

doi: 10.3969/j.issn.1006-1576.2010.01.006

粒子群算法在武器装备保障资源优化中的应用

张凯, 赵国荣, 姜静

(海军航空工程学院 控制工程系, 山东 烟台 264001)

摘要: 为简单方便地实现武器装备保障资源优化, 采用粒子群优化算法求解资源约束项目调度问题。具体方法主要包括: 确定不违反前后约束的活动进行次序、确定资源约束项目调度问题的解空间、确定修复策略、计算粒子适应度。同时, 编制了粒子群优化程序, 对某型武器装备技术保障资源进行了优化。优化结果有助于保障资源合理配置, 对提高装备战斗力有重要的军事价值。

关键词: 资源约束项目调度问题; 粒子群优化算法; 资源优化

中图分类号: TP301.6; O224 **文献标识码:** A

Application of PSO in Weapon Equipment Safeguard Resource Optimization

ZHANG Kai, ZHAO Guo-rong, JIANG Jing

(Dept. of Control Engineering, Naval Aeronautical & Astronautical University, Yantai 264001, China)

Abstract: To simplify weapon equipment safeguard resource project optimization, particle swarm optimization (PSO) algorithm was chosen to solve the resource-constrained project scheduling problem (RCPSP). The determinations of activities sequence with relation constraint, the solution space of RCPSP and the repair strategy and the calculation of particle fitness were introduced in detail. Application to a certain type of weapon equipment safeguard resource project optimization example using the PSO program given shows us the validity of using PSO to solve RCPSP and can bring us great military benefit to improve the fighting ability of weapon equipment.

Keywords: RCPSP (Resource-Constrained Project Scheduling Problem); PSO (Particle Swarm Optimization); Resource optimization

0 引言

武器装备资源优化配置是指在保证装备可靠性的前提下, 对武器装备技术保障资源合理地进行分配使用, 达到某种优化目标, 如资源限制下的最短保障时间、资源使用最小均衡方差或最小保障成本等。武器装备资源优化配置属于资源约束的项目调度问题 (Resource-Constrained Project Scheduling Problem, RCPSP), 即资源限制—工期最短问题^[1]。目前, 求解该问题的主要方法是启发式方法, 可以看作优先调度规则, 在为解决资源冲突而进行顺序决策时, 根据临时或与资源相关的启发式规则来分配活动的优先权。文献[2-3]采用遗传算法用于求解资源约束的项目调度问题, 但由于算法比较复杂, 较难实现。而粒子群优化算法 (Particle Swarm Optimization, PSO) 比遗传算法简单, 没有交叉算子和变异算子, 更容易实现^[4]。故采用粒子群优化算法来求解资源约束的项目调度问题, 并应用于某型武器装备的技术准备资源优化。

1 粒子群优化算法简介

粒子群优化算法是由 James Kennedy 和 Russell

Eberhart^[5-6]提出的一种智能仿生进化优化算法, 源于对鸟群捕食的行为研究。在 PSO 系统中, 每个备选解被称为一个粒子 (Particle), 多个粒子共存、合作寻优, 每个粒子根据自身的“经验”在问题空间中向更好的位置“飞行”, 搜索最优解。

标准 PSO 算法 (Shi and Eberhart)^[7]的数学表示如下: 一个由 m 个粒子组成的群体 (Swarm) 在 D 维搜索空间中以一定的速度飞行, 每个粒子在搜索时, 考虑到了自己搜索到的历史最好点和群体内其他粒子的历史最好点, 在此基础上进行位置 (状态, 也就是解) 的变化。

第 i 个粒子的位置表示为: $X_i=(x_{i1}, x_{i2}, \dots, x_{id}, \dots, x_{iD})$, 第 i 个粒子的速度表示为: $V_i=(v_{i1}, v_{i2}, \dots, v_{id}, \dots, v_{iD})$, 第 i 个粒子经历过的历史最好点表示为: $P_i=(p_{i1}, p_{i2}, \dots, p_{id}, \dots, p_{iD})$, 群体内 (或邻域内) 所有粒子所经过的最好的点表示为 P_g 。粒子的速度和位置分别根据进化式 (1)、式 (2) 进行更新:

$$v_{id}^{k+1} = \omega v_{id}^k + c_1 r_1 (p_{id}^k - x_{id}^k) + c_2 r_2 (p_{gd}^k - x_{id}^k) \quad (1)$$

$$x_{id}^{k+1} = x_{id}^k + v_{id}^{k+1} \quad 1 \leq i \leq m, 1 \leq d \leq D \quad (2)$$

