

# 一种求解 Job Shop 调度问题的改进遗传算法<sup>①</sup>

沈镇静<sup>1,2</sup>, 郑 湃<sup>2,3</sup>, 李家霖<sup>2,3</sup>

<sup>1</sup>(中国科学院研究生院, 北京 100049)

<sup>2</sup>(中国科学院 沈阳计算技术研究所, 沈阳 110171)

<sup>3</sup>(常州数控技术研究所, 常州 213164)

**摘 要:** 传统遗传算法在求解 Job Shop 调度问题时存在收敛速度慢, 易于早熟的缺点。在病毒遗传算法(VEGA)和灾变遗传算法的基础上提出了一种带有灾变因子的病毒遗传算法(IVEGA-C)。该算法在传统遗传算法的基本结构上加入了病毒感染操作和灾变操作, 病毒感染操作实现了同代个体之间横向传递进化信息, 灾变操作采用灭绝操作。正是这种改进加快了遗传算法的收敛速度, 避免了早熟现象和陷入局部最优解。通过仿真实验验证了 IVEGA-C 算法在解决 Job Shop 调度问题中的性能优于传统 GA 算法和 VEGA 算法。最后给出了应用该算法的一个实例。

**关键词:** Job Shop 调度问题; 病毒遗传算法; 灾变算子; 收敛性

## Improved Genetic Algorithm for Job Shop Scheduling Problems

SHEN Zhen-Jing<sup>1,2</sup>, ZHENG Pai<sup>2,3</sup>, LI Jia-Ji<sup>2,3</sup>

<sup>1</sup>(Graduate University, Chinese Academy of Sciences, Beijing 100049, China)

<sup>2</sup>(Shenyang Institute of Computing Technology, Chinese Academy of Sciences, Shenyang 110168, China)

<sup>3</sup>(Changzhou Institute of Numerical Control Technology, Changzhou 213164, China)

**Abstract:** Traditional Genetic Algorithm for solving Job Shop Scheduling Problems has some shortcomings such as slow convergence and easy to bring immature convergence. On the basis of Virus Evolutionary Genetic Algorithm (VEGA) and Genetic Algorithm with Catastrophe factor, an improved Virus Evolutionary Genetic Algorithm with Catastrophe factor (IVEGA-C) was proposed. IVEGA-C adds virus infection operation and catastrophe operation to the basic structure of traditional Genetic Algorithm. Virus infection operation passes the evolutionary information between the populations in the same generation and an improved extinction operation was used as the strategy of catastrophe. The improved algorithm speeded up the convergence rate of the Genetic algorithm, avoided the premature phenomena and to fall into local optimal scheduling solution. The simulation results verify that IVEGA-C on solving the Job Shop Scheduling Problems is better than traditional Genetic Algorithm and VEGA. At last we give an example of using this algorithm to solve scheduling problems in our real-world.

**Key words:** Job Shop scheduling; virus evolutionary genetic algorithm; cataclysm operator; convergence

Job Shop 调度问题是对实际生产过程中复杂调度问题的简化, 是一个典型的 NP 困难问题, 因此对于此类问题的研究具有很高的理论和实际应用价值。求解 Job Shop 的主要方法有:①确定性最优化算法, 如基于枚举的分支定界法、线性规划法; ②智能优化算法,

如禁忌搜索算法、模拟退火算法、蚁群算法、遗传算法。遗传算法由于其具有良好的全局搜索性能和隐并行性, 受到研究人员的广泛关注。郝东等提出了一种利用增强时间 Petri 网的激发序列作为遗传算法染色体的方法<sup>[1]</sup>, 顾擎明等讨论了遗传算法的参数值优化问

① 基金项目:江苏省科技攻关项目(BE2010070)

