

An Approach of Maintenance Task Scheduling Considering Rest Time

Xuezhi Lü^{1, a *}, Xinhui Zhao², Baoxin Fan¹, Xianwen Wang¹ and Yü Gü²

¹Artillery Training Base of General Staff, China

² Artillery Academy of Nanjing, China

^aghostsheep@tom.com

Keywords: maintenance resource; maintenance task; scheduling; PSO

Abstract. As maintenance resources have periodicity, namely, rest must be after certain working time, maintenance task usually is pressing, which must be going on without interruption, how to give out maintenance task scheduling plan and rest time of maintenance resources at same time is a problems deserving discussion. This paper introduces maintenance task scheduling model considering rest time and puts forward a solving algorithm based on PSO. Firstly, it expounds the model assumption, and establishes a hybrid integer-programming model. Secondly, it puts forward a solving algorithm based PSO, and it includes algorithm framework, particle representation, resources-skills allocation algorithm, particle decoding algorithm, update methods and. Thirdly, it provides an example to verify the validity and feasibility of the model and resolving algorithm.

Introduction

Equipment are the material foundation for executing actions of whole organization, and equipment maintenance activities are the important part of operation, and the quality of maintenance activities directly affects success of actions. The problem of maintenance task scheduling is one important decision-making problem that determines maintenance efficiency of equipment support activities, and its main purpose is get the answer of beginning time, required maintenance resource of maintenance tasks. When equipment are repaired, maintenance resources usually are periodic, for example, maintenance personnel(maintenance equipment) has to rest for a period of time after working certain time, but in currently maintenance task research , still no consideration of periodicity for maintenance resources. The maintenance task scheduling concept does not include rest time of maintenance resources. This paper will establish maintenance task scheduling optimizing model considering rest time and design its solving algorithm.

The remainder of this paper is organized as follows. In section 2, model is formulated. Section 4, a model solving algorithm based on PSO is described. Section 5, a example is provided.

Mathematical model

Maintenance task scheduling optimizing model considering rest can be described as following.

(1) Structure of whole maintenance task can be represented by Activity On Node $G=(J, Q)$, where J denotes set of maintenance task, Q denotes relation between maintenance tasks. $|J|$ represents number of maintenance tasks in maintenance task set. There may be temporal constraining between maintenance tasks because of request for technique etc, P_j represents predecessor tasks set of task j ; task 1 is only earliest starting task, task $|J|$ is only latest finishing task, and they are dummy task that need no resource and no time, and they can be called *source* and *sink* respectively. Number of non-dummy task is $|J|-2$. Duration of maintenance task j is d_j .

(2) C denotes capability set related with maintenance task performance, $|C|$ represents number of capability in capability set C . As for every maintenance task in maintenance tasks, their requirement to capability are different, namely, maintenance task j need a set of capability C^j , $C^j \in C$. Relation between maintenance task and capability can be represented by Job capability Matrix M^j , $M^j = [m_{jc}^j]_{|J| \times |C|}$, where m_{jc}^j represents requirement of maintenance task j to capability c .

(3) Let R denote maintenance resource set, $|R|$ represents number of maintenance resources in resource set. In this paper, maintenance resources belong to flexible resources, which have flexibility. The flexible resources means to be applied to more than one different situations of updating resources. Flexible is the attribute that the resources has various skills, which can be represented, and defined $\mathbf{M}^R = [m_{rc}^R]_{|R| \times |C|}$, where m_{rc}^R denotes level of capability c of resource r , $m_{rc}^R \in \{0,1\}$, $r \in R$, $c \in C$. If $m_{rc}^R = 1$, it means resource r possesses capability c completely. If $m_{rc}^R = 0$, it means resource r possesses no capability c . Resources perform maintenance task under the constraining of resources-skill matrix.

