文章编号: 1001-0920 (2011) 01-0044-05

基于信息熵和混沌理论的遗传-蚁群协同优化算法

薛 锋, 王慈光, 牟 峰
(西南交通大学 交通运输学院，成都 610031)

摘 要: 为了融合遗传算法和蚁群算法在解决组合优化问题方面的优势, 提出一种基于信息熵和混沌理论的遗传-蚁群协同优化算法. 利用信息熵产生初始群体, 增加初始群体的多样性, 并将混沌优化的遍历特性引入融合的遗传-蚁群算法, 改进相关参数, 实现参数的自适应控制以及遗传算法与蚁群算法混合优化策略的有机集成. 通过仿真实例表明了混合智能算法在解决旅行商问题 (TSP) 50 座城市最短路径寻优时的有效性.

关键词: 信息熵；混沌映射函数；遗传算法；蚁群算法；协同优化
中图分类号: TP18　　**文献标识码:** A

Genetic and ant colony collaborative optimization algorithm based on information entropy and chaos theory

XUE Feng, *WANG Ci-guang*, *MU Feng*
(College of Traffic and Transportation，Southwest Jiaotong University，Chengdu 610031，China．Correspondent：XUE Feng，E-mail：xuefeng.7@163.com)

Abstract: In order to merge the advantage of genetic algorithm and ant colony algorithm in solving combinatorial optimization problem, a kind of genetic and ant colony collaborative optimization algorithm based on information entropy and chaos theory is proposed. This algorithm produces the initial colony by information entropy to increase the variety of the initial colony, and introduces the traversal characteristic of chaos optimization to the integrated genetic and ant colony algorithm. Some relevant parameters in the algorithm are improved, the adaptive control of the parameters is realized, and the hybrid optimization strategy of genetic algorithm and ant colony algorithm is integrated. Simulation example shows that this hybrid intelligent algorithm is valid in solving traveling salesperson problem of 50 cities.

Key words: information entropy；chaos mapping function；genetic algorithm；ant colony algorithm；collaborative optimization

1 引 言

各种组合优化问题均有其特殊性和复杂性, 在解决这些问题时, 每一种智能优化算法都表现出自身的优势和缺陷, 多种算法进行融合、互补、协同工作以建立混合型算法是目前智能算法的发展方向之一. 在诸多混合智能算法中, 将遗传算法和蚁群算法相结合是其中的一个研究热点. 遗传算法和蚁群算法作为两种模拟生物进化的最优化智能算法, 二者都具有本质并行性、自组织性和较强的鲁棒性, 均不依赖于优化问题本身的严格数学性质, 均依靠群体的能力来搜索最优解. 遗传算法依赖于种群之间的交叉、变异、选择操作进化获得最优解; 蚁群算法则依赖群体的协同通讯寻找最优解. 遗传算法与蚁群算法之间这些类似的特性为二者的融合奠定了良好的基础.

本文以传统的遗传算法和蚁群算法为依托, 将信息熵和混沌优化的遍历特性引入融合的遗传-蚁群算法, 用于初始可行解的产生或作为特殊的算子改进相关系数, 实现参数的自适应控制以及遗传算法与蚁群算法混合优化策略的有机集成. 提出一种基于信息熵和混沌理论的参数自适应遗传-蚁群协同优化算法 (ECGACO), 以充分发挥算法寻优的整体功能, 使之既能发挥遗传算法与蚁群算法在寻优搜索中各自的优势, 又能克服遗传算法在搜索到一定阶段时最优解搜索效率低以及蚁群算法初始信息素匮乏等不足.

收稿日期: 2009-10-30；**修回日期:** 2010-01-20.
基金项目: 国家自然科学基金项目(60776824)；中央高校基本科研业务费专项资金项目(SWJTU09BR134)；西南交通大学青年教师科研起步项目(2009Q040).
作者简介: 薛锋(1981–), 男, 讲师, 博士, 从事交通运输规划理论与系统优化的研究；王慈光(1946-), 男, 教授, 博士生导师, 从事运输组织理论与系统优化等研究.

