

Disruption management for resource-constrained project scheduling based on differential evolution algorithm

Weiming Chen^{*}, Xiaoyang Ni, Hailin Guo

Faculty of Engineering, China University of Geosciences, Wuhan, 430074, China

Received 1 March 2014, www.tsi.lv

Abstract

In this paper, we study the problem of how to react when an ongoing project is disrupted. The focus is on the resource-constrained project scheduling problem with finish–start precedence constraints and the recovery strategies based on disruption management for the different types of disruptions are proposed. The goal is to get back on track as soon as possible at minimum cost, where cost is now a function of the deviation from the original schedule. The problem is solved with a differential evolution (DE) algorithm that can be solved more perfectly on the objective function. The new model is significantly different from the original one due to the fact that a different set of feasibility conditions and performance requirements must be considered during the recovery process. Project scheduling problem library (PSPLIB) has been taken into account so as to test the effect of novel hybrid method. Simulation results and comparisons determine the effects of different factors related to the recovery process and show that the differential evolution algorithm is competitive and stable in performance.

Keywords: disruption management, scheduling, resource-constrained, differential evolution

1 Introduction

Project scheduling has attracted an ever-growing attention in recent years both from science and practice. Nowadays, enterprises have to focus more on improving product development efficiency due to the global economic crisis and the increasingly intense market competition. To gain more market share, it is critical for enterprises to reduce product development time and cost with limited resources and shorten the time-to-market. One of decision problems that in practice often involve uncertain information is resource-constrained project scheduling problem (RCPSP).

Projects are often performed under high levels of uncertainty related to such factors as resource availability, unproven technology, team competence, and the commitment of upper management. Sometimes, even the project goal is not well defined when the work begins. For most projects, though, a schedule specifying the implementation details must be developed before uncertainties are resolved. Without any historical data or past experience, expert opinion and rough estimates might be the only way to quantify activity costs and durations in the initial planning stages. What results is an initial schedule designed to optimize some objective within the limits of uncertainty. Eden et al [1] pointed out that it is very hard to estimate the cost of delay and disruption for most real world projects. The primary purpose of this paper is to apply the growing field of disruption management (DM) [2, 3] for examining and resolving this type of problem. Disruption management is

an emerging field in which operations research techniques are applied to help resolve uncertainties as they unfold. The problem that must be solved may be significantly different from the initial planning problem because it contains new decision variables, new constraints, and a new objective. Beside air traffic and airline-related scheduling, the domains of machine scheduling and production planning have been at the centre of research in disruption management [4]. Bean et al. [5] were among the first to consider deviation costs in their approach to match up scheduling, which is based on the idea of identifying an updated schedule that converges with the original one at some early point in the future. While Clausen et al. [1] discuss disruption management in the execution of shipbuilding processes; Xia et al. [6] investigate DM in the context of a two-stage production and inventory system, evaluating solutions for fixed and flexible setup epochs as well as different forms of penalty functions. Yang et al. [7] consider cost and demand disruptions occurring on a single-product manufacturing plant and propose a pseudo-polynomial dynamic programming procedure for the general cost case and present advanced solution procedures for specific forms of cost functions. Additional information and comprehensive overviews of DM in the context of production planning can be found in [8] and [3]. Apart from the areas of application mentioned above, disruption management plays a crucial role in the context of many other real-world processes. Research, for example, has been conducted in the domains of telecommunication [2], project management [9, 3, 10], supply chain coordination [11, 12] and logistics management [3]. Al-Fawzan et al

^{*}Corresponding author e-mail: chenpool@126.com

[13] introduce the concept of schedule robustness and we develop a bi-objective resource-constrained project scheduling model and several variants of the algorithm are tested and compared on a large set of benchmark problems. Hur et al [14] define such a setting as the real-time work schedule adjustment decision, proposes mathematical formulations of the real-time adjustment and develops efficient heuristic approaches for this decision. Herroelen et al [15] review the fundamental approaches for scheduling under uncertainty: reactive scheduling, stochastic project scheduling, fuzzy project scheduling, robust (proactive) scheduling and sensitivity analysis and discuss the potentials of these approaches for scheduling under uncertainty projects with deterministic network evolution structure. Howick [16] et al use System dynamics (SD) to analyse disruption and delay, making an informed decision about the appropriateness of SD as a modelling approach to support any specific claim for compensation. Vonder et al [17] discuss computational results obtained with priority-rule based schedule generation schemes, a sampling approach and a weighted-earliness tardiness heuristic on a set of randomly generated project instances. Demeulemeester et al [18, 19, 20] review the fundamental approaches for scheduling under uncertainty and discuss the potentials of various approaches for scheduling under uncertainty projects with deterministic network evolution structure.