(4) Except for flexibility, maintenance resources also have usage periodicity. Personnel must take a rest τ hour after each λ hour. l is index of time periods ($l \in L = \{1, \dots, |L|\}$), $|L|$ is number of time periods within scheduling time range. $|L| = \text{ceil}(T_{|J|}/\lambda)$, $T_{|J|}$ is deadline of whole maintenance task in hours. The $\text{ceil}()$ is a function that is to round toward positive infinity, for example $\text{ceil}(2.2) = 3$.

(5) We further make the following assumptions: A_1) No preemption is allowed, i.e., maintenance task cannot be interrupted once it starts; A_2) the set of capability C_j have to be present simultaneously for maintenance task 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. A_4) rest time of maintenance resources cannot be interrupted. A_5) the duration of maintenance task cannot be longer than the longest time of each time period.

(6) The solution of problem includes starting time and finishing time of maintenance task and rest, as well as resource-capability assignment project.

(7) Decision variables are following: 1) X_{jrc} : if maintenance personnel r undertakes maintenance task j with capability c , then X_{jrc} is 1, or else X_{jrc} is 0. 2) $Y_{jj'}$: if task j finishes before task j' start, then $Y_{jj'}$ is 1, else $Y_{jj'}$ is 0; 3) T_j : starting time of task j . 4) T_{rl} : starting time of personnel r in rest period l . 5) Y_{rl}^j : if maintenance task j precedes rest time of personnel r in time period l , then Y_{rl}^j is 1, else Y_{rl}^j is 0. 6) Y_j^{rl} : if rest time of personnel r precedes maintenance task j in time period l , then Y_j^{rl} is 1, else Y_j^{rl} is 0. In above variables, $r \in R$, $j, j' \in J$, $c \in C$. When \mathbf{M}^R , G , \mathbf{M}^J , d_j , λ , τ are given, maintenance task scheduling optimizing model considering rest can be formulated as hybrid integer programming as following:

$$\min T_{|J|}; \quad (1)$$

$$\sum_{r \in R^c} X_{jrc} = m_{jc}^J, \quad \forall j \in J, c \in C^j; \quad (2)$$

$$X_{jrc} + X_{jrc'} \leq 1, \quad \forall j \in J, c, c' \in C^j, c \neq c', r \in R; \quad (3)$$

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

$$T_{rl} \geq (l-1) \cdot \lambda, \quad \forall r \in R, l \in L; \quad (5)$$

$$T_{rl} + \tau \leq l \cdot \lambda, \quad \forall r \in R, l \in L; \quad (6)$$

$$T_{j'} \geq T_j + d_j - \Delta \cdot (1 - Y_{jj'}), \quad \forall (j, j') \in J \times J, j \neq j'; \quad (7)$$

$$T_{rl} \geq T_j + d_j - \Delta(1 - Y_{rl}^j), \quad \forall j \in J, r \in R, l \in L; \quad (8)$$

$$T_j \geq T_{rl} + d_j - \Delta(1 - Y_j^{rl}), \quad \forall j \in J, r \in R, l \in L; \quad (9)$$

$$Y_{jj'} + Y_{jj} \geq X_{jrc} + X_{j'rc'} - 1, \quad \forall (j, j') \in J \times J, j \neq j', c \in C^j, c' \in C^{j'}, r \in R; \quad (10)$$

$$Y_{rl}^j + Y_j^{rl} \geq X_{jrc}, \quad \forall j \in J, c \in C^j, r \in R, l \in L; \quad (11)$$

