一种结合多目标免疫算法和线性规划的双行设备布局方法

左兴权 [1, 2] 王春露 [1, 2] 赵新超 [3]

摘 要 设备布局对于提高生产效率和降低运营成本具有重要意义. 本文针对半导体加工制造中常见的双行设备布局问题, 提出了一种结合多目标免疫算法和线性规划的双行设备布局方法来同时优化物料流成本和布局面积两个目标. 首先, 建立了问题的混合整数规划模型; 其次, 针对问题既含有组合方面 (机器排序) 又含有连续方面 (机器精确位置) 的特点, 分别设计了一种多目标免疫算法来获取非支配的机器排序集合, 提出了一种基于线性规划的方法来构造任一非支配机器排序对应的连续的非支配解集; 最后, 由所有连续的非支配解来构造最后 Pareto 解. 实验结果表明, 该方法对于小规模问题能获得最优 Pareto 解, 对于大规模问题能够获得具有良好分布性的 Pareto 解且其质量远好于 NSGA-II 和精确算法获得的解.

关键词 设备布局问题, 免疫算法, 多目标优化, 线性规划

引用格式 左兴权, 王春露, 赵新超. 一种结合多目标免疫算法和线性规划的双行设备布局方法. 自动化学报, 2015, **41**(3): 528−540

DOI 10.16383/j.aas.2015.c140082

Combining Multi-objective Immune Algorithm and Linear Programming for Double Row Layout Problem

ZUO Xing-Quan[1, 2] WANG Chun-Lu[1, 2] ZHAO Xin-Chao[3]

Abstract Facility layout is very significant for improving production efficiency and decreasing operational cost. Aimed at the double row layout problem commonly encountered in the context of semiconductor manufacturing, an approach combining a multi-objective immune algorithm with a linear programming is proposed to simultaneously optimize the two objectives of material flow cost and layout area. Firstly, a mix-integer programming model is established for this problem. Secondly, based on the problem′s characteristic of involving both combinatorial (machine sequence) and continuous (exact machine position) aspects, a multi-objective immune algorithm is devised to obtain a set of non-dominated machine sequences, and then a linear programming based method is proposed to construct a set of continuous non-dominated solutions for an arbitrary non-dominated machine sequence. Finally, the set of final Pareto solutions is created from all the continuous non-dominated solutions. Experimental results show that for small size problems our approach is able to obtain the optimal Pareto solutions, and for large size problems our approach can achieve Pareto solutions with good distribution, which are far better than those obtained by NSGA-II and an exact approach.

Key words Facility layout problem, immune algorithm, multi-objective optimization, linear programming

Citation Zuo Xing-Quan, Wang Chun-Lu, Zhao Xin-Chao. Combining multi-objective immune algorithm and linear programming for double row layout problem. *Acta Automatica Sinica*, 2015, **41**(3): 528−540

设备布局问题是指将生产车间中的设备 (机器) 进行合理布局, 达到提高生产效率和节约运营成本的目的[1]. 优良的设备布局能加快物料处理速度, 减少在制品停留时间和工件缓冲区容量, 提高生产效率; 反之, 不合理的设备布局将导致生产周期延长和生产效率降低[2]. 有研究表明, 15～70% 的运营成本和设备布局有关, 良好的设备布局可降低 10～30% 的运营成本[3].

Heragu 等[4] 总结了制造领域中常见的 4 种设备布局类型, 即单行设备布局、多行设备布局、U 型布局和环形布局, 其中单行和多行设备布局在实际中应用最为广泛[5].

单行设备布局问题 (Single row layout problem, SRLP) 是把若干机器在一行上排列, 相邻机器间以给定的最小间距分隔, 优化目标为最小化物料流成本[6]. 该问题只需确定机器排序, 因而是纯组合优化问题. 一些学者提出了该问题的精确求解方法.

收稿日期 2014-02-19 录用日期 2014-10-16
Manuscript received February 19, 2014; accepted October 16, 2014
国家自然科学基金 (61374204, 61375066) 资助
Supported by National Natural Science Foundation of China (61374204, 61375066)
本文责任编委 宋士吉
Recommended by Associate Editor SONG Shi-Ji
1. 北京邮电大学计算机学院 北京 100876 2. 可信分布式计算与服务教育部重点实验室 北京 100876 3. 北京邮电大学理学院 北京 100876
1. Computer School, Beijing University of Posts and Telecommunications (BUPT), Beijing 100876 2. Key Laboratory of Trustworthy Distributed Computing and Service (BUPT), Ministry of Education, Beijing 100876 3. School of Science, Beijing University of Posts and Telecommunications, Beijing 100876

例如, Heragu 等[7] 建立了 SRLP 的混合整数规划模型, 采用基于非约束规划的惩罚方法来求解模型. Anjos 等[8] 建立了 SRLP 的二次规划模型, 提出了一种结合模型的半正定规划松弛和割平面的求解方法. 求解 SRLP 的启发式算法包括构造启发式算法 (Constructive heuristic) 和现代启发式算法 (Meta-heuristic). 构造启发式算法利用规则逐一安排机器, 以得到机器在一行上的排序[9]. 现代启发式算法通过循环迭代来寻求 SRLP 的最优或次优解. 例如, Datta 等[10] 提出一种置换遗传算法求解 SRLP, 利用机器的随机置换和规则来产生候选个体, 设计了一种特殊的交叉和变异算子来进化群体. Kothari 等[11] 提出两种禁忌搜索算法用于 SRLP, 一种禁忌搜索采用 2-opt 邻域, 另一种采用插入邻域. Samarghandi 等[12] 提出了一种用于 SRLP 微粒群算法.

多行设备布局问题 (Multiple row layout problem, MRLP) 是把机器分配到多行上并确定每行上的机器排序, 机器间仍以给定的最小间距分隔, 优化目标为最小化物料流成本[13]. 通常把 MRLP 建模为二次分配问题 (Quadratic assignment problem, QAP)[14]. Kouvelis 等[15] 建立了 MRLP 的整数规划模型, 用动态规划方法获得机器排序. Gen 等[16] 利用遗传算法解决 MRLP, 用模糊集合来表示机器间的最小距离. Sadrzadeh[17] 提出一种结合启发式过程的遗传算法用于 MRLP. 个体表示为矩阵编码, 通过两个启发式过程来产生初始群体, 设计了启发式交叉和变异算子.

SRLP 和 MRLP 仅需确定机器在一行或多行上的排序, 因而是组合优化问题. Chung 等[18] 提出了双行设备布局问题 (Double row layout problem, DRLP), 该问题在半导体加工和精密仪器制造等领域具有广阔应用背景. 与 SRLP 和 MRLP 不同, DRLP 不仅需确定机器在两行上的排序, 而且需确定机器的精确位置, 因而比 SRLP 和 MRLP 更为复杂和难以求解. Chung 等[18] 提出了 5 个启发式算法, 然而, 即使对于小规模问题, 这些算法获得的解与最优解也有很大差距. Zhang 等[19] 指出文献 [18] 中的 DRLP 模型可能忽略相邻机器间的最小间距, 提出一种 DRLP 的修正模型. Murray 等[20] 研究了带有不对称物料流的 DRLP, 建立了其混合整数规划模型并提出了一种结合构造启发式算法和局部搜索的求解方法. 张则强等[21] 提出了 DRLP 的 3 种优先规则启发式算法并与文献 [18] 中的最好算法进行比较. Amaral[22] 提出一种不考虑相邻机器间最小间距的 DRLP 及其精确求解算法.

以上关于 DRLP 的研究仅以物料流成本作为优化目标, 而没有考虑布局的面积. 然而, 在半导体加工等领域, 生产车间的建造成本非常高昂[23], 因此, 除了物料流成本, 布局面积也是设备布局需要考虑的重要因素. Murray 等[24] 提出了一种扩展的双行设备布局问题 (Extended double row layout problem, EDRLP), 其中假设双行间的距离为非零且同时考虑了物料流成本和布局面积两个优化目标, 将两目标线性加权组合为单个目标, 然后利用 IBM 的优化软件包 CPLEX 求解. 然而, 由于物料流成本和布局面积这两个目标的量纲不同, 文献 [24] 中将两目标线性组合的方法很难给定每个目标合适的权值. 此外, 由于 EDRLP 的 NP-hard 性质, 文献 [24] 中的 CPLEX 求解方法只能获得小规模问题 (包含不大于 12 个机器) 的最优解, 而不能在合理时间内求解大规模问题.

本文提出一种结合多目标免疫算法 (Multi-objective immune algorithm, MIA) 和线性规划 (Linear programming, LP) 的方法 (MIA-LP) 来求解 EDRLP. 首先, 建立了 EDRLP 的混合整数规划模型; 然后, 基于模型把问题分解为组合 (机器排序) 和连续 (机器精确位置) 两方面, 分别设计了 MIA 用于获取非支配的机器排序, 提出了一种基于 LP 的方法来构造任一非支配机器排序对应的连续非支配解集; 最后, 由所有的连续非支配解集来产生最后 Pareto 解.

1　扩展的双行设备布局问题

EDRLP 是将 m 个机器排列于两行上, 需确定机器在两行上的分配, 每一行上的机器排序, 以及每个机器在其所在行上的精确位置.

