

型号项目工期风险管理决策模型与自适应遗传算法*

高峰 陈英武

(国防科技大学 信息系统与管理学院,湖南 长沙 410073)

摘要 :针对型号项目的过程特点,建立了型号项目工期风险的管理决策模型。模型考虑了型号项目中活动重叠、活动迭代、活动执行时间的不确定性和可更新资源总量等重要的工期风险影响因素。从模型的特点出发,给出了问题求解的基于自适应遗传算法的仿真优化方法。算例显示,该算法能较好地求解本文的工期风险管理决策问题。

关键词 :型号项目 ;工期风险 ;自适应遗传算法 ;仿真优化

中图分类号 :C391 文献标识码 :A

The Risk Management Decision Model of the Model Project Duration and Adaptive Genetic Algorithm

GAO Feng ,CHEN Ying-wu

(College of Information System and Management , National Univ. of Defense Technology , Changsha 410073 , China)

Abstract :According to the characteristics of the process of the model project , the risk management decision model of the model project is presented . Some essential risk factors of the project duration such as activity overlapping , activity iterations , uncertain activity durations and non-consumable resource amount are integrated into the decision model . Considering the characteristics of the model , a simulation optimization method based on adaptive genetic algorithm is proposed . A case study shows that the algorithm is beneficial .

Key words :model project ; duration risk ; adaptive genetic algorithm ; simulation optimization

在型号项目过程中,经常面临一些风险事件,如任务需求更改,在计划时间内没有完成预定的活动,还有管理和人员方面的问题。如果在制定计划时不能充分考虑这些风险因素,将很可能导致型号项目不能按时完成,并带来一系列的损失。因为不同的型号项目计划具有不同的工期风险,所以本文的风险管理决策是一个广义的概念,它既包括制定一个科学合理的项目计划,又包括制定工期风险管理措施的选择计划,两者相辅相成,融为一体。

在资源约束条件下制订项目计划的科学方法是项目调度。当活动的执行时间不确定时,问题被归结为一类随机资源受限项目调度问题(stochastic resource constrained project scheduling problem, SRCPSP),目前该类问题正受到越来越多的关注^[1~3]。在工期风险管理措施方面,典型的是压缩活动的执行时间。在 PERT/CPM 型网络中,时间压缩被称为时间费用交换问题,并已经得到较好的解决^[4,5],但它们没有考虑资源的有限性。

由于本文的模型中含有多个随机变量,模型的输入输出关系缺少结构信息,不存在解析表达式,仅能通过仿真得到,而一次仿真仅给出对应某输入的一次性能估计,通常存在误差。在理论界,本文的模型被归结为仿真优化问题^[6],仿真优化的最新发展是大量启发式算法的应用^[7]。

1 模型假设与符号

模型做如下假设:

(1)型号项目的活动重做次数是可估计的^[8],它是服从某一离散分布的随机变量。

* 收稿日期:2005-07-20
基金项目:国家自然科学基金资助项目(70272002)
作者简介:高峰(1976—),男,博士生。

(2)型号项目活动的重做有两种模式:一是上游活动*i*发生改变,导致下游活动*j*重做,称为前馈型迭代;二是下游活动*j*在进行过程中发现上游活动*i*存在错误,需要上游活动*i*重做,称为反馈型迭代。

(3)批处理。在型号项目过程中,上游活动*i*可能在执行过程中发生多次变化,不是在每次变化后都要对下游活动*j*重做,而是等上游活动*i*完成后,对这些变化进行批处理,然后执行下游活动*j*的重做。同样,在下游活动*j*执行过程中发现上游活动*i*存在错误也是要等待下游活动*j*做完后进行批处理。

(4)项目的不可更新资源数量无限,可更新资源数量有限,活动完成后可更新资源立即释放;允许活动因资源冲突而等待,活动一旦执行,不可中断,活动进行过程中所需要的资源恒定。

(5)活动在事先没有指定优先级,每个活动只限一种执行模式。

模型中用到的符号如下:

