一种求解车间调度的混合算法

张长胜 [1] 孙吉贵 [1] 杨轻云 [2] 郑黎辉 [1]

摘 要 针对流水车间作业调度问题, 提出了一种基于 "alldifferent" 约束的混合进化算法 (Hybrid particle and genetic algorithm, HPGA), 将粒子群算法、遗传操作及模拟退火策略有效地结合在一起. 为了提高算法的求解质量, 引入了一种随机邻域搜索策略. 最后将此算法在不同规模的实例上进行了测试, 并与其他几种最近提出的具有代表性的算法进行了比较. 结果表明, 无论是在求解质量还是收敛速度方面都优于其他几种算法.

关键词 车间调度, 粒子群算法, 变异
中图分类号 TP18

A Hybrid Algorithm for Flowshop Scheduling Problem

ZHANG Chang-Sheng[1] SUN Ji-Gui [1]
YANG Qing-Yun[2] ZHENG Li-Hui[1]

Abstract A hybrid particle and genetic algorithm (HPGA) based on the "alldifferent" constraint is proposed to solve the flowshop scheduling problem, which combines the particle swarm optimization algorithm, genetic operators, and annealing strategy together. To improve the algorithm′s performance further, a neighborhood based local search strategy is proposed and introduced into HPGA. Finally, the HPGA is tested on different scale benchmarks and compared with the recently proposed efficient algorithms. The result shows that both the solution quality and the stability of the HPGA precede the other two algorithms.

Key words Flowshop scheduling, partical swarm optimization (PSO), mutation

调度问题是很多实际流水线生产调度问题的简化模型, 因此其研究具有极高的理论价值和实践价值. 本文研究的置换流水车间作业调度问题是在满足工件约束和机器约束条件下, 使得最小完工时间及总流水时间尽可能少. 该问题一般可以描述为: n 个待加工的作业 $\{N = J_1, \cdots, J_n\}$, 需要在 m 台机器上加工 $M = \{M_1, \cdots, M_m\}$, 每个作业包含 m 道工序 $J_j = \{O_{j1}, \cdots, O_{jm}\}$, 其中 O_{ik} 代表作业 i 在机器 k 上的加工时间为 t_{ik} $(1 \le i \le n, 1 \le k \le m)$ 的工序. 求解目标是得到一个可行调度 $\pi = \{\pi_1, \cdots, \pi_n\}$, 使得加工完所有作业所花的总流水时间 TFT 及最小完工时间 $C_{\max}$ 尽可能少.

该问题已被证明是 NP (Non-deterministic polynomial) 难度问题[1]. 因此, 精确方法[2−3] 在合理的时间内只能求解小规模问题, 而启发式算法能够在可接受的时间内, 求得问题近似最优解或最优解. 粒子群算法 (Partical swarm optimization, PSO) 是受鸟群觅食启发提出的一种元启发式方法, 其收敛速度快、易于实现, 被成功应用在多个领域中[4−6]. 目前, 应用 PSO 算法求解调度问题的研究还很少. 实验表明, 在求解调度问题时, 较 GA (Genetic algorithm) 算法更为有效, 但已提出的算法存在早熟收敛, 易陷入局部最优、进化后期算法收敛速度明显下降等缺点[7−8]. 这主要由于进化过程中群体能量不断下降, 导致粒子快速聚合, 群体多样性过低造成的. 为了克服这些不足, 本文提出了一种混合元启发式算法 (Hybrid particle and genetic algorithm, HPGA), 将 PSO 算法与 GA 算法[9] 结合在一起, 利用遗传操作不断引入新的信息指导群体进化. 为了能够有效度量群体当前的能量, 定义了群体活性指标, 当群体活性低于事先给定的阈值时, 对粒子的个体最优解进行变异, 以增加群体能量, 提高算法的全局搜索能力. 在更新每个粒子的个体最优解时, 引入了退火策略, 以此有效抑制粒子快速聚合, 保持群体的多样性. 为了进一步提高算法的求解质量, 提出了一种随机邻域搜索策略并将其引入到 HPGA 算法中, 用于对每个粒子的当前位置执行进一步的搜索. 最后将 HPGA 算法与最近提出的 GA[9] 算法 NPSO (Novel partical swarm optimization)[10] 算法进行了实验对比, 进一步证明了算法的有效性.

1 混合粒子群算法

为了使用 PSO 算法求解调度问题, Rameshkumar 提出了一种置换离散粒子群算法[7], 粒子在更新过程中只交换相应位置的元素, 而不引入新元素. 受其启发, 我们将置换的思想引入到所提出的算法中. 对于一个含有 n 个作业的流水调度问题, 粒子 i 的位置及速度均被表示为一个满足 "alldifferent" 约束[11] 的 n 维向量, 即所有作业的一个全排列, 然后结合 GA 算法中的交叉及变异操作不断更新粒子的位置及速度.