令 $I=\{1,\cdots,m\}$ 为机器集合, $R=\{1,2\}$ 为行集合. 每对机器 $i\in I^1$ 和 $j\in I^2$ 间的最小距离为 a_{ij}, 它们间物料流为 f_{ij}, 其中 $I^1=\{1,\cdots,m-1\}$, $I^2=\{i+1,\cdots,m\}$. 每个机器 $i\in I$ 的宽度和深度分别为 w_i 和 d_i. 两行间距为 c. EDRLP 的决策变量见表 1, 其混合整数规划模型如下.

$$\text{Min}\quad \{Obj_1, Obj_2\} \tag{1}$$

$$Obj_1=\sum_{i\in I^1}\sum_{j\in I^2}(f_{ij}+f_{ji})(v_{ij}^{+}+v_{ij}^{-}+c(1-q_{ij})) \tag{2}$$

$$Obj_2=A \tag{3}$$

s. t.

$$x_{ir}\le My_{ir},\quad \forall\, i\in I, r\in R \tag{4}$$

$$\sum_{r\in R}y_{ir}=1,\quad \forall\, i\in I \tag{5}$$

表 1 决策变量

Table 1 Decision variables

决策变量	描述
x_{ir}	连续决策变量; 表示机器 $i \in I$ 在行 $r \in R$ 上的位置; 若机器 i 不在 r 行上, 则 $x_{ir} = 0$.
y_{ir}	二进制决策变量; 若机器 $i \in I$ 在行 $r \in R$ 上, 则 $y_{ir} = 1$; 否则 $y_{ir} = 0$.
z_{rij}	二进制决策变量; 若机器 $i \in I$ 在行 $r \in R$ 上且在机器 $j \in \{I \setminus i\}$ 的左侧, 则 $z_{rij} = 1$; 否则 $z_{rij} = 0$.
W	连续决策变量; 布局的宽度.
s_r	连续决策变量; 行 $r \in R$ 上的布局面积.
A	连续决策变量; 布局的面积, 即包含所有机器的最小矩形的面积.
q_{ij}	二进制决策变量; 若机器 $i \in I^1$ 和 $j \in I^2$ 在同一行, 则 $q_{ij} = 1$; 否则 $q_{ij} = 0$.
v_{ij}^+, v_{ij}^-	辅助连续决策变量, 用于确定机器 i 和 j 的水平距离的绝对值.

$$\frac{w_i y_{ir} + w_j y_{jr}}{2} + a_{ij} z_{rji} \le x_{ir} - x_{jr} + M(1 - z_{rji}),\ \forall\, i \in I^1, j \in I^2,\ r \in R \quad (6)$$

$$\frac{w_i y_{ir} + w_j y_{jr}}{2} + a_{ij} z_{rij} \le -x_{ir} + x_{jr} + M(1 - z_{rij}),\ \forall\, i \in I^1, j \in I^2,\ r \in R \quad (7)$$

$$\sum_{r \in R} x_{ir} - \sum_{r \in R} x_{jr} = v_{ij}^- - v_{ij}^+,\ \forall\, i \in I^1, j \in I^2 \quad (8)$$

$$z_{rij} + z_{rji} \le y_{ir},\quad \forall\, i \in I^1, j \in I^2, r \in R \quad (9)$$

$$z_{rij} + z_{rji} \le y_{jr},\quad \forall\, i \in I^1, j \in I^2, r \in R \quad (10)$$

$$z_{rij} + z_{rji} + 1 \ge y_{ir} + y_{jr}\ \forall\, i \in I^1, j \in I^2,\ r \in R \quad (11)$$

$$W \ge x_{ir} + \frac{1}{2} w_i y_{ir},\quad \forall\, i \in I, r \in R \quad (12)$$

$$x_{ir} - \frac{1}{2} w_i y_{ir} \ge 0,\quad \forall\, i \in I, r \in R \quad (13)$$

$$s_r \ge d_i W - d_i M(1 - y_{ir}),\quad \forall\, r \in R, i \in I \quad (14)$$

$$A = s_1 + s_2 + cW \quad (15)$$

$$q_{ij} = \sum_{r \in R} (z_{rij} + z_{rji}),\quad \forall\, i \in I^1, j \in I^2 \quad (16)$$

$$x_{ir} \ge 0,\quad \forall\, i \in I,\ r \in R \quad (17)$$

$$v_{ij}^+, v_{ij}^- \ge 0,\quad \forall\, i \in I^1,\ j \in I^2 \quad (18)$$

$$y_{ir} \in \{0, 1\},\quad \forall\, i \in I,\ r \in R \quad (19)$$

$$z_{rij} \in \{0, 1\},\quad \forall\, i \in I, j \in \{I \setminus i\}, r \in R \quad (20)$$

$$q_{ij} \in \{0, 1\},\quad \forall\, i \in I^1,\ j \in I^2 \quad (21)$$

$$s_r \ge 0,\quad \forall\, r \in R \quad (22)$$

$$A \ge 0 \quad (23)$$

$$W \ge 0 \quad (24)$$

目标函数 (1) 包含两个目标, 即物料流成本 (2) 和布局面积 (3). 约束条件 (4) 和 (5) 用于确保每个机器只能被放于一行上. 约束条件 (6) 和 (7) 用以保证任意两个机器满足最小间距约束. 约束条件 (8) 确定机器间的水平距离. 约束条件 (9) ∼ (11) 用于确定决策变量 z_{rij}, 当机器 i 和 j 位于同一行 r 上 (即 $y_{ir} = y_{jr} = 1$), 则 z_{rij} 和 z_{rji} 二者之一必为 1; 否则二者皆为 0. 约束条件 (12) ∼ (14) 用于确定布局宽度和每一行布局面积的下界. 约束条件 (15) 用于计算布局的面积, 即两行上机器所占面积和通道面积之和. 约束条件 (16) 用于确定机器 i 和 j 是否位于同一行. 式 (17) ∼ (24) 给定决策变量的定义域.

M 为一个充分大的常数, 给定为

$$M = \sum_{i \in I} \left\{ w_i + \max_{\substack{j \in I \\ j \ne i}} (a_{ij}) \right\} \quad (25)$$

2 结合 MIA 和 LP 的双行设备布局方法

对于 EDRLP, 若两个布局的机器排序相同但机器绝对位置不同, 则代表两个不同的解. 因此, 每个机器排序 S 对应实数空间中一个解集 $AD(S)$, 其中包含无数个解, 这些解的机器排序相同, 但机器的精确位置不完全相同. 这一特点使得 EDRLP 的 Pareto 前沿在目标空间中为多个片段, 每个片段对应一个机器排序, 其上的解是连续的 (由于机器位置的连续性). 当采用多目标优化算法解决这类问题时, 有研究表明[25]: 优化过程中算法会花费较大计算代价来寻求 Pareto 前沿上各片段间那些不存在的区域, 由此导致难以获得高质量的 Pareto 解集. 此外, 已有的多目标优化算法只能获得分布在 Pareto 前沿上的有限个解[26−27], 即用有限个解来近似连续的 Pareto 前沿, 而不能获得全部的连续的 Pareto 前沿.

EDRLP 的模型中包含二进制决策变量 (机器排序相关) 和连续决策变量 (机器精确位置相关). 本文把问题分解为组合部分 (即机器排序) 和连续部分 (即机器精确位置). 组合部分是 NP-hard 问题,

精确算法难以求解, 为此设计了多目标启发式算法来获取非支配的机器排序集合 $Nondomi$. 给定机器排序则意味着固定模型中的所有二进制决策变量 (q_{ij}、y_{ir}、z_{rij}), 此时模型变为纯线性规划模型, 众所周知, 线性规划能快速求解该模型以获得机器的最优精确位置. 每个机器排序 $S \in Nondomi$ 对应一个解集 $AD(S)$, 需要确定 $AD(S)$ 中所有非支配解 (表示为集合 $ND(S)$). 本文利用线性规划来构造任一非支配机器排序 $S \in Nondomi$ 所对应的非支配解集 $ND(S)$, 然后由 $Nondomi$ 中所有机器排序对应的 $ND(\cdot)$ 集合来产生最后 Pareto 解. 算法的整体流程见图 1.

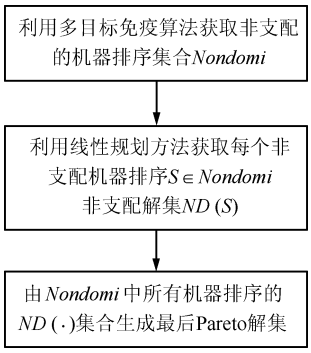

图 1　算法流程图

Fig. 1　Algorithm flow chart

2.1　利用 MIA 获取非支配的机器排序

多目标免疫算法是近年来发展起来的一种多目标优化算法, 具有很强的寻求 Pareto 前沿的能力[28−30]. 多目标免疫算法中的克隆选择算子具有局部搜索的特性[31], 而局部搜索是解决设备布局问题最有效的方法之一[11, 14, 32]. 为此, 本文针对 EDRLP, 设计了一种多目标免疫算法来寻求非支配的机器排序集合.