$$d_j \leq \lambda - \tau, \quad \forall j \in J; \quad (12)$$

$$T_j, T_{rl} \geq 0; \quad (13)$$

$$X_{jrc}, Y_{jj'}, Y_{rl}^j, Y_j^{rl} \in \{0,1\}; \quad (14)$$

Objective (1) is to minimize span of maintenance task. Constraint set (2) assigns exactly resource quantity to maintenance task according to each capability requirement of maintenance task. According to assumption A_2, A_3 , it prevents maintenance personnel using different capability to undertake the same maintenance task. Constraint set (3) limits each maintenance personnel at most use one kind of capability in each maintenance task. Constraint set (4) denotes that if maintenance task j precedes maintenance task j' , then maintenance task j is finished before starting of task j' . Constraint set (5) and (6) limits each rest time of resource r in period l , according to assumption A_4 , rest time of maintenance resource cannot be interrupted. Constraint set (7) defines $Y_{jj'}$, Δ is a infinite number, which denotes if $Y_{jj'}$ is 1, namely j finishing before j' , then j' starting time at least later d_j than j starting time, if $Y_{jj'}$ constantly is 0, according to assumption of A_3 . Constraint set (8) and (9) differently define Y_{rl}^j, Y_j^{rl} respectively. Constraint set (10) and (11) restrict sequence of maintenance task and rest time. Constraint set (12), according to assumption A_5 , limits duration of maintenance task cannot exceed longest duration of each time segment. Constraint set (13) and (14) are constraint for variable.

Model solving algorithm based on PSO

Intelligence Algorithm is effective method for solving NP-hard problem. Particle swarm optimization has already been applied to solving of classic RCPSP, and obtains good effect^[1-2]. So, the next part will put forward one kind of solving algorithm based on particle swarm optimization.

Algorithm framework

Particle swarm optimization framework is following:

- (1) Generate initial particle swarm. Assess particle of initial particle swarm, and get fitness value of each particle x_i , personal best p_i of individual and global best pg 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 pg 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) Generate project (maintenance task) scheduling plan (including starting time of maintenance task, rest starting time of maintenance personnel in each time period, resource-capability assignment concept) with known global best solution, and the algorithm ends.

Particle presentation and initialization

Algorithm adopts particle presentation method based on task sequence. The task sequence presented by particle corresponds to a precedence feasible task sequence. The task position in the sequence presents task scheduling priority order. Particle dimension is the total amount of task, namely $D=|J|$, and particle fitness value corresponds to maintenance task duration. such that $\underline{x}_j=(x_{i,1}, \dots, x_{i,j}, \dots, x_{i,|J|})(i=1, \dots, N, N \text{ denotes size of particle swarm})$; position of task in task sequence represents its priority order, i.e. priority order of task $x_{i,j}$ is j ; and fitness value of particle is span of project, i.e., $f(\underline{x}_j)=T_{|J|}$. Figure 1 is AON of scheduling project example, task 1 being virtual starting task, task 13 being virtual finishing task, one feasible particle of project is $\underline{x}_1=(1, 2, 3, 4, 10, 8, 5, 6, 11, 7, 9, 12, 13)$.

Priority rule based flexible resource-capability assignment algorithm.

Flexible resource-capability assignment problem can be formulated as follows: when \underline{M}_j^j (capability requirement vector of task j , $\underline{M}_j^j=[m_{j1}^j, \dots, m_{j|C|}^j]$), d_j (task duration), \underline{A} (status vector of flexible resources, $\underline{A}=[A_1, \dots, A_r, \dots, A_{|R|}]$, $A_r \in \{0,1\}$; $A_r=1$ if $r \in R^A$, here R^A denotes available resource set), \underline{M}^R (Resource-Capability Matrix) are given, X_j (resource-capability assignment matrix

$X_j=[X_{jrc}]_{|R| \times |C|}$, $X_{jrc} \in \{0,1\}$, $r \in R$, $c \in C$ and \underline{Z}_j (required resource vector $\underline{Z}_j=[Z_{jr}]_{1 \times |R|}$, $Z_{jr} \in \{0,1\}$; $Z_{jr} = \sum_{c \in C} X_{jrc}$) of task j are got.

(1) If $\sum_{c \in C} m_{jc}^J > \sum_{r \in R} A_r$, turn to step (9), else turn step (2).

(2) Let $c=1$, resource assignment project $X_{jrc}=0$, $Z_{jr}=0$ ($\forall r \in R$, $c \in C$).

