# Project Scheduling Problem Constrained by Flexible Resource with Capability Difference

Xinhui Zhao, Xuezhi Lü, Qiyi Mei, and Weidong Feng

***Abstract*—Along with the resource in resource-constrained project scheduling problem being extended to flexible resource with capability difference, a mix-integer programming model is established for project scheduling problem constrained by flexible resource with capability difference (PSPFRCD). Formulated model is a practical extension of traditional resource-constrained project scheduling problem (RCPSP). To solve the model, it presented particle swarm optimization algorithms based on activity list. In decoding process of particle swarm optimization (PSO), a Dispatching rule based Flexible Resource Assignment Algorithm (DFRAA) is presented, and it discusses Modified Serial Schedule Generation Scheme (MSSGS) and Modified Parallel Schedule Generation Scheme (MPSGS), algorithm framework and selective particle update methods in detail. Comprehensive computational experiments were performed on adapted standard project sets in PSPLIB. Experimental results verify feasibility and effectiveness of algorithm, results also show that PSO algorithm which uses *MSSGS* and *most suitable dispatching rule* has high solution quality.**

***Index Terms*—Flexible resource, capability difference, resource-constrained, project scheduling problem, particle swarm optimization.**

## I. Introduction

21 century is a new economy era of globalization, information network, knowledge capitalization, flat organization and flexible management. In this era, more and more enterprises adopt project management mode, and good project management capability can increase competition capability. Project scheduling is critical to project management, well-know project scheduling tools includes Critical Path Method, Program Evaluation and Review Technique and Resource-Constrained Project Scheduling Problem. Nevertheless, in real world, resource constraining project management has flexibility, such that programmer masters Delphi and .Net, cutting machine has various cutting capability. Moreover, capability of flexible resource is definitely different, how to assign resource in project management considering flexible resource with capability difference is worthwhile in both theory and practice.

Currently, some researchers begin to study the flexible-resource-constrained project scheduling problem. Vairaktarakis developed model of resource-constrained job assignment problem (RCJAP), proposed heuristics algorithm to get lower bound of problem and analyzed value of flexible resource [1]. French scholar Bellenguez addressed multi-skill project scheduling problem (MSPSP), in his model skill level are classified, for example, software development maybe includes junior, intermediate and senior programmer, personnel have some unavailable time, which is to say personnel cannot be assigned to job when they are on vacation [2], [3]. French scholar Kadrou and Najid researched on multi-skilled labor multi-mode resource constrained project scheduling problem (MSL-MRCPSP), the model developed not only considered multi-mode, but also considered constraint of activity center [4]. Wu researched on flexible resource constrained multi-mode project scheduling problem [5]. Yu proposed flexible resource constrained resource leveling project scheduling problem and developed mathematical model, presented a Path Relinking (PR) algorithm which was based on an Adaptive Serial Schedule Generation Scheme (ASSGS) and Maximum network flow based Flexible Resource Assignment Model (MFRAM) [6]. Huang proposed a modified genetic algorithm to deal with flexible resource constrained product development project scheduling problem. The algorithm was constructed on the basis of standard genetic algorithm and maximal flow theory with respect to precedence, skills and flexible resource constrained [7].

From review above, it is obvious that researchers have not considered capability difference. As for some capability, resource was assumed either absolutely be with, or absolutely be without. Moreover, it was assumed that resource assignment does not affect activity duration. However, in reality, more capable resource will shorten duration, while less capable resource will lengthen duration. So, first objective of this paper is extend capability level form set of {0,1} to range of [0,1] and formulate relationship between capability and activity duration, making developed model more practical. Second objective is to present a solving algorithm based on particle swarm optimization. To deal with shortcoming of existing flexible resource assignment model based on maximal flow theory, a flexible resource assignment algorithm based on dispatching rule is presented. The new method not only decrease complexity of programming adopted flexible resource assignment model based on maximal flow theory significantly, but also make resource assignment more reasonably.

The remainder of this paper is organized as follows. In Section II, flexible resource with capability difference is introduced. Section III, model of project scheduling problem constrained by flexible resource with capability difference is formulated. Section IV, a model solving algorithm based on

Manuscript received May 9, 2014; revised August 15, 2014.
Xinhui Zhao is now with Nanjing Artillery Acedemy, Nanjing, Jiangsu, 211100, China (e-mail: enihc@ tom.com).
Xuezhi. Lü, Qiyi Mei, and Weidong Feng are with the Artillery Training Base of General Staff, Xuanhua, Hebei, 075100, China (e-mail: ghostsheep@tom.com, meiqyi@hotmail.com, qdpingzi@tom.com).

