

带释放时间的并行机调度问题的 ILS & SS 算法¹⁾

罗家祥¹ 唐立新²

¹(东北大学教育部暨辽宁省流程工业综合自动化重点实验室 沈阳 110004)

²(东北大学信息科学与工程学院 沈阳 110004)

(E-mail: neuluojia@163.com)

摘 要 研究工件带释放时间的两类并行机最小化总完成时间的调度问题. 针对问题提出了一种新的基于变深度环交换邻域结构的 Iterated local search(ILS) 算法. 1) 提出了变深度环交换邻域结构. 2) 基于变深度环交换和传统 Swap 的混合邻域, 提出了带有两种 *kick* 策略的 ILS 算法. 3) 为了加强 ILS 逃出局部最优的能力, 将 Scatter search (SS) 搜索方法引入了 ILS 算法中; 算法将当前最好解和次好解进行分散处理, 再从处理后的解开始继续迭代. 为了验证算法的有效性, 对两类并行机问题分别随机产生 100 组数据进行试验. 实验结果表明: 对于同构并行机问题, 引入 SS 的 ILS 算法的计算结果与下界的平均偏差为 0.99%, 而没有引入 SS 的 ILS 算法的为 1.06%; 对于无关并行机问题, 引入 SS 搜索方法后, ILS 算法的计算结果改进了 6.06%, 并明显优于多点下降算法.

关键词 并行机, 变深度环交换, ILS 算法, SS 算法

中图分类号 N49

A New ILS & SS Algorithm for Parallel-machine Scheduling Problem

LUO Jia-Xiang¹ TANG Li-Xin²

¹(Key Laboratory of Process Industry Automation, Ministry of Education, Shenyang 110004)

²(School of Information Science and Engineering, Northeastern University, Shenyang 110004)

(E-mail: neuluojia@163.com)

Abstract The problems of scheduling jobs with different ready time on parallel machines to minimize the total completion time are addressed. A new iterated local search (ILS) is proposed based on a new neighborhood structure named variable-depth cycle exchange. First, we define the variable-depth cycle exchange neighborhood structure. Based on the variable-depth cycle exchange neighborhood and the traditional Swap neighborhood, a new ILS algorithm with two kick strategies is then proposed. Scatter search (SS) is embedded into ILS to enhance its power of getting away from the local optima, in which the algorithm continues work after combining the best and the second best solutions found so far. Computational experiments on 10 problems with up to 50 jobs and 20 machines for unrelated parallel machines and identical machines respectively are carried out to test the performance

1) 国家杰出青年科学基金 (70425003), 国家自然科学基金 (70171030, 60274049), 高等学校优秀青年教师教学科研奖励计划 (教育司 [2002]383) 资助

Supported by the Excellent Young Faculty Program of the Ministry of Education, P. R. China (70425003), National Natural Science Foundation of P. R. China (70171030, 60274049), the Science Foundation for Excellent Youth Teachers of Universities by the Ministry of Education of P. R. China ([2002]383)

收稿日期 2004-5-31 收修改稿日期 2005-3-17

Received May 31, 2004; in revised form March 17, 2005

of the algorithm. For the identical parallel-machine scheduling problem, the average deviation of the ILS with SS to the lower bound obtained by Lagrangian Relaxation is 0.99%, but 1.06% for the ILS without SS. For the unrelated parallel-machine scheduling problem, the average performance of the ILS with SS make an improvement 6.06% over that of the ILS without SS, and also is better than that of multi-start descent algorithm.

Key words Parallel-machine scheduling, variable-depth cycle exchange neighborhood, iterated local search, scatter search

1 引言

本文研究了两类并行机——同构并行机和无关并行机上工件带释放时间的调度问题, $P|r_i|\Sigma C_i$ 和 $R|r_i|\Sigma C_i$. 设带有释放时间的工件集合 $J = \{1, 2, \dots, n\}$ 里的工件将在并行机器集合 $M = \{M_1, M_2, \dots, M_m\}$ 里的任意台机器上处理. 工件处理不可中断; 每个工件在同一时刻只能由一台机器处理; 每台机器在同一时刻最多只能处理一个工件; 工件 i 的释放时间为 r_i , 在机器 M_k 上的处理时间为 p_{ik} . 目标函数是最小化总完成时间.