Despite of noticeable progress in the researches on DM (disruption management), DM applications to date to solving practical difficulties are still in great constraints, especially in the field of project scheduling, where fruits of multi-mode DM study are yet to be added to present few. With hope of advancing further, this section introduces a framework for solving the problem of multi-mode project scheduling with uncertainties, including task-disruption and resource-disruption. In section 2, we give an outline of DM studies on project scheduling with uncertainties, followed by section 3, adjustment strategies of DM and its mathematical model. In the section 4, differential evolution algorithm is introduced to tackle project scheduling under uncertainties. Differential evolution algorithm is employed to calculate examples and analyse the outcomes in section 5. The last section is conclusion of the whole paper.

2 Principle of Project Scheduling in DM

Project scheduling in DM is to bring the project deviating from the original schedule back on track with recovery strategies when the project is still in progress, as well as to minimize the effect of disruption. As is shown in Figure 1, the flow of project scheduling mainly includes disruption types, recovery strategies, recovery objectives and recovery constraints. This section will explain and analyse the first two, based on which, the latter two aspects will be explained in the next section.

2.1 DISRUPTION TYPES ANALYSIS

This section will make further analysis and explanation on the specific types of disruptions.

1) Task delay disruption

Task delay disruption is mainly caused by project structure disruption. During the implementation of project, some disruptions due to requirements change or other environmental factors may contribute to new task needed to be added or to task stop, which triggers shift in task number. Meanwhile, some change in task priority will result in difference of project structure in project scheduling, which may bring about disruption in the project structure itself. In this way, project structure disruption would make project scheduling vulnerable and infeasible, and eventually tasks in original schedule could not proceed as planned so as to produce delay disruption.

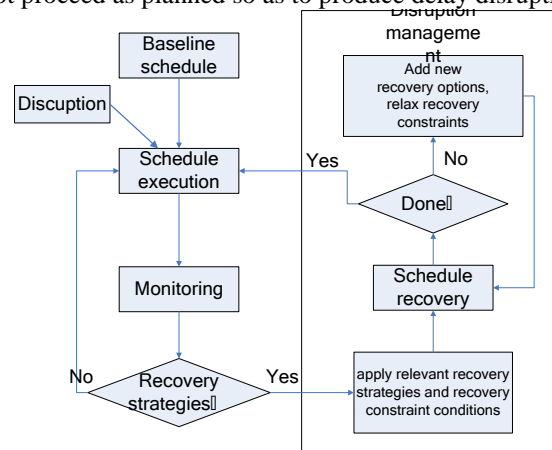


FIGURE 1 Flow of project scheduling in DM

2) Task completion time disruption and resource consumption disruption

Task disruption main consist of completion date disruption and resource consumption disruption. Task completion date disruption refers to the failure of task completion on time as scheduled; in most cases, delay is expected. Resource consumption disruption is the overuse of resource compared with that allocated in original schedule. The impacts of task disruption are as follows: for one thing, delay of task completion will postpone the opening of next task, thereby impairing the implementation of original schedule and raising the cost of the project; for the other, the postponed task will grab the resource initially allocated to other task, which may put off the execution of parallel task.

3) Resource available disruption

Resource disruption refers to that of current available resource, usually denoting the resource available cannot meet the needs of project implementation, namely resource deficiency. This is the commonest type of disruption in project scheduling. The main reasons for this disruption are equipment malfunction, stuff insufficiency, over-consumption of resource by other tasks or projects. The impacts of resource available disruption are the following: on the one hand, the task

about to begin needs to be postponed, in order to wait for the completion of other tasks when available resource is restored; on the other hand, backup resource needs to be manoeuvred to meet the requirement of task execution.

4) Delivery time disruption

Delivery time disruption will not impact the implementation of project; however, its influence on project itself is evident. The delay of delivery will cause the offence of project against the contract, thus leading to the failure of the project to some extent. Mainly affected by previous three disruptions, delivery time disruption is valuable when acting as a cost punishment measure to restrict the options of disruption recovery strategies, as those strategies are usually used to curb the disruption in project execution.

2.2 DISRUPTION RECOVERY STRATEGIES

As the key to disruption management, disruption recovery strategies are flexible decision related to disruptive events. These strategies share the same model parameters with disruptive events, while the essence is distinct. Disruption is the fact that project scheduling deviate from the original under the impact of external uncertainties, whereas disruption recovery strategies serve as a decision variable to amend the disrupted process in project scheduling.

