A special resource constrained project scheduling problem: model update and ant colony optimization solving

Mingwei Li[1], Kun Jiang[2], Sheng Han[1] and Xingye Dong[1,a]

[1]Beijing Key Lab of Traffic Data Analysis and Mining, School of Computer and IT, Beijing Jiaotong University, Beijing 100044, China

[2]Systems Engineering Research Institute, Beijing 100094, China

[a]xydong@bjtu.edu.cn

Keywords: resource constrained project scheduling; meta-heuristic; max-min ant system.

Abstract. A special resource constrained scheduling problem is proposed to satisfy the needs of an enterprise. The constraints of sites and the constraints among different resources are added. With a SSGS-based heuristic algorithm generating an initial solution for the proposed model, the max-min ant system is applied in selecting proper sites for jobs to optimize the solution. The experimental results show that the algorithm can achieve good performance by improving the initial solution significantly, which proves the effectiveness of the algorithm.

Introduction

To achieve the project goal, some tasks need to be executed, and the work of determining the process order of these tasks and the involved resources is called project scheduling. During the last 30 years, RCPSP (Resource Constrained Project Scheduling Problem) has become a standard problem in the field of project scheduling area [1].

Though the model has been extended by a number of researchers [2,3], it still cannot meet the demand of specific practical industrial problem. Based on the fundamental need of an enterprise, a special RCPSP model, SRCPSP, has been proposed in the previous study [4], and a heuristic algorithm based on the SSGS (Serial Scheduling Generated Scheme) is proposed to solve the SRCPSP [4]. In this work, the above SRCPSP model is updated according to more realistic constraints. The main updates are: firstly, some sites and machines may be unavailable in some given time-slots; secondly, instead of being fixed in one position, some machines can move from one site to another, and so they can serve all sites. So the above SRCPSP model is updated in this work by adding the unavailable periods to the sites and renewable resources, and movable renewable resource types are considered. RCPSP has been proved NP-hard in strong sense [5]. Therefore, the use of heuristics and meta-heuristics is a rational choice. So, a heuristic algorithm is designed to satisfy the additional constraints, and then the max-min ant system [6] is applied to optimize the solution. The experimental results show that the proposed method is effective.

The rest of the paper is organized as follows. After the detailed description of the updated SRCPSP model in Section 2, the proposed optimization algorithm using the max-min ant system is presented in Section 3. In Section 4, the computational results and analysis are reported. Finally, the paper is concluded in Section 5.

The Model of SRCPSP

In our previous work [4], the SRCPSP has been modeled as an extension of classic RCPSP, with the introduction of the concepts of job and site, and the objective is to minimize the makespan. In this work, the SRCPSP is extended by adding some new constraints, and the objective is changed to minimize the total flow time and deviation of the project as well. In this section, the model is presented in three parts: the constraints of resources, the constraints of operations, and the objectives.

Resources Constraints. In the scheduling environment of SRCPSP, there are non-renewable and renewable resources, and all the operations need to be processed on a suitable site.

Non-renewable resources, such as raw materials, are usually limited throughout the entire project horizon. There is a set of non-renewable resource types $NR = \{nr_1, nr_2, \ldots, nr_{NR}\}$ in SRCPSP. In this work, the amount of each non-renewable resource type is assumed unlimited.

Sites are the places where the job can be processed. There are W types of sites, and the number of each type is denoted by w_i, $i = 1, 2, \ldots, W$. The site cannot be used sometimes, on account of the occupation or some other reasons. So, for each site s_j, it has a release time r_j and a set of unavailable periods $US_j = \{[t_s,t_e]^*\}$, i.e. the site s_j cannot be used before time r_j or during the unavailable periods. Transferring the following described jobs and movable resources from site j to site k requires some time. As the speeds vary from one job (or movable resource) to another, the transfer time of different jobs may be different. However, the time is in proportion to the distance between the sites and the speed of the job or movable resource. As a result, there is a factor s_{jk} reflecting the distance from site j to site k, and the real transfer time of different jobs or movable resources will be calculated according to the corresponding speed factor. Some sites are in mutex relationship, which means if there is a job being processed on site i, no job can be processed on site j when site i and site j are in mutex relation. At any period, one site can only serve one job, and preemption is not allowed.