MIA 中的每个抗体表示为一个机器排序. 每一世代中随机选取一个目标作为活动目标, 用活动目标值来评价抗体. MIA 的步骤如下:

步骤 1. 初始化. 随机产生 N 个抗体组成当前群体 Pop; 令外部归档集 $Archive = \emptyset$, 进化的世代数 $gens = 1$.

步骤 2. 选择活动目标. 从式 (1) 中随机选择一个目标作为活动目标.

步骤 3. 选择. 用活动目标评价抗体, 按评价值从高到低对群体 Pop 中的抗体排序; 选出 $round(\alpha \times N)$ 个评价值最高的抗体组成群体 Pop_s, 其余 $N - round(\alpha \times N)$ 个抗体组成群体 Pop_l, 其中 $round(\cdot)$ 表示取整.

步骤 4. 克隆. 群体 Pop_s 中每个抗体进行克隆, 抗体克隆的数目与其评价值成正比, $round(\alpha \times N)$ 个抗体共产生 N 个克隆组成克隆群体 Pop_c.

步骤 5. 超突变. 群体 Pop_c 中的每个克隆执行交换或插入突变.

步骤 6. 更新. 群体 Pop_s 中对每个抗体利用其克隆对其进行更新; 更新 $Archive$; 当前群体更新为 $Pop = Pop_s \cup Pop_l$.

步骤 7. 每隔若干世代, 从 $Archive$ 中随机选取一个抗体来替换 Pop 中评价值最低的抗体.

步骤 8. 令 $gens = gens + 1$; 若 $gens \leq maxGens$, 则返回步骤 2; 否则, 算法结束, 由 $Archive$ 集合中的抗体 (机器排序) 构成非支配机器排序集合 $Nondomi$.

2.1.1　抗体的编码

抗体表示为机器在两行上的排序. 例如, 一个抗体 S 的编码如图 2 所示. 该抗体为实验中问题实例 $P10$ 的一个机器排序. 机器在第一行上的排序为 [3 2 10 8 1 7], 在第二行上的排序为 [6 4 5 9]. 这种编码方式易于执行交换和插入变异操作.

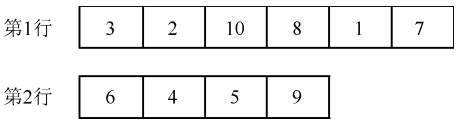

图 2　抗体的编码

Fig. 2　Antibody coding

该编码包括两方面信息: 1) 所有机器的排序; 2) 机器在两行上的分配 (两行上分别有多少机器). 若抗体编码长度为 n, 则机器排序的搜索空间为 $n!$, 机器分配的搜索空间为 $(n+1)$. 二者组合, 编码的搜索空间为 $(n+1) \times n! = (n+1)!$.

2.1.2　抗体的解码与评价

抗体 S 对应一个解集 $AD(S)$. 用 $AD(S)$ 中的解所能达到的最小面积和最小成本来评价抗体 S. 按照抗体 S (图 2) 中的机器排序, 依次将机器排列于两行且机器间以最小间距 a_{ij} 分隔, 得到的设备布局 S_a 如图 3 所示. 这种解码方式产生的解 S_a 能满足第 1 节中数学规划模型中的所有约束条件, 因而是可行解. 利用问题参数, 可计算 S_a 的物料流成本和面积. 显然, S_a 为 $AD(S)$ 中面积最小的解. 因此, 当活动目标为面积时, 用 S_a 的面积来评价抗体 S.

当活动目标为成本时, 利用线性规划方法 (见第 2.2 节) 可获得 $AD(S)$ 中成本最小的解 S_b. 然而, 这种方法需要评价每个抗体都求解一个线性规划问题, 非常耗时. S_a 是由机器间以最小间距分隔而构造, 而机器间距与成本相关, 小的机器间距会倾向于构造一个小成本的解, 因此 S_a 与 S_b 的成本值接近. 实验中发现, 一个抗体的 S_b 较小时, 通常其 S_a 也较小. 由于在抗体选择时只需要对抗体进行两两比较, 因此, 当成本为活动目标时, 用 S_a 的成本来近似评价抗体 S.

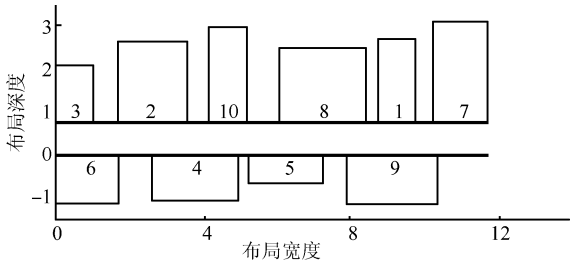

图 3 抗体 S 代表的设备布局 S_a

Fig. 3 Facility layout S_a represented by antibody S

2.1.3 克隆操作

将群体 Pop_s 中的抗体按评价值从高到低进行排序, 则第 k $(k = 1, \cdots, round(\alpha \times N))$ 个抗体的克隆概率为

$$P_k = \frac{2 \times round(\alpha \times N) + 1 - k}{\sum\limits_{i=1}^{round(\alpha \times N)} (i + round(\alpha \times N))} \tag{26}$$

Pop_s 中每个抗体产生若干克隆, 利用轮盘赌方法根据克隆概率来确定其产生克隆的数目.

2.1.4 突变操作

克隆群体 Pop_c 中的每个克隆执行交换或插入突变. 交换突变是交换任选两个机器的位置. 例如, 若图 2 所示的抗体执行交换突变, 则任选两个机器 (机器 2 和 9) 交换它们的位置而得到一个新抗体 (见图 4).

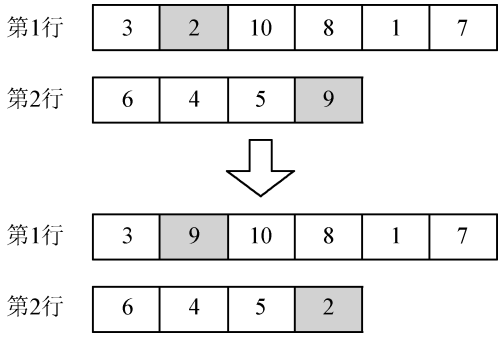

图 4 交换突变

Fig. 4 Swap mutation

交换突变只能改变两行上的机器排序, 不能改变每行上的机器数目. 为此, 采用插入突变来改变机器在两行上的分配. 以图 2 所示的抗体为例, 任选一个机器 (如机器 1), 将其插入到另一行的任一位置 (如机器 5 和 9 之间) 而得到一个新抗体 (见图 5).

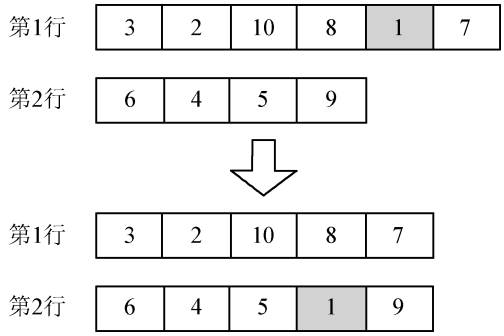

图 5 插入突变

Fig. 5 Insert mutation

交换和插入突变分别用于改变机器在两行上的排序和分配. 由于机器排序的组合数要远大于机器分配的组合数, 因此需要更多的克隆进行交换突变 (搜索机器排序). Pop_c 中每个克隆以 80 % 的概率执行交换突变, 以 20 % 的概率执行插入突变.

2.1.5 群体更新

假设任一抗体 $Ab \in Pop_s$ 产生 $cloneN(Ab)$ 个克隆, 其第 i $(i = 1, \cdots, cloneN(Ab))$ 个克隆变异为 Cl_i. 用 $\succ$ 和 $\prec$ 表示支配关系, 若 Ab 支配 Cl_i, 则表示为 $Ab \succ Cl_i$ 或 $Cl_i \prec Ab$. Pop_s 中每个抗体 Ab 执行以下更新过程:

用 Ab 更新 $Archive$
if $cloneN(Ab) = 0$ **then**
  用随机抗体替换 Ab
else
  for $i = 1$ to $cloneN(Ab)$ **do**
    if $Cl_i \succ Ab$ **then**
      $Ab = Cl_i$
      用 Cl_i 更新 $Archive$
    else
      if $Ab \succ Cl_i$ **then**
        舍弃 Cl_i
      else
        if $\nexists\, x \in Pop_s : x \succ Cl_i$ **then**
          以 0.1 的概率令 $Ab = Cl_i$
          用 Cl_i 更新 $Archive$
        else
          舍弃 Cl_i
        end if
      end if
    end if
  end for
end if

2.1.6 归档集更新

群体更新过程中也更新 $Archive$. 对于用于更新 $Archive$ 的任一抗体或变异克隆 x, $Archive$ 更新过程为

if *Archive* 中存在支配 x 的抗体 **then**
　舍弃 x
else
　if *Archive* 中存在被 x 支配的抗体 **then**
　　清除 *Archive* 中被 x 支配的抗体
　　将 x 加入 *Archive* 中
　else
　　if *Archive* 集合未满 **then**
　　将 x 加入 *Archive* 中
　　end if
　end if
end if

2.2　任一机器排序的连续非支配解集的构造方法

每个抗体 S 对应一个解集 $AD(S)$. 图 3 所示的解 s_a 是 $AD(S)$ 中具有最小面积的解. 用以下方法来获得 $AD(S)$ 中具有最小成本的解.