1) Task Execution Mode Substitution Strategy

With respect to multi-mode project scheduling, the task execution modes are diverse implementation mode of the project, each of which responds to different completion date and resource consumption. When the task execution is delayed, alternative execution mode could be employed to restore the disrupted scheduling, so as to ensure the task completion on schedule and keep other tasks free from disruption. On the other hand, when the task is delayed for the deficiency of resource, the execution modes of some tasks could be altered to restore the execution of subsequent task. The change in execution mode of tasks may accelerate the task-implementing speed, bringing project scheduling back on track faster, at the expense of project cost, however. To keep the balance between project scheduling and cost control, we define a variable C_{im} as mode-switching cost, the extra cost produced when the execution mode of task i is converted from m in original schedule to m' in disruption management. C_{im} reflects the decision cost on the conversion of execution mode of the task.

2) Resource Substitution Strategy

Resource is the foundation of project implementation. The impact of resource insufficiency on project is decisive. Considering this, when disrupted, additional resource could be provided to ensure the project free from impedance of disruption. However, increase of available resource may bring about rise in cost, which means that additional resource provision is restricted within the project budget. Based on the balance between project scheduling and cost control, we define $g(r)$ as resource

punishment coefficient, to express the cost of resource r in units.

3 Disruption management model for project scheduling

The current universal form of disruption management model is

$$\min f(x), \quad (1)$$

$$\text{subject to } x \in X. \quad (2)$$

Expression (1) is the objective function, and $f(x)$ is the disruption degree function; expression (2) is the constraint condition. The objective of disruption management is to minimize the degree of deviation of new scheme from the original after disruption.

According to uncertain disruption types, we propose a disruption management model based on disruption recovery strategies in this section. This model consists of disruption recovery objective function and disruption recovery constraint condition. The disruption recovery objective function is to minimize the project disruption, the component variable includes not only recovery cost caused by the adopted disruption recovery strategy, but also punishment for execution delay and delivery delay due to disruption; the disruption recovery constraint condition defines the constraint conditions for each factor in amended project scheduling.

3.1 OBJECTIVE FUNCTION

We define recovery objective function as the following:

$$\min Q(x) = D(x) + C(x) + G(x) + P(x), \quad (3)$$

where $D(x)$ is the deviation between project scheduling and implementation after delay of task start, and we define w_i as a weight of delay in start-up punishment, which means the cost caused by the deviation of start delay for per time unit from the original schedule; let the start time of task i in the original schedule be s_i , the real start time s'_i .

According to the definition above, $D(x)$ can be expressed as below,

$$D(x) = \sum_i w_i \left([s'_i - s_i]^+ + [s_i - s'_i]^+ \right), \quad (4)$$

where $[z] = \max\{0, z\}$.

$C(x)$ is the extra cost produced when substitution strategy of task execution mode is employed; for multi-mode task, we define the following resource substitution decision variable

$$x_{imt} = \begin{cases} 1 & \text{Task } i \text{ is implemented in mode } m \text{ at the time } t \\ 0 & \text{Otherwise} \end{cases} \quad (5)$$

According the definition above, we have

$$C(x) = \sum_{i,m} C_{im} \sum_t x_{imt} \quad (6)$$

$G(x)$ is the extra cost of additional resource increase when resource substitution strategy is employed, where we define resource substitution decision variable y_r

$$y_r = \begin{cases} 1 & \text{Resource substitution strategy } r \text{ is employes} \\ 0 & \text{Otherwise} \end{cases} \quad (7)$$

Therefore, $G(x)$ can be expressed as below

$$G(x) = \sum_r g(r)y_r \quad (8)$$

$P(x)$ is the contract punishment for project delivery delay; let variable t_θ be the delivery time on schedule, variable η be the contract punishment cost of delivery delay for per time unit, we have

$$P(x) = \eta_i ([t_\theta - \sum_{i,m,t} tx_{imt}]^+ + [\sum_{i,m,t} tx_{imt} - t_\theta]^+), \quad (9)$$

The objective function is to minimize the sum of $D(x)$, $C(x)$, $G(x)$ and $P(x)$, so as to minimize the effect of disruption on project.

