

DOI: 10.3901/JME.2010.22.182

基于差分进化与块结构邻域的作业车间调度优化*

潘全科^{1,2} 王凌³ 高亮¹ 桑红燕¹

(1. 华中科技大学数字制造装备与技术国家重点实验室 武汉 430074;

2. 聊城大学计算机学院 聊城 252059;

3. 清华大学信息科学与技术国家实验室 北京 100084)

摘要: 作业车间调度问题是一类非常重要的生产调度问题, 具有广泛的工程应用背景。针对优化最大完成时间指标, 研究通过交换或插入关键路径块结构中的工序产生可行解的方法, 提出基于块结构的交换邻域和插入邻域。结合问题的结构特点和基本差分进化算法的原理, 提出一种解决作业车间调度问题的离散差分进化算法。该算法采用基于操作的编码方式, 设计了离散的变异算子和交叉算子, 能够直接产生可行调度解。进而提出一种自适应的多邻域局部搜索算法, 并将其嵌入离散差分进化算法中来改善优化质量; 基于典型算例的仿真试验验证了所提算法的有效性和优越性。

关键词: 作业车间调度问题 差分进化算法 最大完成时间 局部搜索

中图分类号: TN43

Differential Evolution Algorithm Based on Blocks on Critical Path for Job Shop Scheduling Problems

PAN Quanke^{1,2} WANG Ling³ GAO Liang¹ SANG Hongyan¹

(1. State Key Laboratory of Digital Manufacturing Equipment & Technology, Huazhong University of Science & Technology, Wuhan 430074;

2. School of Computer Science, Liaocheng University, Liaocheng 252059;

3. National Laboratory for Information Science and Technology, Tsinghua University, Beijing 100084)

Abstract: Among all kinds of production scheduling problems, job shop scheduling problem (JSSP) is a key one with extensive engineering application background. The minimization of maximum completion time or makespan for JSSP is addressed. Based on some properties associated with the blocks of jobs on the critical path in the direct graph of JSSP, the way to generate feasible solution by an interchange or insertion operator is presented, and two specific search neighborhoods are obtained. Then a discrete differential evolution (DDE) algorithm with operation-based representation is developed for solving JSSP. The proposed DDE algorithm adopts a newly designed mutation and crossover operators and it can directly generate feasible solutions in the search space. An effective local search based on the newly obtained neighborhoods is presented and imbedded in the DDE algorithm to enhance the local search ability. Computational simulations and comparisons demonstrate the effectiveness and superiority of the proposed DDE algorithm.

Key words: Job shop scheduling problem Differential evolution algorithm Maximum completion time

Local search algorithm

0 前言

作业车间调度问题 (Job shop scheduling problem, JSSP) 是最典型最困难的组合优化问题之一, 它来源于许多实际问题, 诸如加工制造、电气

布线和网络交通等。因此, JSSP 的算法研究一直是学术界和工程界共同关注的重要课题^[1-2]。按照优化机制和优化行为的不同, 解决 JSSP 的算法可区分为精确求解算法、构造式启发算法和智能优化算法。由于 JSSP 属于 NP 难题, 目标解的搜索涉及解空间的组合爆炸, 故线性规划、动态规划和分支定界等精确求解算法只适合于小规模问题, 对于一般规模问题则难以在合理时间内求得最优解。优先分配规则等构造式方法虽然能快速求解, 但是解的质量往

* 国家自然科学基金(60874075, 70871065, 60774082, 60834004)、教育部新世纪优秀人才支持计划(NCET-10-0505)和华中科技大学数字制造装备与技术国家重点实验室开放课题资助项目。20091125 收到初稿, 20100615 收到修改稿

往不高。因此，能在合理时间内求得满意解的智能优化算法就成为求解调度问题的主要方法。

差分进化(Differential evolution, DE)是一种新兴的智能优化算法，已在人工神经网络、电力系统、系统辨识与故障诊断和信号处理等领域得到了成功应用^[3-4]。由于 DE 具有结构简单、容易实现、快速聚合和鲁棒性强等优点，本文针对 JSSP 的特点，提出了一种解决 JSSP 的离散差分进化(Discrete DE, DDE)算法。首先，在分析关键路径块结构特性的基础上，提出了两种新型邻域结构，即基于块结构的交换邻域和插入邻域；其次，通过采用基于操作的编码方式、设计新的变异算子和交叉算子，提出了一种能够直接在离散域内工作的 DDE 算法；第三，提出了一种自适应的多邻域局部搜索算法，并将其嵌入 DDE 算法来改善优化质量；最后，仿真试验验证了所提算法的有效性和优越性。

