

航空多项目资源约束下计划优化的改进型粒子群算法*

Improved Particle Swarm Algorithm for Aviation Multi-Project Resource-Constrained Scheduling Optimization

西北工业大学现代设计与集成制造技术教育部重点实验室 王凯 李原 张杰

[摘要] 建立了并行多项目在资源约束条件下计划优化问题的数学模型,并提出一种改进型混合粒子群算法对多项目实施并行调度来解决网络计划的优化问题。最后,对某型飞机的2个并行装配计划进行了实例计算,验证了方法的有效性。

关键词: 多项目 资源约束 粒子群算法 项目调度

[ABSTRACT] A mathematical model for aviation concurrent multi-project resource-constrained scheduling optimization is built and an improved particle swarm algorithm is put forward to carry out concurrent scheduling for multi-project and to solve the net optimization problem. Finally, the simulation calculation is carried out for concurrent assembly plan of a certain aircraft, which validates this method.

Keywords: Multi-project Resource-constrained Particle swarm algorithm Project scheduling

多项目资源约束下计划优化问题指的是一类在满足各并行项目资源约束和任务前后约束的条件下合理安排各项目任务的开始时间以最小化项目总工期的调度问题。航空产品项目工序数量庞大,任务并行度比较高,传统的优化方法,如分枝定界法^[1-2]和启发式算法^[3]随着网络规模的增加,优化效果并不理想;文献[1][2]中的分枝定界法是一种精确类算法,可以精确地求解小规模项目的资源约束项目调度问题,但计算量会随着网络计划中节点数目的增多而呈指数曲线上升,会出现所谓的组合爆炸现象;文献[3]中的启发式算法计算简单、可操作性强,发展得较为成熟。但不同的启发式方法处理同一问题的效果有很大的差别^[4],很难找出一种合适的方法来处理不同航空项目

资源计划的优化问题。同时,随着网络计划任务数目的增加,约束关系的复杂化,启发式方法处理问题的效果有时会发生明显的变化,计算精度逐渐降低^[5]。遗传算法^[6]适合于求解大型复杂网络的调度问题,但是由于简单遗传算法的适应度值决定了个体被复制的概率,并且算法对于系统中的反馈信息利用不够,因此在进化的早期容易使个别个体的后代充斥整个种群,产生大量冗余迭代和早熟收敛现象,效率不高。

针对以上问题,本课题采用一种改进型混合粒子群算法来求解多项目中资源约束下的计划优化问题,以适应航空并行多项目网络寻优的需要。

1 航空并行多项目资源受限项目调度的问题描述

航空企业的项目计划往往是以分级网络计划图的形式展现的,而企业内部由于自身的资源有限,不可能在项目计划上无限地进行资源投入,往往只有一个固定的资源供应量。为了保证制定的项目计划能够实施,需要在资源受限制的条件下,对各项目任务的施工进度计划进行调整,使各单位时间资源需求量在均满足资源限量的要求上,达到最小化的项目总工期,为了描述资源约束下网络计划的优化问题,进行如下的定义:

- (1) 项目中的任务一旦启动,就必须进行到完工,中途不得中断。
- (2) 资源供应能力在其工期使用范围内为均匀分布。
- (3) 项目中每一个任务对某种资源的需求量都小于该资源的最大供应量。

建立如下的数学模型:

航空并行多项目在资源约束下网络计划的调度问题涉及 n 个独立的并行项目,第 i 个项目包含 n_i+2 个任务,其中第 0 和第 n_i+1 个任务为虚拟项目的起始和终止任务,其作用是标记项目开始时间和结束时

* 国家科技支撑计划资助项目(2006BAF01A03)。

间,有一定的持续时间但资源消耗量均为 0;项目 i 的任务 j 对第 k 种资源的消耗量为 r_{ijk} , 持续时间为 d_{ij} ;任务 ij 的开始时间为 S_{ij} ; A_t 为时刻 t 正在进行的任务集合; P_{ij} 为任务 ij 的紧前任务集合; R_k 是资源 k 的总量; k 是不同资源类型的数量。则问题可以用数学语言描述如下:

$$\begin{cases} \min \sum_{i=1}^n T_i = \sum_{i=0}^n | (S_{(n+1)} - S_{i1}) | \\ \text{s.t. } s_{ij} + d_{ij} \leq s_j, \forall ij \in P_{ij} \\ \sum_{A_{ij} \in I_i} r_{ijk} \leq R_k, \forall k, t \\ \sum_{i=1}^n w_i = 1 \end{cases}, \quad (1)$$