2　信息熵和混沌映射函数

对于离散型随机变量, 其信息熵[1]为

$$S = -k\sum_{i=1}^{n}(p_i \ln p_i), \tag{1}$$

其中 p_i 表示各状态发生的概率, $p_i \geqslant 0$ 且 $\sum_{i=1}^{n} p_i = 1$.

混沌优化是以混沌原理为基础的一种新型直接搜索优化算法. Logistic 映射函数是其中较为常用的混沌迭代方程, 但其迭代序列的分布并不均匀. Tent 映射[2]函数 (帐篷映射) 的时间序列具有均匀的概率密度和功率谱密度等特性, 在遍历的均匀性方面优于 Logistic 函数, 其表达式为

$$z_{n+1} = \alpha - 1 - \alpha|z_n|,\ 1 < \alpha \leqslant 2. \tag{2}$$

当 $\alpha = 2$ 时, 此时的映射称为中心 Tent 映射, 其表达式为

$$z_{n+1} = 1 - 2|z_n|,\ -1 \leqslant z_n \leqslant 1. \tag{3}$$

与 Logistic 映射函数产生的序列主要分布在 0, 1 点附近相比, Tent 映射函数产生的混沌序列比较均匀地分布在 [0,1] 之间, 具有更好的遍历均匀性, 迭代速度也快于 Logistic 映射.

3　ECGACO 算法设计

3.1　初始群体的产生方法

考虑组合优化问题的特殊性和约束条件, 采用实数编码, 利用信息熵产生初始群体[3], 可使初始群体在解空间中均匀分布, 避免集中分布在解空间的局部区域, 增加初始群体的多样性.

设初始群体由编码长度为 L 的 N 个染色体组成, 群体中第 j 个基因的熵值 S_j 定义为

$$S_j = \sum_{i=1}^{N}\sum_{k=i+1}^{N}(-p_{ik}\ln p_{ik}), \tag{4}$$

$$p_{ij} = 1 - \frac{|x_j^i - x_j^k|}{A_j^{\max} - A_j^{\min}}. \tag{5}$$

其中: $A_j^{\max}$ 和 $A_j^{\min}$ 分别表示染色体中基因 j 的最大值和最小值, p_{ik} 表示染色体 i 中基因 j 的数值 x_j^i 不同于染色体 k 中基因 j 的数值 x_j^k 的概率.

整个初始群体的熵值 S 定义为群体中所有基因熵值 S_j 的平均值, 即

$$S = \frac{1}{L}\sum_{j=1}^{L} S_j. \tag{6}$$

利用信息熵产生初始群体的过程如下:

Step 1: 设定临界熵值 S_0.

Step 2: 在染色体定义域内随机产生第 1 个染色体.

Step 3: 用同样的方法每次生成一个染色体, 并计算该染色体与已有个体的熵值 S. 若 $S > S_0$, 则接受该染色体; 否则, 拒绝. 重新产生一个新的染色体并计算熵值 S, 直至出现满足条件 $S > S_0$ 的染色体为止.

Step 4: 重复 Step 3, 直至染色体数目达到规定的初始群体为止.

当初始种群产生后, 利用全排列理论, 将每个混沌量对应于一条路径, 采用式 (3) 的 Tent 映射为混沌信号发生器产生混沌变量, 为每条路径上的信息素浓度给出初始值, 以引导蚂蚁进行路径选择.

3.2　遗传及蚁群策略

3.2.1　遗传策略

遗传策略具有较强的全局快速收敛能力, 将遗传的交叉变异策略引入蚁群策略中, 二者有机地融合为一个整体, 可以发挥其整体功能.

1) 选择策略. 采取轮盘赌的形式选择个体可能引发超级个体和相似个体问题, 从而导致“早熟现象”. 若采用联赛选择法, 即每次从群体中随机选取若干个体加以比较, 适应值最大的个体获胜. 既可以避免个体被选中的概率与其适应值大小直接成比例, 又能保证被选中的个体具有较高的适应值.

