Journal of Civil Engineering and Construction 2019;8(1):25-33
https://doi.org/10.32732/jcec.2019.8.1.25

Resource-Constrained Construction Scheduling Using Agent Based Modeling Technique

Ahmed Senouci[1], Karim Abdel Warith[2], Neil Eldin[1]

1. Construction Management Department, University of Houston, Houston, Texas 77204, USA
2. Alghanim International General Trading & Contracting Co. WLL, Doha, Qatar
E-mail:asenouci@uh.edu

Received: 1 August 2018; Accepted: 26 September 2018; Available online: 15 January 2019

Abstract: This paper presents the use of Agent Based Modeling (ABM) technique as a tool for optimum resource constrained scheduling. The model added two features to the standard resource scheduling applications. It allowed activity interruptions when necessary and the impact of the quality of the predecessors on the successors' duration. An illustrative example is offered to demonstrate the performance of the proposed model. ABM technique was confirmed to be a valid approach for seeking alternative solutions in resource constrained schedules. The model proved advantageous to resource-constrained schedules. It illustrated additional flexibility to the standard techniques for resource-constrained problems. The model was proven successful in minimizing the project duration under preset priority rules.
Keywords: Agent based model; Resource-Constrained scheduling; Scheduling models; Simulation.

1. Introduction

Ineffective project planning and scheduling is one of the main causes of project delays and budget overruns [1, 2]. Therefore, improvements to planning and scheduling tools are crucial to the success of the construction industry.

A project schedule is an exercise of mapping the project's logic. It could be viewed as a unidirectional, fully connected graph of the activities defining the project [3]. A construction schedule is a series of tasks sequenced according to functional dependence, zone of work, or similar denominators.

A project schedule is often resource-loaded and cost-loaded to further control the project's progress and expenditures [3,4]. In developing a construction schedule, contractors frequently face resource shortages. For this reason, research efforts have been lunched to determine optimum schedules under resource constraints conditions [5].

This paper explores Agent Based Modeling (ABM) technique in developing resource constrained schedules. It illustrates the use of ABM in what is traditionally referred to as the Resource-Constrained Project Scheduling Problem (RCPSP). The paper explains the use of ABM in creating Activity Agents, and examines the use of these Agents in a project schedule. The scope of this study involved the development of an Agent Based Model capable of predicting the impact of work interruptions due to resource shortages on the project duration, and of the impact of predecessors' quality on the successors' durations. The model was demonstrated using illustrative examples.

2. Background

There are numerous techniques that have been employed to provide solutions for the RCPSPs. Branch and bound methods are a common solution to the problem [6-9]. More sophisticated methods include mathematical programming [4], genetic algorithms [10] and Ant Colony Optimization [3]. Other researchers utilized neural networks [11] and particle swarm optimization [12]. Knotts et al. [13] introduced an agent-based system for project scheduling utilizing priority rules.

The available methods could also be categorized according to their applicability and limitations in handling RCPSPs. Yang et al. [5] categorized these techniques as follows:

1) Basic Single-Mode RCPSP
2) Basic Multi-Mode RCPSP
3) RCPSP problems with Non-regular objective functions
4) Stochastic RCPSP
5) Bin-packing-related RCPSP problems
6) Multi-resource-constrained project scheduling problems (MRCPSP)

The work illustrated in this paper can be categorized as Basic Multi Mode RCPSP, and builds upon the concepts introduced by Knotts et al. [13] and adds a resource constrained dimension.

