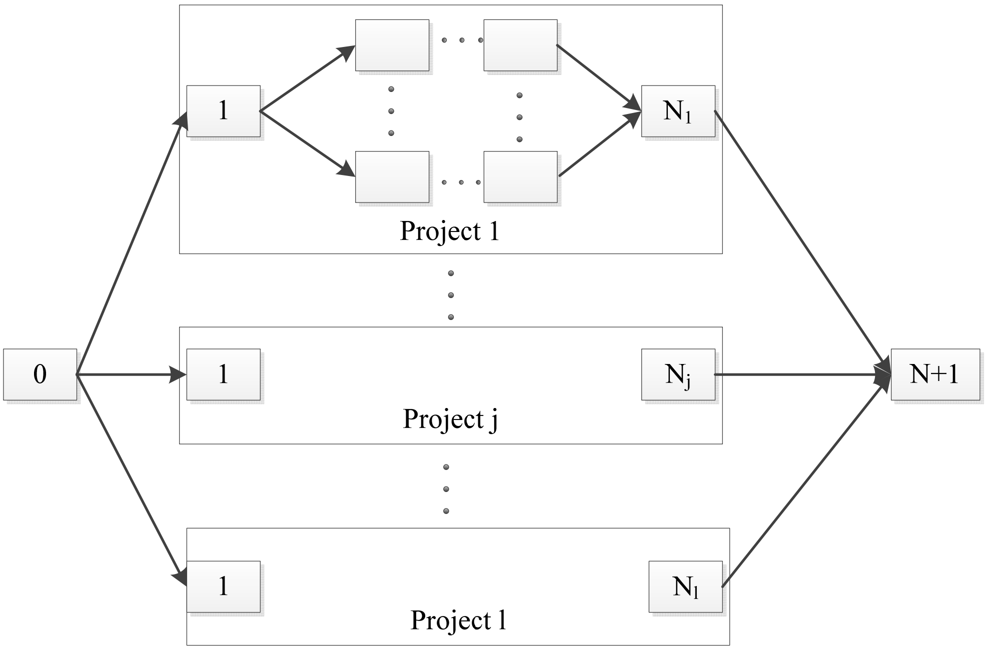


Figure 1.An example of RCMPSP

III Modified Differential Evolution Algorithm

The differential evolution (DE) algorithm is a population based evolutionary algorithm proposed by Price and Storn [22] . It is an efficient and a powerful evolutionary algorithm (EA).

A.Classical DE Algorithm

In the DE algorithm, one population consists of NP individuals $X_{i,G}$, where i=(1,2,3…NP), and G donates one generation. Each individual $X_{i,G}$ consists of D variables which are limited to search range. There are three major operations used in the iteration phase: mutation, crossover and selection.

Mutation Operation

Strategy DE/rand/1/bin [22] is the most wildly used strategy. Three randomly target vectors are chosen to generate mutation vector using the formula:

$$v_{i,G+1} = x_{r1,G} + F(x_{r2,G} - x_{r3,G}), r1 \neq r2 \neq r3 \neq i \quad (2)$$

Where F means mutation parameter. $x_{r1,G}$, $x_{r2,G}$, $x_{r3,G}$ are three randomly chosen vectors, and they neither equal to one another, nor to the target vector $x_{i,G}$.

Crossover Operation

Crossover Operation generates trail vectors by collecting the dimensions from the target vectors and their offspring mutant vectors with a determined possibility. Binomial crossover operation is used:

$$u^{j}_{i,G+1} = \begin{cases} v^{j}_{i,G+1} & if\ r(j) \leq CR\ or\ j = n_j \\ x^{j}_{i,G} & otherwise \end{cases} \quad (3)$$

Where CR means crossover rate and n_j is a randomly selected dimension to make sure at least one dimension of the trial vectors is chosen from the mutant vectors.

Selection Operation

Greedy selection operation is used to determine whether trail vectors or target vectors can survive into the next generation. The formula is described as follows:

$$x_{i,G+1} = \begin{cases} u_{i,G+1} & if\ f\left(u_{i,G+1}\right) \prec f\left(x_{i,G}\right) \\ x_{i,G} & otherwise \end{cases} \quad (4)$$

B. Modified DE Algorithm

Usually, DE/current-best/1/bin [23] strategy is an efficient way to improve the precision of DE algorithm. However, it is not suitable for complex problems because of its slow convergence. In order to strengthen the advantage of DE/current-best/1/bin strategy and avoid the disadvantage in convergence, we change DE/current-best/1/bin strategy to DE/rand-best/1/bin:

$$v_{i,G+1} = x_{r1,G} + F(x_{\text{best},G} - x_{r1,G}) + F(x_{r2,G} - x_{r3,G}), r1 \neq r2 \neq r3 \neq i \quad (5)$$

In this strategy, the target vector $x_{i,G}$ is replaced by $x_{r1,G}$, allowing the method to converge to the best solution, and at the same time not to converge too fast to, or be stuck at, a local optimum.