其中 $\min \sum_{i=1}^n T_i = \sum_{i=0}^n | (S_{(n+1)} - S_{i1}) |$ 定义的是优化的目标函数,即调整后的项目工期与原项目工期之差的加权和最小; $\text{s.t. } s_{ij} + d_{ij} \leq s_j, \forall ij \in P_{ij}$ 定义的是紧前约束关系,表示任务必须在其所有紧前任务均已完工的情况下才能开始进行; $\sum_{A_{ij} \in I_i} r_{ijk} \leq R_k, \forall k, t$ 定义的是资源约束关系,即在时刻 t 进行的所有任务对任何一种资源 k 的消耗量不能超过其限制数量。 w_i 表示项目 i 的权数,并行项目的权重系数之和必须为 1,权重的具体判别方法可参见文献[7]。

2 改进型粒子群算法的设计

2.1 粒子适应度函数的确定

调度模型的目标值是最小化的加权后项目总工期 $\sum_{i=1}^n w_i T_i$,它是衡量调度优化的标准,加权后的总工期 $\sum_{i=1}^n w_i T_i$ 越小越好。而在粒子群算法中,用粒子的位置表示待优化问题的解,即表示一个可行的调度方案,每个粒子性能的优劣程度取决于待优化问题目标函数确定的适应值,因此,必须将原始目标函数转换为适应度函数以确保优秀粒子具有较高的适应度值,在本模型中优化的目标为项目总工期 $\sum_{i=1}^n w_i T_i$, $f(X_i)$ 是粒子 i 的适应度函数, $T(t)$ 是粒子 i 在次迭代时对应的目标函数值,即项目加权后的总工期 $\sum_{i=1}^n w_i T_i$,

为一正数, $X_i(t)$ 为粒子在次迭代的位置,所以在 t 时刻粒子 i 适应度函数为:

$$f(X_i(t)) = \frac{1}{T(X_i(t))} = \frac{1}{\sum_{i=1}^n w_i T_i}. \quad (2)$$

这样就使得资源方差小的优秀个体对应较大的适应度,为一正数。令表示当前抗体的第 i 个个体,则式 (2) 表示当前第 i 个粒子个体的适应度函数值。

2.2 粒子位置信息的编码操作

由于各并行项目任务之间存在着紧前约束关系,任意的排序可能产生不可行的进行次序,为了有效地产生处理前后约束的编码,采用一种基于优先权系数的编码方案来解决该问题。将受某任务影响的任务都排在其后,因而就可以按照染色体中基因的顺序判断任务的合法性。基于优先权系数的并行多项目网络计划拓扑排序的步骤如下:

(1) 将并行的各项目首尾各添加一个虚拟的任务,虚拟任务具有一定持续时间但不消耗资源,所有的起始虚拟任务共用一个起始节点,所有的结束虚拟任务共用一个结束节点,这样就可以将多个双代号网络图联结成一个双代号网络图;

(2) 将并行多项目网络图添加的虚拟起始任务置入部分拓扑排序序列中;

(3) 搜索后续未进行排序的任务,检查其紧前任务是否都位于部分拓扑序列中,如果满足要求,将其置入部分拓扑排序序列中;

(4) 重复 (2) 直到所有任务都置入部分拓扑排序序列中,得到网络图的拓扑排序;

(5) 检查排序好的任务序列的优先权系数,将排序好的任务在满足任务紧前紧后约束的条件下按优先权系数的大小排序。

相对应粒子位置采用三维实数编码,用排序好的任务 r_i 的优先权作为基因, r_i 是区间 $[0, 1]$ 内的一个实数,数值越大则优先权越高;维数 2 对应的是该任务的编号;维数 3 对应的是该任务的紧前关系集合 P_i 和紧后关系集合 P_i ,按照拓扑排序^[8]将所有任务排列成一行形成粒子位置的实数编码串。这样将每一个粒子的位置对应于待求问题的一个可行调度解,如图 1 所示。

其中, d 为多项目集合 T 元素的数目,表示编码的串长, $r_1, r_2, \dots, r_k, \dots, r_d, \dots$ 表示集合 T 中元素的优先权系数,且 j 任务的优先权系数 r_j 要小于所对应的紧前任务的最小优先权系数并且大于它紧后任务的最大优