B_{i_k} ——活动*i*的第*k*-1次重做的开工时间,它作为决策变量; d_i^{ori} ——活动*i*的原始执行时间,是一个随机变量; d_i^0 ——活动*i*被压缩后的执行时间; d_{i_k} ——活动压缩后的第*k*-1次重做的执行时间; w_i ——活动*i*共做次数,是一个随机变量; c_i ——附加在活动*i*上的费用,决策变量; L_i ——学习参数,表示活动*i*在重做过程中由于经验增长等原因产生的进度的加快幅度。学习参数越小,学习能力越强; O_{ij} ——上游活动*i*与下游活动*j*发生重叠时的搭接参数, O_{ij} 越大,重叠程度越小。特别地,当 $O_{ij} = 1$ 时,活动*i*、*j*之间无重叠; r_{ij} ——风险时间参数。它与 O_{ij} 共同决定下游活动在重叠时的风险时间增量; s_{ij} ——上游活动与下游活动*j*之间的迭代模式。 $s_{ij} = 0$,表示活动*i*、*j*之间无迭代发生的原因,属于一般的紧前紧后关系; $s_{ij} = 1$,表示活动*i*、*j*之间属于前馈型迭代; $s_{ij} = 2$,表示活动*i*、*j*之间属于反馈型迭代; C ——附加费用总量; T ——合同规定的工期; P_T ——项目提前或超期的奖惩系数; V ——原始活动网络; $J(j)$ ——活动*j*的紧前活动集; ξ_1, ξ_2 ——分别表示决策者对工期风险的偏好系数和对附加费用的偏好系数; $n(V)$ ——型号项目的原始活动数目; $n(c_i)$ ——附加费用到活动*i*的离散点的数目; c_{im} ——活动*i*用第*m*种时间费用模式执行的布尔值。

2 型号项目工期风险管理决策模型

2.1 目标函数

型号项目工期风险管理决策的目标之一是最小化超期导致的罚款额,由于采用附加费用的方法缩短单个活动的执行时间以最小化超期导致的罚款额,因此还存在附加费用与工期风险之间的权衡问题。本文的型号项目工期风险管理决策的目标为

$$\min\{\xi_1 \cdot E[P_T \cdot (B_{n(V)+1} - T)] + \xi_2 \cdot C | \xi_1 + \xi_2 = 1\} \quad (1)$$

2.2 次序约束

在本文的模型中,因考虑了活动重叠的情况,活动之间不完全是紧前紧后关系,所以有

$$B_{j_1} \geq B_{i_{w_i}} + O_{ij} \cdot d_{i_{w_i}}, s_{ij} = 0 \quad (2)$$

$$B_{j_1} \geq B_{i_1} + O_{ij} \cdot d_{i_1}, s_{ij} \in \{1, 2\} \quad (3)$$

式(2)表示当上游活动*i*与下游活动*j*发生重叠时,如果*i*、*j*之间只是一般的紧前紧后关系,重叠发生在上游活动*i*最后一次重做 i_{w_i} (*i*的重做由别的活动引致)与下游活动的第一次执行 j_1 (*j*的重做由别的活动引致)之间。式(3)表示,如果*i*、*j*之间存在发生重做的原因时,重叠发生在上下游活动的首次执行之间。

当活动之间不存在重叠时,可以根据迭代模式描述活动之间的紧前紧后关系,由假设(2)(3)知

$$B_{j_k} \geq B_{i_k} + d_{i_k}; B_{i_k} \geq B_{i_{k-1}} + d_{i_{k-1}}; B_{j_k} \geq B_{j_{k-1}} + d_{j_{k-1}}, 1 < k \leq w_i, s_{ij} = 1 \quad (4)$$

$$B_{i_k} \geq B_{j_{k-1}} + d_{j_{k-1}}; B_{j_k} \geq B_{i_k} + d_{i_k}, 1 < k \leq w_i, s_{ij} = 2 \quad (5)$$

迭代活动的紧前紧后关系可以用图1形象描述。当活动之间仅为简单紧前紧后关系时,有

$$B_{j_1} \geq B_{i_{w_i}} + d_{i_{w_i}} \quad r_{s_{ij}} = 0 \tag{6}$$

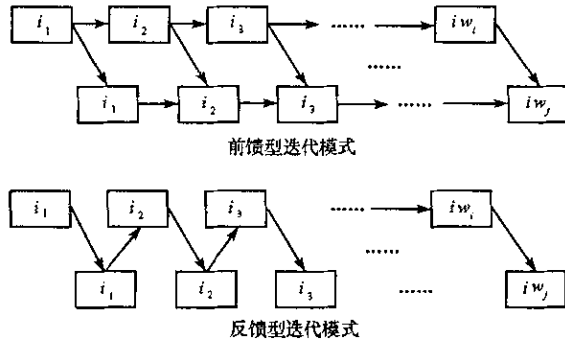


图 1 两种活动迭代模式

Fig.1 Two kinds of activity iteration modes

2.3 时间参数

型号项目的活动在重做时一般会吸取上次执行的经验,同时,技术人员的熟练程度也在提高,并有了一定的工作基础。简单而不失一般性,对于任意活动 i ,不妨设

$$d_{i_k} = d_i^0 \cdot L_i^{(k-1)}, \quad 1 \leq k \leq w_i \tag{7}$$

当活动处于多个重叠关系中的下游时,需要加上一段风险时间,于是有

$$d_{j_1} = d_j^0 + \max\{r_{ij} \cdot (1 - O_{ij}) \cdot d_j^0 \mid i \in J(j)\} \tag{8}$$