2) 交叉和变异规则. 交叉规则采用非常规码常规交配法, 变异规则采用交换变异算子, 随机选择两个变异点, 互相交换两个基因位置. 比如所选位置是 2 和 8, 用下划线表示为 1<u>2</u>34567<u>8</u>9→1<u>8</u>34567<u>2</u>9.

3.2.2　蚁群策略

蚁群策略主要包括蚂蚁的寻优策略和信息素的更新策略. 在求解不同性质的组合优化问题时, 蚁群策略的具体形式会有所不同, 此处以 Dorigo 等人[4]提出的自适应蚁群算法为例说明基本蚁群策略.

1) 蚂蚁的寻优策略

采用自适应伪随机比率选择规则, 在 t 时刻, 每只蚂蚁 k 根据下列状态转移概率选择下一节点:

$$j = \begin{cases} \arg\max\{[\tau_{ij}(t)][\eta_{ij}(t)]^\beta\},\ q(t) \leqslant q(\lambda(t)); \\ S,\ \text{otherwise}. \end{cases} \tag{7}$$

$$q(\lambda(t)) = \lambda(t)/N. \tag{8}$$

其中: $\lambda(t) \in [2, N]$ 表示算法在第 t 次迭代时的 ANB, N 表示节点数, $q(\lambda(t)) \in [2/N, 1]$, $q(t)$ 为 [0,1] 区间上一致分布的随机数. S 根据下式进行选择:

$$p_{ij}^k(t) = \begin{cases} \dfrac{[\tau_{ij}(t)][\eta_{ij}(t)]^\beta}{\sum\limits_{s\in J_k(i)}[\tau_{is}(t)][\eta_{is}(t)]^\beta},\ j \in J_k(i); \\ 0,\ \text{otherwise}. \end{cases} \tag{9}$$

其中: $\tau_{ij}(t)$ 为节点 i 与节点 j 之间的信息素量, $\eta_{ij}(t)$

为节点 i 与节点 j 之间的启发因子, β 为启发因子的强弱, $J_k(i)$ 为蚂蚁 k 下一步允许访问的节点集合. 由于函数 $q(\lambda(t))$ 的值与平均节点数成正比, 当 $\lambda(t)$ 变小时, 说明算法将要出现停滞现象, 此时 $q(\lambda(t))$ 的值也会变小, 按照式 (9) 进行选择的概率增大, 从而提高了算法的随机搜索能力, 使解的搜索空间增大; 当 ANB 变大时, 按照式 (7) 进行选择的可能性增大, 即较好的路径将会被选择.

2) 信息素的更新策略

蚁群算法信息素的更新分为局部信息素更新和全局信息素更新. 局部更新的实质是一种信息素的负反馈机制, 这样可增加未访问路径被选择的机会, 从而扩大算法的搜索空间, 有效地避免算法陷入局部最优; 全局更新的实质是信息素的正反馈机制, 它使得全局最优解所经路径上的信息素增强, 使算法最终收敛于最优解.

对于蚂蚁 k, 如果节点 i 和 j 是它所选择路径上的两个相邻节点, 则蚂蚁从节点 i 转移到节点 j 后, 路径上的信息素按下式进行局部更新:

$$\tau_{ij}(t) = (1-\xi)\tau_{ij}(t) + \xi\tau_0. \tag{10}$$

其中: τ_0 为常数, $\xi \in (0,1)$ 为可调参数. 否则, 将不予更新.

针对全局最优解所属的边, 进行全局更新如下:

$$\tau_{ij}(t+1) = (1-\rho(t))\tau_{ij}(t) + \rho(t)\Delta\tau_{ij}(t). \tag{11}$$

其中: $\Delta\tau_{ij}(t)$ 是在本次迭代中, 节点 i, j 之间边上的信息素增量, 且有 $\Delta\tau_{ij}(t) = 1/L_{\text{best}}$, L_{best} 为当前全局最优路径长度; $\rho(t) \in (0,1)$ 为信息素的挥发系数.

3.3 适应度函数的改进

适应度函数值是算法所得解的优劣程度度量标准. 为了充分利用遗传算法和蚁群算法每次迭代过程中适应度函数所反映出的信息, 提出适应度改进函数.