(3) If $c > |C|$, then turn to step (10), else turn to step (4).

(4) If $m_{jc}^J=0$, then $c=c+1$ turn to step (3), else turn to step (5).

(5) Let $r=1$.

(6) If $r > |R|$, then turn to step (8), else turn to step (7).

(7) If $A_r=0 \cup \sum_{c \in C} X_{jrc}=1$, PRI $\rho_r = \inf$, else, according to priority rule (base on one of formulas (15)

~(17)) calculate ρ_r , $r=r+1$ turn to step (6).

(8) For PRI sequence $\{\rho_1, \dots, \rho_r, \dots, \rho_{|R|}\}$, sort by order from small to big, if the $m_{jc}^{J \text{th}}$ in PRI sequence isn't \inf (infinite) after sorting, then take resource subset that includes first m_{jc}^J resources in PRI as R' , set $X_{jrc}=1, Z_{jr}=1$, ($\forall r' \in R'$), $c=c+1$, turn to step (3) else turn to (9).

(9) Algorithm ends, cannot get feasible resource assignment concept $X_{jrc} = \inf$, $Z_{jr} = \inf$ ($\forall r \in R$, $c \in C$), over.

(10) Output resource-capability assignment project X_j and required resource vector Z_j , over.

In step (7), three priority rules are following:

(1) Most suitable priority rule.

$$\rho_r = -\sum_{c \in C} m_{rc}^R. \quad (1)$$

(2) Least suitable priority rule.

$$\rho_r = \sum_{c \in C} m_{rc}^R. \quad (2)$$

(3) The random selection priority rule.

$$\rho_r = \text{rand}(), \quad (3)$$

In the formula, $\text{rand}()$ is function generating random number between 0 and 1.

Particle decoding method

In this algorithm, a particle represents a feasible solution. The algorithm adopts modified parallel schedule generation scheme to converts particle into feasible concept. Next, it introduces the modified parallel schedule generation scheme, which includes flexible resource-capability assignment algorithm. It can also arrange rest time of maintenance resource while provide maintenance task scheduling concept [3].

The modified parallel schedule generation scheme at best consists of $|J|$ stages, stage n corresponds to a scheduled time t_n , a finished task set E_n , a active task set B_n and a decision task set D_n . Finished task set E_n denotes set that consists of finished task until t_n , namely $E_n = \{j | T'_j \leq t_n\}$. Active task set B_n denotes set that consists of ongoing task at time of t_n , namely $B_n = \{j | T_j \leq t_n < T'_j\}$. Decision task set D_n denotes set that consists of the unsorted task which satisfies temporal sequence constraint at time of t_n , namely $D_n = \{j | j \notin E_n \cup B_n, P_j \subset E_n\}$. Let flexible resources-capability assignment algorithm be represented as function: $[X_j, \underline{Z}_j] = F(\underline{M}_j, \underline{A}, \underline{M}^R)$.

The algorithm can be simply described as follows.

Input: particle \underline{x}_j , task-capability matrix \underline{M}^J and resource-capability matrix \underline{M}^R , λ , τ .

Output: Scheduling project including task starting time T_j , finishing time T'_j and resource concept X_j ($\forall j \in J$), rest starting time of maintenance resource r in time period l , T_{rl} ($\forall r \in R$, $\forall l \in L$).

(1) *Initialization:* $n=1$, $k=1$, $t_1=0$, $E_n=B_n=\{1\}$, $D_n=\Phi$, \underline{A} (unit vector whose element number is $|R|$), $T_j=0$ ($\forall j \in J$), $T'_j=0$ ($\forall j \in J$), $T_{rl}=0$ ($\forall j \in J, \forall l \in L$), $T'_{rl}=0$ ($\forall j \in J, \forall l \in L$), $l=1$.

(2) If $k > |J|$, then algorithm is over, else continue.