3.2 CONSTRAINT CONDITION

The ultimate objective of disruption management is to ensure the smooth execution of the project, in the way that the disruption recovery strategy can tackle the disruption somehow; thereby, the disruption recovery strategy is expected to meet the constraint condition of project scheduling. We express the constraint condition as below

$$\sum_{m \in M_i} \sum_{t \in T_i} x_{imt} = 1, i \in A \cup A_N, \quad (10)$$

$$\sum_{m \in M_i} \sum_{t \in T_i} tx_{imt} - \sum_{m \in M_j} \sum_{t \in T_j} (t - p_{jm})x_{imt} \leq 0, (i, j) \in P_N \cup P_A \cup (P \setminus P_R), \quad (11)$$

$$\sum_{i \in \pi_k} \sum_{i \in A \cup A_N} \sum_{m \in M_i} \sum_{q=t}^{t+p_{im}-1} r_{imk} x_{imq} \leq R_{pk} - \rho_{pk} + y_{\pi_k} R(\pi_k), \pi_k \in \prod k, k \in K, \quad (12)$$

$$x_{im_0f_i} = 1, \forall i \in A_F, \quad (13)$$

$$\sum_{m \in M_i} \sum_{t_1 \leq t \leq t_2} x_{imt} = 1 \quad (14)$$

Expression (10) ensures that each task has a unique completion time; Expression (11) and Expression (12) are task precedence relationship and resource constraint condition respectively; Expression (13) means the task outside of the time-restoring window is implemented as originally scheduled; Expression (14) is task-restoring constraint.

4 Differential evolution for disruption management in project scheduling

The advancements in meta-heuristics in recent years, related mainly to the development of more efficient computational algorithms have enabled the solution of complex problems by means of numerical optimization algorithms [21]. One of these modern meta-heuristics is the Differential Evolution (DE), an evolutionary computation method. The DE developed by Storn and Price [22] is one of the most superior algorithms. The DE have become widely used in engineering optimization [23–28] due to its simple structure, ease of use, convergence speed, versatility, and robustness. The main difference between genetic algorithms and DE is that, in genetic algorithms, mutation is the result of small perturbations to the genes of an individual (potential solution) while in DE, mutation is the result of arithmetic combinations of individuals.

Storn and Price [22] first introduced the DE algorithm a few years ago. DE is similar to genetic algorithms in that a population of individuals is used to search for an optimal solution. DE combines simple arithmetic operators with the classical operators of crossover, mutation and selection to evolve form a randomly generated starting population to a final solution.

DE offers the advantage of incorporating a relatively simple and efficient form of self-adapting mutation. The fundamental idea behind DE is a scheme whereby it generates the trial parameter vectors. The population of a DE is subject to operators of mutation, crossover and selection. In each time step, DE mutates vectors by adding weighted random vector differentials to them. If the cost of the trial vector is better than that of the target, the target vector is replaced by trial vector in the next generation.

Storn and Price [22] proposed 10 different strategies for DE based on the individual being perturbed, the number of individuals used in the mutation process and the type of crossover used. The strategy implemented here was DE/rand/1/bin, meaning that the target vector is randomly selected, and only one difference vector is used. The bin acronym indicates that the recombination is controlled by a binomial decision rule.

The optimization procedure of DE/rand/1/bin is given by the following steps:

Step 1: Choice of the control parameters, including population size (M), boundary constraints of optimization variables, mutation factor (f_m), crossover rate (c_r), and the stopping criterion (t_{max}).

Step 2: Initialization of population with M individuals. Set generation $t=0$. Initialize a population of $i=1,2,\dots,M$ individuals (real-valued N -dimensional solution vectors) with random values generated according to a uniform probability distribution in the N -dimensional problem space as following equation.

$$x_{ij}^{(0)} = x_{\min,j}^{(0)} + rand \cdot (x_{\max,j}^{(0)} - x_{\min,j}^{(0)}), \quad (15)$$

where $i=1,2,\dots,M$ is the individual's index of population; $x_i^{(t)} = [x_{i1}^{(t)}, x_{i2}^{(t)}, \dots, x_{in}^{(t)}]^T$ stands for the i th individual of a population of real-valued N -dimensional solution vectors; t is the generation (time); $rand$ is random value generated according to a uniform probability distribution in $[0,1]$; $x_{\max,j}$ and $x_{\min,j}$ stand for the upper bound and lower bound of the i th individual of j th real-valued vector.

Step 3: For each individual, evaluate its fitness value.

Step 4: Mutation operation (or differential operation). Mutation is an operation that adds a vector differential to a population vector of individuals according to equation:

$$z_i^{(t+1)} = x_{i,r_1}^{(t)} + f_m^{(t)} \cdot [x_{i,r_2}^{(t)} - x_{i,r_3}^{(t)}], \quad (16)$$

where $z_i^{(t)} = [z_{i1}^{(t)}, z_{i2}^{(t)}, \dots, z_{in}^{(t)}]^T$ for the i th individual of a mutant vector, r_1, r_2 and r_3 are mutually different integers and are also different from the running index i randomly selected with uniform distribution from the set $\{1,2,\dots,i-1,i+1,\dots,N\}$. The mutation factor $f_m^{(t)} > 0$ is a real parameter, which controls the amplification of the difference between two individuals with indexes r_2 and r_3 .