先权系数, 即 $\max\{r_k | k \in P_i\} < r_j < \min\{r_i | i \in P_j\}$ 。这样使得粒子编码串上任意一位满足紧前与紧后的约束, 满足优化网络模型的约束。

2.3 粒子位置的初始化

对粒子位置的初始化按照位置编码的顺序从左向右进行。任务 $j(j \in T)$ 的串位值, 即为它的优先权为:

$$\begin{cases} \text{若任务 } j \text{ 的紧前任务集合为空集}(P_j = \emptyset): \\ r_0 = 1; \\ \text{若任务 } j \text{ 的紧前紧后任务集合不为空集}(P_j \neq \emptyset, P_j^+ \neq \emptyset): \\ r_j = \max\{r_k | k \in P_j\} + \text{randon}(\min\{r_i | i \in P_j^+\} - \max\{r_k | k \in P_j\}); \\ \text{若任务 } j \text{ 的紧后任务集合为空集}(P_j^+ = \emptyset): \\ r_n = 0. \end{cases} \quad (3)$$

2.4 粒子个体的学习与认知操作

2.4.1 交叉操作

由于所涉及的问题为工程上的多目标组合优化问题, 本课题取消了粒子速度的概念, 将粒子个体向较优粒子群体的学习转化为粒子个体的位置编码和极值位置编码的交叉, 并采用单点交叉算子来完成交叉操作, 交叉位为随机产生的基因位即 $\text{random}(d-1)-1$, 如图 2 所示, 由于任务的开始时间依赖于其紧前任务的结束时间, 交叉操作后可能会产生非法个体, 因此在交叉之后需要对个体进行合法性检验, 对于非法个体需要进行修正。对于交叉位左右两边的抗体基因, 由于基因值未发生改变, 因而无需修正, 修正的过

r_1	r_2	...	r_k	...	r_d
任务编号 1	任务编号 2	...	任务编号 k	...	任务编号 d
紧前集合 1	紧前集合 2	...	紧前集合 k	...	紧前集合 d
紧后集合 1	紧后集合 2	...	紧后集合 k	...	紧后集合 d

图 1 粒子位置编码

Fig.1 Code for particle position

程为按照染色体中基因的顺序自交叉点向右逐一对基因进行检验, 检验公式为:

$$\max\{r_k | k \in P_i\} < r_j < \min\{r_i | i \in P_j^+\} \quad (4)$$

当基因值满足上述条件时该基因值合理, 否则需要进行调整, 调整方式为按照公式(3)重新向该基因赋值。

2.4.2 变异操作

变异是为了使粒子个体位置的染色体编码发生改变, 从而得到更优的解, 也就是在约束条件下改变某些工序的优先权系数, 变异的过程比较简单, 如图 3 所示。

对于给定的粒子个体, 按照变异概率随机选择其位置的染色体编码中一位或几位基因位 k , 然后对该基因的值按照公式(3)重新赋值, 保证了粒子的多样性。赋值之后, 检验和校正步骤和上文一致。

2.5 粒子位置信息的解码操作

假设有 m 个并行的项目, S_i 令表示项目 i 的开始时间, 染色体 $R=(r_1, r_2, \dots, r_i, \dots, r_n)$ 表示所得到的拓扑排序后的集合中相应任务的优先级系数, 染色体序号 i 所对应的任务编号为 t , 所对应的串行调度方案的调

当前粒子位置编码

r_1	r_2	...	r_k	...	r_d
任务编号 1	任务编号 2	...	任务编号 k	...	任务编号 d
紧前集合 1	紧前集合 2	...	紧前集合 k	...	紧前集合 d
紧后集合 1	紧后集合 2	...	紧后集合 k	...	紧后集合 d

较优粒子位置编码

r_1	r_2	...	r_k	...	r_d
任务编号 1	任务编号 2	...	任务编号 k	...	任务编号 d
紧前集合 1	紧前集合 2	...	紧前集合 k	...	紧前集合 d
紧后集合 1	紧后集合 2	...	紧后集合 k	...	紧后集合 d

交叉位

r_1	r_2	...	r_k	...	r_d
任务编号 1	任务编号 2	...	任务编号 k	...	任务编号 d
紧前集合 1	紧前集合 2	...	紧前集合 k	...	紧前集合 d
紧后集合 1	紧后集合 2	...	紧后集合 k	...	紧后集合 d