(3) $n=n+1$, determine next stage scheduling time $t_n = \min\{\min\{T'_j | j \in B_{n-1}\}, l \lambda\}$; finished task $j = \{j | T'_j = t_n\}$, update $E_n = E_{n-1} \cup \{j\}$, $B_n = B_{n-1} \setminus \{j\}$, $D_n = \{j | j \notin E_n, P_j \subset E_n, \forall j \in J\}$, status vector of resources $\underline{A} = \underline{A} + \underline{Z}_j$, free starting time of resource r , $T_r^i = t_n$ ($r \in \{r | Z_{jr}=1\}$). Arranged rest resource

$r'=\{r|t_n-\max(T_r^i,(l-1)\lambda)\geq\tau\cap T_{r'l}=0\cap T'_{r'l}=0\}$ (T_r^i is free starting time of resource r), rest starting time of resource r' in time period l is $T_{r'l}=\max(T_r^i,(l-1)\lambda)$, rest finishing time $T'_{r'l}=T_{r'l}+\tau$. If $t_n=l\lambda$, then for time period l , then arranges rest of resource $r=\{r|(T_{r'l}=0\cap T'_{r'l}=0)$ which cannot take task, rest starting time $T_{r'l}=l\lambda-\tau$, rest finishing time $T'_{r'l}=l\lambda$, $l=l+1$.

(4) From Particle representing task sequence, select k^{th} scheduled task j , namely $j=x_{i,k}$. Check the resource which can be assigned task, update \underline{A}' (status vector of flexible resources $\underline{A}'=[A'_1,\dots,A'_r,\dots,A'_{|R|}]$, $A'_r\in\{0,1\}$; $A'_r=1$ iff $r\in\{r|(t_n+d_j+\tau\leq l\lambda\cap T_{r'l}=0\cap T'_{r'l}=0)\cup(T_{r'l}\neq 0\cap T'_{r'l}\neq 0)$, $r\in R\}$). Use resource-capability assignment algorithm $[X_j, \underline{Z}_j]=F(\underline{M}^j_j, \underline{A}', \underline{M}^R)$ to get resource assignment concept. If $X_{jrc}=\inf$ and $Z_{jrc}=\inf(\forall r\in R, c\in C)$, turn to step(2). Else determine starting time T_j and finishing time T'_j of task j , $T_j=t_n$, $T'_j=t_n+d_j$. Update $B_n\leftarrow B_n\cup\{j\}$, $D_n=D_n\setminus\{j\}$, $\underline{A}=\underline{A}-\underline{Z}_j$, $k=k+1$. Repeat this step.

We can see from above algorithm that rest time of maintenance resources are assigned before current time, when its free time length be no less than τ , it is assigned, i.e. whenever there is a chance, then rest time is assigned. While arranging maintenance resources to undertake maintenance task, it does not consider maintenance resources which may not rest in this period if it undertake task. When each period ends, assign rest to those resources that have not rested. This algorithm takes maintenance task as center, arranges rest time at idle time of maintenance resource as possible, and rest time must be ensured.

Particle Update Methods

The particle update equation of algorithm is following as:

$$\underline{x}_i = c'_2 \otimes g(c'_1 \otimes g(w \otimes h(\underline{x}_i), \underline{p}_i), \underline{p}_g), \quad (4)$$

where $h(\underline{x}_i)$ denotes to undertake reflecting operation for particle \underline{x}_i . $g(\underline{x}_i, \underline{p}_i)$ and $g(\underline{x}_i, \underline{p}_g)$ both denotes particle \underline{x}_i inheriting information from other particle. Equation (18) consists of 3 parts:

(1) Particle's reflection by itself, namely

$$\underline{x}_i = w \otimes h(\underline{x}_i) = \begin{cases} h(\underline{x}_i), & \text{rand}() < w; \\ \underline{x}_i, & \text{rand}() \geq w. \end{cases} \quad (5)$$