Using a single strategy in the method may lead the particles convergent to one point quickly, so different strategies are blended to improve the global search ability. In this article, we proposed a modified DE algorithm (MDE) by using both DE/rand/1/bin and DE/current-best/1/bin strategies to generate new trail vectors.

Also different from the self-adaptive algorithms for choosing mutation strategies in the literature, two strategies are used in each generation of MDE. Thus, two trail vectors are generated for one target vector. The new trail vector with better fitness value will survive and be compared with the target vector by using selection operation. Pseudocode of MDE is presented in Figure 2.

Two strategies need one more time function value for each individual, so the number of generations may be reduced to half to meet the same number of function values, which will lead to lower convergence speed. However, this method can maintain the diversity of the population by using different mutation strategies, which is more important to solving complex problems.

```
Set parameters: population size ($NP$), maximum number of function evaluations ($max\_FES$), and the value of $F$ and $CR$.
$G$=0; generate an initial population X0={$x_{1,0}$,⋯,$x_{NP,0}$} and evaluate the objective function values.
while $FES<max\_FES$ do
    for i=1:$NP$ do
        Use strategies DE/rand/1/bin and DE/current-best/1/bin to generate two mutation vectors $v_{i1,G}$ and $v_{i2,G}$;
        Use crossover operation to generate two trail vectors $u_{i1,G}$ and $u_{i2,G}$;
        Evaluate the objective function values of the two trial vectors $u_{i1,G}$ and $u_{i2,G}$;
        Choose the best vector from the two trial vector $u_{i1,G}$, $u_{i2,G}$ and target vector $x_{i,G}$;
    endfor
    $G=G+1$
end while
output the best objective function in the population
```

Figure 2.The pseudocode of MDE

IV Numerical Experiments

Three test projects in Ref. [17] are introduced in this section to perform numerical experiments and show the superiority of the proposed DE algorithm. There are 21, 27 and 26 activities, respectively, in these projects. Two types of resources are needed to complete the activities. The maximum accessible number of resource 1 is 10 units, while the availability of resource 2 is limited to 20 units.

For DE algorithm, we set F=0.7, CR=0.7. For MDE, we set F1=0.7, F2=0.7, CR1=0.3, CR2=0.7. The population size is set at 100, while the total function evaluation number is 10,000. The best solution and the average result are based on10-run experiments. The DE algorithm and modified DE algorithm are coded in C++ and experiments run on a computer with a 2.0 GHz CoreTM 2 Duo CPU.

A.Comparisons WithDifferent Algorithms

The modified DE algorithm is compared with the classical DE algorithm and other five algorithms listed in Ref. [17] : genetic algorithm (GA), simulated annealing (SA), modified version of SA (MSA) and GA-SA Hybrid. The comparison results are shown in Table 1. The front five results are results of the above comparison algorithms which are copied Chen and Syed [17] . For DE and MDE algorithms, we use the same experimental condition as in Chen and Syed [17] .

The experiment results show that DE algorithm produces a better average result than GA and SA do. MDE algorithm achieves minimum best solution (or the global optimum) just as logarithmically improved MSA does, while performing better in the average result. When compared with the 10 most popular heuristic algorithms or priority rules categorized and detailed in Kurtulus and Davis [10] , MDE has indubitable advantages. The numerical results of these 10 algorithms for the three test projects are provided in Table 2, again reusing the results from Chen and Syed [17] .

TABLE 1.

RESULTS OF NUMERICAL EXPERIMENTS

Algorithms	Best solution of 10-run	Average of 10-run
GA	133	135.5
SA	134	135.4
GA/SA	132	134.5
MSA (arithmetically improved)	133	134.2
MSA (logarithmically improved)	130	133.0
DE	133	134.5
MDE	130	132.5

TABLE 2.

EXPERIMENT RESULTS OF 10 POPULAR HEURISTIC ALGORITHMS

Heuristic algorithm	Explanation	Solution
FCFS	First come first severed	144
LCFS	Late come first severed	163
SOF	Short operation first	146
MOF	Maximum operation first	150
MINSLK	Minimum slack first	149
MAXSLK	Maximum slack first	154
MINTWK	Minimum total work content	146
MAXTWK	Maximum total work content	157
SASP	Shortest activity from shortest project	155
LALP	Longest activity from longest project	155

B. Selection FromDifferent Parameters

In order to choose the best selection for the parameters (F, CR) of MDE algorithm, three different values are used to assign these parameters. However, testing all combinations will require 81 experiments. So for experiments in this article we make two parameters in one strategy fixed and change the other two parameters.