PSO is described. Section IV, Comprehensive computational experiments are performed on adapted standard project sets in PSPLIB.

## II. Flexible Resource with Capability Difference

Flexible resources are resources that can take multiple tasks and be assigned to multiple positions and possess multiple skills. It's essential character is multiple capabilities. Capability is skill or function that is needed to take a task, for human resource it means skill, for machine it means function. For example, personnel possess welding skill and painting skill; machine has function of bending and cutting. Resource flexibility refers to property of having multiple capabilities, can be represented by resource capability matrix (RCM). Let $R$ represent resource set, $|R|$ represents quantity of resource. Let $C$ represent capability set, $|C|$ represents quantity of resource capability. Let RCM=$[rc_{rc}]_{|R|\times|C|}$, where $rc_{rc}$ represents level of resource $r$ possesses capability $c$, $rc_{rc}\in[0,1]$, $r\in R$, $c\in C$. If $rc_{rc}$=l, resource $r$ possesses capability $c$ completely. If $rc_{rc}$=0, resource $r$ doesn't possesses capability $c$ completely between 0 and 1. If $rc_{rc}$ is between 0 and 1, it means capability $c$ of resource $r$ is not perfect. For example, level of capability $c$ of resource 1 is 1, level of capability $c$ of resource 2 is 0.8, for same task, resource 1 need 1 hour, and then resource 2 has to spend 1.25 hours on it. Capability level influences project quantity and schedule, for this paper it is assumed that it only affect duration of activity. The relation of capability level and duration can be described by

$$d'_j = d_j \ \text{ceil}\Big(\sum_{r\in R^c}\sum_{c\in C^j} X_{jrc} \Big/ \sum_{r^c}\sum_{c^j} rc_{rc}X_{jrc}\Big) \ \forall j\in J,$$

where, $d'_j$ is actual duration of activity $j$; $d_j$ is standard duration of activity $j$, namely it is activity duration when capability level of all resource is 1; if $X_{jrc}$ is 1, it means resource $r$ undertakes activity $j$ with capability $c$, or else resource $r$ doesn't undertake activity $j$ with capability $c$. ceil is a function that round toward positive infinity, for example, ceil(2.2)=3. Certainly, you can use other equation to describe relation of duration, capability of flexible resource and resource assignment.

## III. Model Formulation

Project Scheduling Problem Constrained by Flexible Resource with Capability Difference (PSPCFRCD) can be formulated as following**:**

1) Structure of whole project can be represented by Activity On Node (AON) $G=(J, Q)$, where $J$ denotes set of activity, $Q$ denotes relation between activities. $|J|$ represents number of activities in activity set. There may be temporal relation between activities, $P_j$ represents predecessor activities set of activity $j$; activity 1 is only earliest starting activity, activity $|J|$ is only latest finishing activity, and they are dummy activity that need no resource and no time, and they can be called *source* and *sink* respectively. Number of non-dummy activity is $|J|$-2. Duration of activity $j$ is $d_j$.
2) $C$ denotes capability set related with project undertaking, $|C|$ represents number of capability in $C$. As for every activity in project, their requirement to capability are different, namely, activity $j$ need a set of capability $C^j$, $C^j\in C$. Relation between activity and capability can be represented by Job Capability Matrix (JCM), JCM=$[jc_{jc}]_{|J|\times|C|}$, where $jc_{jc}$ represents requirement of activity $j$ to capability $c$.
3) Let $R$ denote resource set, $|R|$ represents number of resources in resource set. So, RCM=$[rc_{rc}]_{|R|\times|C|}$, where $rc_{rc}$ denotes level of capability $c$ of resource $r$. If $rc_{rc}$=l, it means resource $r$ possesses capability $c$ completely. If $rc_{rc}$=0, it means resource $r$ possesses no capability $c$. if $rc_{rc}$ is between 0 and 1, it means resource $r$ possesses certain level of capability $c$, but there is gap to totally possession.
4) We further make the following assumptions: $A_1$) No preemption is allowed, i.e., an activity cannot be interrupted once it starts; $A_2$) the set of capability $C_j$ have to be present simultaneously for activity $j$ to progress; $A_3$) A resource can perform at most one capability at any time point, i.e., each resource is treated as a unary resource.
5) The solution of PSPCFRCD includes starting time and finishing time of every activity, as well as resource-capability assignment concept.
6) Decision variables are following: 1) $X_{jrc}$: if $X_{jrc}$ is 1, it means resource $r$ undertakes activity $j$ with capability $c$, or else resource $r$ doesn't undertake activity $j$ with capability $c$. 2) $Y_{jj'}$: if activity $j$ finishes before activity $j'$ start, then $Y_{jj'}$=1, or else $Y_{jj'}$=0; 3) $T_j$: starting time of activity $j$. In above variables, $r\in R, j,j'\in J, c\in C$. When RCM, $G$, JCM, $d_j$ are given, PSPCFRCD can be formulated as hybrid integer programming as following:

$$\min T_{|J|} \tag{1}$$

s.t.

$$\sum_{r\in R^c} X_{jrc} = \text{jc}_{jc} \quad \forall j\in J; c\in C^j, \tag{2}$$

$$X_{jrc} + X_{jrc'} \le 1 \quad \forall j\in J; c,c'\in C^j, c\ne c', r\in R, \tag{3}$$

$$Y_{jj'} = 1 \ \ \forall (j,j')\in J, j\in P_{j'}, \tag{4}$$

$$d'_j = d_j\cdot \text{ceil}\Big(\sum_{r\in R^c}\sum_{c\in C^j} X_{jrc} \Big/ \sum_{r\in R^c}\sum_{c\in C^j} \text{rc}_{rc}\cdot X_{jrc}\Big) \ \forall j\in J, \tag{5}$$

$$T_{j'} \ge T_j + d'_j - \Omega\cdot(1-Y_{jj'}) \quad \forall (j,j')\in J\times J, j\ne j', \tag{6}$$

$$Y_{jj'} + Y_{j'j} \ge X_{jrc} + X_{j'rc'} - 1 \quad \forall (j,j')\in J\times J, j>j'; c\in C^j, c'\in C^{j'}; r\in R, \tag{7}$$

$$T_j \ge 0,\ X_{jrc}\in\{0,1\},\quad Y_{jj'}\in\{0,1\} \tag{8}$$

Objective (1) is to minimize span of project. Constraint set (2) assigns exactly resource quantity to each capability required by an activity. Constraint set (3) prevents a resource from being assigned to capability required by the same activity due to assumption $A2$) and $A3$). Constraint set (4) imposes relation between value of decision variable $Y_{jj'}$ and temporal precedence relation. Constraint set (5) imposes relation between standard duration and practical duration due to capability difference. Constraint set (6) imposes relation between value of decision variable $Y_{jj'}$ and starting time of activity $j$ and $j'$. Constraint set (7) imposes sequence of two activities. Constraint set (8) is (0, 1) integrality constraints.

## IV. Model Solving Algorithm Based on PSO

### A. Algorithm Framework

Algorithm framework is following:

1) Generate initial particle swarm, and assess initial particle swarm to get fitness value of each particle $x_i$, personal best $p_i$ of individual and global best $p_g$ of all;

2) Iteration number plus 1 and update particles according to update equation;

3) Reassess particle swarm to get fitness value of each particle $x_i$, personal best $p_i$ of individual and global best $p_g$ of all;

4) Determine whether algorithm meet the terminating requirement, if it is true, then turn to step (5), else turn to step (2);

5) Transform global best particle $p_g$ into task assignment concept.

### B. Particle Presentation and Initialization

Several types of solution representation for the resource constrained project scheduling problem (RCPSP) have been proposed in the literature and reviewed in the paper of Kolisch [8]. In this paper, we represent a particle with a precedence feasible activity list (An activity list is precedence feasible if any predecessor of an activity appears before this activity in the list.), such that $x_i=(x_{i,1}, \ldots x_{i,j}, \ldots, x_{i,|J|})$ ($i=1,\cdots,N$, $N$ denotes size of particle swarm); position of activity in activity list represents its priority order, i.e. priority order of activity $x_{i,j}$ is $j$; and fitness value of particle is span of project, i.e., $f(x_i)=T_{|J|}$. $x_1$ represents an example of a feasible activity list for a project with 13 activities, $x_1$=(1, 2, 3, 4, 10, 8, 5, 6, 11, 7, 9, 12, 13), AON of example project can be seen from Fig. 1.