There is a set of R renewable resource types $R = \{r_1, r_2, \ldots, r_R\}$, and r_i represents the amount of type i. The renewable resource type i could deal with m_i types of non-renewable resource, denoted by $\{nr_{ij} \mid j = 1, \ldots, m_i\}$. Renewable resource type i has the same processing capacity c_{ij} for the non-renewable resources type j. Usually, renewable resources only can serve some of the nearby sites. However, some machines, depicted as movable resources, can move from one site to another, with an initial site to determine the location and an speed factor to determine the real time of transferring. Similar to the sites, each renewable resource j is under the constraints of the release time rt_j and the unavailable periods URS_j. At any period, one renewable resource can only serve one job, and preemption is not allowed. At any time, the occupation of resource type i should not exceed r_i throughout the scheduling.

Operations Constraints. There is a set of n jobs $N = \{N_1, N_2, \ldots, N_n\}$ in the model, each of which is under the constraints of release time, initial site, and the speed factor. The job N_i consists of a set of k operations $K = \{1, 2, \ldots, k\}$. Operations 1 and k are dummy operations, and they denote the start operation and the sink node. Each of the non-dummy operations can be processed on several given types of sites, requiring some non-renewable resources. The duration of the operation is computed by the needed amount of non-renewable resources and the processing capacity of the renewable resources that serve the job. Besides the precedence constraints, there may be mutual exclusion relationship and mutual inclusion relationship between operations. The mutual exclusion relationship between operation i and j means that they cannot be processed simultaneously even though they satisfy the precedence constraints. Analogously, the mutual inclusion relationship means these two operations must be processed simultaneously. Similarly, preemption is not allowed.

All the operations of a job can be divided into different sets. Each set represents one phase of the process of the job, and phases need to be processed in sequence. Dummy operations are added to each phase to represent the start and the end of the phase. There may also have precedence constraints between the phases of different jobs. For example, if the phase a of job i is the predecessor of the phase b of job j, then the phase b could not begin until the phase a completed.

Objective. The objective of the classic RCPSP is to minimize the makespan. However, the minimization of the makespan cannot ensure every job end the processing as early as possible. So total flowtime is used in this work, i.e., to minimize $\Sigma_{j \in N} C_j$, the sum of the flowtime of every job. In addition, as every job has an expected time to start the process for certain phases, then each job needs to start these phases as close to the given time, so it needs to minimize the absolute deviation of actual start time and the expected start time. The total deviation, computed by $\Sigma_{j \in N} |C_j - E_j|$, ought to be minimized as well.

In summary, the following is updated compared with the model in the previous work [4]: (1)The unavailable periods are added to the sites and renewable resources; (2)Mutex relationship are added between the sites; (3)The movable resources are considered; (4)There are precedence constraints between the phases of different jobs; (5)The scheduling objective is updated.

The Proposed Ant Algorithm

Given the fact that the traditional RCPSP is NP-Complete, and the discussed problem is even more complex, meta-heuristic is considered here. An optimal algorithm based on ant colony optimization is proposed to solve the SRCPSP.

Initial Solution. To generate an initial solution, a constructive heuristic algorithm based on SSGS [7,8] is proposed, in which the following notations and definitions are used:

- ST_i, the candidate operations set of job N_i, consists of all the operations of job N_i that the predecessors have been scheduled already.
- CJ, the candidate job, whose candidate operations set is not empty.
- SJ, the candidate jobs set, consists of all the candidate jobs whose candidate operations set contains one or more operations which can obtain the required sites and resources.
- FJ_i, the front operations set of job i, consists of all the immediate predecessors of the operations in the candidate operations set.