设在第 t 次迭代中, 遗传算法所得种群的最大适应度值为 $f_{\text{GA}}^{\max}(t)$, 最小适应度值为 $f_{\text{GA}}^{\min}(t)$, 平均适应度值为 $\bar{f}_{\text{GA}}(t)$; 蚁群算法所得解的最大适应度值为 $f_{\text{ACO}}^{\max}(t)$, 最小适应度值为 $f_{\text{ACO}}^{\min}(t)$, 平均适应度值为 $\bar{f}_{\text{ACO}}(t)$. 其中 $\bar{f}_{\text{GA}}(t)$ 和 $\bar{f}_{\text{ACO}}(t)$ 分别为

$$\bar{f}_{\text{GA}}(t) = \frac{1}{N_{\text{GA}}}\sum_{i=1}^{N_{\text{GA}}} f_{\text{GA}}^i(t), \tag{12}$$

$$\bar{f}_{\text{ACO}}(t) = \frac{1}{N_{\text{ACO}}}\sum_{i=1}^{N_{\text{ACO}}} f_{\text{ACO}}^i(t). \tag{13}$$

其中: N_{GA} 和 N_{ACO} 分别为遗传算法的种群规模和蚂蚁数量, $f_{\text{GA}}^i(t)$ 和 $f_{\text{ACO}}^i(t)$ 分别为遗传算法和蚁群算法所得解 i 的适应度函数值.

为了将遗传算法和蚁群算法有效融合, 适应度函数 $f(t)$ 改进如下:

$$f(t) = a\bar{f}_{\text{GA}}(t) + b\bar{f}_{\text{ACO}}(t). \tag{14}$$

其中

$$a = \partial_1/(\partial_1+\partial_2),\ b = \partial_2/(\partial_1+\partial_2),$$

$$\partial_1 = f_{\text{GA}}^{\max}(t) - f_{\text{GA}}^{\min}(t),\ \partial_2 = f_{\text{ACO}}^{\max}(t) - f_{\text{ACO}}^{\min}(t).$$

若适应度函数值在连续 T 代迭代中均无明显改进, 则认为算法收敛, 此处定义 ECGACO 算法的终止条件函数为

$$\varphi(t) = \frac{f(t) - f(t+1)}{f(t)}, \tag{15}$$

$$\varphi(t) < \varepsilon, \tag{16}$$

其中 ε 为一个给定的精度. 若在连续 T 代中式 (16) 均成立, 则算法终止.

4 参数自适应控制策略

4.1 遗传策略的参数自适应控制

为了平衡遗传算法的搜索范围与搜索能力, Srinivas 等人[5]提出了自适应遗传算法, 采用种群中最优个体的适应度值与种群平均适应度值的差值来判断种群的多样性. 但是, 有时这一差值并不能及时反映种群个体的过早收敛程度. 例如, 若某一种群是由一些局部最优个体和一些适应度很低的个体组成, 则此时过早收敛现象虽已发生, 但由于适应度很低的个体的存在, 差值依然较大, 这时算法就会误认为种群没有发生过早收敛, 从而无法跳出局部最优解, 导致搜索寻优性能下降. 此时, 可将适应度值大于 $\bar{f}(t)$ 的个体的适应度值再做一次平均得到 $\bar{\bar{f}}(t)$, 定义 $\varPhi = f_{\text{GA}}^{\max}(t) - \bar{\bar{f}}(t)$ 表示种群的过早收敛程度, 有

$$p_c(t) = \begin{cases} \dfrac{k_1(f_{\text{GA}}^{\max}(t) - f'_{\text{GA}}(t))}{f_{\text{GA}}^{\max}(t) - \bar{\bar{f}}(t)},\ f'_{\text{GA}}(t) \geqslant \bar{\bar{f}}(t); \\ k_3,\ f'_{\text{GA}}(t) < \bar{\bar{f}}(t). \end{cases} \tag{17}$$

$$p_m(t) = \begin{cases} \dfrac{k_2(f_{\text{GA}}^{\max}(t) - f_{\text{GA}}(t))}{f_{\text{GA}}^{\max}(t) - \bar{\bar{f}}(t)},\ f_{\text{GA}}(t) \geqslant \bar{\bar{f}}(t); \\ k_4,\ f_{\text{GA}}(t) < \bar{\bar{f}}(t). \end{cases} \tag{18}$$

其中: $k_1, k_2, k_3, k_4 \leqslant 1.0$; $f'_{\text{GA}}(t)$ 为进行交叉的两个个体中适应度函数值较大者; $f_{\text{GA}}(t)$ 为变异个体的适应度函数值.

