Robust Optimization Model of Project Scheduling Problem Based on Genetic Algorithm

Dayong Wang[1], Xiangyang Li[2], Bingxiu Fang[3]and Jinling Chou[4]

[1]Harbin institute Technology, Harbin 150001, China

[2]Harbin institute Technology, Harbin 150001, China

[3]Jilin University, Jilin,130035, China

[4]Jinling Chou，Beijing Aerospace Automatic Control Institute, Beijing 100584

Wangdayong1688@163.com

Keywords: Project scheduling; Robust optimization model; probability distribution; Genetic algorithm.

Abstract. A robust optimization model of resource-constrained critical chain project scheduling problem is introduced. Stochastic programming method is used to describe the uncertainty of activity duration. Based on the traditional critical chain project scheduling mode, a genetic algorithm is introduced, and steps of solving proposed model for RCPSP problem with stochastic activity durations are designed. An actual project is applied to explain the robust optimization model and the proposed genetic algorithm. The numeric experiment showed that the method designed in this paper is robust to deal with activity duration uncertainties.

Introduction

Resource-constrained project scheduling problem (RCPSP) widely exists in construction engineering, software development, aircraft and ship building and other small batch production enterprises. The research on it has a significant practical value. In theory, RCPSP is mostly NP-hard problem and difficult to solve. It has attracted attention of many researchers[1,2].

In recent years, many studies on RCPSP are based on a premise that activity duration is determined. That condition aimed at constructing a baseline schedule for raw material procurement, equipment maintenance and order processing activities[3]. However, these projects are often executed under resource constraints (e.g., limited number of workers), and uncertainty (e.g., uncertain productivity of workers, bad weather, man-made or natural disasters etc.)[4]. Nowadays, the demand to finish a project in a shorter time is increasing. Hence, project managers strongly need an effective method that considers resource constraints and uncertainty for project scheduling, and provides a systematic mechanism for managing schedule during project execution.

Goldratt introduced the critical chain project management (CCPM) method which is based on the probability theory[5-6]. CCPM uses a deterministic schedule integrated by a buffer mechanism to deal with both resource constraints and uncertainty. However, many projects in practice have to face a relatively high degree of uncertainty factors, such as accidents or man-made disasters. Many unknown events may result in the useless of the critical chain made before. So this paper introduces how to use the robust optimization algorithm to improve the critical chain when facing the uncertainty.

This paper firstly presents the mathematical model of RCPSP, and some solutions for stochastic task durations are introduced. Then a mathematical robust optimization model is deducted for the RCPSP with stochastic task durations. Genetic algorithm is designed to solve the proposed model. At last, numerical experiments using the model and algorithm before are presented to verify and analysis.

Problem statement

Classic RCPSP problem involves a project network $G(N,A)$, N is project set, A is immediate predecessors set between each project. For the basic PCPSP problem, some assumptions are made as follows:

1. Consider "finish-start" (FS) type of precedence relation only, that is to say that activity *j* may just be started immediately when activity *i* is concluded. There is no delay, feedback and circulation.
2. Consider renewable resources only. That means activities consume resources at each time that are provided in limited quantities. But the total supply don't gradually consumed as the project progressed.
3. The activities cannot be interrupted once they are started.

The symbols of RCPSP model used are shown in the following table:

Table 1 the symbols of RCPSP model

symbol	instruction
j	Activity number, j = 1, 2, …, J; Where J is the total number of activities contained in the project
t	Time number, t = 1, 2, …, T; Where T is upper limit of activities duration.
k	Resource number, k = 1, 2, …, K; Where K is number of renewable species needed by activities.
T	Total time of the project
$d(j)$	Duration of activity *j*
F_j	Immediate predecessors set of activity *j*
S_j	immediate successors set of activity *j*
T_j^s	Start time of activity *j*
T_j	Complete time of activity *j*
I_t	Set of executing activities at time *t*
R_k	The supply of resources *k*, at any time, the total supply needed by activities cannot exceed R_k
P_B	Project buffer time
r_{jk}	Each time the resources needed by activity *j*

The basic RCPSP problem can be described as:

$$\min\left(T+P_B\right) \tag{1}$$

s.t.