Lenstra 等证明了单机条件下最小化带释放时间的工件的总完成时间的调度问题为强 NP 难题^[1]. 当机器条件由单机换成并行机时, 问题变得更为复杂, 每台机器上工件的排序都是强 NP 难题. 探讨此类问题的近似算法成为主要的解决方法. 目前, 对 $P|r_i|\Sigma C_i$ 的研究如 Hall 等人提出的 $(2.89 + \varepsilon)$ 的近似算法^[2], 而对 $R|r_i|\Sigma C_i$ 问题的研究还较少.

影响近似算法的两个重要因素是邻域结构和搜索策略. 一般来讲, 邻域结构越好, 搜索出的局部最优解的质量越好. 传统邻域有交换 (Swap) 和插入 (Insert)^[3], 大规模邻域有 k-opt 移动邻域、发射链邻域和环交换邻域^[4,5] 等. 本文的变深度环交换邻域基于环交换而建立. 变深度环交换与一般环交换的区别在于前者包含多个相关的环交换. 与一般环交换相比, 它不依赖于环中所有元素移动时才能确定该环交换是否能改进目标函数.

ILS 算法是一个稳定性好的新搜索方法. 它对局部最优解进行 *kick* 操作后, 继续搜索. 一些近期的研究^[6,7] 表明了它的有效性. 另一方面, SS 具有较强的跳出局部最优的能力^[8,9]. 本文将 SS 引入 ILS 算法, 尽可能的改进了历史最好解, 增强了算法跳出局部最优的能力.

上述引入 SS 的 ILS 算法与传统的 ILS 算法相比, 前者嵌入了 SS 的组合机制, 提高了 ILS 算法的搜索性能; 同时算法中采用了新的基于变深度环交换的邻域结构, 在邻域结构上有所创新. 本文解决了两类并行机 (同构并行机与无关并行机) 的最小化工件总完成时间问题. 它们在算法上的区别体现在有关工件处理时间的计算上. 对于同构并行机, 工件处理时间不发生变化; 而工件在不同的无关并行机上处理时有不同的处理时间.

2 变深度环交换邻域结构

2.1 变深度环交换

由于并行机问题可归结为集划分问题, 因此以并行机为例, 定义适用于集划分问题的变深度环交换邻域. 集合 $J = \{1, 2, \dots, n\}$, $I_k (k = 1, 2, \dots, m)$ 为 J 的子集, $I_1 \cup I_2 \cup \dots \cup I_m = J$, $I_j \cap I_k = \emptyset (\forall j, k \in \{1, 2, \dots, m\}, j \neq k)$; $I_{\sigma(i)}$ 表示工件 i 所在的集合, $\sigma(i)$ 表示工件 i 所在集合的下标. I_k 表示在机器 M_k 上处理的工件集合; $C[I_k]$ 表示集合 I_k 的费用, 表示在机器 M_k 上处理的工件的总完成时间. 整个并行机问题的目标函数为最小化 $\Sigma_m C[I_m]$. S 表示 J 的集合划分 $\{I_1, I_2, \dots, I_m\}$, 关于 S 的变深度环交换定义如下.

定义 1. 变深度环交换是一个序列 $W = (i_1, i_2, \dots, i_m)$, 其中任意两个工件 i_j 和 $i_k (j \neq k)$ 属于不同集合, 即 $\sigma(i_j) \neq \sigma(i_k), \forall i_j, i_k \in J$. W 代表如下一系列操作: 工件 i_1 从 $I_{\sigma(i_1)}$ 移

到 $I_{\sigma(i_2)}$, i_2 从 $I_{\sigma(i_2)}$ 移到 $I_{\sigma(i_1)}$, 构成环 $W_1 = (i_1, i_2, i_1)$; 工件 i_1 从 $I_{\sigma(i_1)}$ 移到 $I_{\sigma(i_2)}$, i_2 从 $I_{\sigma(i_2)}$ 移到 $I_{\sigma(i_3)}$, i_3 从 $I_{\sigma(i_3)}$ 移到 $I_{\sigma(i_1)}$, 构成环 $W_2 = (i_1, i_2, i_3, i_1)$; 以此类推, 直到环 $W_{m-1} = (i_1, i_2, \dots, i_m, i_1)$ 形成. 定义 i_1 为变深度环的初始工件. 环 $W_j (j = 1, \dots, m-2)$ 称为虚环, 环 W_{m-1} 称为实环, 如图 1 所示. 整个过程中, 环的工件个数逐次增加, 因此称为变深度环交换.