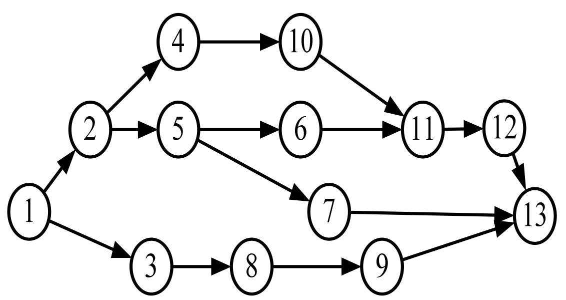

Fig. 1. AON of example project.

### C. Dispatching rule Based Flexible Resource Assignment Algorithm

The key point of flexible resource assignment lies in how to determine resource and capability assignment. Dispatching rule based flexible resource assignment problem can be formulated as follows: when $JCM_j$ (capability requirement vector of activity $j$, $JCM_j=[jc_{j1},...,jc_{j|C|}]$), $d_j$ (standard activity duration), $A$ ($A$ is status vector of resources, that $A=[A_1,\ldots,A_r,\ldots A_{|R|}]$, $A_r\in\{0,1\}$; $A_r=1$ iff $r\in R^A$, here $R^A$ denotes available resource set), RCM (Resource Capability Matrix) are given, $X_j$ (resource assignment Matrix $X_j=[X_{jrc}]_{|R|\times|C|}$, $X_{jrc}\in\{0,1\}$, $r\in R$, $c\in C$), $d'_j$ (real duration) and $Z_j$ (required resource vector $Z_j=[Z_{jr}]_{1\times|R|}$, $Z_{jr}\in\{0,1\}$; $Z_{jr}=\sum_{c\in C} X_{jrc}$) of activity $j$ are got.

Pseudo code of flexible resource assignment algorithm is following:

If $\sum_{c\in C^j} \mathrm{jc}_{jc} > \sum_{r\in R} A_r$
  $X_{jrc}=0(r\in R, c\in C)$; $Z_{jr}=0$ $(r\in R)$;
  For $c\in C^j$
    For $r\in R$
      If $A_r=0 \| Z_{jr}=1 \| r\notin R^c$
        $\mathrm{PRI}_r=inf$
        Continue
      End if
      According to (9) or (10) or (11), caculating PR $I_r$
    End for
    [PR $I'_1,\ldots,$PR $I'_{r'},\ldots,$PR $I'_{|R|}$]=sort([PR $I_1,\ldots,$PR $I_r,\ldots,$PR $I_{|R|}$])
      If PR $I'|jc_{jc}| \neq inf$
        $X_{jrc}=1$, $Z_{jr}=1$, $r\in\{r|\mathrm{PR}\ I_r=\mathrm{PR}\ I'_{r'}, r'\in\{1,\ldots|jc_{jc}|\}\}$
      Else
        NoSolution=1;
      Break;
    End if
  End for
Else
  NoSolution=1;
End if
If NoSolution=1
  $X_{jrc}=inf\,(r\in R, c\in C)$; $Z_{jr}=inf(r\in R)$; $d'_j=inf$
Else
  $$d'_j = d_j \cdot \mathrm{ceil}\Big(\sum_{r\in R^c}\sum_{c\in C^j} X_{jrc} \Big/ \sum_{r\in R^c}\sum_{c\in C^j} \mathrm{rc}_{rc}\cdot X_{jrc}\Big)$$
End if

Three dispatching rule is following:

1) Most suitable dispatching rule：

$$\mathrm{PR}I_r = -\omega\cdot rc_{rc} + \sum_{c\in C}\mathrm{sign}(rc_{rc}) \qquad (9)$$

2) Least suitable dispatching rule：

$$\mathrm{PR}I_r = \omega\cdot rc_{rc} - \sum_{c\in C}\mathrm{sign}(rc_{rc}) \qquad (10)$$

3) Randomly selection dispatching rule：

$$\mathrm{PR}I_r = rand() \qquad (11)$$

where, $\omega$ is proportional coeffecient representing relative importance between resource flexibility and capability level; rand is a function generating ramdom number; $y=\mathrm{sign}(x)$ is a function, if $x > 0$, then $y=1$; if $x<0$, then $y=-1$; if $x=0$, then $Y=0$.