步骤 1. 用抗体 S 所代表的机器排序来固定模型中的所有二进制决策变量, 即 y_{ir}、z_{rij}、q_{ij}, 使模型变为纯线性规划模型, 表示为 P.

步骤 2. 以成本作为优化目标 (即 Obj_1), 用线性规划方法求解模型 P 以获得决策变量 x_{ir} 的最优值, 即确定机器在两行上的最优位置, 得到 s_b (见图 6).

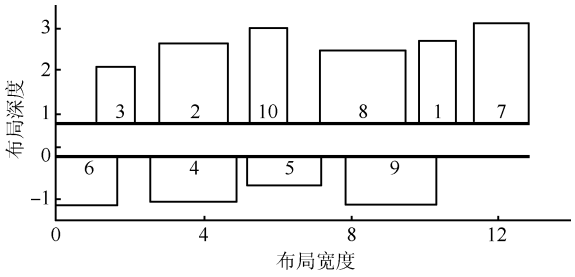

图 6　$AD(S)$ 中成本最小的解 S_b

Fig. 6　The solution with minimum cost, S_b, amongst $AD(S)$

线性规划采用 IBM 的数学规划软件包 CPLEX 实现. 从图 3 和 6 可看出, s_a 和 s_b 在两行上的机器排序相同但机器的精确位置略有不同, 且 s_b 比 s_a 的面积更大. 下面利用 s_a 和 s_b 来获取 $AD(S)$ 中的所有非支配解.

$AD(S)$ 中所有解在两行上机器分配相同, 即在每行上有相同的机器, 因此 $AD(S)$ 中的所有解在每行上的最深机器相同 (例如 s_a 和 s_b 在第一行上的最深机器均为机器 7). 故 $AD(S)$ 中所有解的深度 (从上行最深机器顶部到下行最深机器底部间的距离) 相同, 但宽度 (从最左边机器左侧到最右边机器右侧间的距离) 不同. 布局的面积为其深度和宽度的乘积, 因此 $AD(S)$ 中每个解的面积仅取决于其宽度. 令 $AD(S)$ 中所有解的深度为 D, s_a 和 s_b 的宽度分别为 W_a 和 W_b, 则它们的面积分别为 $W_a \times D$ 和 $W_b \times D$.

显然, $AD(S)$ 中的解所能达到的最小宽度为 W_a. 由于可把机器放在一行上的任意位置, 因此 $AD(S)$ 中的解所能达到的最大宽度为 $+\infty$. $AD(S)$ 中任一宽度大于 W_b 的解必被 s_b 所支配, 故 $AD(S)$ 中的非支配解的宽度必定在 $[W_a, W_b]$ 内. 用以下线性规划方法 (Linear programming method, LPM) 可确定 $AD(S)$ 中宽度不大于 $W \in [W_a, W_b]$ 的具有最小成本的解.

步骤 1. 用抗体 S 所对应的机器排序来固定模型中的所有二进制决策变量, 即 y_{ir}、z_{rij}、q_{ij}, 使模型变为纯线性规划模型 P.

步骤 2. 在模型 P 中加入以下约束条件, 以成本作为优化目标 (即 Obj_1), 得到模型 P_W.

$$A \le D \times W \tag{27}$$

其中, A 为面积决策变量.

步骤 3. 用线性规划方法求解 P_W 以获得 $AD(S)$ 中宽度不大于 W 的具有最小成本的解 $\pi(S, W)$.

对每个宽度 $W \in [W_a, W_b]$, 用以上方法可得到解 $\pi(S, W)$, 由此在理论上可得到一个连续的解集:

$$C(S) = \{\pi(S, W), W \in [W_a, W_b]\} \tag{28}$$

下面首先证明 $C(S)$ 中的解互不支配. 在此基础上, 证明 $C(S)$ 即为 $AD(S)$ 中的所有非支配解构成的集合 $ND(S)$. 由前可知, MIA 获得的每个非支配机器排序 $S \in Nondomi$ 对应一个解集 $AD(S)$, 因此也对应一个解集 $ND(S)$ (即 $C(S)$). 由此, 利用集合 $Nondomi$ 中所有机器排序的 $C(\cdot)$ 来产生最后 Pareto 解, 详见第 2.3 节.

定理 1. 对于任一机器排序 S, 集合 $C(S)$ 中的解互不支配.

证明. 假设 $\pi(S, W_1)$ 和 $\pi(S, W_2)$ 为 $C(S)$ 中任意两个解, 其中 $W_1, W_2 \in [W_a, W_b]$ 且 $W_1 > W_2$.

由于 P_{W_1} 为 P_{W_2} 的松弛, 因此 P_{W_1} 的最优解的目标函数值小于等于 P_{W_2} 的最优解的目标函数值. $\pi(S, W_1)$ 和 $\pi(S, W_2)$ 分别为 P_{W_1} 和 P_{W_2} 的最优解, 令它们的目标函数值分别为 $Cost(\pi(S, W_1))$ 和 $Cost(\pi(S, W_2))$, 则有 $Cost(\pi(S, W_1)) \le Cost(\pi(S, W_2))$.

令 $Wd(\cdot)$ 为解的宽度. 假设 $Wd(\pi(S, W_1)) < Wd(\pi(S, W_2))$, 则有 $Wd(\pi(S, W_1)) < Wd(\pi(S, W_2)) \le W_2$. 由于 P_{W_1} 为 P_{W_2} 的松弛, 则有 $Cost(\pi(S, W_1)) \le Cost(\pi(S, W_2))$. 同时, 又由于 $Wd(\pi(S, W_1)) < W_2$, 则有 $Cost(\pi(S, W_1)) \ge Cost(\pi(S, W_2))$ ($\pi(S, W_2)$ 为布局宽度小于 W_2 的

具有最小成本的解). 因此有 $Cost(\pi(S,W_1)) = Cost(\pi(S,W_2))$.

根据假设 $Wd(\pi(S,W_1)) < Wd(\pi(S,W_2))$, 可知 $\pi(S,W_1)$ 和 $\pi(S,W_2)$ 的宽度不同. 根据布局宽度的定义, 可知 $\pi(S,W_1)$ 和 $\pi(S,W_2)$ 中一定存在不同取值的 x_{ir} . 成本值由式 (8) 和 (2) 计算, x_{ir} 的变化将导致成本目标值的变化, 这与 $Cost(\pi(S,W_1)) = Cost(\pi(S,W_2))$ 矛盾. 因此假设 $Wd(\pi(S,W_1)) < Wd(\pi(S,W_2))$ 不成立, 即有 $Wd(\pi(S,W_1)) \geq Wd(\pi(S,W_2))$.

下面分两种情况讨论:

1) 若 $Wd(\pi(S,W_1)) = Wd(\pi(S,W_2))$, 则有 $Wd(\pi(S,W_1)) = Wd(\pi(S,W_2)) \leq W_2$. 由于 P_{W_1} 为 P_{W_2} 的松弛, 则有 $Cost(\pi(S,W_1)) \leq Cost(\pi(S,W_2))$. 又由于 $Wd(\pi(S,W_1)) \leq W_2$, 则有 $Cost(\pi(S,W_1)) \geq Cost(\pi(S,W_2))$. 因此, $Cost(\pi(S,W_1)) = Cost(\pi(S,W_2))$. 故 $\pi(S,W_1)$ 和 $\pi(S,W_2)$ 互不支配.

2) 若 $Wd(\pi(S,W_1)) > Wd(\pi(S,W_2))$, 则由前所述, $Cost(\pi(S,W_1)) \neq Cost(\pi(S,W_2))$. 考虑前文所述 $Cost(\pi(S,W_1)) \leq Cost(\pi(S,W_2))$, 则有 $Cost(\pi(S,W_1)) < Cost(\pi(S,W_2))$. 因此, $\pi(S,W_1)$ 和 $\pi(S,W_2)$ 互不支配. □

定理 2. 对于任一机器排序 S, 其非支配解集 $ND(S)$ 即为集合 $C(S)$.

证明. 首先证明 $ND(S)$ 中任一解都在 $C(S)$ 中, 然后证明 $C(S)$ 中任一解均在 $ND(S)$ 中.

1) 令 sl_1 为集合 $ND(S)$ 中任一解, 其宽度为 W_1. W_a 为机器排序 S 对应的解集 $AD(S)$ 中具有最小宽度的解的宽度, 因此有 $W_1 \geq W_a$. 若 $W_1 > W_b$, 则由于 $cost(\pi(S,W_b)) \leq cost(sl_1)$ 且 $W_b < W_1$, 故有 $sl_1 \prec \pi(S,W_b)$, 即 $sl_1 \notin ND(S)$. 这与前提矛盾, 故有 $W_1 \in [W_a,W_b]$.

$\pi(S,W_1)$ 为宽度不大于 W_1 的具有最小成本的解, 因此有 $cost(\pi(S,W_1)) \leq cost(sl_1)$. 若 $cost(\pi(S,W_1)) < cost(sl_1)$, 则有 $sl_1 \prec \pi(S,W_1)$, 即 $sl_1 \notin ND(S)$. 这与前提矛盾, 故有 $cost(\pi(S,W_1)) = cost(sl_1)$, 即 sl_1 为宽度不大于 W_1 的具有最小成本的解, 故有 $sl_1 \in C(S)$.