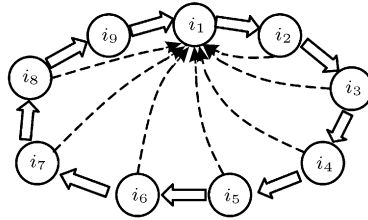


图 1 变深度环交换

Fig. 1 The variable-depth cycle exchange

在变深度环交换 $W = (i_1, i_2, \dots, i_m)$ 中, 若存在任何一个可以使目标函数改进的环 $W_j (j = 1, \dots, m-1)$, 我们都称为该变深度环交换为负费用的变深度的环交换.

性质 1. 对于一个包含 m 个节点的变深度环, 实环的个数为 1, 虚环的个数为 $m-2$.

性质 2. 在变深度环交换 $W = (i_1, i_2, \dots, i_m)$ 中, 环 $W_j (2 \leq j \leq m-1)$ 是通过 W_{j-1} 扩展工件 i_{j+1} 得到的. 用符号表示为: $W_j = W_{j-1} + \{i_{j+1}\}$.

变深度环交换试图使实环中的每个工件 (除初始工件) 都对应一个环. 这些环所包含的工件数逐次增加, 体现了“变深度”的含义. 若每个集合里有 t 个工件, 则集合 S 的邻域大小为: $t^m m!$. 这里 $m!$ 是变深度环中 m 个工件的全排列数目, t^m 是从 m 个集合中可选择工件的组合数目. 显然变深度环交换邻域是指数型的大规模邻域.

在各子集里添加一个无实际意义的工件, 参与变深度环交换, 称为加入虚拟工件的变深度环交换. 加入虚拟工件后的邻域规模扩大, 包含了多个插入. 以下算法中涉及到的变深度环交换邻域中都加入了虚拟工件.

2.2 动态规划方法搜索变深度环交换邻域

为搜索负费用的变深度环交换, 需建立一个辅助图 $G(S)$. 辅助图里的节点与工件一一对应. 当工件 i 和 j 不属于同一集合时, 它们对应的节点间就存在边 (i, j) 连接. 如图 2.

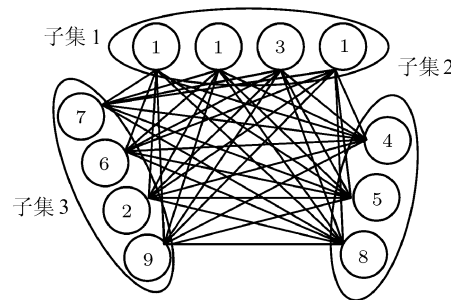


图 2 辅助图

Fig. 2 The auxiliary graph

根据 $G(S)$ 建立动态规划算法, 把负费用变深度环交换的搜索转换为对负长度路径的搜索. 设搜索的集合序列为 (I_1, I_2, \dots, I_m) ; 以 I_1 里的工件对应的节点为初始点开始搜索.

首先定义边的费用. 由于边的费用随着变深度环交换初始节点的变化而变化, 定义边

(i, j) 以 $\beta \in I_1$ 为初始节点的费用为:

$$\text{cost}_{t_{\beta}(i,j)} = C[I_{\sigma(\beta)} + \{j\} - \{\beta\}] + C[I_{\sigma(j)} + \{i\} - \{j\}] - C[I_{\sigma(\beta)} + \{i\} - \{\beta\}] - C[I_{\sigma(j)}] \quad (1)$$

根据边的费用定义得出一个巧妙的结论. 设从初始点 β 到节点 j , 历经集合 $(I_1, I_2, \dots, I_{\sigma(j)})$ 的一条路径为 $(\beta, t_{\beta,j}(2), \dots, t_{\beta,j}(h-1), t_{\beta,j}(h), t_{\beta,j}(h+1), \dots, j)$, 其中, $t_{\beta,j}(h) \in I_h (1 < h < \sigma(j))$ 表示该路径上的节点. 有如下命题成立.