The mutation operation using the difference between two selected randomly individuals may cause the mutant individual to escape from the search domain. If an optimized variable for the mutant individual is outside of the domain search, then this variable is replaced by its lower bound or its upper bound so that each individual should be restricted with the search domain.

Step 5: Evaluate Operation. Evaluate is employed to generate a trial vector by replacing certain parameters of the target vector by the corresponding parameters of a randomly generated donor vector. For each vector $z_i^{(t+1)}$, an index $mbr(i) \in \{1,2,\dots,n\}$ is randomly chosen using uniform distribution, and a trial vector, $u_i(t+1) = [u_{i_1}(t+1), u_{i_2}(t+1), \dots, u_{i_n}(t+1)]^T$, is generated via:

$$u_{ij}(t+1) = \begin{cases} z_{ij}(t+1) & \text{if } randb(j) \leq CR \text{ or } j = mbr(i) \\ x_{ij}(t) & \text{if } randb(j) > CR \text{ or } j \neq mbr(i) \end{cases}, \quad (17)$$

where j is the parameter index; $x_{ij}(t)$ stands for the i th individual of j th real-valued vector; $z_{ij}(t)$ stands for the i th

individual of j th real-valued vector of a mutant vector; $u_{ij}(t)$ stands for the i th individual of j th real-valued vector after crossover operation; $randb(j)$ is the j th evaluation of a uniform random number generation with $[0,1]$; CR is a crossover rate in the range $[0, 1]$.

Step 6: Selection operation. Selection is the procedure whereby better offspring are produced. To decide whether or not the vector $u_i(t+1)$ should be a member of the population comprising the next generation, it is compared with the corresponding vector $x_i(t)$. Thus, if f denotes the objective function under maximization, then

$$x_i(t+1) = \begin{cases} u_i(t+1) & \text{if } f(u(t+1)) > f(x_i(t)) \\ x_i(t) & \text{otherwise} \end{cases}. \quad (18)$$

In this case, the value of objective function cost of each trial vector $u_i(t+1)$ is compared with that of its parent target vector $x_i(t)$. If the objective function f of the target vector $x_i(t)$ is upper than that of the trial vector, the target is allowed to advance to the next generation. Otherwise, the target vector is replaced by a trial vector in the next generation.

Step 7: Verification of the stopping criterion. Set the generation number for $t=t+1$. Proceed to Step 3 until a stopping criterion is met, usually a maximum number of iterations (generations), t_{\max} . The stopping criterion depends on the type of problem.

5 Simulation Experiments

To verify the validity of disruption management and algorithm discussed in this chapter, we conduct the experiments using C++ for encoding, and employ the multi-mode scheduling test packs J20 and J30 in project scheduling standard question bank PSPLIB[29], to make test on the PC with Intel® Core™2 Duo 2.4GHz CPU. Here J20 and J30 include 20 and 30 tasks respectively (not including virtual tasks); for each task, there are 3 types of execution modes for option, in each of which, the task duration is 1-10 time units, consuming 2 kinds of renewable resources and 2 kinds of non-renewable resources. For test questions of each group, randomly generate a question. Therefore, there are 640 questions for each group; however, there exists no feasible solutions for some of these test questions. Because of this, eliminate those insolvable questions, there are 554 project cases left in J20, and 552 left in J30. During the experiment, we make discussion and analysis on the minimal disruption cost of the projects with and without project delivery time limits respectively.

5.1 PARAMETERS SETTING

The cases in question bank J20 and J30 are both about project scheduling under certain circumstance. However, as project scheduling is in the complex environment

during the execution, some disruption parameters in J20 and J30 needs to be set to make the project scheduling disruption management close to reality. Firstly, in dynamic environment, the completion time of project scheduling is no shorter than that of baseline scheduling at planning stage. For the baseline project completion time generated by the Certain Scheduling Algorithm solution in J20, we extend it by 20%, 10% and 0% respectively as the expected completion time; for that in J30, we extend it by 40%, 30% and 20% respectively as the expected completion time. The extension of expected completion time will increase the amount of scheduling schemes, thereby affecting the time cost of solution directly. Secondly, we add two tasks to J20 and J30 respectively, which stand for the disruption of project structure caused by the requirement change by clients at the execution stage. Besides, we define the 10%-extension of completion time of two random tasks of the project caused by disruption, and 10% more resource consumption in the execution of two random tasks than the original. In addition, the resource consumption may

be 5% fewer at some period of execution than the original supply. Lastly, the parameter values are defined for delay in start-up weight w_i , mode-switching cost, delayed delivery punishment, resource punishment coefficient and the time of delay in delivery. The detailed question parameter settings are shown in Table 1.