1 问题模型与邻域结构

1.1 JSSP 的问题模型

JSSP 可描述为^[1]，加工系统中有 n 个工件和 m 台机床，每个工件均需要不重复地经过所有机床加工。工件的加工路线和加工时间已确定。同时假设：
① 在任何时刻，每台机器最多只能加工一个工件，每个工件最多只能由一台机器加工；
② 工件加工过程不能间断；
③ 机器之间缓冲区容量为无穷大。问题是安排生产，使工件的最大完成时间最短。

JSSP 的析取图^[5]可定义为， $G = (V, A, E)$ 。其中， $V = O \cup \{s, e\}$ 是节点集， O 为所有操作集， s 和 e 分别为整个加工过程的起点和终点；除 s 和 e 的权值为零外，其余各节点的权值为其对应操作的加工时间。 A 为连接同一工件不同操作的有向弧集合； V 为连接同一机器上不同操作的边集。JSSP 就是如何确定 V 中各边的方向，使所有有向图 $G(\pi)$ 的关键路径最短。图 1 所示为一个 JSSP 的析取图(图 1 中，①、②、③为工件 1 的工序；④、⑤、⑥为工件 2 的工序；⑦、⑧、⑨为工件 3 的工序)。

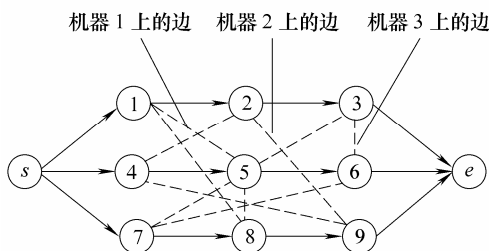


图 1 析取图(含 3 个工件, 3 台机器)

1.2 基于关键路径块结构的邻域

邻域结构的选择对算法的搜索质量和效率有较大影响^[1,6]。由于基于关键路径块结构的邻域可以避免一些不必要的搜索，有效提高算法性能，故采用该类邻域。关键路径的块由同一机器上相邻的关键操作组成，记为 $B = \{b_1, b_2, \dots, b_k\}$ ，其中 b_1 和 b_k 分别称为块首操作和块尾操作， b_2, b_3, \dots, b_{k-1} 称为块内操作。记 $\alpha(x)$ 和 $\gamma(x)$ 分别为操作 x 的工件前任操作和后续操作， $\beta(x)$ 和 $\delta(x)$ 分别为操作 x 的机器前任操作和后续操作， p_x 为操作 x 的加工时间， $L(x, e)$ 为从操作 e 到 x 的最长路径， $x > y$ 表示操作 x 先于 y 加工，有如下定理。

定理 1: 对于一个可行的 JSSP 调度，若关键路径同一块上有两个操作 x 和 y 且 $\delta(x) > y$ ，则交换操作 x 和 y 的位置(记为 $Swap(x, y)$)后，得到的调度仍是可行的，如果式(1)、(2)同时成立

$$L(\gamma(x), e) < L(y, e) + \min(p_{\beta(y)} + p_{\beta(\beta(y))}, p_{\beta(y)} + p_{\alpha(\beta(y))}, p_{\alpha(y)}) + p_{\gamma(x)} \tag{1}$$

$$L(\delta(x), e) < L(\alpha(y), e) + \min(p_{\delta(\delta(x))}, p_{\gamma(\delta(x))}) + p_{\delta(x)} \tag{2}$$

证明：采用反证法。

设可行调度对应非循环图 $G(\pi)$ ，通过交换 (x, y) 生成图 $G'(\pi)$ 。若 $G'(\pi)$ 中存在循环 C ，则 C 必包含有新增加的部分弧(图 2)。

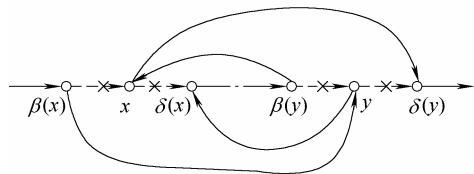


图 2 由交换操作引起的弧的增减

若 C 包含弧 $(\beta(x), y)$ ，则在 $G(\pi)$ 中存在从 y 或 $\delta(x)$ 到 $\beta(x)$ 的弧，这与“ $G(\pi)$ 是非循环图”矛盾。因此， C 不包含弧 $(\beta(x), y)$ 。同理可证， C 不包含弧 $(x, \delta(y))$ 。

若 C 包含弧 $(\beta(y), x)$ ，则在 $G(\pi)$ 中至少存在下述三条路径之一(图 3)

- ① $d_1(\gamma(x), y) = (d(\gamma(x), \beta(\beta(y))), (\beta(\beta(y)), \beta(y)), (\beta(y), y))$
- ② $d_2(\gamma(x), y) = (d(\gamma(x), \alpha(\beta(y))), (\alpha(\beta(y)), \beta(y)), (\beta(y), y))$
- ③ $d_3(\gamma(x), y) = (d(\gamma(x), \alpha(y)), (\alpha(y), y))$

若路径①存在，则有

$$L(\gamma(x), e) \geq L(y, e) + p_{\beta(y)} + p_{\beta(\beta(y))} + p_{\gamma(x)} \tag{3}$$