其中 c_1 、 c_2 称为学习因子或加速度系数, 一般为正常数, 学习因子使粒子具有自我总结和向群体

收稿日期: 2009-07-16; 修回日期: 2009-08-28

作者简介: 张凯 (1974-), 女, 山东人, 讲师, 硕士, 从事导航、制导与控制及先进算法研究。

中优秀个体学习的能力, 从而向自己的历史最优点以及群体内的历史最优点靠近。 c_1 和 c_2 通常等于 2。 ω 称为惯性权重, 其大小决定了对粒子当前速度继承的多少, 合适的选择可以使粒子具有均衡的广域搜索能力和局部搜索能力。 r_1 、 r_2 是[0,1]区间内均匀分布的伪随机数。

2 资源限制—工期最短项目调度数学模型

资源限制—工期最短项目调度问题的基本类型可以描述如下: 一个项目包含许多相互关联的活动, 其中, 每个活动都有确定的持续时间和给定的资源要求。资源的数量是有限的, 但会随着时间而更新。资源之间不可替代, 活动不能被中断。问题的解是确定满足前后约束和资源约束的最小项目工期。该问题的数学模型可以表示为:

$$\min(\lambda) = \min\{t_s(n)\} \quad (3)$$

$$st. \begin{cases} t_s(j) + t_D(j) \leq t_s(i), & \forall j \in IP(i) \\ \sum_{k \in A[t]} r(i, k) \leq b(k), & k = 1, 2, \dots, m, t = 1, 2, \dots, t_s(n) \\ t_s(i) \geq 0, & i = 1, 2, \dots, n \end{cases} \quad (4)$$

式 (4) 中: $t_s(i)$ 是活动 i 的实际开始时间; $t_D(i)$ 是活动 i 的持续时间; $IP(i)$ 是活动 i 的紧前活动集合; $r(i, k)$ 是活动 i 需要资源 k 的数量; $b(k)$ 是资源 k 的总量; $A[t]$ 是在 t 时刻正在进行的所有活动的集合; m 是不同资源类型的数量; 活动 1 和活动 n 分别是虚开始活动和虚结束活动, 不消耗资源, 持续时间为 0, 其作用是标记项目开始和项目结束。

式 (3) 给出项目的目标函数是最小化项目总工期, 即虚结束活动的实际开始时间。式 (4) 给出了项目必须满足的 2 个约束条件, 即前后约束和资源约束: 前后约束为基本约束条件, 体现了项目中各活动的逻辑关系约束; 资源约束为属性约束条件, 要求在任何时段内所有活动消耗的资源 k 的数量不超过其限制的数量。

3 粒子群算法求解资源约束项目调度问题

资源约束项目调度问题本质上包含以下 2 个问题: 1) 确定不违反前后约束的活动进行次序; 2) 确定不违反资源约束的每个活动的开始时间。粒子群优化算法的求解思路是: 先利用简单编码规则来解决第 1 个问题, 然后, 在确定满足资源约束的项目调度最大解空间的基础上, 通过合理设计, 发挥粒子群优化算法群体智能的优势, 在最大解空间中寻找满足资源约束的最短项目工期的项目调度方案, 以解决第 2 个问题。

3.1 确定不违反前后约束的活动进行次序

提出一个简单编码规则, 确定不违反前后约束的活动进行次序。编码规则为: 对于一个含有 n 个活动的活动, 设虚开始活动的活动号是 1, 虚结束活动的活动号是 n , 其他实际活动从小到大 (从左到右) 进行编码的原则: 实际活动集合中任一活动的所有紧前活动都应排在其前面, 任一活动的所有紧后活动都应排在其后面, 并行活动编码的先后顺序可随机排列。

结合项目的网络计划图模型^[8], 按上述简单编码规则对项目中的活动进行编码, 按编码大小排列的活动序列即满足项目前后约束的项目调度方案。

3.2 确定资源约束项目调度问题的解空间

项目调度是在各活动的最早开始时间和最晚开始时间的范围内确定各活动的实际开始时间。采用上述简单编码规则, 对项目中的所有活动进行编码, 并依据项目中的活动前后约束信息和活动持续时间信息, 确定各活动满足前后约束的最早开始时间和最晚开始时间, 即解空间。由于项目中的活动须满足前后约束和资源约束, 同时, 在资源限制—工期最短项目调度过程中, 项目工期是动态变化的, 难以确定各活动真正的最早开始时间和最晚开始时间, 故无法确定准确的解空间。而粒子群优化算法又需要一个解空间, 采用的解决办法是: 先确定一个满足活动前后约束的最大解空间, 然后, 在这个最大解空间里采用粒子群优化算法来寻找满足资源限制的最短工期项目调度方案。