2) 令 sl_1 为集合 $C(S)$ 中任一解, 其宽度为 W_1. 若 sl_1 不在集合 $DN(S)$ 内, 则 $AD(S)$ 中必存在一个支配 sl_1 的解 sl_2, 即 $sl_2 \succ sl_1$. 令 sl_2 的宽度为 W_2, 由前所述, W_2 不小于 W_a. 下面分 $W_2 \in [W_a,W_b]$ 和 $W_2 > W_b$ 两种情况讨论.

a) 若 $W_2 \in [W_a,W_b]$, 则有 $Cost(\pi(S,W_2)) \leq Cost(sl_2)$.

若 $Cost(\pi(S,W_2)) = Cost(sl_2)$, 则 sl_2 为宽度不大于 W_2 的解中具有最小成本的解, 即 $sl_2 \in C(S)$. 由前提可知 $sl_1 \in C(S)$. 根据定理 1, sl_1 和 sl_2 互不支配. 因此, 假设 $sl_2 \succ sl_1$ 不成立, 即 $AD(S)$ 中不存在支配 sl_1 的解, 即 $sl_1 \in DN(S)$.

若 $Cost(\pi(S,W_2)) < Cost(sl_2)$, 由于 $Wd(\pi(S,W_2)) \leq W_2$ 且 $Cost(\pi(S,W_2)) < Cost(sl_2)$, 则有 $sl_2 \prec \pi(S,W_2)$. sl_1 和 $\pi(S,W_2)$ 均在 $C(S)$ 内, 则根据定理 1, 它们互不支配. 由于 $\pi(S,W_2) \succ sl_2$ 且 sl_1 不被 $\pi(S,W_2)$ 支配, 因此 sl_1 必不被 sl_2 支配, 即假设 $sl_2 \succ sl_1$ 不成立. 故 $AD(S)$ 中不存在支配 sl_1 的解, 即 $sl_1 \in DN(S)$.

b) 若 $W_2 > W_b$, 则有 $Cost(\pi(S,W_b)) \leq Cost(sl_2)$ 且 $Wd(\pi(S,W_b)) \leq W_b < W_2$, 因此 $sl_2 \prec \pi(S,W_b)$. $\pi(S,W_b)$ 和 sl_1 均在 $C(S)$ 内, 则根据定理 1, 它们互不支配. 由于 $\pi(S,W_b) \succ sl_2$ 且 sl_1 不被 $\pi(S,W_b)$ 支配, 因此 sl_1 必不被 sl_2 支配, 即假设 $sl_2 \succ sl_1$ 不成立. 故 $AD(S)$ 中不存在支配 sl_1 的解, 即 $sl_1 \in DN(S)$. □

2.3 构造最后 Pareto 解集

由定理 2 可知, 机器排序 S 的非支配解集 $ND(S)$ 即为集合 $C(S)$, 可由式 (28) 获得. 由于 $W \in [W_a,W_b]$ 是连续变量, 因此 $C(S)$ 中包含无数个解, 不可能由 LPM 全部获得. 令式 (28) 中的 W 值按步长 A_{step}/D 从 W_a 增加到 W_b, 其中 A_{step} 为给定的面积变化步长. 对于每个 W 值, 用 LPM 获得 $\pi(S,W)$, 由此可构造一个包含有限个解的集合 $C'(S)$ 来替代 $C(S)$. $C'(S)$ 中包含的解的数目取决于 A_{step} 值.

对 MIA 获得的每个非支配机器排序 $S \in Nondomi$, 用以上方法构造其 $C'(.)$ 集合, 然后由式 (29) 得到所有的非支配解集 NS.

$$NS = \bigcup_{S \in Nondomi} C'(S) \tag{29}$$

最后, 由式 (30) 从 NS 中获得最后 Pareto 解集 PS.

$$PS = \{s | s \in NS \text{ and } \nexists s' \in NS : s' \succ s\} \tag{30}$$

2.4 算法的复杂性分析

MIA-LP 包括 MIA 和 LPM. 首先分析 MIA 的时间复杂性. MIA 包括抗体亲和力计算、选择、克隆、超突变、群体更新、归档集更新等操作. 计算抗体亲和力的时间复杂度是 O(N) (N 为种群规模). 选择操作对 N 个抗体进行排序, 然后选取前 $round(\alpha \times N)$ 个抗体, 根据文献 [33], 该操作的计算复杂度为 O($round(\alpha \times N)$). 克隆和突变操作的计算复杂度均为 O(N). 群体更新操作在最差情况下的复杂度为 O($round(\alpha \times N) \times cN$) ($cN$ 为抗体所产生的最大克隆数目). 归档集更新在最差情况下的复杂度为 O($round(\alpha \times N) \times cN \times Archive\_size$).

根据 O(·) 的意义, MIA 的计算复杂度为 O($N \times cN \times Archive\_size$). 用 LPM 方法来产生非支配解的过程中, 需要求解 $|NS|$ 次线性规划问题, 其时间复杂度为 O($|NS|$). 因此, MIA-LP 的计算复杂度为 O($N \times cN \times Archive\_size + |NS|$). 这表明, MIA-LP 的计算时间取决于群体规模 (N)、抗体的最大克隆数目 (cN)、归档集规模 ($Archive\_size$)、以及集合 NS 中元素数目 ($|NS|$).

3　实验结果与分析

3.1　问题实例

EDRLP 的参数包括物料流 f_{ij}、机器间最小间距 a_{ij}、机器宽度 w_i、机器深度 d_i、两行间距 c. w_i 和 d_i 均为服从 unif[0.5, 2.5] 的随机数. a_{ij} 为服从 unif[0.25, 1.5] 的随机数. f_{ij} 按以下方式产生: 令产品类型数 $p \sim$ unif[8, 10], 加工每个产品类型所需机器数占总机器数的百分比 $r \sim$ unif[0.25, 0.75], 每个产品类型的产品数 $n \sim$ unif[20, 50], 则 f_{ij} 为依次在机器 i 和 j 上加工的产品总数. c 为服从 unif[0, $max_{i\in I}\{d_i\}$] 的随机数.

随机产生三个问题实例 $P10$、$P20$、$P30$, 分别包含 10、20、30 个机器. $P10$ 属于小规模问题, $P20$ 和 $P30$ 属于较大规模问题.

3.2　算法参数

MIA-LP 的参数包括 MIA 的参数和用 LPM 获取 $C'(\cdot)$ 集合所需的面积变化步长 A_{step}.

MIA 的参数包括进化世代数 $maxGens$、群体规模 N、选择率 α、归档集规模 $Archive\_size$. 对于免疫算法, α 一般取为 [0.1, 0.2] 之间的值, 在此取为 0.1. $Archive\_size$ 根据非支配机器排序的估计数目给定. 由于这三个问题实例中的每个实例的非支配机器排序的数目一般都不大于 50, 因此给定 $Archive\_size$ 为 50. 给定 $maxGens$ 适当的值以使算法能够充分收敛. 具体参数见表 2.

表 2　MIA 的参数

Table 2　Parameters of MIA

问题实例	$maxGens$	N	α	$Archive\_size$
$P10$	2 000	1 000	0. 1	50
$P20$	7 000	1 000	0. 1	50
$P30$	10 000	1 000	0. 1	50

A_{step} 的值取决于需要获取的 Pareto 解的数目. 对于 $P10$、$P20$、$P30$, A_{step} 分别给定为 0.1、0.3、0.5.

3.3　实验结果

用 Matlab 实现 MIA-LP, 运行在带有 3.21 GHz AMD Phenom CPU 和 2.0 GB 内存的 PC 机上.

为了能用 CPLEX 求解第 1 节中的模型, 将其两个优化目标按式 (31) 线性组合为单目标.

$$\text{Minimize} \left\{\alpha \times \frac{Obj_1}{z_1^*} + (1-\alpha) \times \frac{Obj_2}{z_2^*}\right\} \quad (31)$$

其中 z_1^* (z_2^*) 为单独优化成本 (面积) 目标而得到的理想目标值. $\boldsymbol{z}^* = (z_1^*, z_2^*)$ 为理想目标向量 (Ideal objective vector). 通过将两个目标函数分别除以其理想目标值, 把它们规范化为同一量纲[34]. α 为目标函数权值. 令 α 以给定步长从 0 增加到 1; 对每个 α 值, 以式 (31) 为优化目标, 用 CPLEX 求解模型以获得多个 Pareto 解. 对于小规模问题实例 $P10$, CPLEX 能快速获得其真实最优解 (平均用时小于 50 秒), 给定 α 步长为 0.02.

由于 EDRLP 的 NP-hard 性质, CPLEX 的运行时间将随着问题规模呈指数增长, 因此不能在合理时间内求解 $P20$ 和 $P30$. 实验中发现, 对于 $P20$ 和 $P30$, 对每个 α 值即使给定 CPLEX 的运行时间为 10 小时, CPLEX 也不能得到满意解. 因此, 对于这两个问题实例, 未将 MIA-LP 与 CPLEX 比较.