若路径②存在，则有

$$L(\gamma(x), e) \geq L(y, e) + p_{\beta(y)} + p_{\alpha(\beta(y))} + p_{\gamma(x)} \quad (4)$$

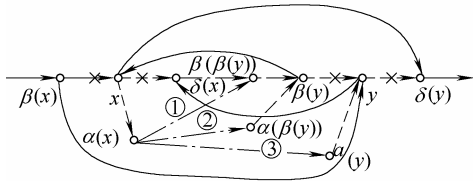


图 3 从 x 到 $\beta(y)$ 的路径①、②和③

若路径③存在, 则有

$$L(\gamma(x), e) \geq L(y, e) + p_{\alpha(y)} + p_{\gamma(x)} \quad (5)$$

综合考虑式(3)~(5)可知, 若式(6)成立, 则路径①、②和③中至少有一条存在。

$$L(\gamma(x), e) \geq L(y, e) + \min(p_{\beta(y)} + p_{\alpha(\beta(y))}, p_{\beta(y)} + p_{\alpha(y)} + p_{\gamma(x)}) \quad (6)$$

显然, 这与式(1)矛盾。故路径①、②和③均不可能存在, 即 C 不包含弧 $(\beta(y), x)$ 。

若 C 包含弧 $(y, \delta(x))$, 则在 $G(\pi)$ 中至少存在下述两条路径之一(图 4)

- ④ $d_4(\delta(x), y) = ((\delta(x), \delta(\delta(x))), d(\delta(\delta(x)), \alpha(y)), (\alpha(y), y))$
- ⑤ $d_5(\delta(x), y) = ((\delta(x), \gamma(\delta(x))), d(\gamma(\delta(x)), \alpha(y)), (\alpha(y), y))$

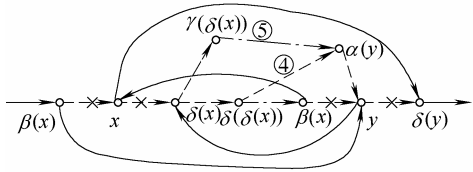


图 4 从 $\delta(x)$ 到 y 路径④和⑤

若路径④存在, 则有

$$L(\delta(x), e) \geq L(\alpha(y), e) + p_{\delta(\delta(x))} + p_{\delta(x)} \quad (7)$$

若路径⑤存在, 则有

$$L(\delta(x), e) \geq L(\alpha(y), e) + p_{\gamma(\delta(x))} + p_{\delta(x)} \quad (8)$$

综合考虑式(7)、(8)可知, 若式(9)成立, 则路径④和⑤中至少有一条存在

$$L(\delta(x), e) \geq L(\alpha(y), e) + \min(p_{\delta(\delta(x))}, p_{\gamma(\delta(x))}) + p_{\delta(x)} \quad (9)$$

显然, 这与式(2)矛盾。故路径④和⑤均不可能存在, 即 C 不包含弧 $(y, \delta(x))$ 。

定理证毕。

定理 2: 对于一个可行的 JSSP 调度, 若关键路径同一块上有两个操作 x 和 y 且 $\delta(x) > y$, 则将 x 插入 y 之后(记为 $Inserta(x, y)$)得到的调度仍是可行的, 如果下列条件成立

$$p_{\beta(y)} + p_{\beta(\beta(y))} + \dots + p_{\delta(x)} < \min(p_{\alpha(y)}, p_{\beta(y)}) + p_{\gamma(x)} \quad (10)$$

证明: 采用反证法。

设可行调度对应非循环图 G , 将 x 插入 y 之后生成图 G' 。若 G' 中存在循环 C , 则 C 必包含有新增的部分弧(图 5)。

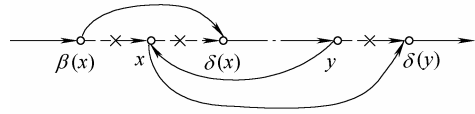


图 5 由插入操作引起的弧的增减

若 C 包含弧 $(\beta(x), \delta(x))$, 则在 G 中存在从 $\delta(x)$ 到 $\beta(x)$ 的弧, 这与“ G 是非循环图”矛盾。因此 C 不包含弧 $(\beta(x), \delta(x))$ 。

同理可证, C 不包含弧 $(x, \delta(y))$ 。

若 C 包含弧 (y, x) , 则在 $G(\pi)$ 中至少存在下述两条路径之一(图 6)

- ① $d_1(x, y) = ((x, \gamma(x)), d(\gamma(x), \beta(y)), (\beta(y), y))$
- ② $d_2(x, y) = ((x, \gamma(x)), d(\gamma(x), \alpha(y)), (\alpha(y), y))$