命题 1. 路径 $(\beta, t_{\beta,j}(2), \dots, t_{\beta,j}(h-1), t_{\beta,j}(h), t_{\beta,j}(h+1), \dots, j)$ 的费用长度为进行环交换 $(\beta, t_{\beta,j}(2), \dots, t_{\beta,j}(h-1), t_{\beta,j}(h), t_{\beta,j}(h+1), \dots, j, \beta)$ 后目标函数的改进量.

证明. 设 $\text{cost}_{t_{\beta}(j)}$ 为路径 $(\beta, t_{\beta,j}(2), \dots, t_{\beta,j}(h-1), t_{\beta,j}(h), t_{\beta,j}(h+1), \dots, j)$ 的费用长度, 因此,

$$\begin{aligned} \text{cost}_{t_{\beta}(j)} &= \text{cost}_{t_{\beta}(\beta, t_{\beta,j}(2))} + \text{cost}_{t_{\beta}(t_{\beta,j}(2), t_{\beta,j}(3))} + \dots + \text{cost}_{t_{\beta}(t_{\beta,j}(\sigma(j)-1), j)} = \\ &= (C[I_1 - \{\beta\} + \{t_{\beta,j}(2)\}] + C[I_2 - \{t_{\beta,j}(2)\} + \{\beta\}] - C[I_1] - C[I_2]) + (C[I_1 - \{\beta\} + \{t_{\beta,j}(3)\}] + \\ &+ C[I_3 - \{t_{\beta,j}(3)\} + \{t_{\beta,j}(2)\}] - C[I_1 - \{\beta\} + \{t_{\beta,j}(2)\}] - C[I_3]) + \dots + (C[I_1 - \{\beta\} + \{j\}] + \\ &+ C[I_{\sigma(j)} - \{j\} + \{t_{\beta,j}(\sigma(j)-1)\}] - C[I_1 - \{\beta\} + \{t_{\beta,j}(\sigma(j)-1)\}] - C[I_{\sigma(j)}]) = \\ &= C[I_1 - \{\beta\} + \{j\}] + C[I_2 - \{t_{\beta,j}(2)\} + \{\beta\}] + C[I_3 - \{t_{\beta,j}(3)\} + \{t_{\beta,j}(2)\}] + \dots + \\ &+ C[I_{\sigma(j)} - \{j\} + \{t_{\beta,j}(\sigma(j)-1)\}] - C[I_1] - C[I_2] - \dots - C[I_{\sigma(j)}] \end{aligned}$$

因此, $\text{cost}_{t_{\beta}(j)}$ 为进行环交换 $(\beta, t_{\beta,j}(2), \dots, t_{\beta,j}(h-1), t_{\beta,j}(h), t_{\beta,j}(h+1), \dots, j, \beta)$ 后 $\sum_m C[I_m]$ 的改进量. 搜索的集合序列为 (I_1, I_2, \dots, I_m) ; 搜索从初始点 β 到任意节点 $j (j \notin I_1)$, 历经集合 $(I_1, I_2, \dots, I_{\sigma(j)})$ 的最优路径. 动态规划方程如下:

$$\begin{aligned} f_k(\beta, j) &= \text{cost}_{t_{\beta}(\beta, j)}, \quad j \in I_k, \quad k = 2 \\ f_k(\beta, j) &= \min_{i \in I_{k-1}} \{\text{cost}_{t_{\beta}(\beta, j)} + f_{k-1}(\beta, i)\}, \quad j \in I_k, \quad 3 \leq k \leq m \end{aligned} \quad (2)$$

动态规划搜索过程如图 3, 设 $\beta = 3$.