收稿时间:2011-11-18;收到修改稿时间:2012-03-30

题,提出了一种自适应遗传算法<sup>[2]</sup>,李书全等提出了一种通过改变交叉和变异率来防止遗传算法成熟前收敛的有效算法<sup>[3]</sup>。将遗传算法应用于求解 Job Shop 的主要问题是容易出现成熟前收敛现象,本文提出了一种带有灾变因子的病毒遗传算法,这一算法能很好的解决成熟前收敛问题。

Job Shop 调度问题可以描述为 n 个工件在 m 台机器上加工,每个工件有 m 道工序(每台机器加工一道工序),给定每个工件使用机器的顺序和每道工序所花费的时间,并满足下面的约束:①每台机器一次只能加工一个工件;②一个工件不能两次使用同一台机器;③工序一旦开始不能中断;④不同工件之间没有先后约束<sup>[4]</sup>。我们以最小完工时间为调度目标,即:

$$\text{Min}F(x) = \text{Max}(c_k) \quad k = 1, 2, \dots, n$$

$c_k$  为工件 k 的加工完成时间。

遗传算法是一种模拟自然进化过程搜索最优解的方法,主要包括编码、适应度函数、遗传算子,遗传算法的运行参数四部分。编码是指在目标问题实际表示与遗传算法的染色体结构之间建立联系,适应度函数即确定评价函数决定当前群体中个体遗传到下一代群体中的机会大小,遗传算子包括选择、交叉和变异操作,遗传算法运行参数主要有 M: 群体大小、T: 遗传算法迭代代数、 $p_c$ : 交叉概率、 $p_m$ : 变异概率。

本文采用基于工序的编码方式来表示 Job Shop 调度问题的解。一个 3 个工件,3 台机器调度问题的工艺约束如表 1 所示,其中  $J_n$  表示工件 n,  $M_k$  表示机器 k,染色体中的数字 n 对应了工件  $J_n$ 。例如染色体“311223213”中的 1 表示工件 J1, 2 表示工件 J2, 3 表示工件 J3。每个工件有 3 道工序,所以每个工件在染色体中出现 3 次,对应表 1 中工件 J1 这一行可知染色体中出现的第一个 1 表示工件 1 的第一道工序在机器 1 上加工,出现的第二个 1 表示工件 1 的第二道工序在机器 2 上加工。

表 1 3×3Job Shop 调度的工艺约束

工件	工序 1	工序 2	工序 3
J <sub>1</sub>	M <sub>1</sub>	M <sub>2</sub>	M <sub>3</sub>
J <sub>2</sub>	M <sub>1</sub>	M <sub>3</sub>	M <sub>2</sub>
J <sub>3</sub>	M <sub>2</sub>	M <sub>1</sub>	M <sub>3</sub>

## 1 病毒遗传算法

1996 年, Kubota 提出了基于病毒的遗传算法 VEGA(Virus Evolutionary Genetic Algorithm),该算法在遗传算法的基础上引入感染因子,来模拟自然界中病毒个体感染主个体的过程。VEGA 中主群体的遗传操作,实为在上下代群体之间纵向传递进化信息,实施解空间的全局搜索;病毒感染操作,实为在同代个体之间横向传递进化信息,实施解空间的局部搜索。

### 1.1 病毒的编码和适应度函数

病毒编码与主群体编码等长,病毒编码中的无效基因用通配符“\*”表示,例如 3×3Job Shop 调度问题中一个可能的病毒染色体编码为“1\*3\*2\*\*23”。病毒个体  $V_i$  的适应度函数为:

$$\text{Fitness}V_i = \sum_{j \in H} \text{fitnessvirus}_{i,j}$$

$$\text{fitnessvirus}_{i,j} = \text{fitnessshost}'_j - \text{fitnessshost}_j$$

其中 H 表示被病毒  $V_i$  感染的主群体集合,  $\text{fitnessshost}_j$  表示主染色体 j 的适应度函数值,其计算公式如下,其中  $C_j$  为由主染色体 j 计算出的最小完工时间。