活动的执行时间和附加费用之间存在离散的关系,有

$$d_i^0 = f(d_i^{ori}, c_i) \tag{9}$$

其中,附加费用满足条件

$$\sum_i c_i = C \tag{10}$$

2.4 执行模式约束

每个活动只能用一种模式执行,于是有

$$\sum_{m=1}^{n(c_i)} c_{im} = 1 \tag{11}$$

其中,当 $c_{im} = 1$ 时,表示活动 i 选择第 im 种时间费用模式执行;当 $c_{im} = 0$ 时,表示活动 i 不选择第 im 种时间费用模式执行。

2.5 资源参数

活动 i_k 在 t 时刻对可更新资源 h 的需求量应不超过 t 时刻系统可获得的可更新资源 h 的总量,即

$$\sum_{i_k} r_{i_k h}(t) \leq R_h(t) \tag{12}$$

3 模型求解

3.1 解的编码

在型号项目工期风险管理决策模型中,决策变量有两个:各原始活动的开工时间和各原始活动附加的费用。我们采取以下的编码方式,将决策变量转化为易求解的活动优先权和活动执行模式:

一个基因包括 $n + \sum_{r=1}^{n(V)} n(c_r)$ 个基因座。前 n 个基因座采用基于优先权的编码方式,即用位置表示

活动的 ID,基因的值用来表示活动的优先权系数,形成了编码的 A 区。后 $\sum_{i=1}^{n(V)} n(c_i)$ 个基因座通过如下操作生成:首先复制 A 区中的基因座,每一个基因座依然代表活动的 ID,将第 i 个基因座(代表活动 i)

分为 $n(c_i)$ 个基因座,这样形成了编码的 B 区。 B 区中每个基因的取值为“0”或“1”,取“1”表示该基因座所对应的活动采用该基因座对应的附加费用投入量,取“0”表示该基因座所对应的活动不采用该基因座对应的附加费用投入量。因为一次决策中某活动的附加费用量一定,所以在解编码中,活动 i 所对应的 $n(c_i)$ 个基因座的基因值之和等于 1。图 2 是一个编码的简单示例。

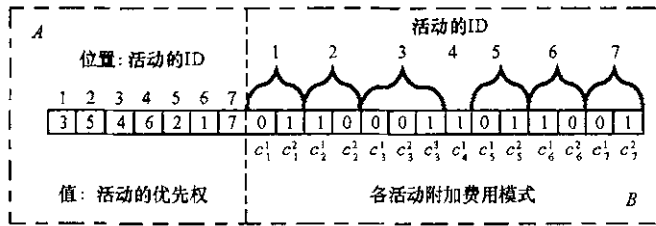


图 2 编码示例图

Fig.2 Encoding example

3.2 解码过程

根据编码得出各原始活动的开工期望及工期期望的过程称为解码,其过程如下:

Step1: 根据编码的 B 区读出各活动的压缩模式,计算各活动在被压缩后的概率分布参数和此时的附加费用总量 C 。

Step2: 运用轮盘赌,为每个活动组随机产生一个共做次数,并生成活动共做次数向量 W 。因为如果两个活动之间产生迭代关系,则两个活动总做次数相同,所以将发生迭代关系的活动编成一个组,这样就保证了它们共做次数相同。

Step3: 建立原始活动网络的矩阵表示 M_0 ,矩阵中的元素为 s_{ij} 的值。

Step4: 在 W 中依次读取各原始活动的共做次数,假设活动 i 重做的次数为 w_i ,则构建新的矩阵 M 。 M 由 M_0 扩展而成。扩展方法是在 M_0 的活动 i 之后(之下)依次添加 $w_i - 1$ 个新活动,它们分别表示活动 i 的第 1,第 2, ..., 第 $w_i - 1$ 次重做。这样新矩阵 M 的维数为 $\sum_i w_i$,置 M 为空矩阵。