为了将 MIA-LP 与多目标启发式算法比较, 我们采用一种流行的多目标优化算法 — NSGA-II[35] 来解决 EDRLP. 文献 [36] 用 NSGA-II 求解块布局 (Block layout) 问题. 该问题只需确定机器的排序, 不需确定机器的精确位置. 我们将文献 [36] 中的 NSGA-II 进行扩展来解决 EDRLP.

文献 [36] 中, 一个解表达为两个整数向量, 第一个向量为所有机器的置换, 第二个向量为断点 (表示每行上的机器数目). 本文采用文献 [36] 中的方法来表达机器排序. 为了表示机器的精确位置, 设计了一个实数向量, 其第 i 个基因表示第 i 个机器与其左边机器的净间距 (不包括 a_{ij}). 由此, 用两个整数向量和一个实数向量来表示 EDRLP 的一个解. NSGA-II 中的整数和实数编码需分别进行不同的遗传操作. 对于整数向量, 采用文献 [36] 中的遗传均匀交叉 (Genetic uniform-based crossover) 和随机变异. 对于实数向量, 采用 NSGA-II[35] 的模拟二进制交叉 (Simulated binary crossover) 和实数变异. 选择和非支配排序等操作直接采用 NSGA-II 中的相应操作.

用 NSGA-II 解决 EDRLP 时, 交叉概率为 0.8, 变异概率为 $1/n$, 其中 n 代表问题中包含的机器数目. 为了使 MIA-LP 与 NSGA-II 进行公平比较, 使得二者在进化过程中的评价次数相同, 即对于 $P10$, $P20$, $P30$, NSGA-II 的进化世代数分别给定为 20 000, 70 000, 100 000.

对于每个问题实例, 分别独立运行 5 次 MIA-

LP 和 NSGA-II 来获取其 Pareto 解集. MIA-LP 包括 MIA 和 LPM 两部分, 前者用于获得非支配的机器排序, 后者用于从这些机器排序中获得 Pareto 解集. 对于 $P10$, 执行 5 次 MIA 获得的非支配机器排序在目标空间的目标函数值如图 7 所示. 由图可见, 每次运行 MIA 均得到相同的非支配机器排序, 其目标函数值在目标空间中分别用 ○、□、·、+、∗ 表示. 这表明 MIA 对于小规模问题具有很强的稳定性. 每次运行 MIA 均得到得到 7 个非支配机器排序, 在目标空间中分别用 $A \sim G$ 来表示, 其中机器排序 D 即为第 2.1 节和第 2.2 节中作为例子给出的机器排序 S.

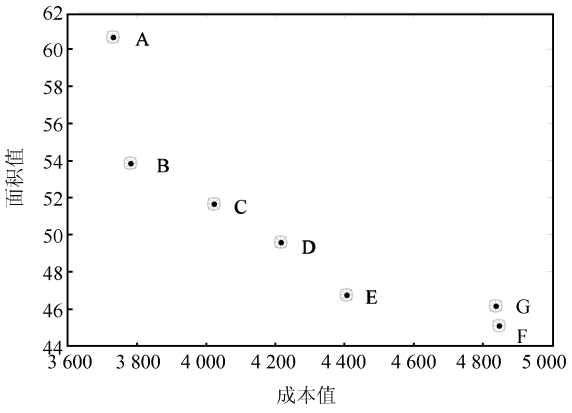

图 7 MIA 获得的 $P10$ 的非支配机器排序

Fig. 7 Non-dominated machine sequences obtained by MIA for $P10$

对于图 7 中的每个机器排序, 用 LPM 构造其对应的非支配解集 $C'(\cdot)$, 如图 8 所示. 图 8 中非支配的机器排序的目标函数值表示为 □. 由图 8 可见, 每个非支配的机器排序或者位于其对应的 $C'(\cdot)$ 的下端 (如 $B \sim G$), 或者位于其对应的 $C'(\cdot)$ 的下端水平偏右位置 (如 A). 这是因为, 对于任一机器排序 S, 是用机器间以最小间距分隔而构造的解 (即 s_a) 对其进行评价, 即将 s_a 的目标函数值作为机器排序 S 的目标函数值. s_a 是 $AD(S)$ 中具有最小面积的解. 若 s_a 也是该面积下成本最小的解, 即 $s_a \in C'(S)$, 则 s_a 位于 $C'(S)$ 的下端; 否则, 在 $C'(S)$ 中存在一个该面积下成本更小的解, 即该解与 s_a 的面积相同, 但比 s_a 的成本更小, 因此 s_a 位于 $C'(\cdot)$ 的下端水平偏右位置. 一般来说, $C'(\cdot)$ 中包含若干个解 (如 $C'(B) \sim C'(G)$), 但也可能仅包含一个解 (如 $C'(A)$). 若 $W_b > W_a$, 则用 LPM 可得到包含多个解的 $C'(\cdot)$, 其中解的数目取决于给定的面积变化步长 A_{step}; 若 $W_a = W_b$, 则表明成本最小的解 s_b 的面积也最小, 因此 $C'(\cdot)$ 中仅存一个非支配解, 即 s_b.

利用式 (30), 由所有 $C'(\cdot)$ 得到的最后 Pareto 解表示为图 9 中的大 "○". 5 次运行 NSGA-II 得到的解分别用小 ○、□、·、+、∗ 表示. CPLEX 得到的 Pareto 解表示为大 "◇". 由图可见, MIA-LP 获得的 Pareto 解是由多个不连续片段组成, 每个片段对应一个非支配的机器排序. CPLEX 仅能获取 4 个解, 这些解都位于各片段的边界点且均被 MIA-LP 所获取, 这说明 MIA-LP 能够获得小规模问题的真实 Pareto 最优解. NSGA-II 仅能得到一部分 MIA-LP 获得的 Pareto 解且结果不稳定. 在理论上 MIA-LP 能得到连续的 Pareto 前沿, 实际操作中能获得均匀分布在 Pareto 前沿上的任意多个解, 而 NSGA-II 和 CPLEX 只能获得有限个解. 多目标优化算法的性能通常用最优性和分布性指标来衡量. 最优性指标反映算法获取的 Pareto 解与真实 Pareto 前沿的接近度, 分布性指标反映算法获得的 Pareto 解能否均匀覆盖整个 Pareto 前沿. 图 9 表明 MIA-LP 对于小规模问题明显优于 NSGA-II 和 CPLEX.

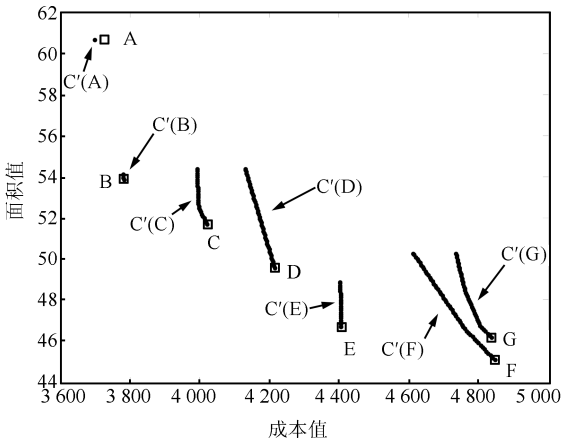

图 8 $P10$ 的非支配解集

Fig. 8 The set of non-dominated solutions for $P10$

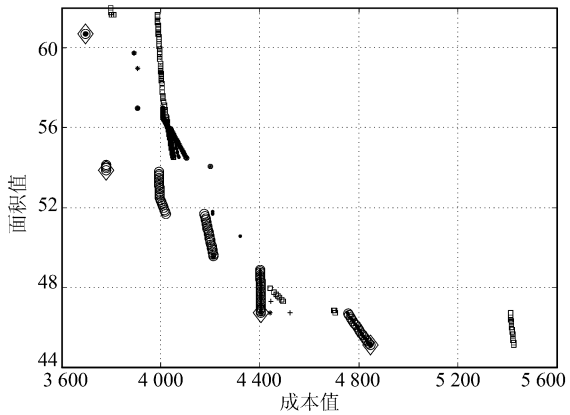

图 9 MIA-LP、NSGA-II、CPLEX 得到的 $P10$ 的 Pareto 解

Fig. 9 Pareto solutions obtained by MIA-LP, NSGA-II, and CPLEX for $P10$

MIA 获得的 $P20$ 的非支配机器排序见图 10, 其中 5 次运行 MIA 获得的非支配机器排序分别用

○、□、·、+、∗ 表示. 由图 10 可见, 每次运行 MIA 均能得到具有相似目标函数值的机器排序. 利用这些非支配机器排序, 由 LPM 获得的最后 Pareto 解见图 11, 其中 5 次运行得到的 Pareto 解分别用大○、□、·、+、∗ 表示. 运行 5 次 NSGA-II 得到的解也在图中给出, 分别用小○、□、·、+、∗ 表示. 由图 11 可见, 与 NSGA-II 相比, MIA-LP 能找到质量更高的 Pareto 解.