4.2 蚁群策略的参数自适应控制

在蚂蚁的状态转移规则中, $q(t)$ 为 [0,1] 区间上一致分布的随机数, 增加了所得解的多样性, 一定程度上削弱了蚁群陷入局部最优的趋势, 但它的产生又过于随机, 不易控制. 此处同样采用式 (3) 的 Tent 映射为混沌信号发生器, 产生 [0,1] 区间上均匀分布的随机

数$q(t)$. 即

$$q(t+1)=\begin{cases} q(t)/\mu,\ q(t)\in(0,\mu); \\ (1-q(t))/(1-\mu),\ q(t)\in(\mu,1); \end{cases}$$
$$0<\mu<1. \tag{19}$$

由于信息素挥发系数 $\rho(t)$ 的存在, 当问题规模较大时, 那些从未被搜索到的边的信息量会减少到接近于 0 的状态, 降低了算法的全局搜索能力. 反之, $\rho(t)$ 过大时, 解的信息量也会较快增大, 以前搜索过的解被选择的可能性很大, 也会影响算法的全局搜索能力. 虽然减小 $\rho(t)$ 可以提高算法的全局搜索能力, 但会降低算法的收敛速度. 因此, 可以自适应地调节 $\rho(t)$ 的值, 此处采用正弦函数逐步降低 ρ 的衰减速度. ρ 的初始值 $\rho(t_0)=\rho_{\max}$, 当算法求得的最优解在 T 次循环内没有明显改进时, ρ 减为

$$\rho(t+1)=\begin{cases} \sin[\rho(t)],\ \sin[\rho(t)]\geqslant\rho_{\min}; \\ \rho_{\min},\ \text{otherwise}. \end{cases} \tag{20}$$

其中 $\rho_{\min}$ 为 ρ 的最小值, 可以防止因 ρ 过小而降低算法的收敛速度.

为了充分利用蚁群算法的正反馈信息, 在全局信息素更新时, 加入混沌扰动. 利用式 (3) 的 Tent 映射为混沌信号发生器产生混沌变量, 以使搜索跳出局部极值区间. 全局信息素更新公式改写为

$$\tau_{ij}(t+1)=(1-\rho(t))\tau_{ij}(t)+\rho(t)\Delta\tau_{ij}(t)+\delta z_{ij}. \tag{21}$$

其中: z_{ij} 为混沌变量, 可由式 (3) 迭代得到; δ 为混沌扰动系数.

5 ECGACO 算法步骤及实例分析

5.1 算法步骤

Step 1: 初始化参数 $\rho(t_0), q(t_0), p_c, p_m, \beta, \xi, \delta$, 临界熵值$S_0$, 群体规模 $N_{\rm GA}$ 和蚂蚁数量 $N_{\rm ACO}$ 以及常数 ε.

Step 2: 利用信息熵产生规模为 $N_{\rm GA}$ 的初始种群.

Step 3: 采用联赛选择规则对初始种群进行筛选, 找出若干优化解, 根据式 (12) 计算平均适应度 $\bar{f}_{\rm GA}(0)$, 记录最优解 $L_{\rm best}$.

Step 4: 利用全排列理论, 采用式 (3) 的 Tent 映射为混沌信号发生器产生混沌变量, 并为每条路径上的信息素浓度给出初始值 $\tau_{ij}(t_0)$.

Step 5: 令 $t=0$.

Step 6: 将 $N_{\rm ACO}$ 只蚂蚁随机放置在 N 个节点上.