1.1 算法进化模型

在 PSO 算法中, 每个粒子的行为主要受其当前动量项、个体认知部分及群体认知部分的影响. 因此, 传统的粒子速度公式可通过将粒子的当前速度与其个体最优解及当前的群体最优解分别进行交叉来取代, 粒子的位置更新也相应地变为将粒子当前位置与当前速度交叉求得. 粒子更新公式可表示如下:

$$\boldsymbol{V}_i(\boldsymbol{k}+1) = \boldsymbol{V}_i(\boldsymbol{k}) \otimes \boldsymbol{P}_{gbest} \otimes \boldsymbol{P}_{ibest} \tag{1}$$

$$\boldsymbol{X}_i(\boldsymbol{k}+1) = \boldsymbol{X}_i(\boldsymbol{k}) \otimes \boldsymbol{V}_i(\boldsymbol{k}+1) \tag{2}$$

其中, 符号 $\otimes$ 表示交叉操作, 由式 (1) 和 (2) 可以看出, 每个粒子追随其当前个体最优解及全局最优解运动. 与传统的 PSO 算法一样, 它具有快速收敛, 计算简单等优点. 但是, 粒子的速度会迅速逼近零, 即粒子的当前速度和其当前位置相同, 易于陷入局部最优解[8]. 因为群体能量不断减小, 有用的指导信息不断丢失, 使得粒子没有能力跳出局部最优位置. 为此, 本文定义了衡量群体能量的群体活性指标, 当群体活性小于某个给定的阈值 α 时, 对粒子的个体最优解执行变异操作, 如式 (3) 所示.

$$\boldsymbol{P}_{ibest}(\boldsymbol{k}+1) = mutation(\boldsymbol{P}_{ibest}(\boldsymbol{k}+1)) \tag{3}$$

通过变异操作能够在群体中重新引入新的信息, 指导粒子搜索那些未曾搜索过的区域, 抑制算法的早熟收敛.

定义 1. 群体活性指标为能量小于给定常量 β ($\beta \in [0,1]$) 的粒子个数占群体规模的百分比. 给定粒子 P_i, 其能量根据当前位置 X_i 和当前速度 V_i 计算如下

收稿日期 2008-01-17 收修改稿日期 2008-04-11
Received January 17, 2008; in revised form April 11, 2008
国家自然科学基金 (60773097), 东北师范大学自然科学青年基金 (20081003) 资助
Supported by National Natural Science Foundation of China (60773397) and Science Foundation for Yong Teachers of Northeast Normal University (20081003)
1. 吉林大学符号计算与知识工程教育部重点实验室 长春 130012 2. 中国科学院长春光学精密机械与物理研究所 长春 130012
1. Key Laboratory of Symbol Computation and Knowledge Engineer, Ministry of Education, Jilin University, Changchun 130012 2. Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, Changchun 130012
DOI: 10.3724/SP.J.1004.2009.00332

$$energy(P_i)=\frac{\sum_{j=1}^{dim} same(X_i(j),V_i(j))}{dim},\quad j=1,\cdots,dim \tag{4}$$

其中

$$same(X_i(j),V_i(j))=\begin{cases}0, & 若\ X_i(j)=X_i(j)\\ 1, & 若\ X_i(j)\neq X_i(j)\end{cases} \tag{5}$$

可以看出群体的活性指标在一定程度上能够反映出当前群体所具有的全局搜索能力. 以上模型在迭代过程中群体多样性会不断减少, 影响群体的进化质量, 因此, 在粒子个体最优解的更新中引入了退火策略, 温度计算如下:

$$Temperature=TE\frac{\sum_{i=1}^{m}\sum_{j=1}^{n}p_{ij}}{nm10} \tag{6}$$

其中, TE 为一可调整参数, 可见温度与具体的问题相关.