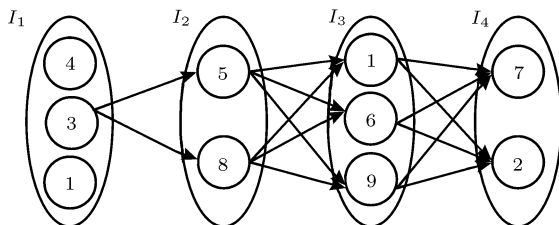


图 3 动态规划搜索过程图

Fig. 3 The search procedure for dynamic programming

若 $\min\{f_k(\beta, j) | \beta \in I_1, j \in A \setminus I_1, k = 2, 3, \dots, m\} < 0$, 动态规划搜索出满意负费用路径. 该负费用路径对应的环可能是虚环. 由于工件在单机上的处理顺序影响目标函数. 本文按照 Liu 和 Maccarthy 的单机近似算法^[10] 进行工件排序.

3 引入 Scatter Search 的 ILS 算法

3.1 基于混合邻域的 ILS 算法

在局部迭代搜索算法 (ILS) 中, 当局部搜索算法陷入局部最优解时, 算法对当前局部最优解实施设定好的随机移动 (*kick*) 对该局优解进行攻击, 再从攻击后得到的新解开始继

续迭代, 跳出局部最优的限制. 这样算法对原来结果中好的特性有继承性, 为充分挖掘当前局部最优解的改进能力提供了可能性. 引入回溯机制, 其目的是使搜索尽可能在有希望改进解的方向上进行. ILS 算法一般由初始解的产生、局域搜索、kick 和接受准则 4 部分组成, 引入回溯机制后, 它的流程框图如图 4 所示.

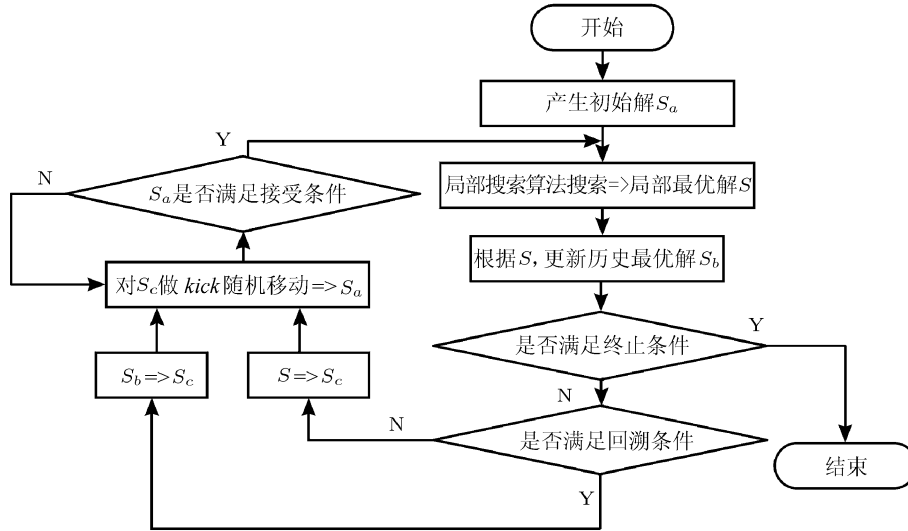


图 4 ILS 流程框图

Fig. 4 The flow chart of ILS

局域搜索算法. 算法首先进行 Swap 邻域的搜索, 当解不能再改进时进行变深度环交换邻域的搜索. 算法过程如下.

Step 1. 根据当前解 S_c , 逐个交换工件 $i (i \in J)$ 与 $j (j \in J), (i \neq j)$, 如果进行最优交换后的解 S' 不能改进解 S_c , 执行下一步, 否则 $S_c = S'$, 重复执行此步骤.

Step 2. 在各集合中加入虚拟工件. 用动态规划算法搜索解 S_c 的变深度环交换邻域.

Step 3. 若动态规划找到负长度路径, 进行相应工件的环交换, 去掉虚拟节点后返回步骤 2; 否则, 终止局域搜索算法.

kick 策略: 1) 在各集合中随机选择一个工件, 按照所在机器顺序连成环, 按照该方式产生两个环进行环交换; 2) 交换动态规划搜索时的机器顺序.

kick 接受准则: 对当前局优解进行 kick 操作后得到的新解均接受.