The proposed SSGS algorithm is presented as follows. At stage i, the candidate jobs set SJ_i is selected. If SJ_i is not empty, then select the optimum candidate operation op from the ST_j of SJ_i, and schedule the operation, i.e., assigning the proper renewable resources and sites for the operation, followed by the updating of the ST_j and FJ_j of the job j that the op belongs to. The SJ is updated at the same time. Repeat above steps until the SJ is empty. If none of the jobs has unscheduled operations, a feasible solution is generated. Otherwise, the scheduling cannot be generated. The pseudo-code of the SSGS algorithm can be found in the literature [4].

In the proposed SSGS, it involves the selection of sites, renewable resources and operations when scheduling. The priority rules used are as follows:

(1) If the job has one or more operations that can be scheduled on its current site, then it will be scheduled on the site. Otherwise, it will be moved to the earliest available site where most operations can be scheduled on to continue the process. One thing to note is that unless no other site could be selected, movable resources would not be considered in the selection.

(2) When selecting proper renewable resource for the operation, the resource with the earliest available time and greatest processing capacity will be preferred.

(3) In each scheduling stage, the candidate operation with the earliest start time and end time is selected.

Optimization Algorithm. In order to improve the initial solution, the max-min ant system (MMAS) [6] is adopted. The construction of a solution for the SRCPSP can be divided into two stages: first, choose an appropriate site for the jobs, and then determine the start time for the operations and assign renewable resources accordingly. As the order of the operations in the practical problems is relatively stable and moving a job from one site to another is very time consuming, selecting proper sites is more important than the selection of renewable resources. As a result, the max-min ant system is adopted in the selection of sites in this work.

In the MMAS-based optimization algorithm, each ant generates a solution according to the algorithm base on SSGS. In contrast to the algorithm that generates the initial solution, the site is selected by the job with a probability consists of the phenomenon and the heuristic information, instead of with the priority rules.

There is a pheromone trail $\tau_{ij}^{k}(t)$, denoting the path between site i and site j, as well as the job k, and the iteration counter t. The heuristic information η_{ij}, being set to the value of $1/s_{ij}$, also influences the choice of ants. In the iteration t, suppose job k is on site i, and need to select another site, then the site j is selected with a probability:

$$p_{ij}^{k}(k) = \frac{[\tau_{ij}^{k}(t)]^{\alpha}[\eta_{ij}]^{\beta}}{\sum_{l \in N_i^k}[\tau_{il}^{k}(t)]^{\alpha}[\eta_{il}]^{\beta}}, \quad (1)$$

where α and β regulate the influence of the pheromone trail and the heuristic information, and N_i^k is the set of sites that job *k* can be moved to.

The pheromone trails are updated after all the ants have generated solutions. Then the pheromone trail is updated according to the ant generated the best solution:

$$\tau_{ij}^{k}(t+1) = \rho\tau_{ij}^{k}(t) + \Delta\tau_{ij}^{best}, \quad (2)$$

where ρ ($0 \le \rho < 1$) is parameter that defines the trail persistence, and $\Delta\tau_{ij}^{best} = 1/f(s^{best})$ where $f(s^{best})$ is the objective value of the best solution that has been generated in the iteration *t*.

To avoid the situation that one pheromone trail becomes significantly greater than other pheromone trails after a number of iterations, all the pheromone trails $\tau_{ij}^{k}(t)$ are limited to $\tau_{min} \le \tau_{ij}^{k}(t) \le \tau_{max}$. In each iteration, if $\tau_{ij}^{k}(t) > \tau_{max}$ ($\tau_{ij}^{k}(t) < \tau_{min}$), then set $\tau_{ij}^{k}(t) = \tau_{max}$ (τ_{min}), where τ_{min} and τ_{max} are dynamically changed with $f(s^{opt})$, the value of the best solution that has already found. According to the literature [6], they are calculated with the equations:

$$\tau_{\max} = \frac{1}{1-\rho} \bullet \frac{1}{f(s^{opt})}, \quad (3)$$

$$\tau_{\min} = \frac{\tau_{\max}(1-\sqrt[n]{p_{best}})}{(1-n/2)\sqrt[n]{p_{best}}}, \quad (4)$$

where p_{best} is the probability that the best solution is constructed when MMAS has converged, and *n* is the number of sites. The initial phenomenon trail is set to τ_{max}, with the best solution is set to the initial solution generated by SSGS algorithm.