1.2　最小完工时间的快速计算

定义 2. 对于给定的 n 个待加工作业, 每个作业包含 m 道工序, 调度 $\boldsymbol{S}=\{s_1,s_2,\cdots,s_n\}$ 的加工时间矩阵 T 为

$$T=\begin{pmatrix} t_{1,s_1} & t_{1,s_2} & \cdots & t_{1,s_n}\\ t_{2,s_1} & t_{2,s_2} & \cdots & t_{2,s_n}\\ \vdots & \vdots & \vdots & \vdots\\ t_{m,s_1} & t_{m,s_2} & \cdots & t_{m,s_n}\end{pmatrix} \tag{7}$$

其中, $s_i\in N$, $i=1,\cdots,n$. 根据以上定义调度 S 对应的最小完工时间, 可通过使用式 (8) 遍历矩阵 T 求得:

$$t_{i,j}=\begin{cases} t_{1,1}, & 若\ i=1,j=1\\ t_{1,j-1}+t_{1,j}, & 若\ i=1,j\neq 1\\ t_{i-1,1}+t_{i,1}, & 若\ i\neq 1,j=1\\ t_{i-1,j}+t_{i,1}, & 若\ t_{i-1,j}>t_{i,j-1}\\ t_{i,j-1}+t_{i,j}, & 若\ t_{i-1,j}<t_{i,j-1}\end{cases} \tag{8}$$

遍历完成后, 新计算出的 $t_{m,n}$ 的值就是调度 S 的最小完工时间.

1.3　算法描述及算法分析

在 HPGA 算法中, 群体的初始化采用随机的方式, 即在搜索空间中随机地产生粒子的初始位置和初始速度, 将每个粒子的历史最优位置设置为其当前位置, 并计算每个粒子当前位置所对应的求解目标的值, 将其作为粒子的适应度值. 根据算法的进化模型, HPGA 算法描述如下:

Algorithm HPGA
begin
initialize(pop);
$gFitness=C_{\max}(pop[1].current)$
for $i=2$ **to** $popNum$ **do**
  if $C_{\max}(pop[i].current)<gFitness$ **then**
    $gBest=pop[i].current$;
  endif
endfor
$currGen=0$;
while ($currGen<MAXGEN$) **do**
  for j =1 **to** $popNum$ **do**
    $pop[j].v[currGen+1]=pop[j].v[currGen]\otimes gBest$
      $\otimes pop[j].pBest$;
    $pop[j].x[currGen+1]=pop[j].x[currGen]\otimes$
      $pop[j].x[currGen]$
    $pop[j].fitness=C_{\max}(pop[j].current)$;
    if $pop[j].fitness<C_{\max}(pop[j].pBest)$ **then**
      $pop[j].pBest=pop[j].current$;
    elseif $random\leq$exp{-($C_{\max}(pop[j].current)$-
        $C_{\max}(pop[j].pBest)$)/Temperature}
      $pop[j].pBest=pop[j].current$;
    endifelse
    endif
    if $pop[j].fitness<C_{\max}(gBest)$ **then**
      $gBest=pop[j].current$;
    endif
  endfor
  $curActivity=$ swarmActivityComp(pop);
  if $curActivity<\beta$ **then**
    for k=1 **to** $popNum$ **do**
      $pop[k].pBest=$mutation($pop[k].pBest$);
    endfor
  endif
  $currGen$++;
endwhile
return $gBest$; **end**

其中, $MAXGEN$ 表示算法的最大迭代次数; $currGen$ 及 $popNum$ 分别为当前迭代次数和群体规模; $random$ 表示 $[0,1]$ 之间的随机数; $curActivity$ 表示当前的群体活性. 收敛通常是指一个系统或过程达到一个稳定状态, 对基于群体的优化算法来说, 算法的收敛可以根据群体的行为来定义[12].

定义 3. 给定待求解的调度问题 p, 其搜索空间 Ω, $gBest\in\Omega$ 为算法在时间 t 或群体的第 t 次进化中求得的最优位置, $gBest^*$ 为 Ω 中的一个固定位置, 收敛的定义可记为

$$\lim_{t\to\infty} gbest(t)=gbest^*$$

上式表明, 如果由算法求得的 $gbest$ 不再变化, 那么就说处于收敛状态. 如果 $gbest$ 为搜索空间的全局最佳位置, 则算法获得了全局最优收敛, 否则算法陷入局部最优位置.

定理 1. HPGA 算法是收敛的.

证明. 将群体中的每个粒子的当前位置视为一个状态, 则所有的粒子当前位置的集合可视为一种状态分布. 这种状态分布会随算法的运行而改变. 由于 HPGA 算法的运行具有随机性, 其基本操作只与当前状态有关, 是无后效性的, 因此可以把群体内的个体视为一个具有不同状态的随机变量的概率分布.