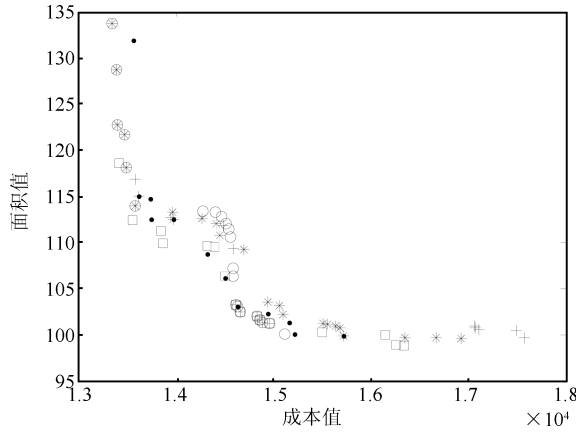

图 10　MIA 获得的 $P20$ 的非支配机器排序

Fig. 10　Non-dominated machine sequences obtained by MIA for $P20$

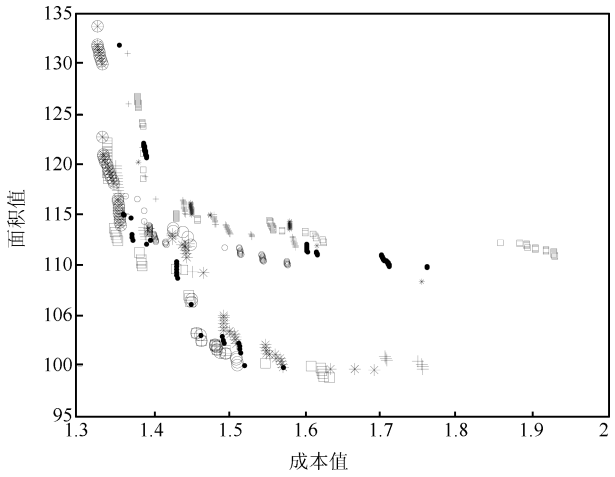

图 11　MIA-LP 和 NSGA-II 得到的 $P20$ 的 Pareto 解

Fig. 11　Pareto solutions obtained by MIA-LP and NSGA-II for $P20$

运行 5 次 MIA 获得的 $P30$ 的非支配机器排序如图 12 所示. 利用这些非支配的机器排序, 由 LPM 获得的最后 Pareto 解如图 13 所示. 图 13 中也给出了 5 次运行 NSGA-II 获得的解. 从图 13 可看出, MIA-LP 获得的 Pareto 解在解的质量和分布性方面远好于 NSGA-II 得到的解.

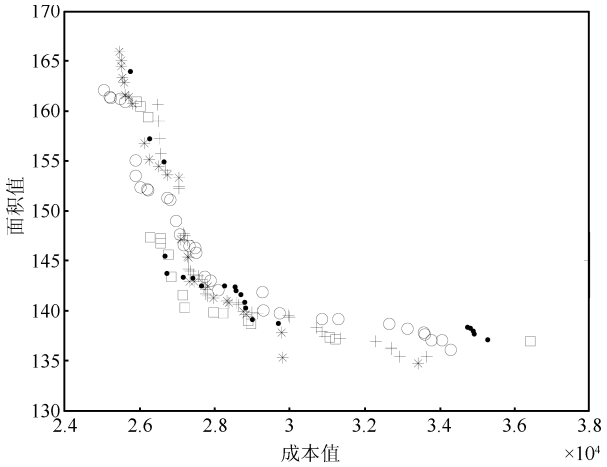

图 12　MIA 获得的 $P30$ 的非支配机器排序

Fig. 12　Non-dominated machine sequences obtained by MIA for $P30$

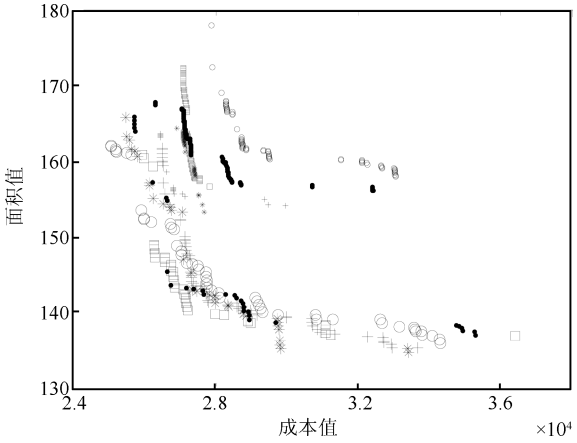

图 13　MIA-LP 和 NSGA-II 得到的 $P30$ 的 Pareto 解

Fig. 13　Pareto solutions obtained by MIA-LP and NSGA-II for $P30$

MIA-LP 和 NSGA-II 的平均计算时间以及 CPLEX 所用时间见表 3. 对于 $P10$, MIA-LP 用时大约为 10 分钟, CPLEX 用时约为 30 分钟, NSGA-II 比 MIA-LP 计算时间略长. 对于 $P20$ 和 $P30$, MIA-LP 比 NSGA-II 的平均用时更少. 由表 3 可见, MIA-LP 的时间主要开销在 MIA 上. 与 NSGA-II 相比, MIA 不需要实数编码和相应的操作且其操作更简单, 因此其计算时间更少.

3.4　算法参数灵敏性分析

MIA-LP 需要调整的参数主要包括选择率 α 和种群规模 N. 为了分析这两个参数对算法性能的影响, 分别给定它们不同的值, 观察 MIA-LP 求解 P20 获得的 Pareto 解的质量.

给定 α 不同的值 (其他参数按第 3.2 节给定), 对每个 α 值, MIA-LP 运行 5 次, 得到的 Pareto 解如图 14 所示. 其中○、□、Δ 分别表示 α 取为 0.1、

表 3 MIA-LP、NSGA-II、CPLEX 的计算时间比较 (秒)

Table 3 Comparison of computational time of MIA-LP, NSGA-II and CPLEX (s)

问题实例	MIA-LP 时间			NSGA-II 时间	CPLEX 时间
	MIA 平均时间	LPM 平均时间	总共时间		
$P10$	704	27	731	921	1 825
$P20$	3 894	9	3 903	5 104	–
$P30$	5 823	34	5 857	7 782	–

0.2、0.3 时 MIA-LP 得到的 Pareto 解. 由图 14 可见, 当 $\alpha = 0.1$ 或 0.2 时, MIA-LP 能得到相似的解. 当 $\alpha = 0.3$ 时, MIA-LP 获得的解的质量略有下降. 这是因为: 当 α 取值较大时, 每个抗体只能在其邻域产生较少的克隆, 使得局部搜索不充分. α 应在区间 [0.1, 0.2] 内取值.

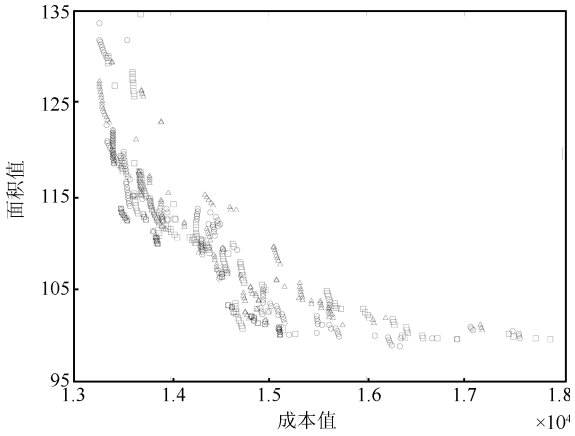

图 14 参数 α 对 MIA-LP 性能的影响

Fig. 14 Influence of parameter α on MIA-LP performance

N 的取值会影响每一世代的评价次数, 而评价次数是衡量进化算法的计算量的指标. 给定 N 分别为 500、1 000、1 500, 为了保持相同的评价次数, $maxGens$ 的值分别给定为 14 000、7 000、4 667, 其他参数按第 3.2 节给定. 对于每个 N 的取值, 运行 5 次 MIA-LP, 得到的 Pareto 解如图 15 所示. 其中 □、○、Δ 分别表示 N 取 500、1 000、1 500 时得到的解. 由图 15 可见, N 在较大范围内取值对 MIA-LP 的性能没有明显影响.

4 结论

本文提出一种基于多目标免疫算法和线性规划的方法 (MIA-LP) 来解决扩展的双行设备布局问题 (EDRLP). 基于该问题的模型, 把 EDRLP 分解为组合 (机器排序) 和连续 (机器精确位置) 两部分, 设计了一种多目标免疫算法 (MIA) 来获取非支配的机器排序集合, 用一种线性规划方法来构造任一非支配机器排序所对应的非支配解集. 最后, 由所有非支配解集产生 Pareto 解. 实验结果表明, MIA-LP 对于小规模问题能快速获得最优 Pareto 解, 对于大规模问题能得到高质量的解.