$\text{fitnessshost}_j$  和  $\text{fitnessshost}'_j$  分别表示第 j 个主染色体被病毒感染前后适应度函数的值,  $\text{fitnessvirus}_{i,j}$  表示第 j 个主染色体被感染前后适应度函数变化值。 $\text{Fitness}V_i$  取正值时病毒的感染力增加。

### 1.2 病毒的生命力

病毒  $V_i$  在 t+1 代的生命力为:

$$\text{life}_{v_i,t+1} = p * \text{life}_{v_i,t} + q * \text{Fitness}V_i$$

其中 p 为衰减系数,  $0 < p < 1$ , q 为  $V_i$  的适应度函数的加权系数,  $0 < q < 1$ , 当  $\text{Fitness}V_i > 0$  时  $V_i$  的生命力有所加强, 当  $\text{life}_{v_i,t+1} < 0$  时该病毒死亡, 需要产生新的病毒<sup>[5]</sup>。

### 1.3 病毒感染操作

病毒个体的感染操作包括侵入和逆取。侵入的目的是产生新的主群体个体, 逆取的目的是产生新的病毒个体<sup>[6]</sup>。

#### 1.3.1 侵入

病毒个体以概率  $p_{\text{infect}}$  用自身基因覆盖主染色体中相应的基因片段, 产生新的主染色体。如图 1 所示主染色体“112323123”, 病毒染色体“2\*12\*\*\*3\*”, 感染后得到新的主染色体“211223133”。感染过程有可能产生非法染色体, 因此需要验证感染后染色体的

有效性。

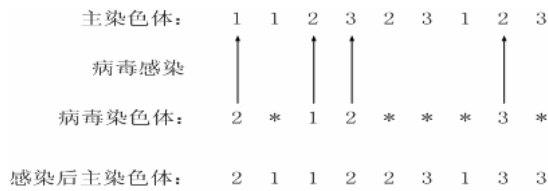


图 1 病毒感染操作

### 1.3.2 逆取

逆取操作在一个病毒死亡时发生,是产生新的病毒个体的过程,有复制和删减两种方式。

① 复制:随机选择一个主染色体,主染色体各位基因以概率  $p_{copy}$  代替病毒个体中的基因以产生新的病毒个体,如图 2 所示主染色体“112323123”,病毒染色体“2\*12\*\*\*3\*”,主染色体的 1,4 位被选中替换病毒染色体中相应的基因,得到新的病毒染色体“1\*13\*\*\*3\*”,在此过程中也有可能产生非法染色体,因此也需要验证新病毒染色体的有效性。

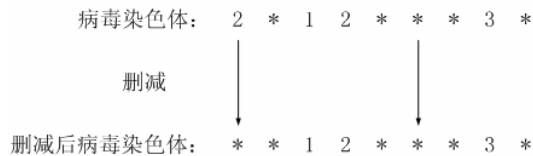


图 2 病毒复制操作

② 删减:病毒个体每个基因位以概率  $p_{cut}$  被通配符‘\*’替换,如图 3 所示病毒染色体为“2\*12\*\*\*3\*”,选中的删减位为 1,6,得到新的病毒染色体为“\*\*12\*\*\*3\*”。

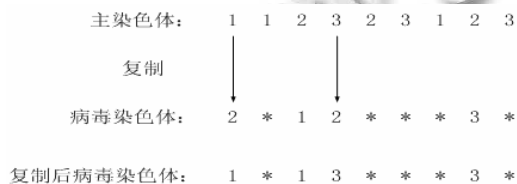


图 3 病毒删减操作

## 2 灾变遗传算法

灾变是自然界中处理进化停滞问题的一种有效方法,每一次灾变都带来了生物进化的巨大飞跃。灾变操作除了可以打破非最优解染色体的垄断地位,还可以产生新的个体,增加了染色体的多样性。灾变遗传算法中

采用的灾变操作主要有两种:一种是对整个种群进行灭绝,随机产生一些新的个体,还有一种是大幅提高遗传算法的变异率。灾变遗传算法的关键是制定合理的灾变条件以及合适的灾变条件检测时间间隔。