首先证明由式 (1) 和 (2) 构成的迭代过程是收敛的. 设群体规模为 m, 问题规模为 l, 群体的状态记为 $(p_1,p_2,\cdots,p_m)$, 最佳调度为 p^*. 所有的群体状态构成一个 $1\times N$ 的向量, 其中 $N=(m\times l)!$, 这个行向量的每个元素由排在一起的 m 个粒子的当前位置构成, 可表示为 $\{(p_1^{(1)},p_2^{(1)},\cdots,p_m^{(1)}),(p_1^{(2)},p_2^{(2)},\cdots,p_m^{(2)}),\cdots,(p_1^{(N)},p_2^{(N)},\cdots,p_m^{(N)})\}$ 假设经过若干代进化后, 已达到种群最优状态 p^*, 不妨设其排在状态向量的第一个分量处, 即 $\{(p_1^{(*)},p_2^{(*)},\cdots,p_m^{(*)}),(p_1^{(2)},p_2^{(2)},\cdots,p_m^{(2)}),\cdots,(p_1^{(N)},p_2^{(N)},\cdots,p_m^{(N)})\}$, 根据粒子的速度及位置更新公式可知, 此时处在最优位置粒子的速度及历史最优位置均为 p^*, 在下一次迭代中求得的粒子当前位置不变. 状态转移矩阵可写为

$$C=\begin{pmatrix}1 & \emptyset\\ C_{21} & C_{22}\end{pmatrix}$$

其中, C 为随机矩阵, 每一行至少有一个正的元素, $\emptyset$ 为 $N-1$ 维向量. 显然 C 为可约随机矩阵, 且 C_{21}、C_{22} 不是零矩阵, 根据可约随机矩阵的性质有: $C^{\infty}=(C_1^{\infty},0,\cdots,0)$ 其中 $C_1^{\infty}>0$. 因为 C^{∞} 可看作是一个状态分布, 所以有 $C_1^{\infty}=1$, 于是 $C^{\infty}=(1,0,\cdots,0)$, 即收敛到 p^*.

当群体活性大于 β 时, HPGA 算法等同于式 (1) 和 (2) 构成的迭代收敛过程; 而当群体活性小于 β 时, 可看作由式 (1) 和 (2) 构成的迭代过程已经收敛, 此刻虽引入变异操作, 但只是对每个粒子的个体历史最优位置进行变异, 并没有改变当前已经得到的历史全局最优位置, 而且在以后的进化中, 只有当得到的位置优于此最优位置时才会被取代, 即全局最优位置总是朝着更好的位置进化. 所以, 根据定义 3, HPGA 算法是收敛的. □

定理 2. HPGA 算法求得的解是有效的.

证明. 算法中使用的交叉及变异操作可参见文献 [9], 每种操作都满足 "alldifferent" 约束, 都是一个有效的调度, 即 HPGA 算法的搜索空间是由所有有效调度构成的, 所以由 HPGA 算法求得的解必然有效. □

定理 3. HPGA 算法的空间复杂度为 $O((3n+2m+2)d)$, 群体每次进化的最坏渐近时间复杂度为 $O(mnd^2)$. 其中 n 为群体规模, d 表示作业数量, m 为处理机个数.

证明. 在 HPGA 算法运行过程中需要存储每个粒子的当前位置、个体最佳位置及速度, 令群体规模为 n, 则它所消耗存储空间为 $O(3nd)$; 在求解粒子适应度时还需要存储处理时间矩阵, 所消耗的空间为 $O(2md)$, 每次执行交叉及变异等操作时还需要一个存储新位置或速度的空间, 此外还需要存储一个全局最优调度; 所以 HPGA 算法的空间复杂度为 $O((3n+2m+2)d)$. 算法运行时, 执行一次交叉操作消耗的时间最多为 $O(2d-1)$, 变异操作最多为 $O(d)$, 所以在 HPGA 算法的一次迭代过程中, 由交叉或变异引起的最坏时间复杂度为 $O(2n(2d-1))$; 计算群体活性时, 需要访问每个粒子并计算其能量, 由此引起的时间复杂度为 $O(nd)$; 求解每个粒子适应度时需要求得及遍历与之相应的时间矩阵, 其时间复杂度 $O(2md)$. 由于在一次迭代中需要计算所有粒子的适应度, 那么需要消耗的时间就变为 $O(2mnd)$, 并且最坏情况下, 在一次迭代中每个粒子都需要更新其个体最优位置时, 此时所消耗的时间就变为 $O(2mnd^2)$. 当问题规模逐渐增大即 mn 的值趋于无穷大时 $2mnd^2 \gg 5nd \gg 2n$, HPGA 算法的最坏渐近时间复杂度为 $O(mnd^2)$. □

1.4 算法改进

实验中发现, 进化后期算法收敛速度明显下降, 当接近最优解时, 易于停止进化. 为此 HPGA 算法中引入一种基于邻域的局部搜索策略, 以进一步提高解的质量.

