

一种求解双目标 flow shop 排序问题的进化算法

师瑞峰^{1,2}, 周泓²

(1. 北京航空航天大学计算机学院, 北京 100083; 2 北京航空航天大学经济管理学院, 北京 100083)

摘要: 提出一种求解双目标 flow shop 排序的递进多目标进化算法. 算法采用改进的精英复制策略, 在实现精英保留的前提下降低了计算复杂性; 通过递进进化模式增加群体多样性, 改善了算法收敛性; 通过群体进化过程中对非劣解集进行竞争型可变邻域启发式搜索, 增强了算法局部搜索性能. 采用新算法和参照算法 NSGA-II 对 31 个标准双目标 flow shop 算例进行优化. 研究结果表明, 新算法在所有算例的求解中均获得了优于 NSGA-II 的非劣解集, 验证了算法的有效性.

关键词: 多目标进化算法; 递进进化; 竞争型邻域搜索; 启发式局部搜索; flow shop

中图分类号: F406.6 **文献标识码:** A **文章编号:** 1007-9807(2007)05-0011-10

0 引言

多目标 flow shop 排序是一类典型的组合优化难题, 传统求解这类问题的方法通常为精确算法或启发式算法^[1]. 利用分支定界等精确方法求解这类问题, 尽管从理论上可以获得问题的最优解, 但由于解空间规模随着问题规模的增长呈指数方式增加, 因此难以求解较大规模问题^[2]. 长期以来, 研究人员针对各类排序问题提出了不同的启发式算法, 这类算法尽管可以在较短时间内获得问题的满意解或次优解, 但由于算法针对特定单目标优化问题所设计, 因而难以获得多个优化目标同时较优的满意解^[3]. 随着现代企业竞争压力的增大, 实际生产计划人员在制定作业排序计划时, 往往需要综合考虑多方面因素后才能确定最终方案, 因此获取这类问题满意解集的优化技术, 就逐渐成为该领域研究人员关注的焦点^[4,5].

多目标进化算法是 20 世纪 80 年代中期发展起来的一类优化技术, 自它诞生起就受到了排序

问题研究领域的广泛关注^[6~9], 特别是 2001 年后国际上召开两年一届的国际多目标进化算法会议, 更是推动了它在生产排序领域的应用^[10,11]. Murata 等人利用变系数加权多目标遗传算法求解多目标 flow shop 排序问题, 提出了 MOGLS 算法, 并将其应用于双目标 flow shop 问题的优化^[12]. Bagchi 采用改进的 ENGA 算法求解多目标 flow shop 问题, 对此做了大量的算例分析与比较研究^[13]. Basseur 等人针对 flow shop 排序问题设计了多目标进化算法系统^[14]. Amentano 等人采用基于禁忌搜索的进化策略求解双目标 flow shop 问题^[15]. Arroyo 等人提出一种求解 flow shop 问题的遗传局部搜索算法^[16]. 宋明华对混合流水生产系统的多目标协同优化问题进行了深入研究^[17]. 这些研究工作的开展极大丰富了多目标排序领域的研究内容.

本文在已有研究基础上, 提出一种求解双目标 flow shop 排序问题的递进多目标进化算法 (Escalating Multi-objective Evolutionary Algorithm, EMEA): 通过精英复制, 有效保留和利用算法已

收稿日期: 2005-12-29; 修订日期: 2007-07-15.

基金项目: 国家自然科学基金资助项目 (70371005; 70521001); 新世纪优秀人才支持计划 (NCET).

作者简介: 师瑞峰 (1977—), 男, 山西人, 博士后, Email: shi_rui_feng@buaa.edu.cn

搜索的非劣解集信息;通过递进进化策略,增强了算法收敛性,避免了算法早熟;通过启发式非劣解邻域搜索,提高了算法的局部搜索性能.采用本文算法对 31个标准的双目标 flow shop 算例进行优化,将其结果与现有代表算法 NSGA-II的优化结果进行比较,验证算法的有效性.