3. Problem formulation

The objective of the work presented here does not include searching for an optimal solution. However, all solutions that are produced by the model must be viable solutions. In order for a solution to be viable, it must fulfill the following:

3.1 Satisfying relationship constraints

All the standard predecessor-successor relationship are supported by the model. These relationships are listed and formalized as follows:

Finish-to-start (FS)

$$T_n + d_n + L_{nn'} \leq T_{n'} \qquad n' \epsilon \{S_n\} \tag{1}$$

Start to Start (SS)

$$T_n + L_{nn'} \leq T_{n'} \qquad n' \epsilon \{S_n\} \tag{2}$$

Start to Finish (SF)

$$T_n + L_{nn'} - d_{n'} \leq T_{n'} \qquad n' \epsilon \{S_n\} \tag{3}$$

Finish to Finish (FF)

$$T_n + d_n + L_{nn'} - d_{n'} \leq T_{n'} \qquad n' \epsilon \{S_n\} \tag{4}$$

where T_n is the start time of activity *n*; d_n is the activity duration; $L_{nn'}$ is the lag/lead time between activities *n* and *n'*; and $T_{n'}$ is the start time of the succeeding activity *n*' and $\{S_n\}$ denotes the set of activities succeeding activity *n*.

3.2 Recognizing Resource Constraints

The following equation presents the formal inclusion of the possible resource constraints:

$$\sum_{n \in \{S_t\}} r_{kn} \leq R_{kt} \tag{5}$$

where $\{S_t\}$ is the set of activities in progress at time t and R_{kt} is the number of type k resources at time *t*.

3.3 Ensuring Non-negativity Constraint

The lags, durations, and resources are not allowed to have negative values.

4. Problem formulation

4.1 Model Architecture

The model follows a modular architecture and object oriented programming. This allows other researchers to build on this work with minimum effort.

4.2 Agent Overview

An agent is an autonomous object that has the ability of satisfying internal goals [14]. Agents have a complex underlying functional architecture such as the belief-desire-intention (BDI) architecture [15]. The agents used in this model range from relatively simple agents to more complex. The following is a list of common agent types suggested by Sycara et al. [16]:

1) Reactive agents: These agents simply react to a stimulus. Their reaction to the stimulus is predetermined by the developer.

2) Adaptive agents: These agents are more sophisticated. They also react to stimulus, but their reaction may change over time. Generally this reaction is also predetermined, either through a set of rules or functions.

3) Goal oriented agents: These agents are similar to adaptive agents, but do not require a stimulus to act. These agents continuously seek their predetermined goal. The goal seeking method is typically predetermined either through rules or functions. This makes these agents relatively predictable, except if the goal seeking method involves learning.

4) Learning agents: These agents can be reactive, adaptive or goal seeking. However, they are less predictable since their decision changes through their respective "experience". The learning mechanism can be straight forward in which the agents "remembers" a given outcome and uses it in a current decision making. If the learning process is complex, the agent is considered intelligent.

5) Intelligent agents: These agents are highly sophisticated agents. They can be reactive adaptive or goal seeking, as well as, having learning abilities. These agents are placed in their class because their decision making process involves complex algorithms, most of which fall under the umbrella of Artificial Intelligence. The model at hand does not contain this type of agent.

It should be noted that the above list is not conclusive. These agents could be divided further. The list is simply meant to provide the context for the discussion presented below. It should also be noted that despite the type of agent, they are all supported by their own internal BDI nucleus. Figure 1 shows an activity state chart.

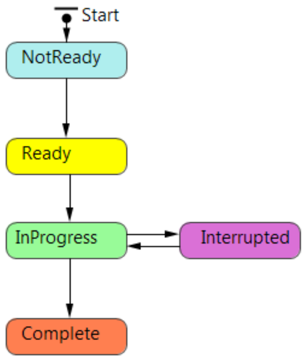


Figure 1. Activity state chart

5. Model components

5.1 Critical path

The model first computes the Critical Path values with no resource constraints. These values include Early Start, Early Finish, Late Start, Late Finish, and Total Float.

5.2 Activity agent

As explained earlier, the model is developed as an agent based model. Thus, agents are the main drivers of the simulation. Activities are not intelligent agents, and thus do not learn. They can be considered goal oriented reactive agents.

The activity agent's goal is to be completed. This is done by performing certain tasks for a given duration then concluding. In some cases the activity may be interrupted. The life of an activity can be translated to a state chart for coding purpose.