$$T=\max_{i=1,2,L\ ,n}\left(T_i^s+d(i)\right) \tag{2}$$

$$T_h\le T_j-d_j,\quad \forall h\in F_j,\ \forall j \tag{3}$$

$$\sum_{j\in I_t} r_{jk}\le R_k,\ \forall k,t \tag{4}$$

$$P_B=\sqrt{\sum_{j\in J}\left(\sigma_j^{\varphi}\right)^2} \tag{5}$$

Where, Eq. 1 means the objective function to minimize total project time limit for a project, Eq. 2 means the length of the critical chain, thus the total project duration. P_B is determined by the root of variance by Eq. 5[7]. σ_j^{φ} is the variance of activity j in the scene φ. Eq. 3 means immediate predecessors relation, Eq. 4 means resources limitation. In the classic RCPSP problem, the duration of activity $d(j)$ is often treated as constant. However, in the real project management situation, it has been recognized that the majority of the activity duration is a variable and obey some probability distribution. According to statistical data and historical experience, we can estimate the probability of the uncertainty activity duration. Activity duration is often a random variable between the minimum duration and maximum duration, which can be described by a discrete probability model. For any activity S∈N, its duration discrete probability model can be described as formula below[8]:

$$P(S)\begin{cases} P_1(S),\ t=t_1(S) \\ P_2(S),\ t=t_2(S) \\ \quad \text{L L L} \\ P_J(S),\ t=t_J(S) \end{cases} \qquad (6)$$

$$\sum_{i=1}^{J} P_i(S)=1$$

The probability of complete time of activity *j* with duration $t_i(S)$ is $P_i(S)$, *J* is the total number of activities contained in the project.

But in the high degree of uncertainty environment, project buffer time P_B may not enough. The critical chain made before may be broken. The change of activity duration time or broken of critical chain would lead the whole project not stable. So it is necessary to rebuild the model above with robust optimization algorithm.

Robust project scheduling optimization model

Scene set $\Omega=\{1,2,K\ ,\Phi\}$ is introduced to describe uncertainty activity duration. Assuming that for the scene φ, suppose there are *J* activities, and the activity is independent of each other. $d_{j\varphi}$ is the duration of activity *j* in the scene φ, vector $d_\varphi=(d_{1\varphi},d_{2\varphi},\text{L}\ d_{J\varphi})$ is activities vector consist of all activity durations in this scene. In each scene φ, the probability of total duration of this activities vector as follows:

$$P_\varphi=\prod_{k=1}^{J} P_k^\varphi \quad \varphi=1,2,K\ ,\Phi; \qquad (7)$$

So the complete time of project T depends not only on the current activities scene vector d_φ, but also depends on the scheduling scheme of the project. The duration time is also depends on the scheduling scheme. It is possible to establish the following stochastic programming (SP) model:

SP:

$$\min E(\pi)=\frac{1}{\Phi}\sum_{\varphi\in\Omega} D_\varphi \qquad (8)$$

s.t.

$$T_{h\varphi}\le T_{j\varphi}-d_{j\varphi},\quad \forall h\in F_j,\ \forall j,\varphi \qquad (9)$$

$$\sum_{j\in I_t} r_{jk}\le R_k,\ \forall k,t \qquad (10)$$

$$P_B=\sqrt{\sum_{j\in J}\left(\sigma_j^\varphi\right)^2} \qquad (11)$$

Where D_φ is total duration time of activity in the scene φ :

$$D_\varphi=P_\varphi\sum_{i=1}^{J} d_{i\varphi} \qquad (12)$$

The stochastic programming above minimizes the total duration time. To further expand the stochastic programming model, we using robust optimization theory[9-10], then construct the following RCPSP robust optimization (RO) model:

RO:

$$\min Q(\pi) = E(\pi) + \lambda V(\pi) \tag{13}$$

s.t.

$$T_{h\varphi} \le T_{j\varphi} - d_{j\varphi}, \quad \forall h \in F_j, \ \forall j, \varphi \tag{14}$$

$$\sum_{j \in I_t} r_{jk} \le R_k, \ \forall k, t \tag{15}$$

$$P_B = \sqrt{\sum_{j \in J} \left(\sigma_j^{\varphi}\right)^2} \tag{16}$$

Where λ is weight coefficient of the model, $V(\pi)$ is variance of duration time:

$$V(\pi) = \frac{1}{\Phi} \sum_{\varphi \in \Omega} \left(D\varphi - E(\pi)\right)^2 \tag{17}$$

As for robust optimization model, we can introduce error term. A small amount to break through the constraint conditions can be tolerated. Because project scheduling scheme π in this paper uses the expression of activity list. A feasible project scheduling scheme corresponding activity list in all scenarios should also be workable. As for the stochastic programming robust optimization model, there is no error term. Instead, we can use the degree of closeness between calculate solutions in the different senses with the optimal solution, means the quality robustness of solutions. The variance of duration time is used in this work to evaluate quality of algorithm[11].

From the perspective of a single activity, the duration of activity *j* in all scenarios can approximate fitting arbitrary probability distribution. We can estimate the probability distribution by using historical data and people's experience. So that RCPSP robust optimization model has the universality and superiority.