3.2.1 最早开始时间的确定方法

由于项目中的活动有紧前约束, 且每个活动的持续时间已知, 在不考虑资源约束的情况下, 按活动编码从小到大 (从左到右) 依次可以确定每个活动的最早开始时间。假定时间单位为天, 具体方法如下: 虚开始活动的最早开始时间 $t_{ES}(1)$ 设为第 0 天; 若活动 i 的紧前活动为虚开始活动, 则最早开始时间 $t_{ES}(i)$ 设为第 1 天, 否则, 取其紧前活动集合中最早结束时间 $t_{EF}(k)$ 的最大值。

$$\begin{cases} t_{ES}(1) = 0 \\ t_{ES}(i) = 1, IP(i) = E \\ t_{ES}(i) = \max_{k \in IP(i)} \{t_{EF}(k)\}, IP(i) \neq E \end{cases} \quad i = 2, \dots, n \quad (5)$$

3.2.2 最晚开始时间的确定方法

先求取所有活动按编码顺序串行的项目工期 λ_{\max} , 这是满足网络计划图的前后约束的最大项目

工期，因为是串行活动，必然满足资源限制约束，然后考虑紧后约束逻辑按活动编号顺序从大到小（从右到左）依次可以确定每个活动的最晚开始时间。具体方法如下：虚结束活动的最晚开始时间设为所有活动串行时的项目工期 λ_{\max} ，其他活动的最晚开始时间取其紧后活动集中最晚开始时间 $t_{LS}(k)$ 的最小值减去该活动的持续时间 $t_D(i)$ 。

$$\begin{cases} t_{LS}(n) = \lambda_{\max} = \sum_{i=1}^n t_D(i) \\ t_{LS}(i) = \min_{k \in IS(i)} \{t_{LS}(k)\} - t_D(i) \end{cases} \quad (6)$$

通过该方法确定各活动的最早开始时间范围是最大解空间，同时，满足前后和资源约束项目最短工期的各活动实际开始时间就在此最大解空间里。

3.3 确定修复策略

在初始化后或进化后，应检验粒子是否满足项目前后约束和时间约束条件，若不满足，将修复策略修复粒子的位置和进化速度。

若某活动 i 没有紧前活动，则依据式 (7) 对粒子进化位置修复；反之，则依据式 (8) 对粒子进化位置进行修复。经过修复的粒子既能满足前后约束，又尽可能地提前了活动的实际开始时间，有助于缩短项目工期。依据式 (9) 对粒子进化速度修复。

$$\begin{cases} t_S(i) = \text{random}\{t_{ES}(i), t_{LS}(i)\}, & \text{if } t_S(i) > t_{LS}(i) \\ t_S(i) = t_{ES}(i), & \text{if } t_S(i) < t_{ES}(i) \end{cases} \quad (7)$$

$$\begin{cases} t_S(i) = \max_{k \in IP(i)} \{t_S(k) + t_D(k)\}, & \text{if } t_S(i) < \max_{k \in IP(i)} \{t_S(k) + t_D(k)\} \\ t_S(i) = \text{random}\left\{ \max_{k \in IP(i)} \{t_S(k) + t_D(k)\}, t_{LS}(i) \right\}, & \text{if } t_S(i) > t_{LS}(i) \end{cases} \quad (8)$$

$$\begin{cases} v(i) = V_{\max}(i), & \text{if } v(i) > V_{\max}(i) \\ v(i) = V_{\min}(i), & \text{if } v(i) < V_{\min}(i) \end{cases} \quad (9)$$

3.4 计算粒子适应度

因优化目标是 minimized 项目工期，需要将项目工期转化为粒子的适应度，适应度越大则该粒子越优秀。如式 (10)。若 1 个粒子的最大日资源消耗量小于等于项目资源限制量，表明该粒子符合资源限制约束，则取串行活动项目工期 $\lambda(\max)$ 减去该粒子的实际项目工期 $t_S(n)$ 作为该粒子的当前适应度，即满足资源限制约束的粒子实际项目工期越短，该粒子的适应度就越大；若 1 个粒子的最大日资源消耗量大于资源限制量，即不满足资源限制约束，该粒子的适应度应最小，则取 0 作为该粒子的当前适应度。

$$\begin{cases} \text{Fitness}(X) = \lambda(\max) - t_S(n), & \text{if } r(i,k)_t \leq b(k), t=1,2,\dots,t(n) \\ \text{Fitness}(X) = 0, & \text{if } r(i,k)_t > b(k), t=1,2,\dots,t(n) \end{cases} \quad (10)$$