交叉后的粒子位置

图 2 交叉操作示意图

Fig.2 Diagram of crossing operation

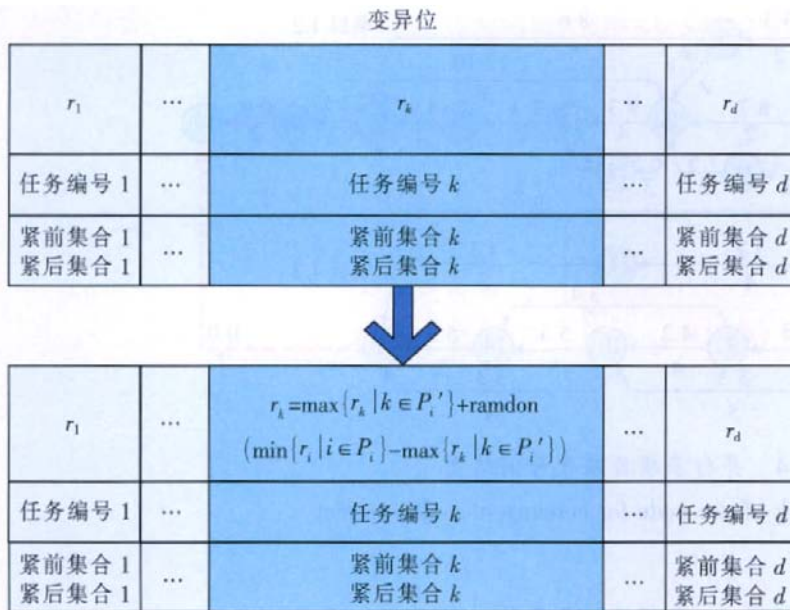


图 3 变异操作示意图

Fig.3 Diagram of variation operation

度过程如下:

Step 1: 令, $S=s_j, j=2$, 转入步骤 2;

Step 2: $j \leq n$ 当时, $j=j+1$, 转入步骤 3, 否则, 转入步骤 6;

Step 3: $s_j = \max\{s_k + d_k | k \in P_j\}$, 转入步骤 4;

Step 4: 判断活动 t_j 进行期间各种资源是否存在资源冲突, 如果存在资源冲突则转入步骤 5, 否则, 转入步骤 2;

Step 5: $s_j = s_j + 1$, 转入步骤 4;

Step 6: 返回所有任务的最早开始时间, 结束。

求得各项目工期为:

$$T_i = s_{i(n+1)} - s_{i1} \quad (5)$$

则加权后项目的总工期为:

$$T = \sum_{i=1}^m w_i T_i \quad (6)$$

2.6 改进型粒子群算法的求解步骤

改进型混合粒子群算法求解多项目资源均衡问题的步骤如下。

Step 1: 初始化粒子群。

(1) 初始化粒子种群的数量 N ;

(2) 初始化粒子群, 按式(3)进行初始化;

(3) 按式(2)计算所有粒子的适应度值 $f(X_i(0))$, 设置每个粒子当前的个体极值 $plbest(i)$ 为 $f(X_i(0))$, 当前的位置 $pxbest(i)$ 为 $X_i(0)$;

(4) 找出适应度值最高的粒子个体作为全局极值 $gbest$, 并记下全局极值的位置 $gxbest$;

Step2: 混合粒子群算法操作。

IF(迭代次数达到最大值)

{

FOR($j=1; j \leq N; j++$;))

{

(5) 第 j 个粒子的位置 $X_j(k)$ 与个体极值的位置 $pxbest(k)$ 交叉得到交叉后的个体位置 $X_j(k+1)$;

(6) 交叉得到的个体位置 $X_j(k+1)$ 与全局极值的位置交叉得到 $X_j(k+1)$ 与全局极值的位置交叉得到 $X_j(k+1)$, 并对交叉的个体按式(4)进行检验, 非法个体按公式(3)进行校正;

(7) 对交叉后的个体如式(3)所示进行变异操作;

(8) 重新计算第 j 个粒子的适应度

$f(X_j(k+1))$, 如果 $f(X_j(k+1)) > plbest(j)$

则 $plbest(j) = f(X_j(k+1))$, $pxbest(j) = X_j(k+1)$, 如果 $f(X_j(k+1)) > gbest$, 则 $plbest = f(X_j(k+1))$, $pxbest = X_j(k+1)$