### 3 灾变-病毒遗传算法

病毒遗传算法应用于求解 Job Shop 调度问题取得较好的效果,其缺点是没有办法及时检测是否已经陷入局部最优解,灾变遗传算法求解 Job Shop 调度问题时效果不是很好而且计算灾变条件需要花费大量时间,但它可以及时检测是否已经陷入局部最优解。因此我们提出了一种带有灾变因子的改进病毒遗传算法 IVEGA-C(Improved Virus Evolutionary Genetic Algorithm with Catastrophe factor)。该算法保留了传统遗传算法的基本结构,以病毒遗传算法为主进行解空间的搜索,灾变算子为辅检测是否陷入局部最优解。

#### 3.1 算法设计

IVEGA-C 的编码采用上文提到的基于工序的编码,初始化种群中五分之一的染色体由剩余工作时间最多(MWR,一种工件优先排列规则)生成的一个初始解通过交叉变异产生,其余的染色体随机产生<sup>[7]</sup>。

主群体中的适应度函数为  $f(x)=1/C_{max}$ ,其中  $C_{max}$  为由该染色体对应加工序列计算出的最早完工时间。选择操作采用轮盘赌,交叉操作采用 POX 交叉算子<sup>[8]</sup>,该算子能确保产生合法的子代染色体。假设父代染色体为 parent1 和 parent2,子代染色体为 child1 和 child2。该算法具体步骤如下:

- ① 将工件集(1,2...n)随机划分为 J1、J2 两个子集;
- ② 复制 parent1、parent2 包含在 J1 中的工件到 child1 和 child2 中的相同位置;
- ③ 复制 parent2 包含在 J2 中的工件到 child1, parent1 包含在 J2 中的工件到 child2,并保持顺序不变。

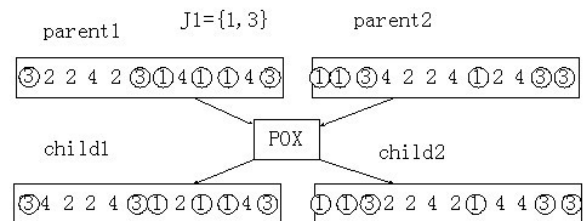


图 4 POX 交叉算子实例

图 4 是一个 4×3 调度问题的 POX 交叉实例,其

中  $J1=\{1,3\}, J2=\{2,4\}$ 。

变异操作采用随机交换两个不同基因位置的方法, 变异概率  $p_m$  是自适应的, 当个体适应度  $f$  小于种群平均适应度时, 个体变异概率  $p_m$  变大, 具体计算公式如下:

$$p_m = \begin{cases} m_1 \frac{f_{Max} - f}{f_{Max} - f_{Agv}} & f \geq f_{Agv} \\ m_2 & f < f_{Agv} \end{cases}$$

$f_{Max}$  为主群体中个体的最大适应度,  $f_{Agv}$  为主群体中个体的平均适应度,  $f$  为当前个体的适应度。  $m_1, m_2$  为常数且  $m_2 > m_1$ 。

IVEGA-C 中的病毒感染操作采用上文提出的方法, 灾变算子采用改进的灭绝操作。设  $C_{ik}$  和  $C_{jk}$  分别为染色体  $i, j$  上的第  $k$  位的基因, 则两染色体的欧式距离为:

$$d_{ij} = \sqrt{\sum_{k=1}^N (C_{ik} - C_{jk})^2}$$

两染色体的相似度为:

$$S_{ij} = \frac{1}{1 + d_{ij}}$$

灾变条件有两个并需要同时满足:

- ① 主群体中相似度达到灾变标准  $S$  的染色体对数超过一定比例。
- ② 主群体中的最优解在整个群体中所占的比例极小。

在灾变后产生的新主群体中, 灭绝前的最优解及其变异所产生的染色体数不超过四分之一, 其余染色体随机产生。因为病毒遗传算法本身就不易陷入局部最优解而且计算染色体的相似度很费时间, 因此我们采用每迭代 20 轮进行一次灾变操作的方法。

