文章编号：1000 - 8152(2003)03 - 0445 - 04

Flow-shop 调度问题的自适应模拟退火算法

陈 雄[1,3]，杨凤霞[2]，吴启迪[3]
（1. 复旦大学 电子工程系，上海 200433；2. 河南职业技术师范学院 化学工程系，河南 新乡 453003；
3. 同济大学 CIMS 研究中心，上海 200092）

摘要：为求得一个强 NP-难问题——flow-shop 调度问题的最优解或近优解，提出一种自适应模拟退火算法.本算法采用一种基于区段特性的特殊邻域结构、简便的目标函数计算方法和自适应退火策略.通过 Flow-shop 调度问题的基准测试问题的实验，数值结果证实了该方法的有效性.

关键词：flow-shop 调度问题；自适应模拟退火算法；启发式算法
中图分类号：TP301.6 **文献标识码**：A

Adaptive simulated annealing algorithm for flow-shop scheduling problem

CHEN Xiong[1,3]，YANG Feng-xia[2]，WU Qi-di[3]
（1. Department of Electronic Engineering, Fudan University, Shanghai 200433, China;
2. Department of Chemical Engineering, Henan Vocation-Technical Teachers College, Henan Xinxiang 453003, China;
3. Research Center for CIMS, Tongji University, Shanghai 200092, China）

Abstract: An adaptive simulated annealing algorithm is proposed for the optimal or sub-optimal solution of flow-shop scheduling problem. A special neighborhood structure based on block property, simple computed method of object function and an adaptive annealing strategy are adopted in the algorithm. The experiment shows the effectiveness of this algorithm.

Key words: flow-shop scheduling; adaptive simulated annealing algorithm; heuristic

1 引言(Introduction)

一个 n 工件、m 台机器的 flow-shop 调度问题描述如下：n 个工件必须按同一机器排序在 m 台不同的机器上进行加工，工件 i 在 j 机器上的加工时间为 $p_{ij}(i = 1,\cdots,n, j = 1,\cdots,m)$，加工时间已知，且假设：1) 每个工件在每台机器上只能加工一次；2) 每台机器一次只能加工一个工件；3) 工件的加工不能发生中断；4) 每台机器按相同的工件排序进行加工；调度的目标就是寻找一个工件加工排序，使最后一个工件在最后一台机器上的完工时间最小.

Flow-shop 调度问题是 NP-hard 问题 ($m \geqslant 3$)[1,2]. 求解 flow-shop 问题的启发式算法有：1) 构造式算法[3,4]；2) 改进式算法[5,6]. NEH 算法[4]是目前为止较好的构造式算法，而模拟退火算法、禁忌搜索算法是改进式算法的代表.

本文提出一种特定邻域结构的自适应模拟退火算法，邻域的产生是基于关键路径的区段特征进行的，退火温度 T 采用自适应方式，目标函数的计算应用一种简洁的方法，可快速、有效地求得 flow-shop 问题的近优解或最优解.

2 Flow-shop 调度问题描述(Description of the flow-shop scheduling problem)

设 π 是一个工件加工排序，Π 是所有工件加工排序的集合，问题的优化目标是寻找一个工件排序 $\pi^* \in \Pi$，使所有工件的完工时间 $C_{\max}(\pi)$ 最小，$C_{\max}(\pi)$ 为

$$C_{\max}(\pi) = \max_{1 = i_0 \leqslant i_1 \leqslant \cdots \leqslant i_{m-1} \leqslant i_m = n} \sum_{j=1}^{m} \sum_{i=i_{j-1}}^{i_j} p_{\pi(i)j}. \quad (1)$$

将工件排序 $\pi \in \Pi$ 用图型 $G(\pi) = (N, E)$ 表示，$N = \{1,\cdots,m\} \times \{1,\cdots,n\}$ 是节点集，其节点 (i,j) 代表工件 i 在机器 j 上进行加工的工序，加工时间为 p_{ij}，$E = \bigcup_{j=1}^{m} \bigcup_{i=1}^{i-1} \{((i,j),(i+1,j))\} \cup \bigcup_{i=1}^{n} \bigcup_{j=1}^{m-1} \{((i,j),(i,j+1))\}$ 是有向弧集. 方程(1) 描述从节点(1,1)到节点(m,n)的关键路径，从节点(1,

收稿日期：2001 - 03 - 14；收修改稿日期：2002 - 10 - 10.
基金项目：国家自然科学基金(69774032)资助项目.