First of all, we set the parameters of DE/rand/1/bin, to(0.7,0.7), the results are show in Table 3. Secondly, the parameters of DE/best-rand/1/bin are fixed at(0.3,0.7), the results are show in Table 4.

From the results, we can see that, parameters are a crucial factor influencing the results. For strategy DE/best-rand/1/bin, the lower the F value and the higher the CR value, the better the results. The opposite is true for strategy DE/rand/1/bin - higher F values and lower CR values produce better results. But parameters (0.7, 0.7) for strategy DE/rand/1/bin, performs better than other parameters. So the combination of (0.7,0.7) and (0.3,0.7) is the best choice.

TABLE 3

.RESULTS OF DIFFERENT VALUES OF DE/BEST-RAND/1/BIN PARAMETERS

Parameters	Best	Mean	Parameters	Best	Mean	Parameters	Best	Mean
(0.3,0.3)	131	132.9	(0.5,0.3)	129	133.4	(0.7,0.3)	133	134
(0.3,0.5)	131	132.9	(0.5,0.5)	131	132.9	(0.7,0.5)	133	133.7
(0.3,0.7)	130	131.6	(0.5,0.7)	130	132.6	(0.7,0.7)	132	133.2

TABLE 4.
RESULTS OF DIFFERENT VALUES OF DE/RAND/1/BIN PARAMETERS

Parameters	Best	Mean	Parameters	Best	Mean	Parameters	Best	Mean
(0.3,0.3)	131	131.9	(0.5,0.3)	130	132.6	(0.7,0.3)	132	132
(0.3,0.5)	131	132.3	(0.5,0.5)	129	132.5	(0.7,0.5)	131	132.3
(0.3,0.7)	130	132.1	(0.5,0.7)	130	132	(0.7,0.7)	130	131.6

C.Comparison BetweenDifferent MutationStrategies

In Section B in chapter III, a modified strategy and its corresponding test results were presented. In this section a comparison between that strategy and DE/best-current/1/bin strategy is proposed to show the efficiency of the modified strategy. Similar to MDE, different parameters are used in DE/best-current/1/bin strategy, while fixed parameters of(0.7,0.7) are adopted in DE/rand/1/bin strategy. As shown in the Table 5, DE/best-rand/1/bin strategy is better than DE/best-current/1/bin strategy for almost all parameters tested.

TABLE 5.
COMPARISON OF TWO DIFFERENT STRATEGIES

	DE/best-current/1/bin		DE/best-rand/1/bin	
Parameters	Best	Mean	Best	Mean
(0.3,0.3)	132	133.5	131	132.9
(0.3,0.5)	131	133.3	131	132.9
(0.3,0.7)	130	132.6	130	131.6
(0.5,0.3)	133	133.8	129	133.4
(0.5,0.5)	131	133.5	131	132.9
(0.5,0.7)	131	133.2	130	132.6
(0.7,0.3)	132	133.8	133	134
(0.7,0.5)	132	133.3	133	133.7
(0.7,0.7)	133	133.9	132	133.2

V. CONCLUSION

In this paper, an improved differential evolution algorithm is introduced to solve the resource constrained multi-project scheduling problem. The next time frame method is used for resource allocation. Modified DE/rand-best/1/bin strategy and DE/rand/1/bin are used in two parallel mutation operations to achieve better search ability. The experiment results show that the modified DE algorithm can not only outperform the classical DE algorithm but also the hybrid GA-SA algorithm. Furthermore, the modified DE algorithm performs much better than some popular heuristic algorithms.

Future studies could rely on the hybrid of GA and DE to improve the search ability.

REFERENCES

[1] R. Kolisch,"Project scheduling under resource constraints", *Physica*, Heidelberg,1995

[2] P.Brucker,K.Mohring,R.Neumann, and Pesch, R. "Resource-constrained project scheduling: notation, classification, models, and methods", *European Journal of Operational Research*, vol. 112, no. 1, pp 3-41,1999

[3] E.L.Demeulemeester, W.S.Herroelen,"Project Handbook", *Kluwer Academic Publishers*, Boston, 2002

[4] S.E.Fricke, A.J.Shenhar, "Managing multiple engineeringprojects in a manufacturing support environment", *IEEE Trans-actions on Engineering Management,* vol. 47 no. 2pp.258 –268, 2000

[5] J.H.Payne, "Management of multiple simultaneous projects: a state of the art review", *International Journal of Project Management*, vol. 13, no. 3, 1995

[6] A.Lova,P.Tormos, "Analysis of scheduling schemes and heuristic rules performance in resource-constrained multiproject scheduling", *Annals of Operations Research*,vol. 102, pp.263-286, 2001

[7] A.Pritsker, B.Allan, L.J.Watters, P.M.Wolfe, "Multiproject scheduling with limited resources: a zero-one programming approach", *Management Sciences*, vol.16, pp.93-108, 1969