}

}

ELSE 输出全局极值 $gbest$ 和全局极值的位置 $pxbest$ 。

3 仿真分析

针对某型号机身装配项目如图 4 所示的 2 个子项目(主起落架装配, 项目编号 1.2; 左进气道装配, 项目编号 1.3)的实际情况, 按照本文 2.2 节所述的多项目合并方法, 通过添加的虚拟任务, 使这 2 个二级的装配项目合并为该航空制造企业一个虚拟的一级计划装配项目(项目编号 1), 合并后 2 个二级项目保持各自的开始时间以及各自的约束条件不变, 如图 4 所示: 箭头下方表示任务的工期, 上方表示该任务对 R_1 、 R_2 两种资源的需求量。

项目各任务参数如表 1 所示。

本例中需要对某 2 种可更新资源进行约束优化, 资源 R_1 的限量为 27, 资源 R_2 的限量为 20, 定义项目 1.2 的权重为 0.6, 项目 1.3 的权重为 0.4, 粒子群数量 $N=50$, 最大迭代次数为 100, 取值为 50, 若分别对这 2 个项目进行工期优化, 得到的只是分别针对这 2 个项目的局部最优解, 项目的总工期并不处于优化

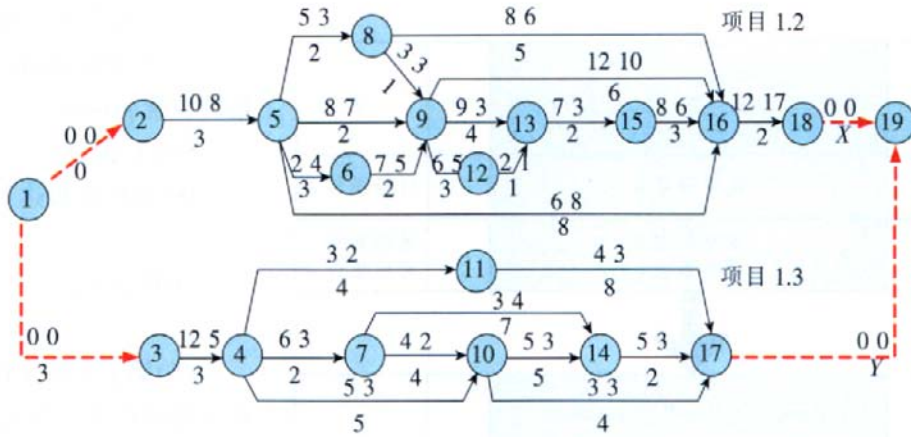


图4 并行多项目双代号网络图

Fig.4 Network of two code for concurrent multi-project

表1 各项目任务参数

任务编号	持续时间	资源 R ₁ 的强度	资源 R ₂ 的强度	所属项目	任务编号	持续时间	资源 1 的强度	资源 2 的强度	所属项目
1-2	0	0	0	1.2	1-3	3	0	0	1.3
2-5	3	10	8	1.2	3-4	3	12	5	1.3
5-6	3	2	4	1.2	4-7	2	6	3	1.3
5-8	2	5	3	1.2	4-11	4	3	2	1.3
5-9	2	8	7	1.2	4-10	5	5	3	1.3
5-16	8	6	8	1.2	7-10	4	4	2	1.3
6-9	2	7	5	1.2	7-14	7	3	4	1.3
8-16	5	8	6	1.2	11-17	8	4	3	1.3
8-9	1	3	3	1.2	10-14	5	5	3	1.3
9-12	3	6	5	1.2	10-17	4	3	3	1.3
9-16	6	12	10	1.2	14-17	2	5	3	1.3
12-13	1	2	1	1.2	17-19	Y	0	0	1.3
9-13	4	9	3	1.2					
13-15	2	7	3	1.2					
15-16	3	8	6	1.2					
16-18	2	12	17	1.2					
18-19	X	0	0	1.2					

状态, 合并项目后, 总体优化得到如下结果, 优化后的各个任务的开始时间见表 2。

由公式(6) 计算得优化后项目 1.2 的工期为 $T_1 = S_{18-19} - S_{2-5} = 27$, 而项目 1.3 优化后的的工期为 $T_2 = S_{17-19} - S_{3-4} = 24$, 加权后的项目总工期为 $T = T_1 + T_2 = 25.8$, 采用以下启发式准则对该实例进行资源约束项目调度优化: 最多的后继任务准则 (MostSA)、最小总时差准则 (MinTS)、以及最大资源准则 (MaxR), 所得到的优