*D. Particle Decoding Method*

A potential solution to the FRCPSP should be transformed to a schedule, so that the status of each particle during the search process can be evaluated in terms of the project duration corresponding to generated schedule. Here, we adopt a modified serial schedule generation scheme (MSSGS) and a modified parallel schedule generation scheme (MPSGS).

Like the serial generation scheme for the RCPSP [9], modified serial schedule generation scheme for the PSPCFRCD also consists of $|J|$ stages. There are two activity sets, scheduled set $PS_i$ ($i$=1,2,…, $|J|$) and decision set $D_n$ ($i$=1,2,…, $|J|$) associated with each stage. The scheduled set $PS_n$ includes the activities which are already scheduled, and the decision set $D_n$ contains the unscheduled activity whose predecessors are already in the scheduled set. $B(t)$ denotes processing activity at time of $t$; $U(t)$ is set of remained flexible resource at time of $t$; $[X_j, Z_j, d'_j]=F(\text{JCM}_j, A, \text{RCM}, d_j)$ is function to get resource assignment based on dispatching rule. The procedure is following:

*Input*: particle $x_i$, JCM and RCM.

*Output*: starting time $T_j$, finishing time $T'_j$ and resource assignment $X_j$ of activity $j$, $j \in J$.

*Initialization*: $PS_1=\{1\}$, $D_1=\Phi$, $n=2$, $T_j=0$，$T'_j=0$.

While $n \le |J|$ do
    Update $D_n$, $D_n=\{j|\, j \notin PS_n, P_j PS_n, j \in J\}$;
    Let $n^{\text{th}}$ activity in activity list of particle be activity $j$: $j=x_{i,n}$;
    Get earliest feasible starting time of activity $j$: $ES_j=\max\{T'_{j'}|j' \in P_j\}$;
    Get starting time of activity $j$: $T_j=\min\{t\ |t \ge ES_j, Z_{jr} \le A_r (r \in R), [X_j, Z_j, d'_j]=F(\text{JCM}_j, A, \text{RCM}, d_j), A_r=1 \text{ iff } r \in U(\tau), \tau=[t, t+d'_j)\}$;
    Get finishing time of activity $j$: $T'_j=T_j+d'_j$;
    Get resource assignment and real duration of activity $j$: $[X_j, Z_j, d'_j]=F(\text{JCM}_j, A, \text{RCM}, d_j)$ ($A_r=1$ iff $r \in U(\tau)$, $\tau=[T_j, T'_j]$);
    Update $PS_n$: $PS_n=PS_n \cup \{j\}$;
    $n=n+1$;
End

Modified parallel schedule generation scheme for the PSPCFRCD consists of $|J|$ stages at most. Every stage corresponds to a scheduling time $t_n$. There are three activity sets, finished activity set $E_n$ ($n$=1,2,…, $|J|$), active activity set $B_n$ and decision set $D_n$ ($n$=1,2,…, $|J|$) associated with each stage. The finished activity set $E_n$ includes the activities which are already finished at $t_n$, i.e., $E_n=\{j|\ T'_j \le t_n\}$. The active activity set $B_n$ contains the processing activity at $t_n$, i.e., $B_n=\{j|\ T_j \le t_n < T'_j\}$. Decision set $D_n$ contains the unscheduled activity whose precedence is feasible at $t_n$, i.e., $D_n=\{j|j \notin E_n \cup B_n, P_j \subset E_n\}$. $[X_j, Z_j, d'_j]=F(\text{JCM}_j, A, \text{RCM}, d_j)$ is function to get resource assignment based on dispatching rule. The procedure is following:

*Input*: particle $x_i$, JCM and RCM.

*Output*: starting time $T_j$, finishing time $T'_j$ and resource assignment $X_j$ of activity $j$, $j \in J$.

*Initialization*: $n=1$, $k=2$, $t_1=0$, $E_1=B_1=\{1\}$, $D_n=\Phi$, $A=R$, $T_j=0$, $T'_j=0$.