定义 4. 给定一个含有 n 个作业的调度问题, 它的任意一个有效调度 $\boldsymbol{\pi}=\{\pi_1,\cdots,\pi_n\}$ 可看作是 n 维空间中的一个点, 通过随机选择一个作业 π_i $(1\le i\le n)$, 将其插入到任意其他位置, 可求得一个新的有效调度, 即 n 维空间中一个新的点. 所有以此方式求得的点就构成了调度 π 的邻域, 显然此邻域共包含 $n(n-2)+1$ 个点.

根据以上定义, 我们构造了如下的一种贪心邻域策略:

Procedure NeighborhoodBestSearch($\boldsymbol{\pi}$)
begin
for i =1 **to** n **do**
  $\boldsymbol{\pi}'$ = best schedule obtained by inserting π_i in any position of $\boldsymbol{\pi}$
  if $C_{\max}(\boldsymbol{\pi}')<C_{\max}(\boldsymbol{\pi})$ **then**
    $\boldsymbol{\pi}=\boldsymbol{\pi}'$
  endif.
endfor.
return $\boldsymbol{\pi}$; **end**

在 HPGA 算法群体的每次进化过程中, 我们使用上面的局部搜索算法, 进一步搜索每个粒子当前位置的邻域, 由此得到的算法我们称之为 L-HPGA 算法.

2 实验仿真与结果分析

本节首先分析了 HPGA 中各种参数对算法性能的影响, 然后在不同规模的 Tailliard 数据集[13] 上对 HPGA 及 L-HPGA 算法进行了测试, 并与最近提出的 GA 算法[9] 及 NPSO 算法[10] 进行了实验对比. 为说明算法每次求得的最优解与已知最优解的差距, 我们定义了算法所求得的最优解的相对偏离度 (Relative percentage deviation, RPD), 即算法每次运行得到的最优解与已知最优解的差的均值, 计算如下:

$$RPD=\frac{\sum_{i=1}^{L}(Sol_i-Optimal)}{L} \tag{9}$$

其中, L 为针对某个问题算法的运行次数, Sol_i 表示算法每次运行求得的最好解, $Optimal$ 表示已知最优解, 它是算法求得的最优解偏离已知最优解平均程度的指标, 能够反映出算法的平均求解质量. 当求解目标为总流水时间时, $Optimal=\min\{Sol_i\}$, $i=1,\cdots,L$. 本文实验中每个测试问题上各算法运行次数都为 20, 即 $L=20$.

2.1 各种参数对算法性能的影响

在 HPGA 算法中, 主要存在两个因素对算法性能影响较大, 退火策略中的温度 TE 和群体活性阈值. 为了测试群体活性阈值对算法性能的影响, 令 $TE=0.5$ 并将群体活性的值分别设置为 0.05, 0.15, 0.45, 在每个所选测试集分别运行 HPGA 算法 20 次. 实验中采用两点交叉策略, 群体规模设置为 60, 最大迭代次数设为 600, 另外, 还分别测试了六种变异操作对算法的影响. 这六种变异操作分别是近邻变异 (M1)、随机交换变异 (M2)、移位变异 (M3)、置换变异 (M4)、逆转变异 (M5)、逆转/置换变异 (M6). 图 1 给出了采用不同的群体活性阈值及各种变异策略时置信区间为 0.95 的双因子方差统计分析结果, 响应变量 ARPD (Average relative percentage deviation) 为测试集上求得的总流水时间和最小完工时间的相对偏离度的平均值.

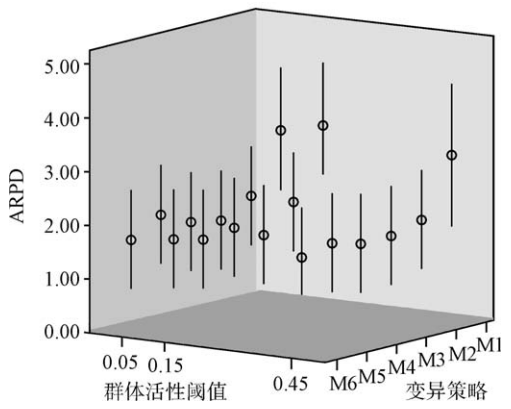

图 1 变异策略及群体活性阈值对算法性能影响的统计分析图

Fig. 1 The statistical analysis result of using different mutation strategies and swarm activity thresholds