Step5: 在关系集 S 中查找 s_{ij} 的类型。然后用式(2)~(6)解析新的矩阵 M 中各活动之间的关系。矩阵中的元素共有 3 类:当元素所在列的活动是所在行的活动的紧前活动时,该元素的值为“1”;当元素所在列的活动是所在行的活动的紧前活动且所在行的活动被提前(有重叠)时,该元素的值为“ O_{ij} ”;当元素所在行的活动和所在列的活动之间没有紧前紧后关系时,该元素的值为“0”。此时矩阵 M 为实矩阵,它是有活动重做时的活动网络的矩阵表示。

Step6: 随机生成一组原始活动执行时间向量,根据式(7)(8)计算矩阵 M 中其它活动的执行时间。

Step7: 检查 M 是否全为零,如果是,则输出各活动的开工时间和工期,转 Step2,进入下一次仿真直至达到指定的仿真次数,转 Step11;否则,检查当前正在执行的活动集合,修改当前活动在给定时间步长下的执行时间,并且修改当前活动在给定时间步长下的重叠时间。

Step8: 如果是正在执行的重叠关系的上游活动,且已经满足下游活动的开始时间,则更新矩阵 M ,将对应的 O_{ij} 的值改为“0”;否则,继续更新选定活动的重叠标示及重叠时间,直至有活动完成,转 Step9。

Step9: 如果有活动完成,则释放资源,更新矩阵 M ,寻找下一步可以执行的活动,检查资源约束及优先级约束(A 区),提交执行活动。

Step10: 计算最近一次活动结束的时间,转 Step7。

Step11: 根据工期的提前或超期情况,计算工期风险。

经典调度算法调度的活动之间遵循严格的紧前紧后关系,只有当上游活动全部执行完毕,下游活动才可以开工,所以只有当上游活动结束的突跳点出现时,下游活动进入可执行的活动集。当存在活动重叠时,我们在算法中通过 Step7~8 将上游活动的突跳点前移,当上游活动执行 $t_{ij} = O_{ij} \cdot d_i^0$ 时长后,突跳点出现,此时,置邻接矩阵中相应 O_{ij} 值为零,即表示下游活动 j 在此时刻进入可执行活动集。这样,活动重叠执行就可以在算法中体现出来。

3.3 自适应遗传算法

(1)产生初始解。读取原始活动的个数 $n(V)$,生成一个顺序随机的从 1 到 $n(V)$ 的整数排列,在 Matlab 6.5 中的命令是“randperm($n(V)$)”,如[4 2 1 3]是 randperm(4)的一个可能值。这样就保证了 A 区中所有基因座的值为从 1 开始连续的整数,且两两不同;对于 B 区中的基因座,采取分活动抽样的方法。具体说来,如果活动 i 有 $n(c_i)$ 种附加费用的模式,则产生一个随机数 μ ,如果 $\varepsilon/n(c_i) \leq \mu < (\varepsilon + 1)/n(c_i)$ ($1 \leq \varepsilon < n(c_i)$)则固定活动 i 对应的基因座自左向右第 $\varepsilon + 1$ 个的值为“1”,其它 $n(c_i) - 1$ 个基因座的值为“0”,这样就随机生成了一个该活动的附加费用模式。其它活动依此类推。如此重复,可以产生多个初始解 $\theta_1^1, \theta_1^2, \dots, \theta_1^N$ 。

(2)排序操作。利用解码技术,计算每个解对应的目标函数值的大小。假设 $\pi_k(i)$ 为 i 种群中的第 k 个个体,则将它们按从小到大的自左向右排列:

$$\hat{f}(\pi_1(i)) \geq \hat{f}(\pi_2(i)) \geq \dots \geq \hat{f}(\pi_N(i))$$

(3)交叉算子。A 区的交叉算子采用 Syswerda^[9]提出的基于位置的交叉算子,这种方法随机地从一个父代中提取一些基因,子代中余下的基因通过另一个父代进行从左到右的扫描并填充空位得到;B 区的交叉算子采用单点交叉的方法。即任意产生一个活动 i ,交换两个父代中该活动对应的 $n(c_i)$ 个基因座的值。

(4)变异算子。A 区的变异算子采用交换变异,及随机选择两个基因座,将它们的值进行交换;B 区的变异算子采用任意产生一个活动 i ,在其对应 $n(c_i)$ 个基因座中任意交换两个基因座的值。