采用这种方法计算粒子适应度，不仅体现了粒子接近或达到目标函数的程度，也巧妙地避开了对资源限制约束的修复，不需要制定复杂的资源优先调度规则，充分发挥了粒子群优化算法的群智能优化特点，在解空间内寻找能同时满足前后约束和资源限制约束的最短项目工期的项目调度安排。

3.5 算法流程

资源约束项目调度的粒子群优化算法如图 1。

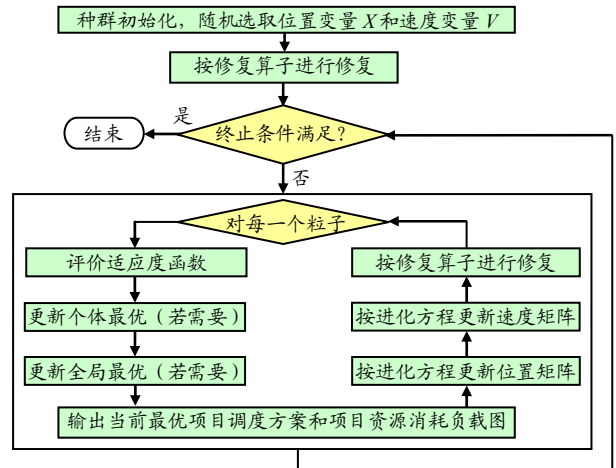


图 1 资源约束项目调度的粒子群优化算法流程

4 某型装备技术准备流程的资源优化实例

某型装备技术准备流程的网络计划图如图 2。装备技术准备项目信息如表 1。

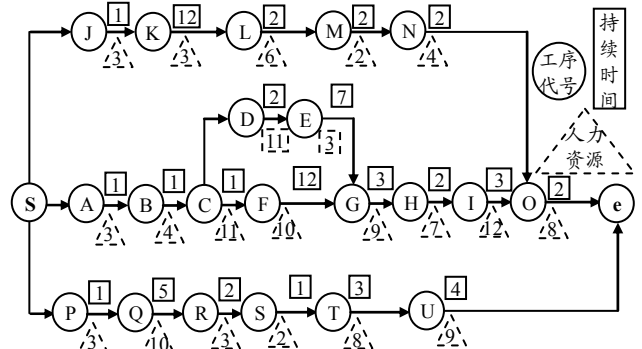


图 2 某型装备技术准备流程的网络计划图

表 1 某型装备技术准备项目信息表

工序代号	人力资源	工序时间	紧前工序	紧后工序
A	3	1	-	B
B	4	1	A	C
C	11	1	B	D、F
D	11	2	C	E
E	3	7	D	G
F	10	12	C	G
G	9	3	E、F	H
H	7	2	G	I
I	12	3	H	O
J	3	1	-	K
K	3	12	J	L
L	6	2	K	M
M	2	2	L	N

续表 1

工序代号	人力资源	工序时间	紧前工序	紧后工序
N	4	2	M	O
O	8	2	I、N	-
P	3	1	-	Q
Q	10	5	P	R
R	3	2	Q	S
S	2	1	R	T
T	8	3	S	U
U	9	4	T	-

1) 算法参数设置

粒子群优化算法程序中的算法参数设置为: 种群数目为 40, 学习因子 $c_1=c_2=2$, 惯性权重 $\omega=1$, 进化代数为 1 000 代。

2) 优化结果

粒子群优化算法程序对装备技术准备进行资源限制—最短工期项目优化, 由于初始化具有随机性, 粒子群进化过程同样也具有随机性, 不是每次优化都能达到全局最优, 需要对多次优化结果进行统计分析, 计算不同优化结果的发生概率, 取概率最大的优化结果作为最终的优化结果。本例中, 优化后项目最短工期为 28 个时间单位, 优化后的项目调度方案之一(工序实际开始时间)按工序编号顺序为 [1, 2, 3, 4, 6, 7, 19, 22, 24, 1, 2, 18, 20, 22, 27, 12, 13, 18, 20, 21, 24], 对应的项目资源负载图如图 3, 纵坐标为人力资源数目, 横坐标为整个项目工期。