(2) Particle inheriting information from p_i

$$\underline{x}_i = c'_1 \otimes g(\underline{x}_i, \underline{p}_i) = \begin{cases} g(\underline{x}_i, \underline{p}_i), & \text{rand}() < c'_1; \\ \underline{x}_i, & \text{rand}() \geq c'_1. \end{cases} \quad (6)$$

(3) Particle inheriting information from p_g

$$\underline{x}_i = c'_2 \otimes g(\underline{x}_i, \underline{p}_g) = \begin{cases} g(\underline{x}_i, \underline{p}_g), & \text{rand}() < c'_2; \\ \underline{x}_i, & \text{rand}() \geq c'_2. \end{cases} \quad (7)$$

In equation (19) ~ (21): c'_1 、 c'_2 、 w denotes control parameters of algorithm. Particle's reflection by itself and particle inheriting information can adopt mutation or crossover to accomplish respectively:

(1) 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 $a1$ generate in range of $[2, |J|-1]$, suppose last position of predecessors of activity \underline{x}_{ia1} in particle is $a2$, and first position of successors of activity \underline{x}_{ia1} in particle is $a3$. Next select a position between $a2$ and $a1$, or between $a1$ and $a3$ randomly, insert to that position.

(2) Particle's information inheriting. Here adopting updating method of particle's reflection by itself is the same as PEPESO algorithm^[4]. Let $\underline{x}_1, \underline{x}_2$ denote inheriting particle and inherited particle respectively, through cross operation generates a new particles \underline{x}' , the methods is following. A random integer a is generated from 2 to $|J|-1$. The first a dimension of \underline{x}' inherits \underline{x}_1 , namely $x'_j=x_{1j}$, $j=1,2,\dots,a$. But from the $a+1$ to $|J|-a$ dimension are came from \underline{x}_2 , the existed task among \underline{x}' are no longer considered, and keep relatively position of each task, namely $x'_j=x_{2k}$, $k=\min\{k|x_{2k}\notin\{x'_1, x'_2, \dots, x'_{j-1}\}\}$, $k=1,2,\dots,|J|$, $j=a+1,\dots,|J|$. Fig.2 shows progress of particle inheriting information.

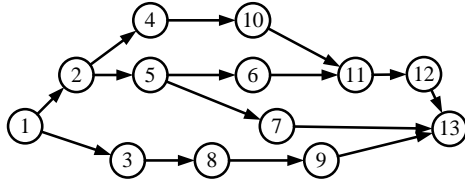


Fig. 1. Example of project scheduling

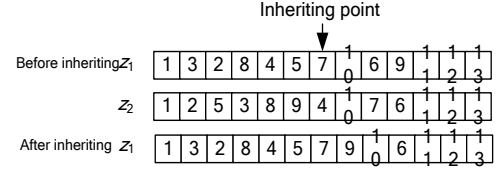


Fig.2. Information inheriting of particle

Calculating example

Next, we will consider a example of maintenance task scheduling. This maintenance task consists of 13 maintenance tasks, is completed by 12 maintenance personnel, maintenance resources rest at least 8 hours working per 24 hours. Table 1 shows the resource-capability matrix of maintenance personnel. Table II shows maintenance task information that includes duration in hours, resource requirement, and temporal relation, where the first task and the last task are virtual starting and finishing task respectively. It uses particle swarm optimization put forward in the text to get solution. Least suitable priority rule is used in resource-capability assignment algorithm. Size of particle swarm is 20, iteration limit is 80, $w=0.75$, $c'_1=0.25$, $c'_2=0.75$.