All activities start in a "NotReady" state. Each activity then assesses whether its predecessors are complete or not. If they are completed, then the activity becomes "Ready". Once resources for this activity are available, then the activity can start to become in an "InProgress" state. If the activity is interrupted for any reason, it moves to an "Inter" state. Once the activity is finished it is transformed to a "Complete" state.

Activities that are ready compete for the available resources through preset priority rules. Three priority rules were coded into the model, namely, shortest remaining float, earliest early start, and latest late finish. More rules can be added in the future. The model allows the user to choose the priority rule to apply. The activity that is ready and has priority will then check if there are enough resources available for it. If there are enough resources the activity will commence. Otherwise, the activity will remain "Ready".

The model also continues to assess the need for interruption. During the priority check, activities can be interrupted in favor of more 'important' activities. In this case, the state of the activity would switch to "Interrupted".

In addition to states, the activity agent contains a number of parameters such as: Predecessors, Duration, Resources, Interruptability, Quality, Early Start, Early Finish, Late Start, Late Finish, Total Float, Actual Start, and Actual Finish. These dates are calculated through the resource constrained ABM simulation. Quality refers to the dependency on the predecessors' quality. It is an input parameter that reflects whether the duration of an activity is affected by the finish quality of a predecessor.

5.3 Duration object

Duration was modeled as a separate object to allow the activity duration to be manipulated as an aggregate. For instance, the user may decide to apply a certain distribution that would calculate the duration given certain limits. This can be employed easily when durations are treated as a separate object.

5.4. Resource pool

The resource pool contains a number of resources as predefined by the user. Activities book these resources and the balance remains in the resource pool.

5.5 Model flowchart

Figure 2 depicts the model and the step-by-step process.

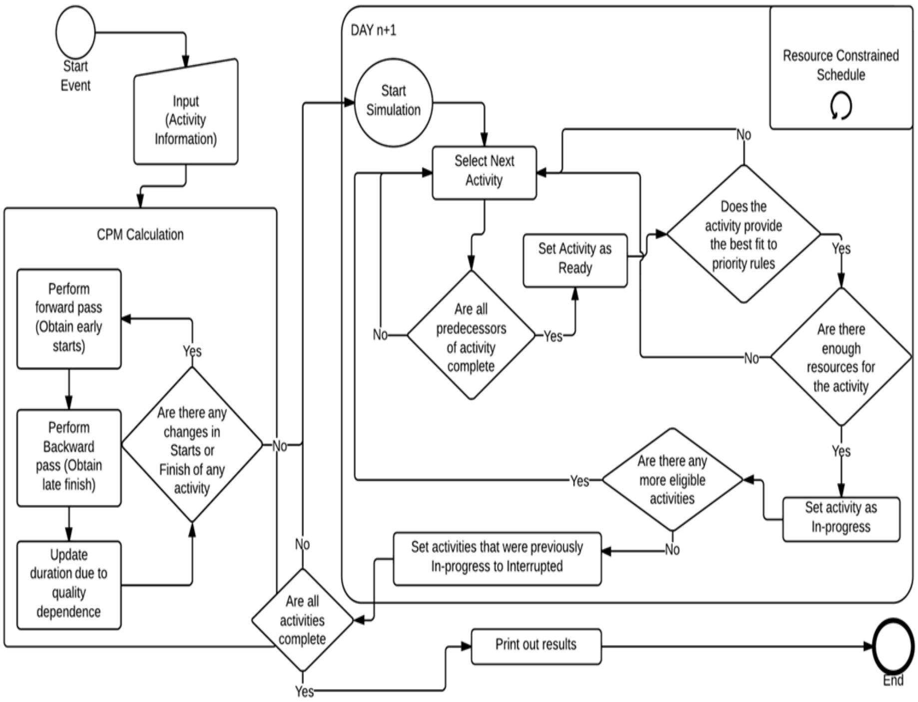


Figure 2. Concept flowchart

6. Illustrative example

Figure 3 is an example project offered by Maroto et al. (1994). This example project was used to illustrate the aspects of the ABM model, and to validate our results. Table 1 summarizes the project inputs. To simplify the tracking of the results, only finish-start relationships were used in the ABM model. However, the user has total freedom on specifying other relationships.