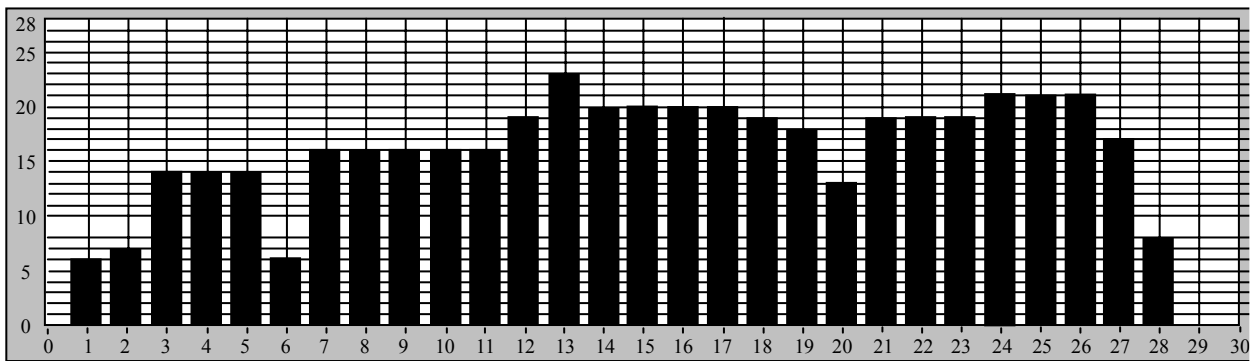


图 3 装备技术准备资源限制最短工期项目调度资源负载图

3) 收敛分析

经过多次试验, 一般需要 400 代左右可收敛于最优解 (28), 典型收敛曲线如图 4, 纵坐标为项目

工期(时间单位), 横坐标为进化代数。由图 4 可知: 曲线总体上是收敛的, 但不是连续收敛, 而是断续、跳跃收敛的, 符合粒子群优化算法的特点。

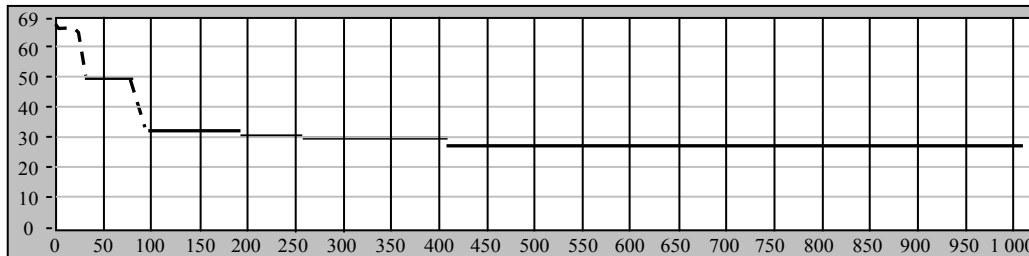


图 4 装备技术准备资源限制工期最短项目优化收敛曲线

5 结论

该算法对装备技术准备进行了优化。借助优化结果, 可合理地配置设备、规划操作场地以及适当地调整人员配备, 对提高装备保障单位的技术准备能力和装备战斗力有一定的现实意义。

参考文献:

[1] 玄光男, 程润伟. 遗传算法与工程优化[M]. 于歆杰, 周根贵 译. 北京: 清华大学出版社, 2004: 200-201.
 [2] 骆刚. 遗传算法在工程项目网络计划优化中的应用[D]. 天津: 天津大学, 2002.
 [3] 杨维, 李歧强. 粒子群优化算法综述[J]. 中国工程科学, 2004, 6(5): 87-94.

[4] Cheng, R. and M. Gen, Resource constrained project scheduling problem using genetic algorithms, International Journal of Intelligent Automation and SoftComputing[J]. 1997, 3(3): 273-286.
 [5] Eberhart R.C, Kennedy J. A new optimizer using particle swarm theory. Proc. On 6th International symposium on micromachine and human science[C]. Piscataway, NJ: IEEE service center, 1995: 39-43.
 [6] Kennedy J., Eberhart R.C. Particle swarm optimization.Proc IEEE International Conference on Neural Networks[C]. Perth, 1995: 1942-1948.
 [7] Shi Yuhui, Eberhart R. A modified particle swarm optimizer[J]. Proc IEEE Int Conf on Evolutionary Computation[C]. Anchorage, 1998: 69-73.
 [8] 崔小杰, 杨军, 苏京晶. 装备维修保障信息数据规划[J]. 四川兵工学报, 2008(6): 125-127.