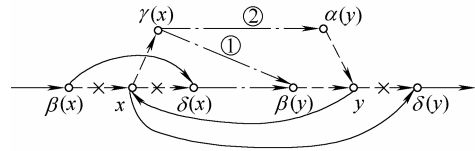


图 6 从 x 到 y 的路径①和②

若路径①存在, 则有

$$L(x, e) \geq L(y, e) + p_{\beta(y)} + p_{\gamma(x)} + p_x \quad (11)$$

若路径②存在, 则有

$$L(x, e) \geq L(y, e) + p_{\alpha(y)} + p_{\gamma(x)} + p_x \quad (12)$$

综合考虑式(11)、(12)知, 若式(13)成立, 则路径①和②中至少有一条存在

$$L(x, e) \geq L(y, e) + \min(p_{\alpha(y)}, p_{\beta(y)}) + p_{\gamma(x)} + p_x \quad (13)$$

显然, 这与定理的条件矛盾, 故 C 不可能包含弧 (y, x) 。

定理证毕。

定理 3: 对于一个可行的 JSSP 调度, 若关键路径同一块上的两个操作 x 和 y 且 $\delta(x) > y$, 则将 y 插入 x 之前(记为 $Insertb(y, x)$)得到调度仍是可行的, 如果下列条件成立

$$p_{\beta(y)} + p_{\beta(\beta(y))} + \dots + p_{\delta(x)} < p_{\alpha(y)} + p_{\gamma(x)} \quad (14)$$

证明同定理 2。

研究表明^[1]: 交换关键路径工序块上的相邻工序仍产生可行解, 但交换块内相邻的工序不能缩短最大完成时间。因此, 仅考虑基于块首工序和块尾工序的移动。定义基于关键路径工序块的两种邻域结构如下。

(1) 交换邻域为

$$Z_{\text{Swap}} = \{\text{Swap}(b_1, x) | x \in B - \{b_1\}\} \cup \{\text{Swap}(x, b_k) | x \in B - \{b_k\}\} \quad (15)$$

式中， $\text{Swap}(b_1, x)$ 和 $\text{Swap}(x, b_k)$ 满足定理 1 或 $x = \delta(b_1)$ 或 $x = \beta(b_k)$ 。

(2) 插入邻域为

$$Z_{\text{Insert}} = \{\text{Inserta}(b_1, x) | x \in B - \{b_1\}\} \cup \{\text{Insertb}(b_k, x) | x \in B - \{b_k\}\} \quad (16)$$

式中， $\text{Inserta}(b_1, x)$ 满足定理 2 或 $x = \delta(b_1)$ ； $\text{Insertb}(b_k, x)$ 满足定理 3 或 $x = \beta(b_k)$ 。

上述两种基于块结构的邻域，其邻居数量非常少，并且有效地排除了不能缩短最大完成时间的移动，因此能提高算法的效率。

2 基本 DE 算法

基本 DE 是一种基于群体进化的智能算法。在进化过程中，DE 维持三个群体，即目标种群、变异种群和试验种群。从初始目标种群开始，DE 利用个体之间的距离和方向信息产生侯选解，并通过“一对一”贪婪竞争机制决定后代目标个体。记

$$\begin{aligned} X_i^t &= (x_{i1}^t, x_{i2}^t, \dots, x_{im}^t) \\ V_i^t &= (v_{i1}^t, v_{i2}^t, \dots, v_{im}^t) \\ U_i^t &= (u_{i1}^t, u_{i2}^t, \dots, u_{im}^t) \end{aligned}$$

分别为第 t 代目标种群、变异种群和试验种群的第 i 个体。则有变异操作、交叉操作和选择操作(对最小化问题)分别为

$$V_i^t = X_a^t + Z(X_b^t - X_c^t) \quad (17)$$

$$u_{ij}^t = \begin{cases} v_{ij}^t & \text{rndn}(i) < C_R \vee j = \text{rndn}(i) \\ x_{ij}^t & \text{其他} \end{cases} \quad j = 1, 2, \dots, n \quad (18)$$

$$X_i^{t+1} = \begin{cases} U_i^t & f(U_i^t) < f(X_i^t) \\ X_i^t & \text{其他} \end{cases} \quad (19)$$

式中， a 、 b 和 c 为 $[1, P_S]$ 内的随机整数； Z 为变异系数； C_R 为交叉系数； P_S 为种群规模； $\text{rndn}(i)$ 为 $[1, n]$ 内的随机整数； $f(\cdot)$ 为个体目标值。

3 求解 Job Shop 问题的 DDE

3.1 个体矢量编码与解码

编码和解码是设计和应用 DDE 算法的首要 and 关键问题。采用基于操作的编码，即对于 n 个工件 m 台机床的 JSSP，用长度为 $n \times m$ 的工件号序列表示个体，其中每个工件号重复 m 次。在解码时，先