1)到节点(m,n)的每条路径可用一个排序$(i_0,i_1,\cdots,i_m)$表示，由水平子路径$(j,i_{j-1}),(j,i_{j-1}+1),\cdots,(j,i_j),j=1,\cdots,m$和垂直弧$(j,i_j),(j+1,i_j),j=1,\cdots,m-1$.令$s=(s_0,s_1,s_m)$是图$G(\pi)$上定义关键路径的一个排序，其中$s_0=1$和$s_m=n$，即关键路径为$(1,s_0),\cdots,\cdots,(1,s_1),(2,s_1),\cdots,(2,s_2),\cdots,\cdots,(m,s_{m-1}),\cdots,(m,s_m)$.对机器$j$，水平子路径$(j,s_{j-1}),(j,s_{j-1}+1),\cdots,(j,s_j)$是在该机器上进行连续加工的工件$\pi(s_{j-1}),\pi(s_{j-1}+1),\cdots,\pi(s_j)$的排序，$j=1,\cdots,m$.如果$s_{j-1}<s_j$，称这工件排序为一个"区段".设排序$u=(u_0,u_1,\cdots,u_k)$，$u_0=1$，$u_l=s_{m_l}$，$l=1,\cdots,k$，则$1=u_0<u_1<\cdots<u_k=n$.这里，$k$是关键路径中的区段数，$m_l$是第$l$个区段的机器指数，$l=1,\cdots,k$.这样，第$l$个区段可用排序$B_l=(\pi(u_{l-1}),\pi(u_{l-1}+1),\cdots,\pi(u_l))$表示，$l=1,\cdots,k$，$B_l$中的工件数为$b_l=u_l-u_{l-1}+1$，$l=1,\cdots,k$，$\sum_{l=1}^{k}b_l=n+k-1\leqslant n+m-1$，$B_l$中的最后一个工件也是$B_{l+1}$中的第一个工件，$l=1,\cdots,k-1$.根据区段的定义，$C_{\max}(\pi)$可重表示为

$$C_{\max}(\pi)=\sum_{l=1}^{k+1}\sum_{j=m_{l-1}+1}^{m_l-1}p_{\pi((u_{l-1})j}+\sum_{l=1}^{k}\sum_{i\in B_l}p_{i\,m_l}. \quad (2)$$

其中$m_0=0$，$m_{k+1}=m+1$.

3 模拟退火算法(Simulated annealing algorithm)

应用模拟退火算法[7]，需根据不同的问题确定不同的邻域结构和退火策略.

3.1 受限的邻域结构(Limited neighborhood structure)

排序问题的邻域结构常采用移动方式生成:

1) 交换移动方式:交换不同位置上的两个工件;

2) 位移移动方式:将某一位置上的工件转移到另一位置上.

本文采用位移移动方式.设$v=(a,b)$是排序π中的一对位置，$a,b\in\{1,\cdots,n\}$，$a\neq b$，而排序π_v是通过排序π中的a位置工件$\pi(a)$转移到b位置而生成的.每一对$v=(a,b)$确定π中的一个位移移动，设Ω是任意$v=(a,b)$的集合，则π的邻域结构可采用Ω集内的位移移动$v=(a,b)$产生的，设为$N(\Omega,\pi)=\{\pi_v:v\in\Omega\}$.由文献[2]知，最大的邻域结构是由位移移动集$V=\{(a,b):b\notin\{a-1,a\},a,b\in\{1,2,\cdots,n\}\}$产生的.当$|a-b|=1$时，移动$v=(a,b)$和$v'=(b,a)$产生同一排序$\pi_v=\pi_{v'}$，避免冗余，$V$只取$v=(a,b)$和$v'=(b,a)$中的一个.$N(V,\pi)$有$(n-1)^2$邻域并满足如下性质：$\forall\pi^{(1)}\in N$，存在一个有限排序$\pi^{(1)},\pi^{(2)},\cdots,\pi^{(r)}$满足$\pi^{(r)}$是一个最优排序，且$\pi^{(i+1)}\in N(V,\pi^{(i)})$，$i=1,\cdots,r-1$.