While $|E_n \cup B_n| \le |J|$ do
    $n=n+1$;
    $t_n=\min\{T'_j|j \in B_{n-1}\}$; $j=\{j|T'_j=t_n\}$; $E_n=E_n \cup \{j\}$; $B_n=B_n \backslash \{j\}$; $D_n=\{j|\, j \notin E_n, P_j \in E_n, j \in J\}$; $A=A+Z_j$;
    While $D_n \ne \Phi$ do
        Let $k^{\text{th}}$ activity in activity list be activity $j$: $j=x_{i,k}$;
        $[X_j, Z_j, d'_j]=F(\text{JCM}_j, A, \text{RCM}, d_j)$;
        If $Z_{jr} \le A_r$ ($r \in R$)
            $T_j=t_n$; $T'_j=t_n+d'_j$;
            $k=k+1$;
            $B_n \leftarrow B_n \cup \{j\}$, $D_n=D_n \backslash \{j\}$, $A=A-Z_j$;
        Else
            Break;
        End
    End
End

*E. Particle Update Methods*

The particle update equation is following as:

$$x_i = c_2 \otimes G(c_1 \otimes G(w \otimes H(x_i), p_i), p_g) \tag{12}$$

where, $H(x_i)$ denotes particle $x_i$ reflecting on current status, $G(x_i, p_i)$ and $G(x_i, p_g)$ denotes particle $x_i$ inheriting information from other particle . Equation (22) consists of 3 parts:

*1) Particle's reflection on current status, namely*

$$O(x_i, x'_i, m, \kappa) = \min(1, \exp(1 - \frac{f(x'_i) - f(x_i)}{M - \kappa \cdot m}))$$

$$x'_i = H(x_i)$$

$$x_i = w \otimes H(x_i) = \begin{cases} x'_i, \ O(x_i, x'_i, m, w) < rand(); \\ x_i, \ otherwise. \end{cases}$$

where, $x'_i$ represents operated particle; $f(x_i)$ is fitness value of particle $x_i$; $m$ is iteration number; $M$ is iteration limit of PSO; $\kappa$ is coefficient, it may be $w$, $c_1$, $c_2$; $Q(x_i, x'_i, m, \kappa)$ is function to control particle update probability according to particle fitness value of new particle and original particle, iteration number, algorithm control parameter. Purpose of this function is that accept inferior particle at prophase of algorithm to increase diversity, while only accept superior particle at anaphase of algorithm to strength constringency.

*2) Particle inheriting information from personal best particle* $p_i$

$$x'_i = G(x_i, p_i)$$

$$x_i = c_1 \otimes G(x_i, p_i) = \begin{cases} x'_i, \ O(x_i, x'_i, m, c_1) < rand(); \\ x_i, \ otherwise. \end{cases}$$

*3) Particle inheriting information from global best particles* $p_g$

$$x'_i = G(x_i, p_g)$$

$$x_i = c_2 \otimes G(x_i, p_g) = \begin{cases} x'_i, \ O(x_i, x'_i, m, c_2) < rand(); \\ x_i, \ otherwise. \end{cases}$$

The reflection and information inheriting of particle $x_i$ are following.

*4) Particle's reflection by itself*

This paper presents a randomly inserting method based on activity list to realize reflection on itself. The specific methods are following. Let one random integer $r_1$ generate in range of [2, |$J$|-1], suppose last position of predecessors of activity $z_{ir1}$ in particle is $r_2$, and first position of successors of activity $z_{ir1}$ in particle is $r_3$. Next select a position between $r_2$ and $r_1$, or between $r_1$ and $r_3$ randomly, insert to that position.

*5) Particle's information inheriting*

Let $x_1$, $x_2$ denote inheriting particle and inherited particle respectively, the methods is following. A random integer $r$ is generated in range of [1, |$J$|]. Particle $x_1$ keeps before $r$ (includes $r^{th}$ position), and inherits $x_2$, and keep the activity order in $x_2$, that is $x_{1,i}=x_{2,d}$, $k=\min\{k \mid x_2 \notin \{x_{1,1}, x_{1,2},\ldots, x_{1,i-1}\}, k=1,2,\ldots,|J|\}$, $n=r+1,\ldots,|J|$ 。