6.1 Priority rules

Three priority rules (i.e., earliest start date, latest finish date, and shortest total float) were implemented. Three different runs were performed, and each run used one of the priority rules (PR) as the BDI of the agents. The results are shown in Table 2.

As shown in Table 2, each priority rule yielded a unique project duration (i.e., 183, 188, and 195). The three priority rules can be used as a preliminary step towards finding an optimum solution. However, the optimization of the solution is beyond the scope of this study.

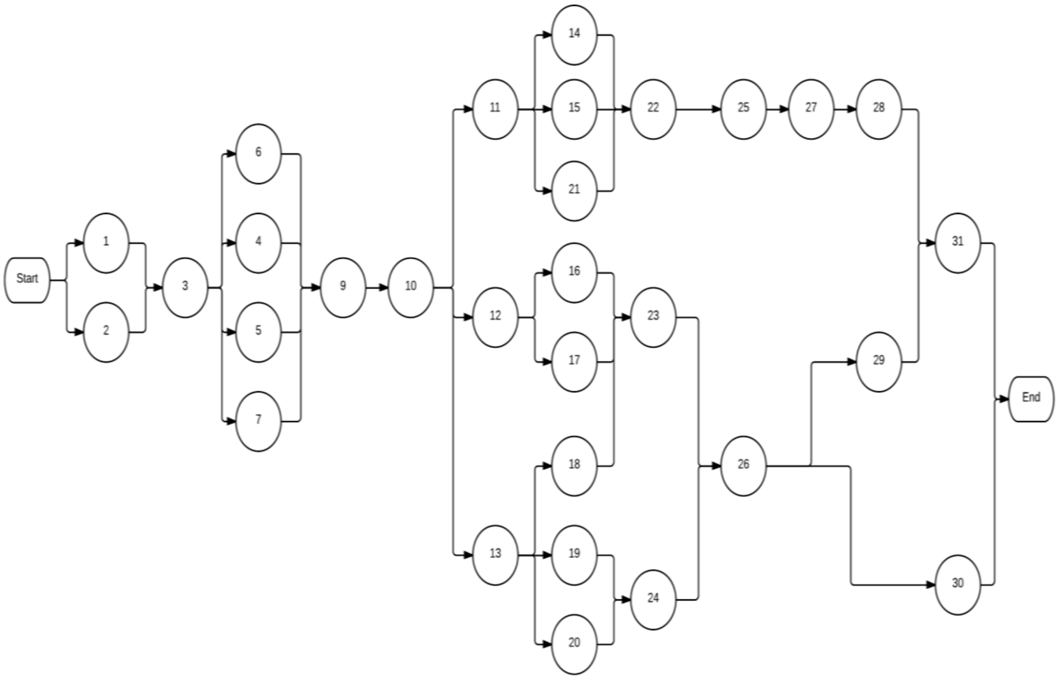


Figure 3. Project activities relationships

Table 1. Project input data

Activity ID	Duration	Resources
1	5	8
2	7	5
3	2	8
4	14	6
5	1	4
6	5	6
7	3	8
8	8	4
9	20	2
10	3	7
11	4	5
12	7	6
13	6	4
14	25	2
15	4	8
16	3	8
17	8	4
18	12	3
19	5	5
20	5	4
21	8	7
22	6	4
23	4	8
24	3	5
25	5	3
26	5	2
27	3	4
28	3	8
29	20	4
30	5	4
31	11	4