对$\forall B_l$，考虑位移移动的一个子集$W_l(\pi)\subset V$，满足

$$W_l(\pi)=\begin{cases}\{(a,b)\in V:a,b\in\{u_0,\cdots,u_1-1\}\},\\ \text{如果 } l=1 \text{ 且 } m_1=1;\\ \{(a,b)\in V:a,b\in\{u_{k-1}+1,\cdots,u_k\}\},\\ \text{如果 } l=k \text{ 且 } m_k=m;\\ \{(a,b)\in V:a,b\in\{u_{l-1}+1,\cdots,u_l-1\}\},\\ \text{其他}.\end{cases} \quad (3)$$

令$W(\pi)=\bigcup_{l=1}^{k}W_l(\pi)$，子集$W(\pi)$中的元素个数设为$\delta$，则$\delta$满足如下关系

$$\sum_{l=1}^{k}(\max\{0,b_l-3\})^2\leqslant\delta\leqslant(\max\{0,b_l-2\})^2+\sum_{l=2}^{k-1}(\max\{0,b_l-3\})^2+(\max\{0,b_k-2\})^2.$$

由文献[8,9]知邻域$N(W(\pi),\pi)$具备如下性质：

$\forall\beta\in N(W(\pi),\pi)$满足$C_{\max}(\beta)\geqslant C_{\max}(\pi)$.

由左移与右移的相似性，本文仅分析右移情况.设j属于某一区段B_l内，即$u_{l-1}<j<u_l$，则只能从位置u_l开始向右移，可插入的位置为$u_l,u_l+1,\cdots,u_{l+1}$，理由是(j,t)属于集合$W_l(\pi)$，$t=j+1,\cdots,u_l-1$.如果$\exists l\in\{1,\cdots,k\}$使$j=u_{l-1}$，则工件$\pi(j)$将从位置$u_{l-1}+1$开始往右移动(性质不包括移动$(u_{i-1},t)$，$t=u_{l-1}+1,\cdots,n$).由于右移$(u_{l-1},u_{l-1}+1)$和左移$(u_{l-1}+1,u_{l-1})$是相等的，仅考虑左移$(u_{l-1}+1,u_{l-1})$而删除右移$(u_{l-1},u_{l-1}+1)$.如果最后一个区段发生在机器$m$上，即$m_k=m$，则工件$\pi(j)$就无需右移，其中$j=u_{k-1}+1,\cdots,u_k-1$.由对称性知，如果$m_1=1$，则工件$\pi(j)$就无需左移，其中$j=u_0+1,\cdots,u_1-1$.如是$\exists l\in\{1,\cdots,k\}$使$u_l=u_{l-1}+1$，则从相等的移动$(u_{l-1},u_l)$和$(u_l,u_{l-1})$中任意删除其中一个移动，本文规定删除右移$(u_{l-1},u_l)$.

$\forall\pi(j)$可能属于一个区段或两个区段.令$l_r(j)$为整数，满足$1\leqslant l_r(j)\leqslant k$和$u_{l_r(j)-1}\leqslant j<u_{l_r(j)}$，即

$l_r(j)$ 是含有工件 $\pi(j)$ 的最大区段指数，其中 $j = 1, \cdots, n-1$. 相似地，令 $l_l(j)$ 为整数，满足 $1 \leqslant l_l(j) \leqslant k$ 和 $u_{l_l(j)-1} < j \leqslant u_{l_l(j)}$，即 $l_l(j)$ 是含有工件 $\pi(j)$ 的最小区段指数，其中 $j = 2, \cdots, n$. 这里，当 $j \notin \{u_0, u_1, \cdots, u_k\}$ 时，$l_r(j) = l_l(j)$；当 $j \in \{u_1, \cdots, u_{k-1}\}$ 时，$l_r(j) = l_l(j) + 1$. 可移动插入的位置数由 γ_l 决定，$\gamma_l = u_l - u_{l-1}, l = 1, \cdots, k$. 设移动集 $Z(\pi) = \bigcup_{j=1}^{n-1} ZR_j(\pi) \cup \bigcup_{j=2}^{n} ZL_j(\pi)$，$m_1 \neq 1$ 和 $m_k \neq m$，右移动的集合为 $ZR_j(\pi), j = 1, \cdots, n-1$，左移动的集合为 $ZL_j(\pi), j = 2, \cdots, n$.