## V. Comprehensive Computational Experiments

**Experimental Design**. Experimental project problem is adapted project scheduling problem J30, J60, J90, J120 from PSPLIB [10]. Adaptation includes 2 steps. 1) It is assumed that project has 4 kinds of capability. Let resource requirement of original problem to be capability requirement of new problem, i.e., if $c=k$, $jc_{jc}=\alpha_{jk}$, where $\alpha_{jk}$ represents demand of activity $j$ for resource of type of $k$ in original problem. 2) Generate RCM (Resource Capability Matrix) randomly, selects number from {0, 0.6, 0.7, 0.8, 0.9, 1} randomly and assign it to $rc_{rc}$ in RCM in turn. Moreover, if $c=k$ then $rc_{rc}=1$, assuring adapted project problem includes original problem.

For PSPCFRCD is unique, so there is no existing algorithm, say nothing of standard optimum solution to compare algorithms. So, we use an existing algorithm (PEPSO) to get the lower bound of experimental problem as basis to assess performance of three kind of PSO [11]. How to get lower bound of experimental problem? We know that adapted project problem is partial flexible resource constrained project problem, and its solution is not less than solution of completely flexible resource constrained project problem. Therefore, it is feasible to solve completely flexible resource constrained project problem with PEPSO to get lower bound of experimental problem.

Transform J30, J60, J90, J120 of PSPLIB into completely flexible resource constrained project problem includes 2 steps. 1) it is assumed that project need only one kind of resource. as for every activity in project problem, let resource requirement of activity of new problem be sum of resource requirement of original problem. 2) Let available resources of new problem be sum of all kind of resources of original problem.

**Comparison of algorithms**. In experiments, size of particle swarm is 10, iteration limit is 40, $w$=0.75, $c_1$=0.75, $c_2$=0.75, $\omega$=8. Here we consider three dispatching rules, modified serial schedule generation scheme (MSSGS) and modified parallel schedule generation scheme (MPSGS), so there six algorithms. We named them as SMPSO, SLPSO, SRPSO SMPSO, SLPSO and SRPSO respectively (Obviously, S denotes modified serial schedule generation scheme, P denotes modified parallel schedule generation scheme, M denotes most suitable dispatching rule, L denotes least suitable dispatching rule, R denotes randomly selection dispatching rule).

First, get solution of completely flexible resource constrained project scheduling problem with PEPSO as lower bound of experimental problem. Next, solve four problem sets with 6 kinds of PSO. At last, according to different problem sets and algorithms, get average deviation and average computing time corresponding to problem sets and algorithms, we can see results form table I.

From table I, we can find that algorithm adopting Modified Serial Schedule Generation Scheme (MSSGS) are better than algorithm adopting Modified Parallel Schedule Generation Scheme (MPSGS) on average. As for the three dispatching rule, most suitable dispatching rule is best, while least suitable dispatching rule is worst. Therefore, the SMPSO is best in six algorithms. The algorithm deviation not only includes performance difference of algorithm, but also contains difference of solution of two kind of problem set. Since most of resource capability level is less than one, so the optimum solution may be considerably bigger than lower bound, which is the reason why there are considerable average deviation.

TABLE I: Performance of Three Algorithms

| performence | Problem set | SMPSO | SLPSO | SRPSO | PMPSO | PLPSO | PRPSO |
|---|---|---|---|---|---|---|---|
| Average Deviation to lower bound | J30′ | 4.39 | 42.11 | 28.95 | 15.79 | 77.19 | 74.56 |
| | J60′ | 5.81 | 34.20 | 19.05 | 26.84 | 72.73 | 41.99 |
| | J90′ | 16.42 | 53.23 | 43.78 | 80.10 | 136.82 | 120.40 |
| | J120′ | 11.22 | 41.03 | 36.54 | 67.63 | 124.36 | 98.72 |
| Average Computing time /s | J30′ | 20.95 | 21.68 | 20.24 | 18.59 | 18.91 | 17.15 |
| | J60′ | 65.72 | 69.63 | 71.44 | 59.02 | 59.31 | 58.10 |
| | J90′ | 141.60 | 163.12 | 162.04 | 125.68 | 129.08 | 123.90 |
| | J120′ | 263.04 | 298.92 | 322.99 | 249.16 | 241.72 | 313.24 |