Table 2. Results using different priority rules

Activity	CPM Calculations				Resource Constrained Schedule*						Is the Activity Critical
					Late Finish PR		Early Start PR		Total Float PR		
	ES	EF	LS	LF	AS	AF	AS	AF	AS	AF	
0	0	0	0	0	0	0	0	0	0	0	Y
1	0	5	2	7	0	5	0	5	7	12	N
2	0	7	0	7	5	12	5	12	0	7	Y
3	7	9	7	9	12	14	12	14	12	14	Y
4	9	23	9	23	14	28	14	28	14	28	Y
5	9	10	22	23	28	29	28	29	31	32	N
6	9	14	18	23	29	34	29	34	39	44	N
7	9	12	12	15	34	37	34	37	28	31	N
8	12	20	15	23	37	45	37	45	31	39	N
9	23	43	23	43	45	65	45	65	44	64	Y
10	43	46	43	46	65	68	65	68	64	67	Y
11	46	50	47	51	97	101	68	72	73	77	N
12	46	53	49	56	68	75	72	79	87	94	N
13	46	52	46	52	78	84	79	85	67	73	Y
14	50	75	51	76	101	126	97	122	94	119	N
15	50	54	72	76	126	130	122	126	150	154	N
16	53	56	61	64	75	78	134	137	127	130	N
17	53	61	56	64	78	86	79	87	119	127	N
18	52	64	52	64	86	98	85	97	73	85	Y
19	52	57	60	65	89	94	87	92	77	82	N
20	52	57	60	65	84	89	92	97	82	87	N
21	50	58	68	76	130	138	126	134	137	145	N
22	75	81	76	82	138	144	144	150	154	160	N
23	64	68	64	68	155	159	140	144	130	134	Y
24	57	60	65	68	94	97	137	140	134	137	N
25	81	86	82	87	144	149	155	160	160	165	N
26	68	73	68	73	159	164	144	149	145	150	Y
27	86	89	87	90	149	152	160	163	165	168	N
28	89	92	90	93	152	155	169	172	174	177	N
29	73	93	73	93	164	184	149	169	154	174	Y
30	73	78	99	104	164	169	150	155	168	173	N
31	93	104	93	104	184	195	172	183	177	188	Y

*Resource pool set at 8 resources

Figure 4 illustrates the total duration of the project given no resource constraints compared to a resource constrain of 8 resources under the three priority rules.

It is worth noting that the above results are specific to this network and the preset resource constraints. Changes to the network or resource constraints may change the total duration and the ranking of the priority rules.

Figure 5 (a-c) compares the resource profile under each priority rule to the resource profile of the non-constrained schedule.

Table 3 compares the results where all activities were allowed to be interrupted to that when no interruption was allowed.

As can be seen in the Table, two activities benefitted from interruption. Allowing interruption resulted in shortening the project duration by 17 days.

6.2 Impact of predecessor quality

Figure 6 shows the project duration profile resulting from the impact of predecessors' quality. Such impact is manifested as extra time added to the successor's duration. The model allows the user to define the activities having quality issues that impacted the predecessor activity.

It can be observed in Figure 6 that as the percentage of predecessors of lower quality increases the total project duration increases. The relation is expected to be non-linear and unique to a given set of activities.