Step 7: 采用式 (3) 的 Tent 映射为混沌信号发生器, 产生 [0,1] 区间上均匀分布的随机数 $q(t)$, 每只蚂蚁根据式 (7) 和 (9) 选择下一访问节点.

Step 8: 根据式 (10) 更新局部信息素.

Step 9: 若每只蚂蚁 k 都访问了所有 N 个节点, 则计算解的适应度值, 记录本次迭代最优解 $L_{\rm best}(t)$; 若其优于历史最优解 $L_{\rm best}$, 则令 $L_{\rm best}=L_{\rm best}(t)$; 否则, 返回 Step 7.

Step 10: 利用式 (3) 的 Tent 映射为混沌信号发生器产生混沌变量, 根据式 (20) 和 (21) 自适应控制蚁群策略参数, 进行全局信息素更新.

Step 11: 根据式 (13) 计算并记录蚁群策略本次迭代所得解的平均适应度值 $\bar{f}_{\rm ACO}(t)$, 根据式 (14) 计算适应度函数 $f(t)$ 的值以及平均值 $\bar{f}(t)$ 和$\bar{f}(t)$. 找出蚁群策略得到的适应度值大于 $\bar{f}(t)$ 的个体并替代原有本代初始种群中适应度值较低的个体.

Step 12: 根据式 (17) 和 (18) 自适应控制遗传策略参数, 得到本次迭代的交叉概率 $p_c(t)$ 和变异概率 $p_m(t)$.

Step 13: 进行遗传策略的交叉和变异操作.

Step 14: 根据式 (14) 计算各解的适应度函数 $f(t+1)$ 的值, 记录本次迭代最优解, 将其与历史最优解做比较. 若优于历史最优解, 则用其替换历史最优解; 否则, 转至 Step 11.

Step 15: 计算适应度函数 $f(t+1)$ 的平均值 $\bar{f}(t+1)$, 然后根据式 (15) 计算函数 $\varphi(t)$ 的值, 并按式 (16) 判断算法是否终止. 若算法终止, 则输出历史最优解 $L_{\rm best}$; 否则, 令 $t=t+1$, 转至 Step 6.

5.2 实例仿真

鉴于 TSP 问题已成为测试新算法的标准问题, 本文采用 TSPLIB 中 50 座城市的数据进行实验, 结果取 10 次实验的平均结果, 如表 1 和表 2 所示. ECGACO 算法的参数选择为: $\rho(t_0)=0.15, q(t_0)=0.2, p_c=0.3, p_m=0.08, \beta=1, \xi=0.5, \delta=0.5$, 临界熵值 $S_0=0.2$, 群体规模 $N_{\rm GA}=100$, 蚂蚁数量 $N_{\rm ACO}=20$, 常数 $\varepsilon=0.000\,5$.

表 1　ECGACO 算法所得的结果

最短路径长度	平均路径长度	迭代次数
426	427.1	146

表 2　TSP 问题的多种优化算法对比[6]

	GA	ACS	EP	SA	AG
Eil50	428	153	426	443	436
迭代次数	25 000	2 412	100 000	68 512	28 111

通过对比 50 个城市路径优化的结果可以看出, ECGACO 算法明显减少了迭代次数, 能够较快地收敛于解空间的集中区域, 避免陷入局部最优, 且能找到更优的最短路径值.

6 结 论

本文设计的ECGACO算法, 利用信息熵产生初始群体, 避免了初始群体集中分布在解空间的局部区域, 增加了初始群体的多样性. 通过融合信息熵和混沌理论, 将遗传策略和蚁群策略综合集成为一个系统整体, 可以充分发挥遗传算法的快速全局收敛能力以及蚁群算法精确寻优和正反馈机制下的快速收敛能力, 动态地实现两种算法优势互补. 对TSP问题50座城市的数据进行仿真实验, 取得了满意的结果.

参考文献(References)

[1] 许国志, 顾基发, 车宏安. 系统科学[M]. 上海: 上海教育出版社, 2000.
(Xu G Z, Gu J F, Che H A. System science[M]. Shanghai: Shanghai Education Publishing House, 2000.)