将个体矢量转化为工件号序列，同一工件的不同操作根据工件号在序列中的次序区分。然后在满足工艺约束的前提下，根据操作序列将每个操作以其最早允许的加工时间安排到相应的机床上加工。显然，这种方法总能产生活动调度。

3.2 DDE 的变异算子

上述编码方式不仅可以表征全部调度空间，而且操作的任意排列均能够对应可行调度。然而，在这种编码方式下，却无法采用基本 DDE 的操作算子进化。由 DE 的进化机理知，变异算子体现了个体在寻优过程中的信息交换和相互学习行为，它是个体从当前群体中获取信息的有效方式。变异个体由三个随机选择的目标个体按照一定的方式产生。目标个体之间的差异决定了所得变异个体不同于任何目标个体。因此，结合基于操作的编码方式有变异算子

$$V_i^t = X_a^t \oplus Z \otimes (X_b^t - X_c^t) \quad (20)$$

式中， Z 是变异系数， $Z \in [0, 1]$ 。

式(20)由两部分构成，第一部分对应式(17)中加权的两目标个体之差，即

$$\begin{aligned} \Delta_i^t &= Z \otimes (X_b^t - X_c^t) \Leftrightarrow \\ \delta_{i,j}^t &= \begin{cases} x_{b,j}^t - x_{c,j}^t & \text{rand}(\bullet) < Z \\ 0 & \text{其他} \end{cases} \end{aligned}$$

式中， Δ_i^t 是临时矢量， $\Delta_i^t = (\delta_{i,0}^t, \delta_{i,1}^t, \dots, \delta_{i,m}^t)$ 。

第二部分通过目标个体 X_a^t 与临时矢量 Δ_i^t 相加得到变异个体，即

$$\begin{aligned} V_i^t &= X_a^t \oplus \Delta_i^t \Leftrightarrow \\ v_{ij}^t &= x_{aj}^t \oplus \delta_{ij}^t = \text{mod}((x_{aj}^t + \delta_{ij}^t + n), n) \end{aligned}$$

式中， mod 表示求余操作。

图 7 所示为变异操作的实例($n=3, m=2$)。

X_b	0	1	2	2	1	0
X_c	2	0	1	0	1	2
$X_b - X_c$	-2	1	1	2	0	-2

(a) $X_b - X_c$

$\text{rand}()$	0.2	0.6	0.4	0.5	0.1	0.7
$X_b - X_c$	-2	1	1	2	0	-2
Δ_i^t	-2	0	1	0	0	0

(b) Δ_i^t

X_a	1	0	0	2	2	1
Δ_i^t	-2	0	1	0	0	0
$V_i = X_a \oplus \Delta_i^t$	2	0	1	2	2	1

(c) V_i

图 7 变异算子($Z=0.5$)

该变异算子保证了所得变异个体 V_i^t 为一个

$n \times m$ 的工件号序列, 但每个工件号重复的次数并不一定均为 m 次。即该变异个体并不表示一个合法调度。由基本 DE 知, 变异个体的作用在于对目标个体产生扰动, 使其避免陷入局部极小。因此, 变异个体没有必要一定为合法个体, 可通过设计合理的交叉操作使所得试验个体为表示合法的调度方案。

3.3 DDE 的交叉算子

交叉算子通过变异个体与对应的目标个体相互作用得到试验个体。这体现了差异进化算法对个体极值信息的利用。为了较好地继承已有目标个体的优良结构, 试验个体按照如下方法产生(图 8)。

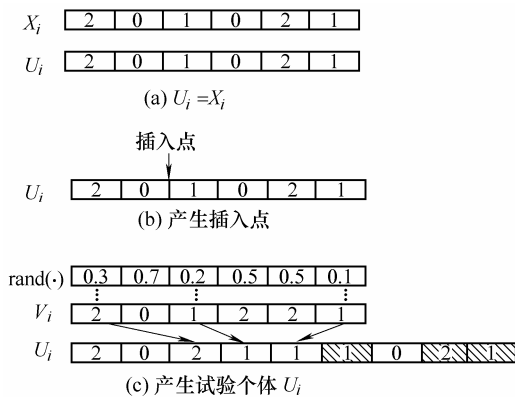


图 8 DDE 的交叉算子($C_r = 0.5$)

步骤 1: 令 $U_i^j = X_i^{j-1}$, $j = 0$, 并随机产生一个插入点 $i' \in [1, n \times m]$ 。

步骤 2: 如 $\text{rand}(\bullet) < CR$, 则将 $v_{i',j}^j \in V_i^j$ 插入 U_i^j 的位置 i' , 并令 $i' = i' + 1$ 。