Robust Optimization Genetic Algorithm

For the robust optimization model of RCPSP with stochastic activity durations, usually we can get Φ scenarios in advance, which is part of the whole scenarios. But we also can estimate the whole scenarios based on that. In this paper, we use genetic algorithm to solve the robust optimization model of RCPSP by small scene samples.

Coding Scheme. Genetic algorithm[12] has been widely applied to solve RCPSP problem. Coding is the most important issue when using genetic algorithms, and also a key step in genetic algorithm design, the purpose of this step is to generate activity scheduling. Generally adopt the activity list to encode chromosome, and decode by serial schedule generation scheme. That can ensure the solutions meet the limitations of precedence and resources at the same time[13]. In this paper, we use activity list as encoding plan.

Fitness function. Fitness function is also referred to as evaluation function. It is used to find the better from the groups according to the objective function. It's the driving force of the evolution process, and also the only basis for natural selection. The objective function of original RCPSP problem is to minimize total activity duration time. So for stochastic programming model and robust optimization model, the fitness functions designed as follow:

SP:

$$f(V) = \frac{1}{E(\pi)} \tag{18}$$

RO:

$$f(V) \frac{1}{E(\pi) + \lambda V(\pi)} \tag{19}$$

Where $f(V)$ is the fitness value of individual *V*. That means in Φ scenarios, the lower the objective function value, the higher the fitness of individuals.

Select operation. Use the roulette wheel method to select individual from the population. Winning probability of each individual $P(V)$ is directly proportional to its fitness value:

$$P(V)=\frac{f(V)}{\sum f(V)} \tag{20}$$

Crossover operation. Crossover operation is to simulate the process of natural evolution, by mating two homologous chromosomes form new chromosome recombination, new species or individual process. This paper adopts the single point crossover operator. Crossover probability is p_c. Two individuals participate in the crossover operation, one is called mother V_M, the other is called father V_F. Daughter V_D and son V_S are generated by crossover operation. The first *j* genes of daughter V_D inherit from mother V_M; and position *j*+1 to J come from father V_F, the existing activity in V_D is no longer considered, And keep the relative position of each activity in V_F. V_S can be obtained by the same way.

Mutation probability. Every time switched on a chromosome mutation, mutation probability is p_m. Switched variation means exchange two activities in the activity list; if switched variation breaks the precedence relation constraint, and then cancel it.

Steps of Algorithm. The specific algorithm steps are as follows:

1) Read all scenarios sample data, and set the genetic algorithm parameters.
2) According to the selected task precedence rules generating initial population, based on Eq. 18 or Eq. 19 to calculate the value of each individual.
3) Chose the individual according to Eq. 20; doing crossover operation with probability p_c, and variation Operation with probability p_m.
4) Repeat step 3 until a new generation of population.
5) According to the Eq. 18 or Eq. 19, Calculate the fitness value of each individual in the current population, retain and record the best one.
6) If the number of iterations reaches a predetermined number of generations, then go to step 7, otherwise, the number of iterations add 1, and return to step 3.
7) Output the best individual, the robust solution of project scheduling problem; terminate the program.

Numerical example

As an example, we consider a project with 10 activities[13]. The data of each activity and the relationship between activities are shown in Table 2. The total availability of resource is 30 workers per day.

Table 2 Data and results for activities in the sample project

number	1	2	3	4	5	6	7	8	9	10
Days	5	3	4.5	6	3	4	6.5	8.5	5	7
probability	0.3	0.4	0.5	0.2	0.3	0.1	0.3	0.2	0.5	0.1
Immediate predecessor	-	1	1，2	2	3，2	5	2，3	3，6	7，8	5，9
r_{jk}	80	120	65	70	85	40	20	50	100	30

When the activity duration time is determined, we use traditional critical chain project management (CCPM) method to calculate the whole project time. Schedule duration is 25d. Project buffer time is 4.5d. So the final project time is 29.5d.

Next, we suppose that there are uncertainty with activity duration time, and the probability of each activity as shown in the table 2. At the best situation, the result is 26.5d including the buffer time. When face the worst situation, the result is 41.5d. The fluctuation range of the duration is large with poor robustness. Therefore, if we make other decision based on that result, the risk also increases.

According to the design method in this paper, this project has 10 activities, so $J = 10$. 30 scenarios are randomly selected as actual scenarios project collection from whole scenarios. Then 10 scenarios are randomly selected as the small sample for solving, $\Phi = 10$.