$$ZR_j(\pi) = \{(j,t): u_{l_r(j)} \leqslant t \leqslant u_{l_r(j)} + \gamma_{l_r(j)+1}\},$$

$$ZL_j(\pi) = \{(j,t): u_{l_l(j)-1} - \gamma_{l_l(j)-1} \leqslant t \leqslant u_{l_l(j)-1}, - \omega(u_{l_l(j)} - u_{l_l(j)-1})\}.$$

这里，当 $x > 1$ 时 $\omega(x) = 0$，其他情况 $\omega(x) = 1$. 在 $m_1 = 1$ 时令 $ZL_j(\pi) = \phi, j = 2, \cdots, u_1 - 1$，如果 $m_k = m$，$ZR_j(\pi) = \phi, j = u_{k-1} + 1, \cdots, n-1$.

3.2　自适应退火策略(Adaptive annealing strategy)

自适应退火策略根据最近参数 T_i 值的搜索性能调整温度参数的变化. 搜索性能的评价采用在第 i 个迭代集(参数 T_i 下)目前最好解减少的次数，设为 $n(i)$. 如果 $n(i) \geqslant n(i-1)$ 且 $n(i) \neq 0, n(i-1) \neq 0$，则 T_{i+1} 按文献[6]的非增函数进行，否则，暗示着搜索区域已接近一个局部或全局最优解. 此时采用两种 T 值：T_L 和 T_S. n_L 和 n_S 分别为两种参数下的最好解减少的次数，再分别运行模拟退火算法，以判定问题的解的性质. $T_L = \alpha T_i, T_S = T_i/\alpha, \alpha > 1$. 较大的 T_L 允许接受较大量的“劣化解”，试图搜索跳出可能的局部最优解陷井，而较小的 T_S 可加速搜索最优解. 如果 $\max(n_L, n_S) > n(i)$，产生较大的 T_L，令 $T_{i+1} = T_L, T_{i+2} = T_{i+1}$，目的是增加搜索力度. 如果 $\max(n_L, n_S) \leqslant n(i)$，采用如下参数加速搜索

$$T_{i+1} = \begin{cases} \frac{1}{2}(T_i + T_L), \text{ if } n_L > n_S, \\ \frac{1}{2}(T_i + T_S), \text{ if } n_L > n_S. \end{cases}$$

自适应退火策略确定的参数序列 $\{T_i\}$ 保证每个温度 T_i 不为零. 当 $n(i) = n(i-1) = 0$ 或 $n(i) < n(i-1)$，参数 T_i 允许分别采用较大值 T_L 和较小值 T_S. 当 $\max(n_L, n_S) \leqslant n(i)$，参数 T_i 将不会无限期的增加，原因如下：即使 T_i 增大，而最坏情况 $n(i) = n_L = n_S = 0$ 发生的可能就更大，由自适应策略知，T_{i+1} 由 $T_{i+1} = \frac{1}{2}(T_i + T_S) = \frac{1}{2}(T_i + (1/\alpha)T_i) < T_i$ 确定，则参数序列 $\{T_i\}$ 在每个迭代阶段有可能存在增加的情况，但在性能没有改进时，依然保持渐近非增的序列，保证自适应模拟退火算法渐近收敛于全局最优解.

初始温度 $T_1 = \sum_{i=1}^{n} \sum_{j=1}^{m} p_{ij}/(5mn)$，停止条件采用固定的最大迭代数 $N = 2000$ 和 5000，以便在相同的计算 $C_{\max}(\pi)$ 次数下评价不同算法的性能.

3.3　算法的加速器(Algorithm accelerator)

本文采用基于图 $G(\pi)$ 的关键路径计算 $C_{\max}(\pi)$ 加速算法运行. 设 ρ_{gh} 是节点(1,1)和节点(g,h)间最长路径的长度，η_{gh} 是节点(g,h)和节点(m,n)间最长路径的长度，$(g,h) \in N$. ρ_{gh} 和 η_{gh} 应用回归公式以复杂度 $O(nm)$ 获得

$$\rho_{gh} = \max\{\rho_{g,h-1}, \rho_{g-1,h}\} + p_{\pi(h)g}, \quad g = 1, \cdots, m,\ h = 1, \cdots, n,$$

$$\eta_{gh} = \max\{\eta_{g,h+1}, \eta_{g+1,h}\} + p_{\pi(h)g}, \quad g = m, \cdots, 1,\ h = n, \cdots, 1.$$

其中 $\rho_{g,0} = 0, g = 1, \cdots, m, \rho_{0,h} = 0, h = 1, \cdots, n, \eta_{g,h+1} = 0, g = m, \cdots, 1, \eta_{m+1,h} = 0, h = n, \cdots, 1$. 显然 $C_{\max}(\pi) = \rho_{m,n} = \eta_{1,1}$.