步骤 3: 令 $j = j + 1$, 如 $j \leq n$ 则返回步骤 2。

步骤 4: 从后向前依次删除 U_i^j 中多余的分量, 使每个工件号重复 m 次。

3.4 DDE 的选择算子

DDE 采用“一对一”贪婪选择方式决定后代目标个体, 其选择方法与基本 DE 相同。

3.5 DDE 的局部搜索

在搜索过程中, 充分利用问题信息增强 DDE 的局部搜索能力是提高其优化性能的有效方式之一^[7]。鉴于当前群体中最优个体往往携带最有价值的信息, 因此可在每次迭代中对最优个体执行局部搜索算法。为了合理利用各种邻域结构, 采用 meta-Lamarckian 学习策略^[8], 根据学习到的知识从多个邻域结构中自适应的选择最适合于当前搜索的邻域来实施局部搜索。记插入邻域为第一邻域, 交换邻域为第二邻域, 则自适应多邻域局部搜索算法如下所述。

步骤 1: 令 $c'_b = c_b$, 其中 c_b 为当前最优个体。

步骤 2: 计算各邻域的选择概率

$$P_i = \eta_i / \sum_{i=1}^2 \eta_i \quad i=1, 2$$

式中, η_i 为邻域 i 奖励值。初始化 $\eta_1 = \eta_2 = 0.5$ 。

步骤 3: 选择邻域。若 $\text{rand}(\bullet) < P_i$, 选择插入移动邻域, 否则选择交换移动邻域。

步骤 4: 局部搜索。

① 搜索 c'_b 的整个邻域, 得到最优邻居 t_b ; 若 $f(t_b) < f(c'_b)$, 令 $c'_b = t_b$, 返回重新搜索; 否则执行步骤 5。

步骤 5: 更新邻域的奖励值。若 $f(c'_b) \leq f(c_b)$, 则令

$$\eta_i = \eta_i + (f(c_b) - f(c'_b)) / (mn)$$

步骤 6: 更新最优解。若 $f(c'_b) \leq f(c_b)$, 则令 $c_b = c'_b$ 。

记采用上述局部搜索的 DDE 为 DDE₁ 算法。

DDE₁ 虽然收敛速度快, 局部开发能力强, 但容易陷入局部极值。为了避免这一缺陷, 可改进局部搜索算法如下(记为 DDE₂)。

步骤 1: 扰动操作。对当前最优个体 c_b 执行几次随机交换或插入移动得到新个体 c'_b 。

步骤 2: 执行自适应多邻域局部搜索算法。

步骤 3: 基于模拟退火机制的选择操作。

$$\text{rand}(\bullet) < \exp((f(c_b) - f(c'_b)) / T) \quad (21)$$

式中, $T = F_r \sum_{i \in O} p_i / (mn)$ 。

(1) 若 $f(c'_b) \leq f(c_b)$, 令 $c_b = c'_b$ 。

(2) 若 $f(c'_b) > f(c_b)$, 则考察式(21)。若式(21)成立, 则令 $c_b = c'_b$ 。

改进后的算法对当前最优个体的扰动解来执行局部搜索, 这样可使算法有效避免循环搜索; 在局部搜索阶段, 只有在当前解的邻域内发现了更好解才进行更新, 这样算法能够快速的达到局部极值; 在选择阶段采用模拟退火算法的概率性劣向转移机制, 有效提高了算法摆脱局部极值的能力。

3.6 DDE 的基本流程

基于上述各部分的分析, 求解 JSSP 的 DDE 的基本流程如图 9 所示。

图 9 中 DDE 仍具有 DE 的基本特征: ① 通过三个随机选择的目标个体产生变异个体, 所选个体的差异决定了变异个体不同于对应的目标个体; ② 通过对所有目标个体与其对应的变异个体执行交叉操作产生试验群体; ③ 一对一的选择决定了新一代目标群体优于当前群体。但 DDE 算法又不同于 DE, 因为它考虑到了 JSSP 的特点, 采用面向 JSSP 的编码和操作算子, 直接工作于离散域, 并强化了算法的局部搜索能