从图 1 中可以看出, 对于不同的群体活性阈值, 采用移位变异粗略 M3 时, 算法的性能都是最好的. 当变异策略采用 M3 时, 群体活性阈值取 0.15 时算法的性能最佳, 即求解的平均偏离度的均值和标准差最小. 此外, 对于同一问题, 选取不同的变异操作, 群体活性值对算法性能的影响是不一样的, 对于问题 ta060, 当变异操作为 M6 时, 其值取 0.05 较好, 而当变异操作为 M5 时, 其值取 0.45 较好, 这主要是由于每种变异操作执行一次所引入的新的信息量不同造成的. 在测试参数 T 算法性能的影响时, 本文将群体活性阈值设置为 0.15. 在不同规模问题上, TE 取不同值时, 算法在每个问题上运行 20 次时求得的 RPD 如表 1 所示. 可以看出, 对于小规模问题如 ta020, $TE = 0.25$ 时算法求得的 RPD 最小, 对于其他规模的问题 $TE = 0.50$ 时, 求得的 RPD 最小, 即算法的平均求解质量最佳. 通过对算法的分析可知, 当 TE 的值比较小时, 粒子个体最优解更新比较频繁, 降低了算法的收敛速度; 当 TE 值较大时, 粒子个体最优更新较慢, 不能及时引入新信息指导群体进化, 使群体易于陷入局部最优位置. 从表 1 中可知, $TE = 0.5$ 是可以接受的.

2.2　与其他算法的比较

为了进一步说明 HPGA 算法与 L-HPGA 算法的有效性, 我们将其与最近提出的 GA 算法[9] 及 NPSO 算法[10] 进行了比较, 为了方便, 所有算法中采用的遗传操作均为两点交叉和移位变异, 群体规模设为 60, 最大迭代次数为 600. 通过对算法的分析可知, 此时 GA 算法、NPSO 算法及 HPGA 算法具有相同的时间复杂度. 实验中, 在每个测试问题上, 针对每个目标、每个算法均运行 20 次, 所得结果分别在表 2 和表 3 中给出. 可以看出在所有的测试集上, 无论求解目标是最小完工时间, 还是总流水时间, L-HPGA 算法的求解质量最高, 所求得的最优解、最优解均值、最差解及 RPD 都最小, 对于问题 ta010, L-HPGA 算法具有全局最优收敛性, 而 HPGA 算法的求解质量也明显优于另外两种算法. 图 2 是群体规模为 60 时, HPGA 算法与 L-HPGA 算法进化一代所消耗的平均时间比较. 各种算法在 Ta080 上的最优解均值的收敛曲线如图 3 (见下页) 所示. 图 3 (a) 表示求解目标为最小完工时间时的平均进化曲线, 图 3 (b) 为求解目标是总流水时间时的平均进化曲线. 对于所有的测试问题, L-HPGA 算法的收敛速度都快于其他算法, 而且所求得的最优解均值的质量也都明显是最优的, HPGA 算法次之. 但在 L-HPGA 算法中, 由于每次进化都执行局部搜索策略, 因此其求解速度低于 HPGA 算法.

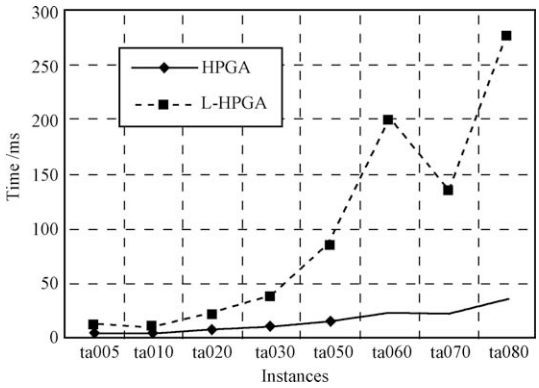

图 2　HPGA 算法与 L-HPGA 算法的平均求解时间对比

Fig. 2　Comparative results of times for HPGA and L-HPGA

表 1　TE 取不同值时 HPGA 算法的对比结果

Table 1　Comparative results with different values of TE

Problems	Size	$TE = 0.0$	$TE = 0.25$	$TE = 0.50$	$TE = 0.75$	$TE = 1.0$
ta020	20 × 10	1.270	**1.266**	1.304	1.364	1.398
ta050	50 × 10	3.323	3.346	**3.284**	3.378	3.383
ta080	100 × 10	1.415	1.340	**1.234**	1.673	1.311

表 2　求解目标为总流水时间时, 与 GA 及 NPSO 算法的求解结果比较

Table 2　Comparative result with GA and NPSO algorithms with the objective of minimizing total flow time