According to the literature [6], $\alpha = 1$, $\beta = 2$, $\rho = 0.98$, $p_{best} = 0.05$, as these values are proved to achieve better performance more likely.

Based on these rules and equations, the pseudo-code of the MMAS-based algorithm is presented as follows:

1. Generate an initial solution π with the SSGS algorithm, let the optimal solution $\pi^* = \pi$;
2. Set the parameters of the MMAS algorithm: *max_iter*, *ant_num*, α, β, ρ and p_{best};
3. Initial the pheromone trails between sites of each job, being set to τ_{max};
4. For *iter* = 1 to *max_iter* Do
5. For *i* = 1 to *ant_num* Do
6. Generate a solution by the SSGS algorithm that selects the sites using E q. 1;
   EndFor
7. Let π' be the best solution that found in step 6;
8. If (π' is better than π^*)
9. Update $\pi^* \leftarrow \pi$';
   EndIf
10. Update the pheromone trails according to the Eq. 2;
    EndFor
11. Output π^* and stop.

Computational experiments

Test Instances. Based on the real business process, two instances of different scales have been designed to test the MMAS-based algorithm. The brief information about the instances is presented in

Table 1. In each instance, the operations of each job have similar precedence constraints, which can be depicted in the Fig. 1. With two dummy operations in each phase, there are 13, 4, 3, 4 and 3 operations in the phase 1 to 5, respectively. From Phase 2, all of the operations are in linear precedence constraints.

Table 1. The summary of test instances

The environment element	Instance 1	Instance 2
Nonrenewable-resource type	20	20
Site type	7	8
Site	18	26
Mutex relation between site	----	(1,2)
Renewable resource type	25	20
Renewable resource	102	112
Job	4	8
Phase (in per job)	10	20
Operation (in per job)	81	108
The precedence between phases	----	7

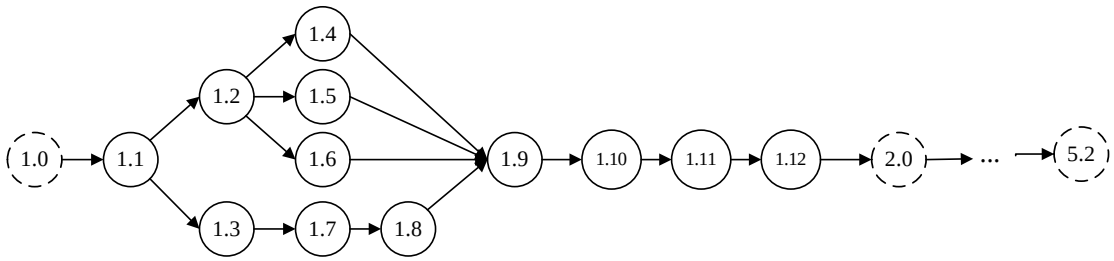

Fig. 1 The AON of operations

The Algorithm Performance. The algorithm is programmed using C++. All the experiments in this paper are carried on a Dell R730 Server with 2.60GHz Intel Xeon processors and 16G RAM running the CentOS7 operating system.

In the case of choosing proper value for *ant_num* and *max_iter*, some preliminary experiments were conducted. The results show that: (1)when the number of ants in each colony is similar to the number of sites, the algorithm are more probably generating better solutions; (2) the algorithm may be converged after 50 to 100 iterations. So, considering the efficiency and effectiveness of the algorithm, *ant_num* is set to the number of sites, and *max_iter* is set to 100 in the following experiments.