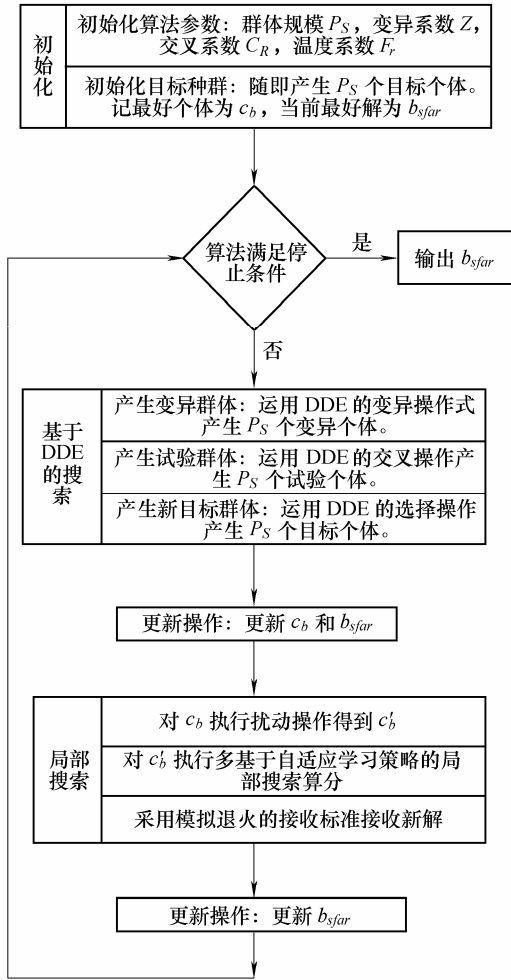


图 9 DDE 算法的示意图

力。DDE 与遗传算法(Genetic algorithm, GA)也有本质的区别：① 从算法结构上看，DDE 在进化过程中始终维持三个群体，即目标群体、变异群体和试验群体，信息在这三个群体之间流动和交换，从而推动 DDE 的进化，而 GA 则在一个群体内交换信息；② 从进化过程来看，DDE 具有记忆个体最优的机制，并且个体最优向试验个体传递信息，指导算法的进化，而 GA 则没有此机制；③ 从新群体的生成方式来看，所有目标个体均参与进化操作，而 GA 则是部分经过选择的个体参与进化；④ 从选择机制来看，DDE 采用一对一的选择机制，而 GA 则在整个种群内进行优胜劣汰。

4 数值仿真研究

4.1 算法试验设置

以 VC++为开发环境，采用 PIV/3.0 G、512 MB RAM 的计算机和 11 个不同规模的测试问题，验证所得算法的性能。算法参数设置如下： $P_S = 50$ ， $Z = 0.2$ ， $C_R = 0.4$ ， $F_r = 0.8$ 。每个测试问题均独

立计算 10 次，并记录算法的运行时间(单位：s)。

4.2 DDE 类算法比较

首先比较 DDE、DDE₁ 和 DDE₂，以考察在 DDE 中嵌入局部搜索算法的有效性。各算法的计算结果如表 1 所示。

表 1 DDE 类算法比较

算法	问题	ft06	ft10	ft20	la01	la06	la11
DDE	平均值 a	55	986.7	1 231.6	666	926	1 222
	方均差 δ	0	11.61	11.63	0	0	0
	平均时间 t/s	0.06	100	100	0.44	0.03	0.19
DDE ₁	平均值 a	55	977.3	1 221.3	666	926	1 222
	方均差 δ	0	10.82	14.57	0	0	0
	平均时间 t/s	0.02	100	100	0.14	0.01	0.02
DDE ₂	平均值 a	55	935.80	1 173.2	666	926	1 222
	方均差 δ	0	5.43	3.55	0	0	0
	平均时间 t/s	0.01	64.67	93.33	0.01	0.01	0.01
算法	问题	la16	la21	la26	la31	la36	
DDE	平均值 a	979.3	1 134.4	1 321.3	1 811.9	1 367.3	
	方均差 δ	1.06	5.10	9.78	8.63	10.58	
	平均时间 t/s	100	500	500	500	1 000	
DDE ₁	平均值 a	977.30	1 116.3	1 283.1	1 792.3	1 353.4	
	方均差 δ	4.69	8.73	12.91	8.93	9.28	
	平均时间 t/s	100	500	500	316.28	1 000	
DDE ₂	平均值 a	945.30	1 046.6	1 218	1 784	1 290.7	
	方均差 δ	0.48	1.35	0	0	3.53	
	平均时间 t/s	77.26	320.25	37.55	0.53	1 000	

(1) 比较 DDE₁ 和 DDE 可知：对于所有测试问题，DDE₁ 的求解结果均优于或等于 DDE，这表明嵌入局部搜索能有效改善 DDE 算法的性能。

(2) 比较 DDE₂ 和 DDE₁ 可知：对于所有测试问题，DDE₂ 的求解结果均优于或等于 DDE₁，这表明在局部搜索算法中采取扰动措施能有效避免算法陷入局部极值。

(3) DDE₂ 的方均差较小，表明 DDE₂ 对初始种群具有较好的鲁棒性。

4.3 DDE 与文献[1]中的混合 PSO 比较

文献[1]提出了一种解决 JSSP 的混合微粒群优化(Particle swarm optimization, PSO)算法。作者的仿真试验表明该方法优于一些著名的算法，如混合 PSO(Hybrid PSO, HPSO)^[9]、遗传—退火算法(Genetic algorithm-simulated annealing, GASA)^[10]和 PSO-priority^[11]等。因此将 DDE₂ 与文献[1]的混合 PSO 相比较，结果如表 2 所示。

(1) DDE₂ 所得解的平均值中，有 4 个测试问题优于混合 PSO，6 个测试问题等于混合 PSO，只有 1 个测试问题劣于混合 PSO。这表明 DDE₂ 具有较强的搜索质量。