Here are the parameter settings in DE algorithm, the maximum evolutionary generation T is set to 100, the population size N is 40, the crossover probability P_c is 0.9, and the scaling factor F is 0.7. The proposed algorithms adopt penalty function method to deal with constraints using following equations:

$$f(w) = F(w_i, u_{hj}, s_{ih}) + M(\sum_{i=1}^m w_i - 1)^2, \tag{19}$$

where M is the penalty impact. In the numerical experiments, M is set to 105 for the J30 and J60, and 106 for the J60 and J120.

TABLE 1 Parameter settings

Benchmark	J20, J30
Question bank size	554, 552
Project structure disruption	2 new tasks
New task 1	Execution duration:3, resource consumption, precedence task, subsequent task
New task 2	Execution duration:3, resource consumption, precedence task, subsequent task
Execution duration extension	2 random tasks, 10% extended
Resource consumption	2 random tasks, random resource 10% more consumed
Resource disruption	5% fewer resource, resource recovery time randomly got from [1, 5]
Delay in start-up punishment weight w_i	Uniform distribution between [1,10]
Mode switching cost	Uniform distribution between [0,5]
Delivery delay punishment	20
Resource punishment coefficient	3
Delayed time in delivery	No longer than 40% of original execution duration

5.2 COMPUTING RESULTS AND ANALYSIS

For the computing process, at first project scheduling method is employed to get the optimal/suboptimal scheduling scheme as the planning one, and then computation is implemented on the minimal disruption cost of projects with and without delivery time limit, based on the question parameters and algorithm parameters discussed above. Meanwhile, to simulate the

real situation in project execution and analyse the influence of different disruption strategies on project execution, single and combined disruption strategies are taken respectively in simulative computations.

1) No delivery time limit

Table 2 shows the computing results for minimal disruption cost without delivery time limit with different disruption strategies.

TABLE 2 Computing results

Benchmark	Strategy	Average disruption cost	Average execution duration	Scheduled average execution duration	Real deviation rate (%)
J20	Task execution mode	238.46	34.52	27.71	24.58%
	Resource substitution	295.73	33.29	27.71	20.14%
	Combined	172.39	32.12	27.71	15.91%
J30	Task execution mode	484.24	46.85	33.38	40.35%
	Resource substitution	566.73	44.27	33.38	32.62%
	Combined	401.96	42.43	33.38	27.11%

As can be seen in Table 2, size of question bank exerts great influences on disruption in project execution. The deviation rate of average disruption cost in benchmark J20 from original scheduling is superior to that in J30, which indicates that the more complex the

project structure is, the greater disruptions received in dynamic environment, thus the more difficult to implement disruption recovery.

Meanwhile, different disruption recovery strategies also have enormous influences on the disruption recovery

cost. With respect to this cost, that of projects with combined strategies is the minimal, followed by that with Task Execution Mode Substitution Strategies, while that with Resource Substitution Strategies is the largest. Concerning the deviation rate of total real execution duration from the original scheduling that with Combined Disruption Recovery Strategies is the least, and next is the one with Resource Substitution Strategies, while that with Task Execution Mode Substitution Strategies is the greatest.

Based on the results above, for the disruption during project execution, resource supply increase can shorten the total real execution duration which entails dramatic

cost rise, thus not the optimal scheme; given the resource condition, change of task execution mode can bring down the disruption cost to some degree at the expense of execution duration, thus leading to delay in delivery; combined strategies can reduce the risk of project execution and disruption cost to the largest extent by changing the execution mode with the permission of resource substitution.

2) Given delivery time limit

Table 3 and 4 reveals the computing results for the questions in J20 and J30 on the minimal disruption cost with delivery time limit with different disruption strategies.