对某一 $l(1 \leqslant l \leqslant k)$，集合 $\bigcup_{j: l_r(j) = l} ZR_j(\pi)$ 内的区段 B_l 的右移动：工件 $\pi(j)$ 从位置 j 移走并插入到位置 $t, t = u_l, u_l + 1 \cdots, u_l + \gamma_{l+1}, j = u_l - 1, u_l - 2, \cdots, u_{l-1}$. 令 $x = u_{l-1}, y = u_l, z = u_l + \gamma_{l+1}$，计算$(z - y + 1)(y - x)$个 $C_{\max}(\pi_{(j,t)}), t = y, y+1, \cdots, z, j = y-1, y-2, \cdots, x$. 根据对称性，可知左移动的情况，这里就不重述.

设 j 是一个整数，满足 $x \leqslant j < y$，令 $\pi^j = (\pi(1), \cdots, \pi(j-1), \pi(j+1), \cdots \pi(n))$ 表示 π 中位置 j 上的工件 $\pi(j)$ 移走后得到的 $n-1$ 工件的排序，排序 π^j 的图为 $G(\pi^j)$. 类似于 ρ_{gh} 的定义，ρ^j_{gh} 是图 $G(\pi^j)$ 中节点(1,1)和节点(g,h)间最长路径的长度，$g = 1, \cdots, m, h = 1, \cdots, n-1$；显然 $\rho^j_{gh} = \rho_{gh}, g = 1, \cdots, m, h = 1, \cdots, j-1$. 而 $g = 1, \cdots, m, h = j, j+1, \cdots, n-1$ 的 ρ^j_{gh} 值的计算：

$$\rho^j_{gh} = \max\{\rho^j_{g,h-1}, \rho^j_{g-1,h}\} + \rho_{\pi(h+1)g}, g = 1, \cdots, m. \tag{4}$$

其中 $\rho^j_{0h} = 0, h = j, j+1, \cdots, n-1$. $C_{\max}(\pi_{(j,t)})$ 的计算：

$$C_{\max}(\pi_{(j,t)}) = \max_{1 \leqslant e \leqslant f \leqslant m}(\rho^j_{e,t-1} + \sum_{i=e}^{f} p_{\pi(j)i} + \eta_{f,t+1}). \tag{5}$$

其中 $t = y, y+1, \cdots, z, j = y-1, y-2, \cdots, x$.

应用式(4),(5) 计算 $C_{\max}(\pi_{(j,t)})$ 的程序(右移动情况) 如下:

右移动的 $C_{\max}(\pi_{(j,t)})$ 的计算程序:

1) For $j = y-1$ to x do

开始

2) 用式(4)计算 $\rho_{g,y-1}^{j}, g = 1, \cdots, m$ 对 $h = j, j+1, \cdots, n-1$;

3) 用式(5)计算 $C_{\max}(\pi_{(j,y)})$;

4) For $t = y+1$ to z do

开始.

5) 用式(4)计算 $\rho_{g,t-1}^{j}, g = 1, \cdots, m$, 对 $h = t-1$;

6) 用式(5) 计算 $C_{\max}(\pi_{(j,t)})$;

结束

结束.

程序的复杂度:固定 j, 第 2 步的执行时间为 $O((y-j)m)$, 第 3 步为 $O(m)$, 第 5 步为 $O(m)$, 第 6 步为 $O(m)$, 则 $C_{\max}(\pi_{(j,t)})$ 的计算复杂为 $\alpha[(y-j)m + m + 2(z-y)m] = \alpha(2z-y-j+1)m$, α 为常数. 区段 B_l 的右移动的 $C_{\max}$ 的计算复杂度为

$$\alpha \sum_{j=x}^{y-1} (2z-y-j+1)m = (\alpha/2)(y-x)(4z-x-3y+3)m = (\alpha/2)(b_l-1)(b_l+4\gamma_{l+1}+2)m.$$