表 2 比较 DDE₂ 和混合 PSO 算法

问题	混合 PSO		DDE ₂	
	平均值 <i>a</i>	平均时间 <i>t/s</i>	平均值 <i>a</i>	平均时间 <i>t/s</i>
ft06	55	7.0	55	0.01
ft10	947.1	164.7	935.80	64.67
ft20	1 178.2	310.0	1 173.20	93.33
la01	666	17.8	666	0.01
la06	926	61.1	926	0.00
la11	1222	145.6	1222	0.01
la16	945.4	148.3	945.30	77.26
la21	1 064.3	733.7	1 046.60	320.25
la26	1 218	2 449.5	1 218	37.55
la31	1 784	3 883.3	1 784	0.53
la36	1 283.5	3 541.7	1 290.70	1 000.00

(2) 从平均计算时间来看, DDE₂ 所需的计算时间明显少于混合 PSO。这说明 DDE₂ 有较高的搜索效率。

5 结论

研究了基于差分进化与块结构邻域的作业车间调度优化方法。

(1) 基于 JSSP 的有向图模型, 提出了基于关键路径工序块结构的交换邻域和插入邻域。

(2) 提出了一种解决作业车间调度问题的离散差分进化算法。该算法工作于离散域, 能够直接产生可行调度解。

(3) 提出了一种基于块结构邻域的自适应局部搜索算法, 该局部算法嵌入到离散差分算法中可有效提高求解质量。

(4) 仿真结果和比较表明本文算法优于相关文献上的混合微粒群算法。

参 考 文 献

- [1] 王凌, 刘波. 微粒群优化与调度算法[M]. 北京: 清华大学出版社, 2008.
WANG Ling, LIU Bo. Particle swarm optimization and scheduling algorithms [M]. Beijing: Tsinghua University Press, 2008.
- [2] 闫利军, 李宗斌, 卫军胡, 等. 一种新的混合优化算法及其在车间调度中的应用[J]. 自动化学报, 2008, 34(5): 604-609.
YAN Lijun, LI Zongbin, WEI Junhu, et al. A new hybrid optimization algorithm and its application in job shop scheduling [J]. Acta Automatic Sinica, 2008, 34(5): 604-609.
- [3] 刘波, 王凌, 金以慧. 差分进化研究进展[J]. 控制与决

策, 2007, 22(7): 721-729.

LIU Bo, WANG Ling, JIN Yihui. Advances in differential evolution [J]. Control and Decision, 2007, 22(7): 721-729.

- [4] 周艳平, 顾幸生. 差分进化研究进展[J]. 化工自动化及仪表, 2007, 34(3): 1-5.

ZHOU Yanping, GU Xingsheng. Development of differential evolution algorithm[J]. Control and Instrument in Chemical Industry, 2007, 34(3): 1-5.

- [5] NOWICKI E, SMUTNICKI C. A fast taboo search algorithm for the job-shop problem [J]. Management Science, 1996, 42: 797-813.

- [6] 罗家祥, 唐立新. 带释放时间的并行机调度问题的 ILS & SS 算法[J]. 自动化学报, 2005, 31(6): 917-924.

LUO Jiaxiang, TANG Lixin. A new ILS & SS algorithm for parallel-machine scheduling problem[J]. Acta Automatica Sinica, 2005, 31(6): 917-924.

- [7] PAN Quanke, WANG Ling. A novel differential evolution algorithm for the no-idle permutation flow shop scheduling problems [J]. European Journal of Industrial Engineering, 2008, 2(3): 279-297.

- [8] ONG Y S, KEANE A J. Meta-lamarckian learning in memetic algorithms [J]. IEEE Trans., Evolut. Comput, 2004, 8: 89-110.

- [9] XIA W J, WU Z M. A hybrid particle swarm optimization approach for the job-shop scheduling problem [J]. International Journal of Advanced Manufacturing Technology, 2006, 29: 360-366.

- [10] WANG L, ZHENG D Z. An effective optimization strategy for job-shop scheduling problems [J]. Computer and Operations Research, 2001, 28: 585-596.

- [11] SHA D Y, HSU C Y. A hybrid particle swarm for job shop scheduling problem [J]. Computer and Industrial Engineering, 2006, 51: 791-808.

作者简介: 潘全科, 男, 1971 年出生, 博士后, 教授。主要研究方向为智能计算及其应用。

E-mail: panquanke@lcu.edu.cn

王凌, 男, 1972 年出生, 博士, 教授。主要研究方向为智能优化调度理论与方法。

E-mail: wangling@tsinghua.edu.cn

高亮, 男, 1974 年出生, 博士, 教授。主要研究方向为先进制造系统、运筹学与优化。

E-mail: gaoliang@mail.hust.edu.cn

桑红燕, 女, 1981 年出生, 博士研究生。主要研究方向为优化调度理论与方法。

E-mail: sanghongyanlcu@163.com