### 3.2 算法流程

IVEGA-C 算法可分为传统 GA 操作、病毒感染操作、灾变操作这三部分, 参照流程图 5 具体步骤如下:

步骤 1: 初始化主群体、病毒群体以及各种系统参数;

步骤 2: 判断是否满足算法终止规则, 若满足则算法停止并通过解码算法得到最优解, 否则转步骤 3;

步骤 3: 对  $t$  代种群数为  $N$  的主群体进行交叉操

作, 产生  $N$  个个体的  $cross(t+1)$ , 对  $cross(t+1)$  进行变异操作, 产生  $N$  个个体的  $mut(t+1)$ , 将  $host(t) \cup cross(t+1) \cup mut(t+1)$  中个体按适应度值由大到小排列, 取前  $N$  个个体, 得到  $host(t+1)$ ;

步骤 4: 病毒群体  $virus(t)$  中的个体以概率  $p_{infect}$  对  $host(t+1)$  中的每个个体实施侵入操作, 感染后的子群体集合为  $U$ , 如果  $U$  中个体  $k$  的适应度值在感染后变大了则将  $k$  替代感染前的个体;

步骤 5: 计算病毒  $V_i$  的适应度函数  $FitnessV_i$ , 若  $FitnessV_i > 0$ , 则在  $U$  中随机选择一个主个体以概率  $p_{copy}$  对病毒  $V_i$  实施复制操作, 否则以概率  $p_{cut}$  实施删减操作。将新产生的病毒  $V_i$  加入  $virus(t+1)$ ;

步骤 6: 计算病毒  $V_i$  的生命力  $life_{vi,t+1}$ , 若  $life_{vi,t+1} < 0$ , 则从  $host(t+1)$  中随机选择选择一个主染色体以概率  $p_{copy}$  对病毒  $V_i$  实施复制操作, 将新产生的病毒加入  $virus(t+1)$  替换病毒  $V_i$ , 并令  $life_{vi,t+1} = 0$ ;

步骤 7: 判断迭代次数  $t$  是否是 20 的整数倍, 如果不是跳转到步骤 9;

步骤 8: 计算主群体中染色体对的相似度  $S_{ij}$ , 若相似度达到灾变标准  $S$  的染色体对数超过  $cata_{min}$  并且最优解所占比例小于  $best_{max}$  则采取灭绝操作产生新的主群体;