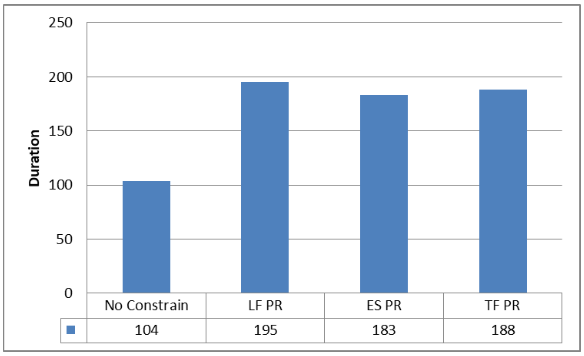
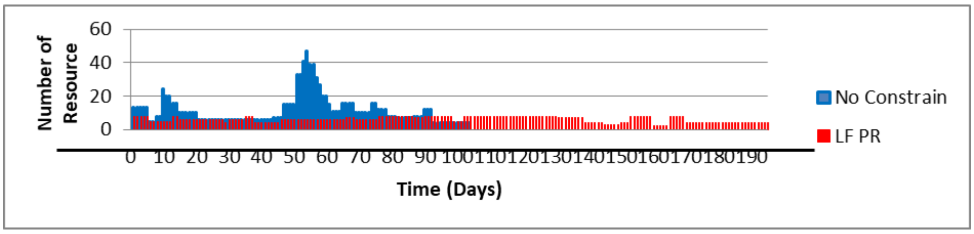
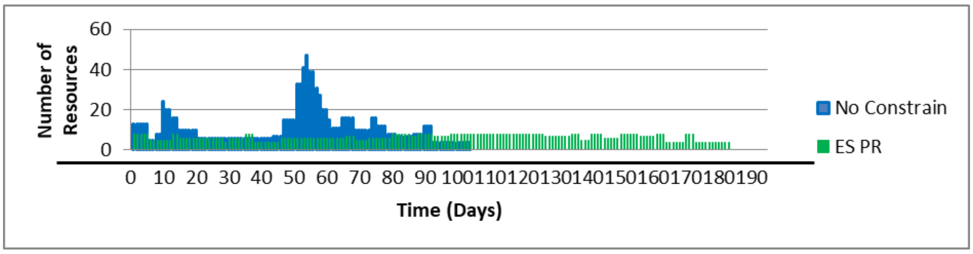


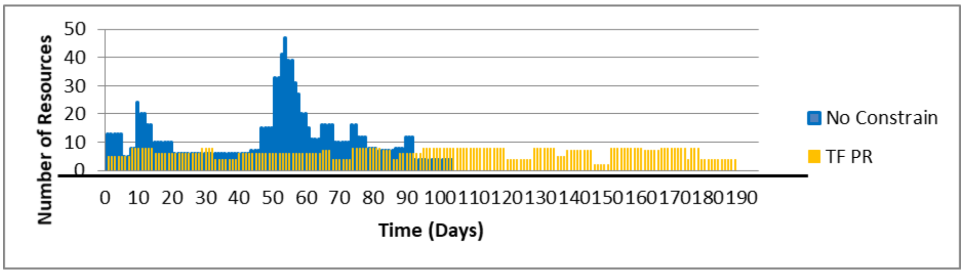
Figure 4. Duration for each priority rule



(a)



(b)



(c)

Figure 5. Resource profiles: (a) late finish, (b) early start, (c) total float

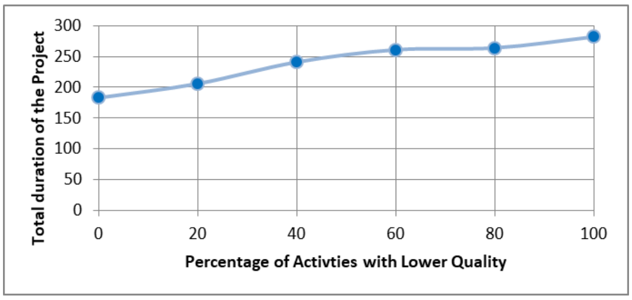


Figure 6. Impact of quality reduction

7. Conclusion

The agent based model presented was capable of predicting the total project duration under three priority rules. The model also made decisions on whether or not to interrupt activities based on the user's input. In addition, the model incorporated the impact of poor quality of the predecessors on the successor's duration.

Agent based modeling proved advantageous to resource-constrained schedules. The model illustrated additional flexibility to the standard techniques for resource-constrained problems. Suggestions for future work include adding the impact of the trades' skill level and complexity of tasks as attributes to the project activities and resources.