TABLE I: RESOURCE-CAPABILITY MATRIX

capability	resource											
	r_1	r_2	r_3	r_4	r_5	r_6	r_7	r_8	r_9	r_{10}	r_{11}	r_{12}
c_1	0	0	1	1	1	0	0	1	0	1	1	0
c_2	0	1	1	0	0	1	0	0	0	1	0	0
c_3	0	0	1	1	0	0	1	1	1	0	0	0
c_4	0	0	1	1	1	1	1	0	0	1	1	0

TABLE II : MAINTENANCE TASKS INFORMATION

Task number	j_1	j_2	j_3	j_4	j_5	j_6	j_7	j_8	j_9	j_{10}	j_{11}	j_{12}	j_{13}
Task duration	0	4	6	5	9	11	9	4	6	10	5	10	0
successors	2,3	4,5	7	10	6	11	8,9	13	13	11	12	13	
c_1	0	2	1	0	0	2	0	1	1	2	0	1	0
c_2	0	0	0	1	2	0	0	1	0	1	0	0	0
c_3	0	1	2	1	2	2	1	2	2	0	1	2	0
c_4	0	0	1	0	2	2	1	1	0	0	0	1	0

After calculation, get the optimized maintenance task sequence (1, 3, 4, 2, 5, 6, 8, 7, 9, 10, 11, 12). Gantt chart of maintenance task scheduling project is shown in figure 3, horizontal coordinate denotes time, and vertical coordinate denotes personnel. Blue rectangle in chart denotes maintenance task, the vertical coordinate of rectangle position corresponds to assigned personnel, three numbers in rectangle divided by “-” denote maintenance task number, personnel number, capability number respectively. Green rectangle in chart denotes rest time, four numbers in rectangle divided by “-” denote maintenance time segment number, personnel number, rest starting time, rest finishing time respectively.

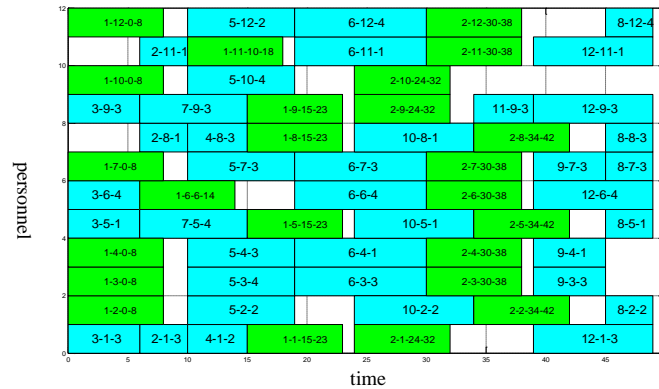


Fig.3 Gantt chart of optimized maintenance task scheduling concept

Summary

This paper establishes a kind of maintenance task scheduling model considering rest time, puts forward its solving algorithm, and provides calculating instance to verify the validity and feasibility

of the model and resolving algorithm. This model can help decision maker to provide maintenance task scheduling concept and maintenance resource rest concept. For the future, we can further research solving algorithm of model, design a decoding algorithm based on serial schedule generation scheme to enhance solving efficiency of model. In addition, we can also research maintenance task scheduling model considering splitting rest time, and restriction of the longest work time, personnel's preference and rest place restriction.

References

- [1] H. Zhang, X. Li, H. Li, Particle swarm optimization –based schemes for resource-constrained project scheduling, *Automation in Construction*, 14(2005) 393-404.
- [2] B. Jarboui, N. Damak, P. Siarry, A combinatorial particle swarm Optimization for solving multi-mode resource -constrained project scheduling problems, *Applied Mathematics and Computation*, 195(2008)299-308.
- [3] A. Sprecher, R. Kolisch, A. Drexl, Semi-active, active, and non-delay schedules for the resource-constrained project scheduling problem , *European Journal of Operational Research* , 80(1995), 94-102.
- [4] L. Y. Deng, Y. Lin , Particle swarm optimization for resource-constrained project scheduling problems with activity splitting, *Control and Decision*, 19(2008) 681-684. (in Chinese).
- [5] R. Kolisch, Serial and parallel resource-constrained project scheduling methods revisited: Theory and computation , *European Journal of Operational Research*, 90 (1996) 320-333, Jan.