回溯机制: 当连续对当前局部最优解进行 t 次 kick 操作后, 仍然不能改进历史最好解, 将 kick 的起点设为历史最好解.

3.2 引入 Scatter Search 的 ILS 算法

SS 是一种基于种群的启发式搜索方法. 首先产生初始种群, 用某种选择法在初始种群里选择一些比较有代表性 (分散性和质量) 的解作为参考集; 再从参考集里选择一子集, 把该子集里的解通过组合机制组合为一个新解; 然后局域搜索算法从新解出发搜索新的局部最优解; 最后根据得出的局部最优解, 更新参考集或者初始种群, 以进行新的搜索. SS 具有如下特点: 1) 保持搜索在大范围内进行; 2) 局域搜索的初始解为两个或者两个以上的具有代表性 (分散性和质量) 的解组合得到的, 易跳出局部最优解.

本文将 SS 的组合机制引入 ILS 算法, 以增强 ILS 算法跳出局部最优的能力. 针对并行机问题, 若 ILS 算法在回溯了 t 次之后, 仍不能改进历史最好解, 就将历史最好解与次优解进行组合. 局域搜索算法从组合解开始新的迭代, 从而增强 ILS 在解决并行机调度问

题时跳出局优解的能力. 组合方法如下.

1) 编码. 用序列 $(\sigma(1), \sigma(2), \dots, \sigma(n))$ 表示问题的一个解, $M_{\sigma(i)}$ 代表处理工件 J_i 的机器, $\sigma(i)$ 为机器下标.

2) 组合. 设历史最好解 $A = \{a_1, a_2, \dots, a_n\}$, 次优解 $B = \{b_1, b_2, \dots, b_n\}$. 从位置 $rand(1$ 到 $n/2$ 间的随机整数) 开始, 进行循环交叉, 得到解 A' 和 B' . 本文选择 A' 为组合解.

假设 3 个机器, 7 个工件, 工件的处理安排 $A: I_1 = \{1, 2\}, I_2 = \{3, 4\}, I_3 = \{5, 6, 7\}$; $B: I_1 = \{3, 5\}, I_2 = \{2, 4\}, I_3 = \{1, 6, 7\}$. 设 $rand = 2$, 则这 A 和 B 循环交叉情况如图 5.

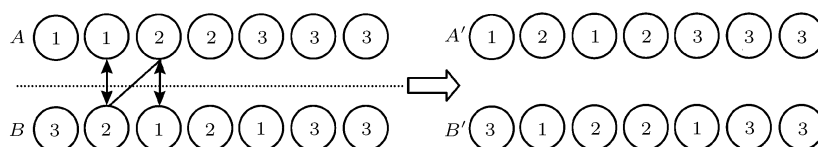


图 5 组合解方法示意图

Fig. 5 The sketch map for combining solution method

3) 解码. 工件 $i(i \in J)$ 由机器 $M_{\sigma(i)}$ 处理.

这样引入 SS 的 ILS 算法尽可能的挖掘了历史最好解的改进潜力, 并将历史最好解与次好解组合, 试图跳出局部最优的限制.

4 实验及算法性能比较

$n \times m$ 表示问题规模, n 为工件数, m 为机器数. 考虑到工件数与机器数的合理比例, 针对两类问题分别设定了 10 种规模, 每种规模 10 组数据来测试算法. 工件的处理和释放时间均为 $[1, 100]$ 之间服从均匀分布的随机数. 回溯参数 t 设为 5. 算法运行时间以混合邻域、采用基于环交换 *kick* 策略的 ILS 算法迭代 100 次为基准, 其他算法 (除拉格朗日算法) 在该时间范围内运行. 对于同构并行机问题, 可通过拉格朗日^[11] 提供下界 LB, 因此用 $(C-LB)/LB$ 评价算法性能, 其中 C 为被评价算法的对该问题的目标函数的计算结果. 对于无关并行机问题, 由于无法获得其最优解或下界, 因此用 C/C^* 规一化的相对指标评价算法的性能, 而 C^* 为被评价的所有算法的对该问题时目标函数的最好值.