Then using robust optimization model, applied the genetic algorithm for solving. Set the population size is 100, generations of 50, crossover probability $p_c = 0.5$, the mutation probability $p_m = 0.15$. Assuming that $\lambda = 1$, genetic algorithm uses crossover operation and mutation operation to update the population gradually, make project scheduling solution delay variance gradually reduced in all scenarios. A better scheduling scheme is obtained finally. After calculation, the expectation duration time $E(\pi)$ of this schedule plan under small sample $(\Phi = 10)$ is 27.5d without project buffer time. And the variance of duration time $V(\pi)$ is 3.2d.

Although the total duration of the project with robust optimization model obtained is increased to 27.5d, bigger than the result from situation of duration determined. But the variance of duration time is only 3.2d, it means fluctuation range greatly reduced, the risk is also reduced.

Gradually adjust the model weighting factor λ . A number of different scheduling schemes can be obtained. The variety of $E(\pi)$ and $V(\pi)$ of those scheduling schemes in the whole scenarios shown as follow:

Table 3 Impact of model weighting coefficient on solutions

λ	$E(\pi)$	$V(\pi)$
0	27.15	4.10
0.1	27.36	3.35
0.5	27.40	3.28
1	27.50	3.20

From the table 3 we can see, with the increase of λ , $E(\pi)$ is increasing, but the $V(\pi)$ is decreasing. This suggests that the project makers can chose different project scheduling schemes by adjusting the weights. If choosing smaller weight coefficient, we can get a smaller duration time $E(\pi)$ of project scheduling schemes, but the variance is larger, that means there exist a higher risk. If we set $\lambda = 0$, it actually becomes stochastic programming model, the variance of this project scheduling scheme is the largest. If we chose a bigger weight coefficient, the deferred expectation wills slightly increasing, but the variance gradually reduces, that means effective inhibition of project risk. Therefore, project makers can choose the appropriate weighting coefficient to generate a suitable project scheduling scheme according to the risk preference.

Conclusion

In the project practice, the activity duration uncertainty brought inconvenience for project scheduling. Robust scheduling scheme proposed in this paper can be solved by robust optimization model with small sample scenarios. Project makers can also weigh between the feasible solution and the optimal solution by controlling the weight coefficient. In a follow-up study can consider other schedule expression to design the robust optimization model, make the robust solution close to the optimal solution under different scenarios better. At the same time, we can improve genetic algorithm or adopt other effective intelligent algorithm, in order to increase the efficiency of solving the robust optimization model.

References

[1] Kolisch R, Padman R. An integrated survey of deterministic project scheduling [J]. Omega, 2001, 3(29): 249～ 272.

[2] Kolisch R, Hartmann S. Experimental investigation of heuristics for resource-constrained project scheduling: An update. European Journal of Operational Research[J]. 2006, 174: 23～ 37.

[3] Demeulemeester E, Herroelen W. Introduction to the special issue: project scheduling under uncertainty[J]. Journal of Scheduling. 2007, 10(3):151～ 152.

[4] Long, L D and Ohsato, A. Fuzzy critical chain method for project scheduling under resource constraints and uncertainty. International Journal of Project Management. 2008. 26(6): 688~698.

[5] Goldratt E. Critical chain. The North River Press Publishing Corporation. 1997. 246.

[6] Rand, G K. Critical chain: the theory of constraints applied to project management[J]. International Journal of Project Management. 2000. 18(3): 173~177.

[7] Daozhi Z, Hua L, Yiliu L. Critical chain method: A new project scheduling approach[J]. Journal of Tianjin University of Technology. 2005. 21(2).

[8] Hanhua L, Xindu C, Lihua W, Liming L. Scheduling of Flexible Flow Shop for Mold Manufacturing with Processing Time Uncertainty[J]. Industrial Engineering Journal. 2012. 15(1).

[9] Mulvey J M, Vanderbei R J, Zenios S A. Robust optimization of large-scale systems[J]. Operations Research, 1995, 43(2):264～281.

[10] Yongyi S, Wei W. A Robust Optimization Model Based Genetic Algorithm for Project Scheduling Policies[J]. Journal of Industrial Engineering Engineering Management. 2009. 23(4).

[11] Valls V, Laguna M, Lino P, Perez A, Quintanilla S. Projec scheduling with stochastic activity interruptions[A]. In: Weglarz J. (Ed.), Project Scheduling: Recent Models, Algorithms and Applications[C].Kluwer, 1999, 333～353.

[12] Grefenstette J J. Genetic algorithms for the traveling salesman problem[C]. Proc. 1st International Conference on Genetic Algorithms and their Applications, Pittsburgh, PA. 1985, 160~168.

[13] Ying Y, Yongyi S, Min L. Hybrid genetic algorithm for resource constrained multi-project scheduling problem[J]. Journal of Zhejiang University . 2009, Vol. 43(1):23～ 27.

[14] Zhigang S. Critical Chain Management Under Resources Constraint[D]. Nanjing: HoHai University. 2007.