## VI. Summary

In this paper, we formulated a mix-integer programming model for project scheduling problem constrained by flexible resource with capability difference (PSPFRCD). Formulated model is a practical extension of traditional resource-constrained project scheduling problem (RCPSP). To solve the model, this paper presented particle swarm optimization algorithms based on activity list. In decoding process of particle swarm optimization (PSO), a Dispatching rule based Flexible Resource Assignment Algorithm (DFRAA) is presented, and it discusses Modified Serial Schedule Generation Scheme (MSSGS) and Modified Parallel Schedule Generation Scheme (MPSGS), algorithm framework and particle update methods in detail. Comprehensive computational experiments were performed on adapted standard project sets in PSPLIB. Experimental results verify feasibility and effectiveness of algorithm, results also show that PSO algorithm which uses MSSGS and most suitable dispatching rule has high solution quality. Our study for the RCMPSP leads to the same conclusion: meta-heuristic strategies are promising approaches to PSPFRCD, and as much problem-specific knowledge as possible should be incorporated into the heuristic. Future research could include the development of further meta-heuristic algorithms for the PSPFRCD and their comparison with the PSO approach presented here.

## REFERENCES


[1] G. L. Valraktarakis, "The value of resource flexibility in the resource-constrained job assignment problem," *Management Science*, vol. 49, pp. 718-732, Jun. 2003.

[2] O. Bellenguez, "Methods to solve multi-skill project scheduling problem," *A Quarterly Journal of Operation Research*, vol. 6, pp. 85-88, Jan. 2008

[3] O. Bellenguez and E. Ne′ron, "Lower bounds for the multi-skill project scheduling problem with hierarchical levels of skills," *Lecture Notes in Computer Science*, vol. 3616, pp. 229-243, Jan. 2005.

[4] Y. Kardrou and N. M. Najid, "A new heuristic to solve RCPSP with multiple execution modes and multi-skilled labor," *Computational Engineering in Systems Applications*, Beijing, 2006, pp. 8-16.

[5] B. Wu, "Research on the flexible resource-constrained multi-mode project scheduling problem," Ph.D. dissertation, Wuhan Technology University, Wuhan, Hubei, Apr. 2008. (in Chinese)

[6] X. G. Yu , D. C. Zhan, L. S. Nie *et al.*, "Flexible resource-constrained resource leveling project scheduling problem," *Computer Integrated Manufacturing Systems*, vol. 16, pp. 1967-1976, Sep. 2010.

[7] M. M. Huang and R. G. Luo, "Study on Flexible-Resource-Constrained Product Development Project Scheduling," *Journal of Industrial Engineering/Engineering Management*, vol. 24, pp. 143-148, Apr. 2010.

[8] R. Kolisch, " Serial and parallel resource-contrained project scheduling methods revisited: theory and computation," *European Journal of Operational Research*, vol. 90, pp. 320-333, Jan. 1996

[9] A. Sprecher, R. Kolisch, and A. Drexl, "Semi-acitive, active, and non-delay schedules for the resource-constrained project scheduling problem," *European Journal of Operational Research*, vol. 80, pp. 94-102, Jan. 1995.

[10] R. Kolisch and A. Sprecher. PSPLIB-A Project Scheduling Problem Library. [Online]. Available: http://www. 129.187.106.231/psplib/indes.html.

[11] L. Y. Deng and Y. Lin, "Particle swarm optimization for resource-constrained project scheduling problems with activity splitting," *Control and Decision*, vol. 19, pp. 681-684, Jun. 2008.


**Xinhui Zhao** was born in Qingyuan, Liaoning China, in 1968. He received his B.E. in Daling University of Technology in 1992, and he received his M.E. degrees in Artillery Command Academy of China, in 2002, and he got Ph.D. degree in Hefei Artillery Academy of China. His research interests include reliability analysis, maintenance decisions, and operation research.

**Xuezhi Lü** was born in Zhangjiakou, Hebei China, in 1979. He received his B.E. and M.E. degrees in artillery command academy of China. He is currently working towards a Ph.D. degree in the ordnance engineering College of China.

**Qiyi Mei** was born in Hubei, China, in 1981. He received his B.E. from National University of Defense Technology, and M.E. from Peking University. His research interests include communication Engineering, control theory and control engineering.

**Weidong Feng** was born in Suozhou, Shanxi, China, in 1976. He received his B.E. and M.E. degrees in artillery command academy of China. His research interests include operation research and management.