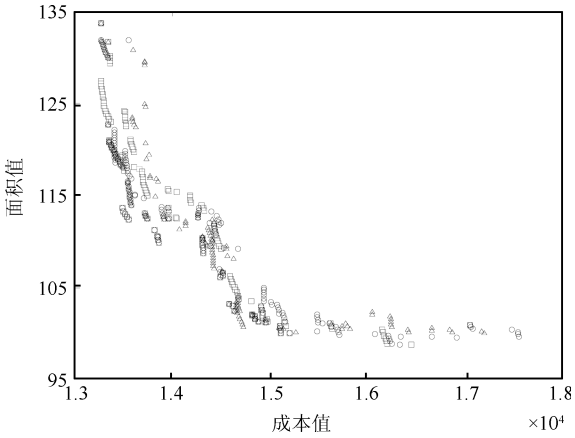

图 15 参数 N 对 MIA-LP 性能的影响

Fig. 15 Influence of parameter N on MIA-LP performance

EDRLP 是一类即有离散决策变量又有连续决策变量的多目标优化问题, 且其 Pareto 前沿由多段不连续的片段组成. MIA-LP 为这类问题的解决提供了有效方法, 其特点在于: 在理论上能够获得针对任一机器排序的最优 Pareto 解集且其中的解是连续的. 而当前广泛采用的多目标进化算法只能得到有限个 Pareto 解来近似连续的 Pareto 前沿.

References

1 Meller R D, Gau K Y. The facility layout problem: recent and emerging trends and perspectives. *Journal of Manufacturing Systems*, 1996, **15**(5): 351−366

2 Drira A, Pierreval H, Hajri-Gabouj S. Facility layout problems: a survey. *Annual Reviews in Control*, 2007, **31**(3): 255−267

3 Braglia M, Zanoni S, Zavanella L. Layout design in dynamic environments: strategies and quantitative indices. *International Journal of Production Research*, 2003, **41**(5): 995−1016

4 Heragu S S, Kusiak A. Machine layout problem in flexible manufacturing systems. *Operations Research*, 1988, **36**(2): 258−268

5 Solimanpur M, Vrat P, Shankar R. An ant algorithm for the single row layout problem in flexible manufacturing systems. *Computers & Operations Research*, 2005, **32**(3): 583−598

6 Djellab H, Gourgand A. A new heuristic procedure for the single-row facility layout problem. *International Journal of Computer Integrated Manufacturing*, 2001, **14**(3): 270−280

7 Heragu S S, Kusiak A. Efficient models for the facility layout problem. *European Journal of Operational Research*, 1991, **53**(1): 1−13

8 Anjos M F, Vannelli A. Computing globally optimal solutions for single-row layout problems using semidefinite programming and cutting planes. *INFORMS Journal on Computing*, 2008, **20**(4): 611−617

9 Kumar K R, Hadjinicola G C, Lin T L. A heuristic procedure for the single-row facility layout problem. *European Journal of Operational Research*, 1995, **87**(1): 65−73

10 Datta D, Amaral A R S, Figueira J R. Single row facility layout problem using a permutation-based genetic algorithm. *European Journal of Operational Research*, 2011, **213**(2): 388−394

11 Kothari R, Ghosh D. Tabu search for the single row facility layout problem using exhaustive 2-opt and insertion neighborhoods. *European Journal of Operational Research*, 2013, **224**(1): 93−100

12 Samarghandi H, Taabayan P, Jahantigh F F. A particle swarm optimization for the single row facility layout problem. *Computers & Industrial Engineering*, 2010, **58**(4): 529−534

13 Sadrzadeh A. A genetic algorithm with the heuristic procedure to solve the multi-line layout problem. *Computers & Industrial Engineering*, 2012, **62**(4): 1055−1064

14 Singh S P, Sharma R R K. Two-level modified simulated annealing based approach for solving facility layout problem. *International Journal of Production Research*, 2008, **46**(13): 3563−3582

15 Kouvelis P, Chiang W C, Yu G. Optimal algorithms for row layout problems in automated manufacturing systems. *IIE Transactions*, 1995, **27**(1): 99−104

16 Gen M, Ida K, Cheng C H. Multirow machine layout problem in fuzzy environment using genetic algorithms. *Computers & Industrial Engineering*, 1995, **29**(1−4): 519−523

17 Sadrzadeh A. A genetic algorithm with the heuristic procedure to solve the multi-line layout problem. *Computers & Industrial Engineering*, 2012, **62**(4): 1055−1064

18 Chung J, Tanchoco J M A. The double row layout problem. *International Journal of Production Research*, 2010, **48**(3): 709−727

19 Zhang Z Q, Murray C C. A corrected formulation for the double row layout problem. *International Journal of Production Research*, 2012, **50**(15): 4220−4223

20 Murray C C, Smith A E, Zhang Z Q. An efficient local search heuristic for the double row layout problem with asymmetric material flow. *International Journal of Production Research*, 2013, **51**(20): 6129−6139

21 Zhang Ze-Qiang, Cheng Wen-Ming. Decomposition strategies and heuristic for double row layout problem. *Computer Integrated Manufacturing Systems*, 2014, **20**(3): 559−568
(张则强, 程文明. 双行布局问题的分解策略及启发式求解方法. 计算机集成制造系统, 2014, **20**(3): 559−568)

22 Amaral A R S. Optimal solutions for the double row layout problem. *Optimization Letters*, 2013, **7**(2): 407−413

23 Turley J. *The Essential Guide to Semiconductors*. Upper Saddle River, NJ: Prentice Hall, 2002.

24 Murray C C, Zuo X Q, Smith A E. An extended double row layout problem. *Progress in Material Handling Research*. Charlotte, North Carolina: Material Handling Industry of America Press, 2012.

25 Chow C K, Yuen S Y. A multiobjective evolutionary algorithm that diversifies population by its density. *IEEE Transactions on Evolutionary Computation*, 2012, **16**(2): 149−172

26 Kong Wei-Jian, Chai Tian-You, Ding Jin-Liang, Wu Zhi-Wei. A real-time multiobjective electric energy allocation optimization approach for the smelting process of magnesia. *Acta Automatica Sinica*, 2014, **40**(1): 51−61
(孔维键, 柴天佑, 丁进良, 吴志伟. 镁砂熔炼过程全厂电能分配实时多目标优化方法研究. 自动化学报, 2014, **40**(1): 51−61)

27 Han Min, Liu Chuang, Xing Jun. A multi-objective evolutionary algorithm based on membrane system theory. *Acta Automatica Sinica*, 2014, **40**(3): 431−438
(韩敏, 刘闯, 邢军. 一种基于膜系统理论的多目标演化算法. 自动化学报, 2014, **40**(3): 431−438)

28 Zuo Xing-Quan, Mo Hong-Wei. *Immune Scheduling Principles with Applications*. Beijing: Science Press, 2013.
(左兴权, 莫宏伟. 免疫调度原理与应用. 北京: 科学出版社, 2013.)

29 Zuo X Q, Tan W, Lin H P. Cigarette production scheduling by combining workflow model and immune algorithm. *IEEE Transactions on Automation Science and Engineering*, 2014, **11**(1): 251−264

30 Zuo X Q, Mo H W, Wu J P. A robust scheduling method based on a multi-objective immune algorithm. *Information Sciences*, 2009, **179**(19): 3359−3369

31 Qi Y T, Liu F, Liu M Y, Gong M G, Jiao L C. Multi-objective immune algorithm with Baldwinian learning. *Applied Soft Computing*, 2012, **12**(8): 2654−2674

32 Yu M, Zuo X Q, Murray C C. A tabu search heuristic for the single row layout problem with shared clearances. In: Proceedings of the 2014 IEEE Congress on Evolutionary Computation. Beijing, China: IEEE, 2014. 819−825

33 de Castro L N, Von Zuben F J. Learning and optimization using the clonal selection principle. *IEEE Transactions on Evolutionary Computation*, 2002, **6**(3): 239−251

34 Miettinen K. *Nonlinear Multiobjective Optimization*. Boston: Kluwer Academic Publishers, 1999

35 Deb K, Pratap A, Agarwal S, Meyarivan T. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 2002, **6**(2): 182−197

36 Aiello G, Enea M, Galante G. A multi-objective approach to facility layout problem by genetic search algorithm and electre method. *Robotics and Computer-Integrated Manufacturing*, 2006, **22**(5−6): 447−455

左兴权 北京邮电大学计算机学院副教授. 2004 年获得哈尔滨工业大学控制科学与控制工程专业博士学位. 主要研究方向为智能优化与调度, 设备布局优化, 进化计算及应用. 本文通信作者.
E-mail: zuoxq@bupt.edu.cn
(**ZUO Xing-Quan** Associate professor at the Computer School, Beijing University of Posts and Telecommunications. He received his Ph. D. degree in control theory and control engineering from Harbin Institute of Technology in 2004. His research interest covers intelligent optimization and scheduling, facility layout optimization, and evolutionary computation with applications. Corresponding author of this paper.)

王春露 北京邮电大学计算机学院副教授. 1994 年获得哈尔滨工业大学计算机科学与技术专业硕士学位. 主要研究方向为计算智能, 云计算, 文件存储系统.
E-mail: wangcl@bupt.edu.cn
(**WANG Chun-Lu** Associate professor at the Computer School, Beijing University of Posts and Telecommunications. She received her master degree in computer science and technology from Harbin Institute of Technology in 1994. Her research interest covers computational intelligence, cloud computing, file, and storage systems.)

赵新超 北京邮电大学理学院教授. 主要研究方向为群体智能, 进化计算, 运筹优化及其工程应用.
E-mail: zhaoxc@bupt.edu.cn
(**ZHAO Xin-Chao** Professor at the School of Science, Beijing University of Posts and Telecommunications. His research interest covers swarm intelligence, evolutionary computation, operation research, and engineering applications.)