Problem	GA	NPSO	HPGA	L-HPGA
ta005	13618/14358.5/14493 (6.13)	13529/13529.0/13529 (0.00)	13529/13529.0/13529 (0.00)	13529/13529.0/13529 (0.00)
ta010	13036/13889.2/14300 (7.31)	12943/13005.1/13021 (0.47)	12943/12943.0/12943 (0.00)	12943/12943.0/12943 (0.00)
ta020	21506/22158.6/22722 (3.93)	21320/21329.5/21352 (0.04)	21320/21321.5/21323 (0.01)	21320/21320.0/21320 (0.00)
ta030	32465/33024.5/34068 (2.36)	32317/32856.4/ 33418 (1.84)	32262/32506.8/33281 (0.75)	32262/32262.0/32262 (0.00)
ta050	89632/93257.2/99552 (5.66)	88631/89192.9/90352 (1.06)	88352/88951.6/89470 (0.79)	88256/88412.2/88906 (0.17)
ta060	127642/132091.6/137056 (6.07)	126582/126958.4/127823 (1.95)	125767/126819.2/127196 (1.84)	124529/124907.6/125208 (0.30)
ta070	246655/258936.2/270922 (6.00)	246261/253058.1/267422 (3.59)	244903/252343.8/258906 (3.30)	244275/245347.6/245735 (0.44)
ta080	298308/315056.5/330834 (6.73)	297028/298793.2/308901 (1.23)	296901/298590.3/308901 (1.16)	295170/297625.8/298562 (0.83)

表 3　求解目标为最小完工时间时, 与 GA 及 NPSO 算法的求解结果比较

Table 3　Comparative result with GA and NPSO algorithms with the objective of minimizing makespan

Problem	GA	NPSO	HPGA	L-HPGA
ta005	1250/1251.2/1258 (1.31)	1250/1250.6/1256 (1.26)	1235/1243.5/1250 (0.68)	1235/1239.8/1244 (0.38)
ta010	1116/1135.6/1161 (2.49)	1108/1117.1/1131 (0.82)	1108/1113.4/1120 (0.49)	1108/1108.0/1108 (0.00)
ta020	1618/1632.8/1654 (2.62)	1610/1629.0/1654 (2.38)	1594/1614.6/1629 (1.48)	1591/1602.7/1611 (0.73)
ta030	2212/2237.5/2282 (2.73)	2185/2220.3/2250 (1.94)	2192/2209.4/2232 (1.44)	2179/2184.5/2194 (0.29)
ta050	3189/3225.5/3280 (5.23)	3167/3192.2/3213 (4.15)	3139/3163.0/3191 (3.19)	3099/3114.3/3131 (1.60)
ta060	3971/4008.6/4093 (6.72)	3900/3942.4/3979 (4.96)	3852/3919.4/3971 (4.35)	3783/3807.8/3854 (1.37)
ta070	5342/5372.3/5403 (0.94)	5334/5351.1/5390 (0.54)	5342/5343.8/5346 (0.40)	5322/5323.2/5328 (0.02)
ta080	5953/6056.3/6106 (3.61)	5903/5934.9/5988 (1.53)	5881/5900.8/5903 (0.95)	5864/5892.9/5903 (0.81)

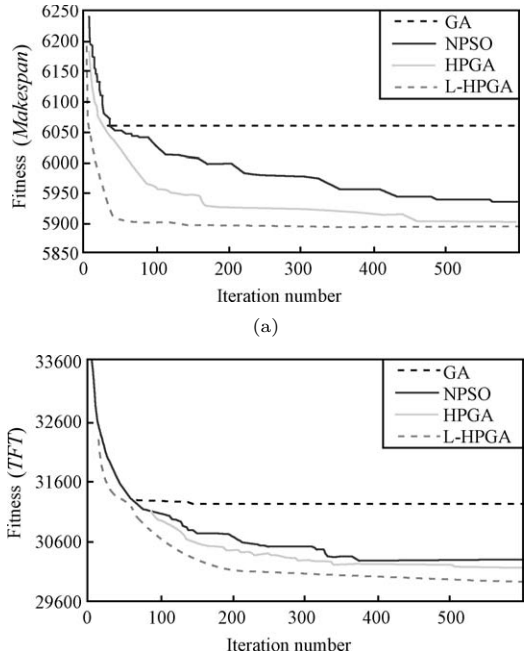

图 3 各种算法的平均进化曲线比较

Fig. 3 Evolution curves of different algorithms

可以看出, 问题的求解空间越复杂, 这两种算法的求解时间差越大. 虽然通过使用文献 [14] 中提出的加速方法可以极大地提高其运行速度, 但还是略慢于 HPGA 算法. 所以在实际应用当中, 是否在 HPGA 算法中引入局部搜索策略, 需要在求解质量与求解速度之间进行权衡. 此外, 对于本文提出的算法, 每代的个体之间是相互独立的, 因此可以利用这种并行特征, 使用 GPU 对算法进行加速, 进一步提高算法的计算效率.