1 双目标 flow shop问题描述

排列型 flow shop问题可描述为: n 项待加工作业 $J = \{1, 2, \dots, n\}$ 按照一定的工艺路线依次通过 m 台不同的机器 $M = \{1, 2, \dots, m\}$ 进行处理,各项作业在每台机器上加工且仅加工一次.通常单目标 flow shop优化目标有最小化 makespan, total tardiness, total flowtime, weighted total tardiness等目标,这些目标分别反映了所有工件的完工指标以及作业安排的紧致程度.由于总完工时间 (makespan) 指标反映了总的加工任务完成情况,而加权总延误时间 (weighted total tardiness) 在所有延误相关 (due date related) 的求解模型中复杂度最高 (总延误时间 (total tardiness) 即为它的一个特例).因此本文双目标 flow shop排序的优化目标为同时最小化总完工时间 (makespan) 和加权总延误时间,优化模型为

$$(\min f_1 \min f_2) = \left(\min(C_{max}), \min \left(\sum_{j=1}^n w_j T_j \right) \right) =$$

$$\left(\min \left(\max_j(C_j) \right), \min \sum_{j=1}^n w_j (\max\{0, c_j - d_j\}) \right) \quad (1)$$

$$s.t. \quad \sum_{g=1}^m t_{gi} + P_{gi} \leq \sum_{h=1}^m t_{hi} + M \cdot (1 - x_{ghi})$$

$$i = 1, 2, \dots, m; \quad g, h = 1, 2, \dots, n \quad (2)$$

$$\sum_{g=1}^m t_{gi} + P_{gi} \leq \sum_{g(i+1)}^m t_{g(i+1)} \quad i = 1, 2, \dots, m;$$

$$g = 1, 2, \dots, n \quad (3)$$

$$\sum_{j=1}^n t_{ji} \leq \sum_{j=1}^n t_{ji}^E \quad 0 \quad i = 1, 2, \dots, m;$$

$$j = 1, 2, \dots, n \quad (4)$$

$$x_{ghi} = 0 \text{ 或 } 1 \quad \text{且满足} \quad x_{ghi} + x_{hgi} = 1$$

$$i = 1, 2, \dots, m; \quad g, h = 1, 2, \dots, n \quad (5)$$

模型中的参数含义: FS 为可行排序集合; o_{ji} 为作业 j 在机器 i 上的加工操作; p_{ji} 为操作 o_{ji} 的加工时间; d_j 为工件 j 的完工期限; w_j 为不同订单的延误权重; M 为任意大的正系数.

模型中的变量含义: t_{ji}^S 为操作 o_{ji} 的开始时间; t_{ji}^E 为操作 o_{ji} 的结束时间,因此: $t_{ji}^E = t_{ji}^S + p_{ji}$; C_j 为作业 j 的完工时间,则 $C_j = \max_i(t_{ji}^E)$;

$$x_{ghi} = \begin{cases} 1; & \text{若机器 } i \text{ 上作业 } g \text{ 为作业 } h \text{ 的紧前作业} \\ 0; & \text{否则} \end{cases}$$

上述模型中,式 (2) 表明在某一个时刻,机器 i 上最多只能有一项操作在加工;式 (3) 表明所有工件的工艺路线均相同;式 (4) 为非负约束;式 (5) 为 0-1 变量及其约束.

2 递进多目标进化算法

2.1 进化编码及其操作

Flow shop排序问题的解可编码为

$$[J_{[1]}, J_{[2]}, \dots, J_{[n]}]$$

其中, $J_{[k]} \in \{1, 2, \dots, n\}$ 且满足

$$\sum_{i=1}^n \dots \dots i(j) = 1, \quad j = 1, 2, \dots, n \quad (6)$$

式 (6) 中,

$$i(j) = \begin{cases} 1, & J_{[i]} = j \\ 0, & \text{otherwise} \end{cases} \quad i = 1, 2, \dots, n;$$

$$j = 1, 2, \dots, n \quad (7)$$

式 (6) ~ (7) 表明,每个工件的编号在编码中出现且仅出现一次.

Gen与 Cheng总结了现有 flow shop排列型基因的进化交叉和变异操作算子,对其进行了总结与分析^[18,19],Murata采用定量分析方法对求解 flow shop的遗传操作算子进行了比较研究^[20].本文基于这些研究结论,选用 PXM交叉算子和移位变异算子作为进化操作算子,这两种操作算子的实现方法详见文献 [18 ~ 20].