[2] 郝柏林. 从抛物线谈起——混沌动力学引论[M]. 上海: 上海科技教育出版社, 1993.
(Hao B L. Speaking of from the parabola — Chaotic dynamics introduciton[M]. Shanghai: Shanghai Scientific and Technolgical Publishing House, 1993.)

[3] 袁晓辉, 袁艳斌, 王乘, 等. 一种新型的自适应混沌遗传算法[J]. 电子学报, 2006, 34(4): 708-712.
(Yuan X H, Yuan Y B, Wang C, et al. A novel self-adaptive chaotic genetic algorithm[J]. Acta Electronica Sinica, 2006, 34(4): 708-712.)

[4] Dorigo M, Caro G D, Gambardella L M. Ant algorithms for discrete optimization[J]. Artificial Life, 1999, 5(3): 137-172.

[5] Srinivas M, Patnaik L M. Adaptive probabilities of crossover and mutation in genetic algorithms[J]. IEEE Trans on Systems, Man and Cybernetics, 1994, 26(4): 656-667.

[6] 李士勇, 陈永强, 李研. 蚁群算法及其应用[M]. 哈尔滨: 哈尔滨工业大学出版社, 2004.
(Li S Y, Chen Y Q, Li Y. Ant colony algorithm and its application[M]. Harbin: Harbin Institute of Technology Publishing House, 2004.)

(上接第43页)

[7] Rostami S, Hamidzadeh B. An optimal periodic scheduler for dual-arm robots in cluster tools with residency constraints[J]. IEEE Trans on Semiconductor Manufacturing, 2001, 17(5): 609-618.

[8] Rostami S, Hamidzadeh B. Optimal scheduling techniques for cluster tools with process-module and transport-module residency constraints[J]. IEEE Trans on Semiconductor Manufacturing, 2002, 15(3): 341-349.

[9] Kim J H, Lee T E. Scheduling analysis of time-constrained dual-armed cluster tools[J]. IEEE Trans on Semiconductor Manufacturing, 2003, 16(3): 521-534.

[10] Li L Y, Hu J T. Scheduling single-blade cluster tools with time window constraints[C]. 2008 Chinese Control and Decision Conf. Yantai: IEEE Press, 2008: 1069-1072.

[11] Wu N Q, Chu C B, Chu F, et al. A Petri net method for schedulability and scheduling problems in single-arm cluster tools with wafer residency time constraints[J]. IEEE Trans on Semiconductor Manufacturing, 2008, 21(2): 224-236.

[12] Rostami S, Hamidzadeh B. Scheduling techniques for flexible semiconductor manufacturing tools[C]. Proc of the IEEE Int Conf on Systems, Man and Cybernetics. Yasmine Hammamet: IEEE Press, 2002: 286-291.

[13] Yoon H J, Lee D Y. On-line scheduling of Integrated single-wafer processing tools with temporal constraints[J]. IEEE Trans on Semiconductor Manufacturing, 2005, 18(3): 390-398.

[14] 李莉, 乔非, 许潇红, 等. 半导体生产线全局修正式重调度方法研究[J]. 计算机集成制造系统, 2006, 12(7): 1022-1027.
(Li L, Qiao F, Xu X H, et al. Complete modification rescheduling method for semiconductor wafer fabs[J]. Computer Integrated Manufacturing Systems, 2006, 12(7): 1022-1027.)

[15] 乔非, 李莉, 马玉敏, 等. 基于模糊推理的半导体生产重调度策略研究[J]. 计算机集成制造系统, 2009, 15(1): 102-108.
(Qiao F, Li L, Ma Y M, et al. Fuzzy-reasoning-based rescheduling strategy for semiconductor manufacturing[J]. Computer Integrated Manufacturing Systems, 2009, 15(1): 102-108.)

[16] Brauner N, Finke G, Kubiak W. Complexity of one-cycle robotic flow-shops[J]. J of Scheduling, 2003, 6(4): 355-371.

[17] Schwalb E, Hamidzadeh B. Processing disjunction in temporal constraints networks[J]. Artificial Intelligence, 1997, 93(1): 29-61.