As the objective of SRCPSP is to minimize the weighted sum of the total flowtime and total deviation, i.e., $min\ \Sigma(\varphi\Sigma_{j\in N}C_j + \omega\Sigma_{j\in N}|C_j - E_j|)$, the solution may be affected by the weight factors φ and ω. Therefore, different combinations of these two factors are adopted in the experiment. The performance of the MMAS-based algorithm is presented in Table 2. For there are more constraints in the instance 2, the optimizing effect of instance 2 is not as good as that of instance 1. But the result shows that the MMAS-based algorithm can achieve a good performance on both instances and all the combinations of ω and φ.

Table 2. The performance of the MMAS-based algorithm

(ω, φ)	INSTANCE 1			INSTANCE2		
	Initial[s]	Result[s]	RPD[1][%]	Initial[s]	Result[s]	RPD[%]
0,10	11692	8132	30.45	72883	57776	20.73
1,9	14834	10458	29.50	119625	94659	20.87
2,8	17976	12544	30.22	166368	130875	21.33
3,7	21118	14953	29.19	213110	167259	21.52
4,6	24260	16501	31.98	259853	204643	21.25
5,5	27402	18969	30.78	306596	239579	21.86
6,4	30544	21051	31.08	353338	277173	21.56
7.3	33686	22679	32.68	400081	313450	21.65
8,2	36828	25532	30.67	446823	349019	21.89
9,1	39970	28217	29.40	493566	386185	21.76
10,0	43112	29274	32.10	540309	423103	21.69

[1] RPD is the relative percentage deviation, calculated by (*initial* - *result*)/*initial*×100.

Conclusions

In this paper, a special resource-constrained project scheduling problem is presented. Based on the classic RCPSP, some new concepts and constraints are added to the model: introducing the concepts of sites and jobs; besides the constraints of resource and precedence, the operations are also under the constraints of sites; the duration of an operation is calculated by the required amount of non-renewable resource and the capacity of the renewable resource; the operations can be divided into different phases, and precedence constraints exist between phases. Two objectives are considered: the total flowtime and total deviation. A MMAS-based algorithm is proposed to optimize the solution. The experimental results show that the algorithm is effective, and the proper selecting of the site for the jobs plays a key role in the optimizaition of the problem.

Acknowledgements

This work is supported by The Fundamental Research Funds for the Central Universities of China (Project Ref. 2014JBM034, Beijing Jiaotong University).

References

[1] S. Hartmann and D. Briskorn: A survey of variants and extensions of the resource-constrained project scheduling problem. European Journal of Operational Research Vol. 207 (2010), p.1

[2] U. Beşikci, U. Bilge and G. Ulusoy: Multi-mode resource constrained multi-project scheduling and resource portfolio problem. European Journal of Operational Research Vol. 240 (2015), p. 22

[3] J. Coelho and M. Vanhoucke: An approach using SAT solvers for the RCPSP with logical constraints. European Journal of Operational Research Vol. 213 (2015), p. 73

[4] Z. Wei, H. Li and M. Li, et al.: Heuristic research on a resource constrained project scheduling problem. Computer Science and Applications - Proceedings of the 2014 Asia-Pacific Conference on Computer Science and Applications (CSAC 2014), Shanghai, China, 27-28 Dec. 2014, p. 99

[5] J. Blazewicz, J. Lenstra and A. Kan: Scheduling subject to resource constraints: classification and complexity. Discrete Appl. Math. Vol. 5 (1983), p. 11

[6] T. Stützle and H. Hoos: MAX–MIN ant system. Future generation computer systems Vol. 16 (2000), p. 889

[7] R. Kolisch: Serial and parallel resource-constrained project scheduling methods revisited: Theory and computation. European Journal of Operational Research Vol. 90 (1996), p. 320

[8] J. Kelley: The critical-path method: Resources planning and scheduling. Industrial scheduling Vol. 13 (1963), p. 347