3 结论

本文提出了一种求解流水调度问题的混合算法, 将遗传操作、退火策略结合到粒子群算法中, 定义了群体活性指标以及通过遍历矩阵的方式快速计算最小完工时间, 此外还分析了算法的收敛性及复杂度. 最后, 将该算法在 8 个不同规模的问题上进行了测试, 并与当前国际文献中最近提出的两种著名算法进行了比较, 结果表明无论是求得的解的质量, 还是算法的稳定性均明显好于这两种算法, 加入局部搜索策略后效果更加明显. 下一步工作主要集中在测试其他交叉策略对算法的影响, 找出交叉操作与变异操作的最佳组合, 以及使用本文提出的算法解决其他组合优化问题.

References

1 Garey M R, Johnson D S, Sethy R. The complexity of flow shop and job shop scheduling. *Mathematics of Operations Research*, 1976, **1**(2): 117−129

2 Dimopoulos C, Zalza A M S. Recent developments in evolutionary computation for manufacturing optimization: problems, solutions, and computations. *IEEE Transactions on Evolutionary Computation*, 2000, **4**(2): 93−113

3 Valente J M S, Alves R A F S. An exact approach to early/tardy scheduling with release dates. *Computers and Operations Research*, 2005, **32**(11): 2905−2917

4 Yang Q Y, Sun J G, Zhang J Y, Wang C J. A hybrid discrete particle swarm algorithm for open-shop problems. In: Proceedings of the 6th International Conference on Simulated Evolution and Learning. Hefei, China: Springer, 2006. 158−165

5 He S, Wu Q H, Wen J Y, Saunders J R, Paton R C. A particle swarm optimizer with passive congregation. *Biosystems*, 2004, **78**(1-3): 135−147

6 Kao Y T, Zahara E, Kao I W. A hybridized approach to data clustering. *Expert Systems with Applications: An International Journal*, 2008, **34**(3): 1754−1762

7 Rameshkumar K, Suresh R K, Mohanasundaram K M. Discrete particle swarm optimization algorithm for permutation flowshop scheduling to minimize makspan. In: Proceedings of International Conference on Advances in Natural Computation. Changsha, China: Springer, 2005. 572−581

8 Lian Z G, Gu X S, Jiao B. A similar particle swarm optimization algorithm for permutation flowshop scheduling to minimize makespan. *Applied Mathematics and Computation*, 2006, **175**(1): 773−785

9 Nearchou A C. The effect of various operators on the genetic search for large scheduling problems. *International Journal of Production Economics*, 2004, **88**(2): 191−203

10 Lian Z G, Gu X S, Jiao B. A novel particle swarm optimization algorithm for permutation flowshop scheduling to minimize makespan. *Chaos, Solitons and Fractals*, 2008, **35**(5): 851−861

11 Lauriere J L. A language and a program for stating and solving combinatorial problems. *Artificial Intelligence*, 1978, **10**(1): 29−127

12 van den B F. An Analysis of Particle Swarm Optimizers [Ph. D. dissertation], University of Pretoria, South Africa, 2002

13 Taillard E. Some efficient heuristic methods for the flow shop sequencing problem. *European Journal of Operational Research*, 1990, **47**(1): 65−74

14 Pan Q K, Tasgetiren M F, Liang Y C. A discrete particle swarm optimization algorithm for the no-wait flowshop scheduling problem. *Computers and Operations Research*, 2008, **35**(9): 2807−2839

张长胜 吉林大学计算机学院博士研究生. 主要研究方向为智能信息处理. 本文通信作者. E-mail: zcs820@yahoo.com.cn

(**ZHANG Chang-Sheng** Ph. D. candidate at the College of Computer Science and Technology, Jilin University. His main research interest is intelligent information processing. Corresponding author of this paper.)

孙吉贵 吉林大学符号计算与知识工程教育部重点实验室教授. 主要研究方向为自动推理、约束程序及智能规划.
E-mail: jiguisun@jlu.edu.cn

(**SUN Ji-Gui** Professor at the Key Laboratory of Symbol Computation and Knowledge Engineer, Ministry of Education, Jilin University. His research interest covers automatic reasoning, constraint programming, and intelligent planning.)

杨轻云 中国科学院长春光学精密机械与物理研究所副研究员. 主要研究方向为群智能. E-mail: qyyang@yahoo.com.cn

(**YANG Qing-Yun** Associate professeur at the Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences. His main research interest is swarm intelligence.)

郑黎辉 吉林大学计算机科学与技术学院硕士研究生. 主要研究方向为智能信息处理. E-mail: zhenglihuii@163.com

(**ZHENG Li-Hui** Master student at the College of Computer Science and Technology, Jilin University. His main research interest is intelligent information processing.)