(5)选择概率,交叉概率和变异概率。采用自适应方法,根据个体的排序值自适应确定,具体方法如下:

选择概率 $p_{sel}(\pi_k(i)) = \frac{1}{N} + \alpha \frac{N+1-2k}{N(N+1)}$, $k=(1, 2, \dots, N)$;交叉概率 $p_{cro}(\pi_k(i)) = \lambda_c(1 - e^{-\beta_c \frac{k-1}{N}})$;

变异概率 $p_{mut}(\pi_k(i)) = \lambda_m(1 - e^{-\beta_m \frac{k-1}{N}})$ 。其中, α 为选择算子的自适应系数, $0 < \alpha < 1$ 。 $\lambda_c, \lambda_m, \beta_c, \beta_m$ 均为常数,其中 $0 < \lambda_c \leq 1, 0 < \lambda_m \leq 1$ 。

4 算例

图 3 为某型号研制过程的原始邻接矩阵。这是一个由 8 个原始活动组成的研制过程,我们对数据进行了一定的处理,活动名称用代号 A, B, C, D, E, F, G, H 表示,但这并不影响演示的效果。

已知 $S = \{s_{BD} = 1, s_{FH} = 2, s_{GH} = 2\}$,活动 D 被提前,活动 B 是其上游活动, $O_{BD} = 0.5, r_{BD} = 0.1$ 。项目不可更新资源无限,可更新资源的种类为 5,总量分别为 30, 30, 40, 40, 50。合同规定的工期 80 天,提前或超期的奖惩系数为 0.5,模型的两个参数 ξ_1, ξ_2 分别为 0.6 和 0.4。自适应遗传算法中的参数值 $\alpha = 0.4, N = 50, \lambda_c = 1, \lambda_m = 0.2, \beta_c = 5, \beta_m = 10$ 。求性能值时仿真次数为 100,算法运行 500 代终止,每代种群数为 20。

其它已知数据如表 1 所示。

	A	B	C	D	E	F	G	H
A	*	0	0	0	0	0	0	0
B	0	*	0	0	0	0	0	0
C	1	0	*	0	0	0	0	0
D	0	1	0	*	0	0	0	0
E	0	0	1	1	*	0	0	0
F	0	0	1	0	1	*	0	0
G	0	0	0	0	1	0	*	0
H	0	0	0	0	0	1	1	*

图 3 某型号研制活动的 DSM
Fig.3 DSM of a model project

表 1 某型号研制项目原始数据表
Tab.1 Original data of the model project

活动 ID	消耗的资源种类及数量					原始执行时间分布律	学习 参数	重做次数分布律					
	1	2	3	4	5			0	1	2	3	4	
A	10		15		20	$N(15, 3^2)$							
B	10	20				$N(10, 5^2)$	0.8						
D			15	15	10	$N(26, 5^2)$	0.8	0.1	0.3	0.3	0.2	0.1	
C	5			15		$N(8, 2^2)$							
E		20		20		$N(12, 3^2)$							
F	15					$N(18, 4^2)$	0.6						
G	8		8		6	$N(10, 5^2)$	0.6	0.6	0.4	0	0	0	
H		12	12			$N(5, 1^2)$	0.6						

表 2 某型号研制项目各活动的可压缩模式
Tab.2 Compression modes of activities in the model project

活动 ID	压缩模式(附加费用额度 ,压缩后时间占原时间的比例)				
	1	2	3	4	5
A	(0, 1)	(2, 0.9)	(5, 0.72)		
B	(0, 1)				
C	(0, 1)	(2, 0.88)			
D	(0, 1)	(3, 0.8)	(4, 0.7)		
E	(0, 1)	(0.5, 0.95)	(1, 0.9)	(1.2, 0.85)	(2, 0.82)
F	(0, 1)	(1, 0.9)			
G	(0, 1)				
H	(0, 1)	(0.3, 0.88)			

算法得到的最优编码为：

6	1	8	2	3	4	7	5	1	0	0	1	1	0	0	0	1	0	0	0	1	0	0	1	1	0	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

解码得活动 A , B , C , D , E , F , G , H 的优先权系数分别为 6, 1, 8, 2, 3, 4, 7, 5。各活动采用的时间压缩模式如表 2 中阴影部分所示。工期风险为 1.75 ;各开工时间期望分别为 0, 0, 14.7588, 4.7332, 48.5839, 58.3149, 58.3149, 74.6033 ;附加费用值为 6.5。如果附加费用值为 0 ,解的编码只为 A 区 ,则经过计算得到最佳调度方案为 A , B , C , D , E , F , G , H 的优先权系数分别等于 1, 2, 7, 3, 6, 4, 5, 8 ,工期风险为 11.63。可以看出 ,根据本文模型的目标函数 ,附加费用是值得的。图 4、图 5 给出了优化结果下各原始活动开工时间与总工期仿真 100 次后的分布图。