步骤 9:  $t=t+1$ , 转到步骤 2。

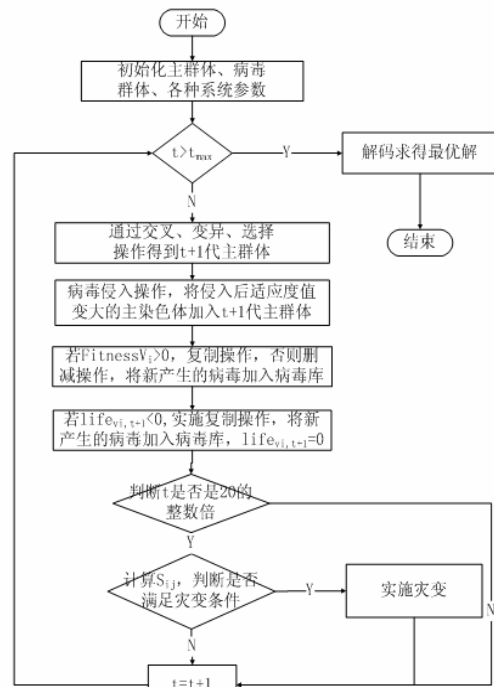


图 5 IVEGA-C 的流程图

### 4 标准算例实验与分析

为了验证算法的有效性,将传统 GA 算法、VEGA 算法和 IVEGA-C 分别应用于部分车间调度标准实例集进行对比。

#### 4.1 实验环境

实验运行环境:硬件为 2.5GHz CPU 和 2G 内存、操作系统 XP、开发工具 VC6.0。

IVEGA-C 中的参数设置为:最大迭代代数  $t_{max}=400$ ,主群体个数  $N=50$ ,病毒群体个数  $M=20$ ,交叉概率  $p_c=0.85$ ,自适应变异概率计算公式中的  $m_1=0.1$ ,  $m_2=0.2$ ,感染概率  $p_{infect}=0.1$ ,复制概率  $p_{copy}=0.2$ ,删减概率  $p_{cut}=0.1$ ,生命衰减系数  $p=0.8$ ,病毒适应度函数加权系数  $q=0.5$ ,灾变标准相似度  $S$  为:

$$S = \frac{1}{1 + \sqrt{0.1L}}$$

$L$  为染色体的长度,达到灾变标准染色体对的最

小比例  $cata_{min}=0.75$ ,最优解所占最大比例  $best_{max}=0.1$ 。

#### 4.2 实验结果与分析

每个算法对于标准实例运行 10 次,其结果如表 2 所示:

表 2 中, best 表示的是算法运行 10 次找到的最优解, Avg 表示 10 次求解的平均值, RD 表示平均解相对于最优解的偏差率,计算公式为:

$$RD = \frac{C_{Avg} - C^*}{C^*} \times 100\%$$

其中,  $C^*$  表示该问题的最优解,通过分析可知:

① IVEGA-C 算法无论是最优解还是平均解都要好于另外两个算法。

在问题规模较小时 IVEGA-C 搜索出的最优解效果较好,平均解可能会有较大偏差(例如 LA21)。

② 当问题规模较大时(LA36, LA39),平均解的偏差率变大,而且求解时间明显大于 GA 和 VEGA。

表 2 相关算法比较分析

标准实例	最优解	GA (best)	GA (Avg)	GA (RD)	VEGA (best)	VEGA (Avg)	VEGA (RD)	IVEGA-C (best)	IVEGA-C (Avg)	IVEGA-C (RD)
FT06	55	55	57.4	4.3	55	57.5	4.5	55	55.8	1.5
LA01	666	666	688.6	3.4	666	676.3	1.5	666	669.4	0.5
LA02	655	675	691.2	5.5	660	667.8	1.9	655	662.7	1.1
FT10	930	967	1021.6	9.8	940	954.2	2.6	930	948.5	2.0
LA21	1046	1105	1174.2	12.2	1053	1118.4	6.9	1048	1109.1	6.0
LA24	935	997	1075.0	15.0	942	977.4	4.5	942	960.9	2.8
LA36	1268	1305	1397.8	10.2	1293	1331.2	5.0	1278	1321.4	4.2
LA39	1233	1324	1362.9	10.5	1278	1315.1	6.7	1235	1297.0	5.2

### 5 工程实例

常州数控所柔性制造实验室目前有 3 台机床(H63 卧式加工中心 1 台,车床 1 台,镗铣一台),1 个 3 层的立体仓库,2 个出入库操作平台(用于自动上下料控制),和 1 台自动堆垛机。现有一批 3 个工件的加工任务,其在每台机器上的加工时间和工艺约束如表 3 所示。

为了简化分析,只考率工件在机床上的加工时间,采用基于工序的编码,按照上述 IVEGA-C 步骤求得最早完工时间为 110,对应的染色体为“231231321”。机器的加工时间如图 6 所示。

表 3 一个 3 工件 3 机器的 JSP 加工数据

项目	工件	操作序号		
		1	2	3
操作时间	J1	30	30	20
	J2	10	50	30
	J3	30	20	30
机器顺序	J1	M1	M2	M3
	J2	M1	M3	M2
	J3	M2	M1	M3