Table 3. Interruption results

Activity	Not Interrupted		Interrupted				Interruption	Is the Interrupted
	AS	AF	AS	AF	Day of Interruption	Resumed on day	Duration	Activity Critical?
0	0	0	0	0				
1	0	5	0	5				
2	5	12	5	12				
3	12	14	12	14				
4	14	28	14	28				
5	28	29	28	29				
6	29	34	29	34				
7	34	37	34	37				
8	37	45	37	45				
9	45	65	45	65				
10	65	68	65	68				
11	97	101	97	105	98	102	4 Days	N
12	68	75	68	75				
13	78	84	78	84				
14	101	126	105	130				
15	126	130	130	134				
16	75	78	75	78				
17	78	86	78	91	84	89	5 Days	N
18	86	98	84	98	89	91	2 Days	Y
19	89	94	84	89				
20	84	89	89	94				
21	130	138	134	142				
22	138	144	142	153	144	149	5 Days	N
23	155	159	98	102				
24	94	97	94	97				
25	144	149	153	158				
26	159	164	102	144	105	142	37 Days	Y
27	149	152	158	161				
28	152	155	161	164				
29	164	184	144	167	161	164	3 Days	Y
30	164	169	144	149				
31	184	195	167	178				

8. References

[1] Kerzner HR. Project management: a systems approach to planning, scheduling, and controlling. John Wiley & Sons; 2013

[2] Assaf SA, Al-Hejji S. Causes of delay in large construction projects. International journal of project management. 2006;24(4):349-357.

[3] Christodoulou S. Scheduling resource-constrained projects with ant colony optimization artificial agents. Journal of computing in civil engineering. 2009;24(1):45-55.

[4] Winston WL, Venkataramanan M. Introduction to mathematical programming. 2nd ed. Pacific Grove, California: Thomson-Brooks/Cole; 2002

[5] Yang B, Geunes J, O'brien WJ. Resource-constrained project scheduling: Past work and new directions. Department of Industrial and Systems Engineering, University of Florida, Tech. Rep. 2001.

[6] Brucker P, Knust S. Lower bounds for resource-constrained project scheduling problems. European Journal of Operation Research. 2003;149(2): 302-313.

[7] Crawford J. An approach to resource constrained project scheduling. In: AI and Manufacturing Research Planning Workshop. Albuquerque, N.M.: G. F. Luger, ed., AAAI Press; 1996

[8] Demeulemeester EL, Herroelen WS. New benchmark results for the resource-constrained project scheduling problem. Management Science. 1997;43(11):1485-1492.
[9] Mingozzi A, Maniezzo V, Ricciardelli S, Bianco L. An exact algorithm for the resource-constrained project scheduling problem based on a new mathematical formulation. Management science. 1998;44(5):714-729.
[10] Senouci AB, Eldin NN. Use of genetic algorithms in resource scheduling of construction projects. Journal of Construction Engineering and Management. 2004;130(6):869-877.
[11] Senouci AB, Adeli H. Resource scheduling using neural dynamics model of Adeli and Park. Journal of Construction Engineering and Management. 2001;127(1):28-34.
[12] Zhang H, Li H, Tam CM. Permutation-based particle swarm optimization for resource-constrained project scheduling. Journal of Computing in Civil Engineering. 2006;20(2):141-149.
[13] Knotts G, Dror M, Hartman BC. Agent-based project scheduling. IIE Transactions. 2000;32(5):387-401.
[14] Iglesias CA, Garijo M, González JC. A Survey of Agent-Oriented Methodologies. Intelligent Agents V: Agents Theories, Architectures, and Languages. 5th International Workshop, ATAL'98 Paris, France. July 4-7, 1998.
[15] Rao AS, Georgeff MP. An abstract architecture for rational agents. In: Rich C, Swartout W, Nebel B, editors. Proceedings of Knowledge Representation and Reasoning. San Mateo, CA, USA: Morgan Kaufmann Publishers Inc.; 1992. p. 439–449.
[16] Sycara K, Pannu A, Willamson M, Zeng D, Decker K. Distributed intelligent agents. IEEE expert. 1996;11(6):36-46.