由开工时间结果可以看出 ,活动 A , B 同时开工 ,而表 1 中两个活动需要的资源量之和确实不超过项目的资源总量 ,可以同时执行。A , B , D 三个活动需要的资源量之和也不超过项目的资源总量。由于活动 B 的执行时间的期望值是 10 ,而活动 D 的平均开工时间为 4.7 左右 ,因此活动 D 确实被提前安排了 ,而且活动 D 开工时 ,活动 A 仍在继续 ,活动 B 基本执行到一半 ,与实际要求一致。活动 D 的执行时间较长 ,这决定了给活动 A 或 C 附加费用只会导致费用的增加 ,而不会减小工期的风险 ,而本文的解中活动 A 和 C 确实没有附加任何费用。活动 D 的开工时间大致为 4.7 左右 ,其紧后活动 E 的开工时间大致为 48.5 左右 ,而 D 的执行时间在压缩后的期望为 18.3 左右 ,说明活动 D 确实发生了重做。

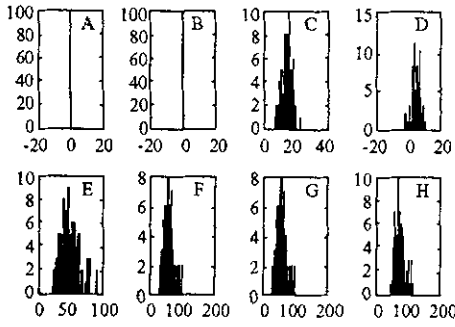


图4 各活动开工时间分布

Fig.4 Distribution of beginning time of each activity

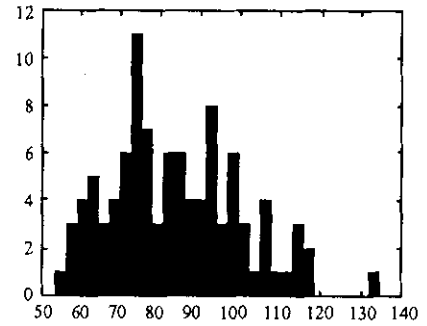


图5 总工期分布

Fig.5 Distribution of project duration

5 结论

本文的型号项目工期风险管理决策模型充分考虑了型号项目中活动重叠、活动迭代和活动执行时间的不确定性等特点,并且考虑了可更新资源约束对工期风险的影响。提出的基于自适应遗传算法的仿真优化方法针对了模型的特点,算例显示,该算法能够较好地求解工期风险管理决策问题。

参考文献:

- [1] Ginzburg D G, Gonik A. A Heuristic for Network Project Scheduling with Random Activity Durations Depending on the Resource Allocation[J]. International Journal of Production Economics, 1998, 55: 149 - 162.
- [2] Tereso A P, Araujo M M T, Elmaghraby S E. Adaptive Resource Allocation in Multimodal Activity Networks[J]. International Journal of Production Economics, 2004, 92: 1 - 10.
- [3] Tsai Y W, Gemmill D D. Using Tabu Search to Schedule Activities of Stochastic Resource-constrained Projects[J]. European Journal of Operational Research, 1998, 111: 129 - 141.
- [4] 李文华, 谭燕秋. 不确定性网络直接优化规划模型建立[J]. 系统工程理论与实践, 2000 (12): 97 - 101.
- [5] Abbasi G Y, Mukattash A M. Crashing PERT Networks Using Mathematical Programming[J]. International Journal of Project Management, 2001 (19): 181 - 188.
- [6] 王凌, 张亮, 郑大钟. 仿真优化研究进展[J]. 控制与决策, 2003, 18(3): 257 - 262.
- [7] Boesel J, Bowden J R, Glover F, et al. Future of simulation optimization[A]. Proceedings of 2001 Winter Simulation Conference[C], Arlington, USA, 2001: 1466 - 1469.
- [8] Bromberg M F. Modeling Design Rework in a Product Development Process[D]. Master's Thesis, MIT, 2000.
- [9] Davis L. Handbook of Genetic Algorithms[M]. Van Nostrand Reinhold, New York, 1991: 332 - 349.