[8] L.G.Fendley, "Towards the development of a complete multiproject scheduling system", *Journal of Industrial Engineering*, vol. 19, no. 10, pp. 505-515, 1968

[9] J.K.Lenstra, A.H.G.R.Kan, ,"Complexity of scheduling under precedence constraints", *Operations Research*, vol. 26, no. 1, pp. 22-35, 1978

[10] I.S.Kurtulus,E.W.Davis,"Multi-project scheduling: categorization of heuristic rules performance", *Management Science*, vol. 28, pp.161-172, 1982

[11] S.R.Lawrence,T.E.Morton,"Resource-constrained multi-project scheduling with tardy costs: comparing myopic bottleneck and resource pricing heuristics", *European Journal of Operational Research*,vol.64, pp.168-187, 1993

[12] A.Lova, C.Maroto, P.Tormos,"A multicriteria heuristic method to improve resource allocation in multiproject scheduling", *European Journal of Operation Research*,vol. 127, pp.408-424, 2002

[13] R.B.Tyson, A.Y.Ali, "Resource-constrained multi-project scheduling: priority rule performance revisited", *International Journal of Production Economics*, vol. 126, pp. 212-218, 2010

[14] K.Doreen,S.Armin,"A heuristic solution framework for the resource constrained (multi-) project scheduling program with sequence-dependent transfer times", *European Journal of Operation Research*,vol.197, pp.492-508, 2009

[15] KwanWoo Kim, YongSu Yun, JungMo Yoon, Mitsuo Gen, Genji Yamazaki, "Hybrid genetic algorithm with adaptive abilities for resource-constrained multiple project scheduling", *Computers in Industry*,vol. 56, pp.143-160, 2005

[16] J.F.Goncalves, J.J.M.Mendes, and M.G.C.Resende,"A genetic algorithm for the resource constrained multi-project scheduling problem", *European Journal of Operation Research*, vol. 189, pp.1171-1190, 2008

[17] Chen Po-Han,Seyed Mohsen Shahandashti,"Hybrid of genetic algorithm and simulated annealing for multiple project scheduling with multiple resource constraints",

Automation in Construction, vol. 18, pp.434-443, 2009

[18] L.Y.Deng,Y.Lin,"A particle swarm optimization for resource constrained multi-project scheduling problem", *Proceedings of2007 International Conference on Computational Intelligence and Security*,pp.1010-1014, 2007

[19] Zhao Man, Tan Wei, Li Xiang, Kang Lishan, "Research on multi-project scheduling problem based on hybrid genetic algorithm", *Proceedings of2008 International Conference on Computer Science and Software Engineering*, pp. 390-394, 2008

[20] CaiZhicheng, Li Xiaoping,"A hybrid genetic algorithm for resource-constrained multi-project scheduling problem with resource transfer time", *Proceedings of2012 IEEE International Conference on Automation Science and Engineering*,pp.569-574, 2012

[21] O. Moselhi, P. Lorterapong, "Least impact algorithm for resource allocation", *Canadian Journal of Civil Engineering*, vol. 20, pp.180-188, 1993

[22] R. Storn, K. Price, "Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces", *Journal of Global Optimization*, vol. 11, pp. 341-359, 1997

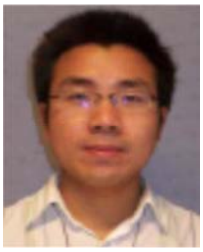
[23] R.Storn, "Differential evolution research – Trends and Open Questions", *Advances In Differential Evolution*, Uday K. Chakraborty, Springer, 2008

Rui Yan received his B. E in computer science and technology in 2004, and his M. E in marine engineering in 2008. Currently he is pursuing his PhD in marine engineering, Huazhong University of Science and Technology.

His research interests include simulation technique in ship manufacturing.

Weijia Li is a professor in the school of Naval Architecture and Ocean Engineering, Huazhong University of Science and Technology.

His research interests include underwater work-system, intelligent control system, etc.



Ping Jiang received his BS and PhD degrees from the school of Mechanical Science and Engineering (MSE), Huazhong University of Science and Technology (HUST) in 2004 and 2009 respectively. He is now an associate professor in MSE of HUST.

He has published more than 30 articles in international journals and conference proceedings.His research interests include multidisciplinary design optimization, process management of product development, etc.

Yinzhi Zhou received his BS degrees from the school of Mechanical Science and Engineering (MSE), Huazhong University of Science and Technology (HUST) in 2011. He is a PhD candidate in the school of Mechanical Science and Engineering, Huazhong University of Science and Technology.

His research interests include intelligent optimization algorithms, scheduling, and robust optimization.

Guoqing Wu received his B. E in computer science and technology in 2003, Hubei University of Technology.

He worked for Wuchang shipbuilding industry co. ltd., from 2004.

His research interests include simulation technique in ship manufacturing.