同理,由对称性知,区段 B_l 的左移动的 $C_{\max}$ 的计算复杂度为 $(\alpha/2)(b_i-1)(b_l+4\gamma_{l-1}+2)m$. 故邻域结构 $N(Z(\pi),\pi)$ 可在时间 τ 内搜索完, τ 为

$$\tau = \alpha \sum_{l=1}^{k} (b_l-1)(b_l+2\gamma_{l-1}+2\gamma_{l+1}+2)m. \tag{6}$$

这里 $b_0 = 0 = b_{k+1}$. 由于 $\sum_{l=1}^{k} b_l = n+k-1, b_l \geqslant 2$, $l = 1, \cdots, k, k \leqslant m$, 而 τ 主要依赖于排序 π 中的区段的分布, 如果存在 m 个长度几乎相同的区段, 则 τ 接近于低界 $O((n+m)^2)$, 如果仅存在一个区段, 则 τ 接近于上界 $O(n^2 m)$.

4 数值结果(Numerical results)

本文采用 Taillard[10] 的 flow-shop 基准问题进行测试,问题的规模分别为: 10×10, 10×20, 20×10, 20×20, 50×10, 50×20, 100×10 和 100×20. 本文的初始解采用 NEH 算法产生,每个迭代温度 T_i 下的马氏链迭代数为定长 $L = 10$, 自适应退火策略中的控制参数 α 取为 2. 算法的性能是基于平均相对百分比偏差 ξ 进行评价的,

$$\xi = 100(C_{\max}(\sigma^H) - C_{\max}(\sigma^B))/C_{\max}(\sigma^B).$$

其中, σ^H 表示由算法 $H \in$ {Taillard 的禁忌搜索算法[5], Osman 的模拟退火算法[6], 本文的适应模拟退火算法} 求得的排序, σ^B 是求得的已知的最优排序. 实验结果如表 1 所示,表中每项数据是每种规模问题的 10 次运行的平均值. 由数值结果可以看出,对所有的测试问题,自适应模拟退火算法不仅寻优性能强于 Osman 的模拟退火算法和 Taillard 的禁忌搜索算法,而且算法的运行时间明显小于其他两种算法,其中 10×10 问题在 $N = 5000$, 50×10 在 $N = 2000$ 和 50×20 在 $N = 2000$ 时,两种模拟退火算法获得相同的效果,而 10×10 在 $N = 2000$ 和 10×20 在 $N = 2000$ 环境下,自适应模拟退火算法与禁忌搜索算法得到同样的结果. 除了有限的几个问题外, Osmam 的模拟退火算法总的趋势是比禁忌搜索算法具有更强的寻优性能,而两种算法的运行时间相差并不很大.

表 1 3 种算法的性能比较

Table 1 Comparison performance among three algorithm

			自适应模拟退火算法		Osman 模拟退火算法		Taillard 禁忌搜索	
n	m	N	ξ	时间/s	ξ	时间/s	ξ	时间/s
10	10	2000	0.00	0.83	0.02	1.28	0.00	1.26
		5000	0.00	2.06	0.00	3.18	0.01	3.14
10	20	2000	0.00	1.25	0.04	1.91	0.00	1.93
		5000	0.00	3.12	0.02	4.73	0.01	4.78
20	10	2000	0.00	1.36	0.09	2.11	0.31	2.24
		5000	0.00	3.40	0.05	5.19	0.31	5.31
20	20	2000	0.00	1.89	0.22	2.95	0.26	2.87
		5000	0.00	4.65	0.36	7.36	0.21	7.15
50	10	2000	0.00	3.16	0.00	4.72	0.21	4.83
		5000	0.00	7.85	0.02	11.63	0.63	12.01
50	20	2000	0.00	7.54	0.00	11.57	0.02	12.67
		5000	0.00	18.81	0.19	28.94	0.69	31.32
100	10	2000	0.00	7.41	0.08	11.23	0.22	11.41
		5000	0.00	18.49	0.03	27.02	0.36	28.29
100	20	2000	0.00	15.51	0.09	22.81	0.32	21.67
		5000	0.00	38.67	0.12	56.52	0.51	54.01

5 结论(Conclusion)

本文的自适应模拟退火算法采用了简化的邻域结构、快速简便的目标函数计算方法和自适应退火策略,加速了算法的运行速度,同时寻优性能明显提高. 数值计算结果证实了本算法的性能明显强于目前著名的 Taillard 的禁忌搜索算法和 Osman 的模拟退火算法.