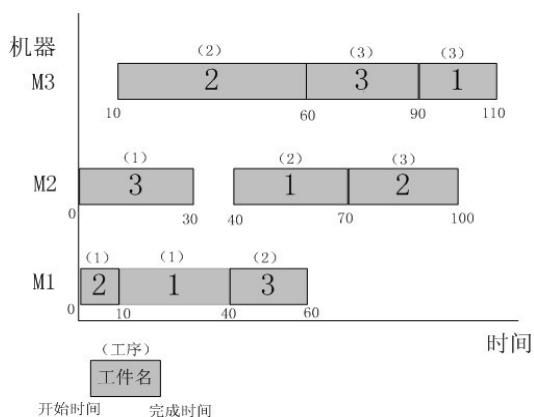


图6 机器加工 Gantt 图

## 6 结论

本文在简要分析病毒遗传算法和灾变遗传算法的基础上提出了一种带有灾变因子的改进病毒遗传算法,用来求解 Job Shop 调度问题,给出了算法的具体步骤,并通过标准算例和工程实例证明了算法的有效性。下一步研究的重点是考虑生产中的实际问题(如机器故障,交货日期变更,紧急工件等),采用动态调

度策略,结合本文提出的算法(用作静态调度策略),应用于求解更复杂的生产调度问题。

## 参考文献

- 1 郝东,蒋昌俊,林琳.基于 Petri 网与 GA 算法的 FMS 调度优化.计算机学报,2008,28(2):201-208.
- 2 顾黎明,宋文忠,曹丽娟.基于遗传的一种 Job-shop 调度方法.东南大学学报,1997,27(5):369-374.
- 3 李书全,赵良英,史克禄,等.一种防止遗传算法成熟前收敛的有效算法.系统工程理论与实践,1999,19(5):72-77.
- 4 洪刘兵,杨艳丽.求解车间调度的问题的一种改进遗传算法.机床与液压,2010,38(5):101-103.
- 5 曹先彬,王本年,王煦法.一种病毒进化型遗传算法.小型微型计算机系统,2001,22(1):59-62.
- 6 刘胜辉,张淑丽,王波,等.单车间调度问题的改进型病毒进化遗传算法.电机与控制学报,2008,12(2):234-238.
- 7 王凌.车间调度及其遗传算法.北京:清华大学出版社,2003.
- 8 尹作海,邱洪泽,周万里.基于改进变异算子的遗传算法求解柔性作业车间调度.计算机系统应用,2009,18(10):156-159.

(上接第 47 页)

- for ERP vendors. Int'l J. Production Economics,2005,97: 279-295.
- 2 Kumar MNV, Suresh AV, Prashanth P. Analyzing the Quality Issues in ERP Implementation:A Case Study. IEEE Emerging Trends in Engineering and Technology,2009 (12):759-764.
  - 3 Ge LP, Voß S. ERP application in China:An overview. Int'l J. Production Economics,2009,122:501-507.
  - 4 李益兵,郭顺生,赵春阳.ERP 中质量管理系统的研究与开发.机械制造,2007(3),45(511):62-64.
  - 5 Barjis J. The importance of business process modeling in software systems design. Science of Computer Programming,2008,71:73-87.
  - 6 Antunes P, Mourão H. Resilient Business Process Management:Framework and services. Expert Systems with Applications,2011,38:1241-1254.

- 7 刘林,罗中先,戴跃洪.基于 ERP 的质量管理系统研究.系统管理,2009,11:55-58.
- 8 Trkman P. The critical success factors of business process management. International Journal of Information Management,2010,30:125-134.
- 9 高伟坤,祝金会,戚银城,杨超.业务流程管理在 ERP 中的实现.河北电力技术,2010,29(4):24-26.
- 10 Glykas MM. Effort Based Performance Measurement in Business Process Management. Knowledge and Process Management,2011,18(1):10-33.
- 11 方江雄,何红波,李义兵.基于 SOA 和 BPM 的 ERP 系统的研究与实现.计算机应用,2007,27(5):1264-1267.
- 12 韩亚利,李长林.质量管理信息系统和 ERP 系统的集成设计与实现.中国制造业信息化,2009,38(7):1-4.