TABLE 3 Computing results

Benchmark	Strategy	Average disruption cost	Average execution duration	Scheduled average execution duration	Real deviation rate (%)
≤40%	Task execution mode	253.61	38.73	27.71	39.77%
	Resource substitution	314.03	37.49	27.71	35.29%
	Combined	199.47	36.53	27.71	31.83%
≤30%	Task execution mode	217.46	35.64	27.71	28.62%
	Resource substitution	279.83	34.76	27.71	25.44%
	Combined	164.35	33.98	27.71	22.63%
≤20%	Task execution mode	249.79	33.18	27.71	19.74%
	Resource substitution	306.02	32.39	27.71	16.89%
	Combined	193.72	31.83	27.71	14.87%
≤10%	Task execution mode	268.45	30.42	27.71	9.78%
	Resource substitution	329.78	29.87	27.71	7.80%
	Combined	228.75	29.12	27.71	5.09%

TABLE 4 Computing results

Benchmark	Strategy	Average disruption cost	Average execution duration	Scheduled average execution duration	Real deviation rate (%)
≤50%	Task execution mode	512.33	49.63	33.38	48.68%
	Resource substitution	589.37	48.47	33.38	45.21%
	Combined	433.92	46.81	33.38	40.23%
≤40%	Task execution mode	473.51	46.71	33.38	39.93%
	Resource substitution	543.39	45.26	33.38	35.59%
	Combined	403.58	43.98	33.38	31.76%
≤30%	Task execution mode	525.46	43.18	33.38	29.36%
	Resource substitution	587.24	42.39	33.38	26.99%
	Combined	436.69	40.43	33.38	21.12%
≤20%	Task execution mode	566.87	39.82	33.38	19.29%
	Resource substitution	609.35	38.87	33.38	16.45%
	Combined	475.43	37.12	33.38	11.20%

It can be seen in Table 3 and 4 that similar to Table 1, size of benchmark and different disruption recovery strategies still have significant influence on the disruption during project execution. With regards to deviation of different delivery time limits, the deviation rate of average disruption cost from original scheduling in question bank J20 is superior to that in J30. As for the solution results of each question bank, different disruption recovery strategies is also greatly influential on recovery cost. The recovery cost with combined strategies has the least recovery cost, followed by that with Task Execution Mode Substitution Strategies, while that with Resource Substitution strategies is the largest. Judging from the deviation rate of total real execution duration from the original scheduling, that with combined strategies is the minimal, and that with Resource

Substitution Strategies comes next, while that with Task Execution Mode Substitution Strategies is the largest.

For the same benchmark, the delivery time limit also has enormous influence on the recovery cost and total execution duration. As we can see from Table 3, in benchmark J20, when the delivery time limit is not 30% longer than the original scheduling, the recovery cost becomes the highest as the delay time in delivery shortens; when the delivery time limit is not 30% longer than the original delivery schedule, the recovery cost is the least, followed by that when delivery time limit is not 20% longer, while that when delivery time limit is not 10% longer is the highest. This indicates that the bigger the deviation of delivery time limit from the scheduled total execution duration, the lower the disruption recovery cost is. That is, loose delivery time limit may

bring low execution cost. The same conclusion could be got from the result in Table 3.

However, the delivery time limit could not be extended without boundaries. We can see from Table 2 and Table 3 that when the delivery time limit is not 40% and 50% longer than the scheduled execution duration, the disruption execution cost increases. This is because despite of the decline of cost of mode switching and resource substitution in project execution, the degree of this kind of decline could not compensate the punishment cost caused by delayed delivery. Therefore, in project management, proper project completion schedule helps reduce the disruption cost that the dynamic environment entails in the project execution process.

Based on the computing results in single and combined strategies respectively with and without delivery time limit, to deal with disruption in execution process of project scheduling, the selection of disruption recovery strategies needs to be in accordance with delivery time and cost of the project. Within the delivery time limit, the total project execution duration can be extended to the largest extent and the execution mode can be switched to ensure a somehow soft execution and to reduce the disruption cost of the project. Meanwhile, within the permission of developing cost, Resource Substitution Strategies could be employed to shorten the overall execution duration and lower the disruption cost. This is one of the reasons why developing task outsourcing is employed currently in a large number of products development projects.

6 Conclusions

This chapter analyses the disruption types in the execution process of project scheduling, namely task

delay disruption, task duration and resource consumption disruption, resource available disruption and delivery time disruption, and raises two corresponding disruption recovery strategies, Task Execution Mode Substitution Strategy and Resource Substitution Strategy. On the basis of this, we establish a disruption management model in dynamic environment, whose target function is to minimize disruption recovery cost.

At the end, we use DE to solve the computation of minimal disruption cost problem and analyse the computing results, with and without delivery time limit in multi-mode scheduling test pack J20 and J30 from the standard question bank in PSPLIB. The analysis indicates that within the permission of resource substitution, the change of task execution mode can reduce the executive risk and decrease the disruption cost to the greatest extent. This partly explains why task development outsourcing is widely used for lots of product development projects.




Through the model and computations in this paper, we can draw the following conclusion for practical engineering applications: task outsourcing for product development and proper scheduling of project execution duration can facilitate the disruption-resistance and risk-resistance capability of product development project, thus improving the product developing efficiency and quality.