(下转第 453 页)

[4] KHARGONEKAR P P, PETERSEN I R, ZHOU Kemin. Robust stabilization of uncertain systems and H_∞ optimal control [J]. *IEEE Trans on Automatic Control*, 1990, 35(3): 351 - 361.

[5] BOYD S, GHAOUI L E, BALAKRISHNAN V, et al. *Linear Matrix Inequalities in System and Control Theory* [M]. Philadelphia, PA: SIAM, 1994.

[6] YANG Guanghong, WANG Jianliang, CHAI S Y. Guaranteed cost control for discrete-time linear systems under controller gain perturbations [J]. *Linear Algebra and Its Applications*, 2000, 312(1/3): 161 - 180.

[7] GARCIA G, BERNUSSOU J, ARIZELIER D. Robust stabilization of discrete-time linear systems with norm-bounded time-varying uncertainty [J]. *Systems & Control Letters*, 1994, 22(4): 327 - 339.

[8] YANG S K, CHEN C L. Observer based robust controller design for a linear system with time-varying perturbations [J]. *J of Mathematical Analysis and Applications*, 1997, 213(2): 642 - 661.

作者简介:

贾新春　(1964 —),男,1988 年获中国科学院系统科学研究所硕士学位,在职博士生.研究领域为鲁棒控制,智能信息处理,广义系统等.　E-mail: xchjia@aiar.xjtu.edu.cn;

郑南宁　(1952 —),男,中国工程院院士.研究领域为智能信息处理,机器人视觉与模式识别等;

袁泽剑　(1971 —),男,博士生.研究领域为鲁棒控制,智能信息处理,图象处理等.

(上接第 448 页)

参考文献(References):

[1] GAREY M R, JOHNSON D S, SETHI R. The complexity of flow shop and job shop scheduling [J]. *Mathematics of Operations Research*, 1976, 1(2): 117 - 129.

[2] LENSTRA J K, RINNOOY Kan AHG, BRUCKER P. Complexity of machine scheduling problems [J]. *Annal of Discrete Mathematics*, 1977, 1(1): 343 - 362.

[3] CAMPBELL H G, DUDEK R A, SMITH M L. A heuristic algorithm for the *n*-jobs, *m*-machine sequencing problem [J]. *Management Science*, 1970, (16): 630 - 637.

[4] NAWAZ M, EMSCORE Jr E E, HAM I. A heuristic algorithm for the *n*-job, *m*-machine flow shop sequencing problems [J]. *Int J of Management Science*, 1983, (11): 91 - 98.

[5] TAILLARD E. Some efficient heuristic methods for the flow shop seqencing problem [J]. *European J of Operational Research*, 1990, (47): 65 - 74.

[6] OSMAN I H, POTTS C N. Simulated annealing for permulation flow shop scheduling [J]. *Int J of Management Science*, 1989, (17): 551 - 557.

[7] KIRKPATRICK S, GELATT C D Jr, VECCHI M P. Optimization by simulated annealing [J]. *Science*, 1983, (220): 671 - 680.

[8] GRABOWSKI J, SKUBALSKA E, SMUTNICKI C. On flow shop scheduling with release and due dates to minimize maximum lateness [J]. *J of Operational Research Society*, 1983, (4): 615 - 620.

[9] GRABOWSKI J, NOWICKI E, ZDRALKA S. A block approach for single-machine scheduling with release dates and due dates [J]. *European J of Operational Research*, 1986, (26): 278 - 285.

[10] TAILLARD E. Benchmark for basic scheduling problems [J]. *European J of Operational Research*, 1993, (64): 278 - 285.

作者简介:

陈　雄　(1964 —),男,博士,现为复旦大学电子工程系副教授.研究方向为智能调度理论及应用,过程控制,移动机器人规则.
E-mail: chenxiong@fudan.edu.cn;

杨凤霞　(1968 —),女,学士,现为河南职业技术师范学院化学工程系讲师.研究方向为智能调度理论及应用,智能方法在化学领域中的应用;

吴启迪　(1947 —),女,博士,现为同济大学校长,博士生导师.研究方向为智能控制,CIMS,智能调度.