化结果见表 3。由表中结果可以看出, 用不同的启发式方法处理同一问题的效果有很大的差别, 利用改进型粒子群算法处理资源约束项目调度问题时要比启发式方法更优秀。

4 结束语

本课题采用了一种改进型粒子群算法来解决并行多项目资源约束下的计划优化问题。算法以最短的

表 2 优化后各项目的开始时间

任务编号	任务名称	开始时间	所属项目	任务编号	任务名称	开始时间	所属项目
1-2	虚工序	0	1.2	1-3	虚工序	0	1.3
2-5	轮舱内无线电安装	0	1.2	3-4	附件安装	3	1.3
5-6	轮舱三角区液压导管和附件安装	5	1.2	4-7	导管安装	6	1.3
5-8	42 框电气安装	5	1.2	4-11	电缆固定	6	1.3
5-9	龙骨梁定位	3	1.2	4-10	进气道部件对接	7	1.3
5-16	附件安装	3	1.2	7-10	安装调节	11	1.3
6-9	主起落架随动护板安装	8	1.2	7-14	电气安装	15	1.3
8-16	主起落架舱门电门机构安装	10	1.2	11-17	支架安装	18	1.3
8-9	龙骨梁电气安装	10	1.2	10-14	上模安装	15	1.3
9-12	无线电缆安装	12	1.2	10-17	固定支架	20	1.3
9-16	主起落架锁机构安装	15	1.2	14-17	下模安装	22	1.3
12-13	主起落架侧杆安装调节	15	1.2	17-19	虚工序	27	1.3
9-13	起落架侧轮机构安装	11	1.2				
13-15	主起落架电气安装	16	1.2				
15-16	主起落架收放机构安装	21	1.2				
16-18	防火传感线和传感器安装	25	1.2				
18-19	虚工序	27	1.2				

表 3 不同方法优化结果的比较

方法	项目 1.2 优化后工期	项目 1.3 优化后工期	加权后的总工期
MostSA	30	25	28
Min	32	20	27.2
MaxR	31	27	29.4
改进型粒子群算法	27	24	25.8

项目总工期为优化目标,对各调度任务的优先权进行编码解码,彻底避免不可行调度解的产生,从而使每个粒子的位置都代表一个可行的调度方案,并引入遗传操作交叉和变异的内容,对各可行的调度方案进行优化,加快了算法的收敛速度。实例计算表明该改进型粒子群算法能快速有效地解决并行多项目资源约束下网络计划的优化问题。

参 考 文 献

- [1] Patterson J H, Slowinski R, Talbot F B, et al. An algorithm for a general class of precedence and resource constrained scheduling problem. *Advances in Project Scheduling*. Amsterdam: Elsevier Publisher, 1989, 3-28.
- [2] Brucker P, Knust S, Schoo A, et al. A branch and

bound algorithm for the resource-constrained project scheduling problem. *Eur J of Ops Res*, 1998, 107: 272-288.

[3] Brucker P, Knust S, Schoo A, et al. A branch and bound algorithm for the resource-constrained project scheduling problem. *Eur J of Ops Res*, 1998, 107: 272-288.

[4] 白思俊. 网络计划的计算机辅助分析. 西安: 陕西科学技术出版社, 1991.

[5] Davisew, Pattersonjh. A comparison of heuristic and optimum solutions in resources constrained project scheduling. *Management Science*, 1975, 21.

[6] 刘士新,王梦光,唐加福. 一种求解资源受限工程调度问题的遗传算法. *系统工程学报*, 2002, 17(1): 1-7.

[7] 王永超. 航空多型号项目计划与协调控制技术研究[D]. 西安: 西北工业大学. 2007: 24-32.

[8] 王为新. 航空产品资源优化方法研究. 西安: 西北工业大学. 2006: 13-14.

[9] 高尚,杨静宇. 群智能算法及其应用. 北京: 中国水利水电出版社. 2006.

[10] Fatemi Ghomi S M T, Ashjari B. A simulation model for multi-project resource allocation. *International Journal of Project Management*. 2002, 20 (2):127-130.

[11] Zhang Hong, Li Heng, Tam C M. Particle swarm optimization for resource-constrained project scheduling. *International Journal of Project Management*. 2006(24): 83-92.

(责编 依然)