1) *kick* 策略的选取

算法采用基于机器顺序的 *kick* 策略. 实验结果表明了该策略优于基于环交换的 *kick* 策略. 说明 *kick* 策略涉及的邻域结构不同于局域搜索的邻域结构有助于提高算法性能.

2) 邻域结构的选取

邻域结构影响局域搜索的有效性, 从而影响 ILS 算法的性能. 加入虚拟工件的变深度环交换邻域包含了多个插入, 但不包含传统的交换邻域, 因此变深度环交换邻域与传统交换邻域结合的混合邻域可能比单邻域效果好. 因此, 分别对混合邻域和单邻域进行了测试, 计算结果如表 1. 根据计算结果, 本文在算法中采用了混合邻域.

综上所述, ILS 算法和引入 SS 的 ILS 算法 (ILS&SS) 采用了交换机器顺序的 *kick* 策略和混合邻域结构. 对这两种算法的测试结果如下. 1) 同构并行机问题. 混合邻域的 ILS 算法、引入 SS 的 ILS 算法 (ILS&SS) 与拉格朗日下界的比较情况如表 2. 2) 无关并行机问题. 混合邻域的 ILS 算法、ILS&SS 和多点下降算法 (MD) 算法性能比较如表 3.

表 3 的数据表明, 对于同构并行机问题, 未引入 SS 的 ILS 算法与下界的平均偏差为 1.06%, 而引入 SS 的 ILS 算法为 0.99%. 表 4 的数据表明, 对于无关并行机问题, 引入 SS 的 ILS 算法比没有引入 SS 的 ILS 算法改进了 6.06%, 比传统的多点下降算法改进了 8.00%, 充分表明了引入 SS 的 ILS 算法的有效性.

表 1 并行机条件下, 混合邻域与变深度环交换邻域的比较

Table 1 Comparison of multi-neighborhood and variable-depth cyclic exchange neighborhood for parallel-machine scheduling problems

规模 $n \times m$	无关并行机			同构并行机		
	算法性能 (%)		运行时间 (s)	平均偏差 (%)		运行时间 (s)
	变深度环交换邻域	混合邻域		变深度环交换邻域	混合邻域	
10×3	1.13	1.00	1.55	0.35	0.48	0.61
20×3	1.36	1.00	12.95	1.35	1.06	3.31
50×3	1.51	1.00	308.45	1.64	1.61	48.11
20×4	1.40	1.00	13.23	1.30	0.64	2.56
50×4	1.63	1.00	262.18	2.31	1.54	30.66
30×5	1.58	1.00	43.71	1.42	0.74	5.73
50×5	1.70	1.00	240.73	2.46	1.07	23.77
50×10	1.63	1.00	193.58	2.84	0.68	12.13
50×15	1.42	1.00	180.96	1.92	1.73	7.87
50×20	1.40	1.00	181.36	3.23	0.34	6.97
平均算法性能	1.48	1.00		1.88	0.99	

表 2 同构并行机问题, ILS 算法、ILS & SS 算法与下界的平均偏差

Table 2 The deviations of ILS and ILS & SS algorithms for the identical parallel-machine scheduling problem

$n \times m$	平均偏差 (%)		运行时间 (s)
	ILS&SS	ILS	
10×3	0.48	0.48	0.61
20×3	1.06	1.11	3.31
50×3	1.61	1.44	48.11
20×4	0.64	0.71	2.56
50×4	1.54	1.42	30.66
30×5	0.74	0.93	5.73
50×5	1.07	1.28	23.77
50×10	0.68	1.15	12.13
50×15	1.73	1.73	7.87
50×20	0.34	0.36	6.97
平均算法性能	0.99	1.06	

表 3 无关并行机问题, ILS & SS 算法、ILS 算法和多点下降算法 (MD) 算法性能比较

Table 3 Computational results of the ILS & SS, ILS and multi-start algorithms for the unrelated parallel-machine scheduling problem