Acknowledgments

This study was provided by China Postdoctoral Science Foundation funded project (Grant no. 2013M542091), and the Fundamental Research Funds for National University, China University of Geosciences (Wuhan) (Grant no. CUG120824).

References

- [1] Colin E, et al 2000 *Journal of the Operational Research Society* **51**(3) 291-300
- [2] Clausen J, Hansen J, Larsen J, Larsen A 2001 *Disruption management* **28** 40-3
- [3] Yu G, Qi X 2004 *Disruption Management: Framework, Models, Solutions and Applications* World Scientific Publishers: Singapore
- [4] Yang, Jian, Xiangtong Qi, Gang Yu 2005 *Naval Research Logistics* **52**(5) 420-42
- [5] Bean J C, et al 1991 *Operations Research* **39**(3) 470-83
- [6] Xia Yusen, et al 2004 *IIE transactions* **36**(2) 111-25
- [7] Qi Xiangtong, Bard J F, Gang Yu 2006 *International Journal of Production Economics* **103**(1) 166-84
- [8] Herroelen W, Roel L 2005 *European journal of operational research* **165**(2) 289-306
- [9] Zhu Guidong, Bard J F, Gang Yu 2005 *Journal of the Operational Research Society* **56**(4) 365-81
- [10] Xu Minghui, et al 2003 *Journal of Systems Science and Systems Engineering* **12**(1) 82-97
- [11] Qi Xiangtong, Bard J F, Gang Yu 2004 *Omega* **32**(4) 301-12
- [12] Al-Fawzan, Mohammad A, Mohamed Haouari 2005 *International Journal of production economics* **96**(2) 175-87
- [13] Hur Daesik, Mabert V A, Bretthauer K M 2004 *Production and Operations Management* **13**(4) 322-39
- [14] Herroelen W, Roel L 2005 *European journal of operational research* **165**(2) 289-306
- [15] Howick S 2003 *Journal of the Operational Research Society* **54**(3) 222-9
- [16] Van de Vonder Stijn, et al 2007 *Computers & Industrial Engineering* **52**(1) 11-28
- [17] Lambrechts O, Demeulemeester E, Herroelen W 2007 *International Journal of Production Economics* **111**(2) 493-508
- [18] Lambrechts O, Demeulemeester E, Herroelen W 2008 *Journal of scheduling* **11**(2) 121-36
- [19] Deblaere F, Demeulemeester E, Herroelen W 2011 *Computers & Operations Research* **38**(1) 63-74
- [20] Taghizadeh N, Neirameh A 2012 *International Journal of Computing Science and Mathematics* **3**(4) 332-40
- [21] Storm R, Price K 1997 *Journal of global optimization* **11**(4) 341-59
- [22] Coelho L S, Mariani V C 2006 *IEEE Trans Power System* **21**(2) 989-96
- [23] Coelho L S, Mariani V C 2006 *IEEE Trans Power System* **21**(3) 1465
- [24] Liu B, Wang L, Jin Y H, Huang D X, Tang F 2007 *Chaos, Solutions & Fractals* **34**(2) 412-9
- [25] Chang Wei-Der 2007 *Chaos, Solutions & Fractals* **32**(4) 1469-76
- [26] Coelho L S, Mariani V C 2007 *Energ. Convers Manage* **48**(5) 1631-39
- [27] Peng Bo, et al 2009 *Chaos, Solutions & Fractals* **39**(5) 2110-8
- [28] Kolisch R, Sprecher A 1996 *European Journal of the Operational Research* **96** 205-16

Authors	
	<p>Weiming Chen, born on April 21, 1981 at Hanchuan City, China</p> <p>Current position, grades: Lecturer at Faculty of Engineering at China University of Geosciences, Wuhan, China University studies: PhD in Industrial Engineering from Huazhong University of Science and Technology in 2011. Scientific interest: modern design theory and methods, complex project management and product development. Publications: 5 papers</p>
	<p>Xiaoyang Ni, born on January 15, 1970 at Shen Yang City, China</p> <p>Current position, grades: vice-professor at Faculty of Engineering at China University of Geosciences. University studies: PhD in China University of Geosciences (Wuhan) in 2006. Scientific interest: safety engineering and management science Publications: 8 papers</p>
	<p>Hailin Guo, born on September 10, 1973 at Tangshan City, China</p> <p>Current position, grades: Lecturer at Faculty of Engineering at China University of Geosciences, Wuhan, China. University studies: PhD in Geological Engineering from China University of Geosciences in 2005. Scientific interest: risk analysis theory and methods Publications: 19 papers</p>