$n \times m$	算法性能 (%)			运行时间 (s)
	ILS&SS	ILS	MD	
10×3	1.00	1.05	1.05	1.55
20×3	1.00	1.06	1.06	12.95
50×3	1.00	1.07	1.09	308.45
20×4	1.00	1.07	1.09	13.23
50×4	1.00	1.09	1.13	262.18
30×5	1.00	1.07	1.10	43.71
50×5	1.00	1.09	1.12	240.73
50×10	1.00	1.03	1.07	193.58
50×15	1.00	1.02	1.07	180.96
50×20	1.00	1.02	1.05	181.36
平均算法性能	1.00	1.06	1.08	

5 结论

本文针对带释放时间的并行机的总完成时间最小化问题进行了研究, 提出了变深度环交换的大规模邻域结构, 将 Scatter search (SS) 搜索方法的组合机制引入了 ILS 算法. 通过大规模的数据测试, 得出了如下结论. 1) 变深度环交换邻域结构与 Swap 邻域结合的混合邻域在解决并行机总完成时间最小化问题时是很有效的. 它可以拓宽到用于求解其他的并行机问题; 2) 嵌入 SS 组合机制的 ILS 算法, 既发挥了 ILS 改进历史最好解的能力, 又提高了算法逃出局部最优的能力; 3) 在 ILS 算法中, 局域搜索的邻域不同于 *kick* 策略涉及的邻域结构更有希望提高 ILS 算法的性能.

References

- 1 Chen B, Potts C N, Woeginger G J. A review of machine scheduling: Complexity, algorithms and approximability. University of Southampton Faculty of Mathematical Studies, Southampton, S017 1BJ, England, 1998
- 2 Hall L A, Schulz A S, Shmoys D B, Wein J. Scheduling to minimize average completion time: Off-line and on-line approximation algorithms, *Mathematics of Operations Research*, 1997, **22**(3): 513~544
- 3 Tang L X, Wu Y P. A genetic descent algorithm for hybrid flow shop scheduling. *Acta Automatic Sinica*, 2002, **28**(4): 637~341.
- 4 Thompson P M, Orlin J B. The theory of cyclic transfers. Working paper, Operations Research Center, MIT, Cambridge, Mass, 1989
- 5 Frangioni A, Necciarì E, Scutellà M G. A multi-exchange neighborhood for minimum makespan machine scheduling problems. *Journal of Combinatorial Optimization*, 2004, **8**(2): 195~220
- 6 Lourenco H R, Martin O, Stützle T. A beginner's introduction to iterated local search. In: Proceedings of MIC'2001-4th Metaheuristics International Conference, Porto, Portugal, 2001.
- 7 Stützle T. Applying Iterated Local Search to the Permutation Flow Shop Problem. Form web: <http://www.citeseer.nj.nec.com>
- 8 García-López F, Melián-Batista B, Moreno-Pérez J A, Moreno-Vega J M. Parallelization of the scatter search for the p-media problem. *Parallel Computing*, 2003, **29**: 575~589
- 9 Angel E, Bampis E. A multi-start dynasearch algorithm for the time dependent single-machine total weighted tardiness scheduling problem. *European Journal of Operational Research*, 2005, **162**(1): 281~289
- 10 Liu J Y, Maccarthy B L. Effective heuristics for the single machine sequencing problem with ready time. *International Journal of Production Research*, 1991, **29**(8): 1521~1533
- 11 Oguz C, Tang L X. The Lagrangian heuristic algorithms for parallel machine scheduling with multiprocessor tasks subject to ready time to minimize total completion time. Report paper of the Hong Kong Polytechnic University. 2003

罗家祥 东北大学信息学院物流优化与控制研究所博士生. 研究方向为智能优化算法和生产计划与调度.

(**LUO Jia-Xiang** Ph.D. candidate in the Logistics Institute of Information Science and Engineering at Northeastern University. Her research interests include intelligent algorithm and production plan and scheduling.)

唐立新 东北大学教授, 博导. 研究方向为生产计划与调度的理论研究、组合最优化的精确与近优化算法研究和物流和供应链建模与优化研究.

(**TANG Li-Xin** Professor in the Logistics Institute of Information Science and Engineering at Northeastern University. His research interests include theory of production plan and scheduling, optimal algorithms and approximate algorithms for combinational problems, and the model and optimization research for the logistics and supply chain.)