2.2 精英复制策略

现有多目标进化算法的复制策略设计思想为:在适应值规范化基础上,以个体适应值大小决定其在后代群体中的生存概率.这些算法构造的适应值计算方式不仅需要依赖于诸多敏感参数,而且多数算法构造的适应值并不能真正表达群体中解个体在目标空间分布的真实情况.因此,本文尝试采用一种不须计算适应值、而仅依据

Goldberg提出的非劣解等级优先及同等级内稀疏个体优先的 Pareto最优性原则,建立一种精英复制策略.该策略思想可描述为:

在每代进化结束时合并新旧进化群体,剔除混合群体中的重复个体,然后按照混合群体中个体的非劣解等级由低到高逐级复制整个等级的个体到下一代进化群体;若某一等级群体复制进入下一代群体后进化群体规模超过最大规模时,按照该等级内每一个体与其他个体距离稀疏优先的原则逐一将其中个体复制到下一代进化群体,直至达到规定的进化群体规模.精英复制策略具体实现过程为:

设当前群体为 P , 群体规模为 N , 对 P 进化操作生成的群体为 P' , 新、旧群体合并后的混合群体为 P'' , 下一代进化群体为 Q :

令 $P' = P \cup P''$, 对 $\forall x_i \in P'$, 若 $\exists x_k \in P$, $k \neq i$ 且满足 $x_i = x_k$, 则 $P' = P' \setminus x_k$;

依据目标值对 P' 进行 Pareto 分级, 得到 $S_i = P'$ 且 $S_1 \succ S_2 \succ \dots \succ S_r$, 其中 r 为 P' 的非劣解等级数, 每一等级内的个体数为 n_1, n_2, \dots, n_r , 令 $t = 1$;

若 $\sum_{k=1}^t n_k < N$, 则 $Q = Q \cup S_t$, $t = t + 1$, 重复; 否则转

计算 S_t 内任意个体距 x_j 其他个体的距离:

$$d_j = \sqrt{\sum_{k=1}^{n_t} d_{jk}^2} \tag{8}$$

并按照 d_j 从大到小顺序重新排列各 x_j 得到 $x_{(1)}$, $x_{(2)}, \dots, x_{(n_t)}$, 令 $l = 1$;

其中

$$d_{jk} = \sqrt{\sum_{i=1}^m [f_i(x_j) - f_i(x_k)]^2} \tag{9}$$

$x_{(1)} \in \{x_j\}$, 且满足 $d_{(1)} > d_{(2)} > \dots > d_{(n_t)}$

若 $\sum_{k=1}^{l-1} n_k + l < N$, 则令 $Q = Q \cup x_{(l)}$, $l = l + 1$, 重复; 否则选择完毕.

在上述算法步骤中, Q 中的新旧群体合并后按照非劣解等级优先复制, 可以保证已搜索到的精英个体不会在后代进化群体中遗失; 而剔除混合群体中的重复个体, 可以限制精英个体在后代

群体中出现且仅出现一次, 隐式地保证了进化群体多样性, 避免了精英个体在后代群体中大量繁殖而导致算法早熟.

2.3 递进进化结构

Coello提出一种基于微小群体进化的小生境群体遗传算法 (microGA), 并利用这种算法求解多目标优化问题, 获得了较好的结果^[21]. 在文献 [22] 初步研究递进进化结构的基础上, 本文提出一种改进的递进进化结构, 其进化思想可描述为:

生物群体从一个随机初始状态开始演化 (群体初始化), 以个体优秀等级的优劣决定其在后代进化过程中的繁殖概率 (依据非劣解等级优先复制个体), 每当灾变期 (可视为自然界的冰川期或大洪水) 来临时 (群体递进), 有一个“诺亚方舟”为生物群体保留其中的优秀个体 (外部伴随群体保留迄今为止搜索到的非劣解集); 度过灾变期后, 部分在“方舟”内度过劫难的生物个体重新回到生物进化群体中 (部分非劣解重新引入进化群体), 并且自然界又一次产生了空缺的生物物种 (通过初始化方式弥补进化群体中的不足个体), 然后, 进化得以延续, 这个过程持续到适应性最强的一个物种群体被找到 (搜索到真实的非劣解集).

以上述算法思想为进化策略的递进多目标进化算法 (EMEA) 可描述为: 采用递进进化结构, 将整个进化过程“分层”优化, 每次递进都以一定策略对群体加以重构, 以克服现有算法可能出现的早熟; 通过外部伴随群体对不同递进进化层之间的非劣解集信息加以保留和交换, 保证了算法优化的延续性; 再通过对每代进化群体的非劣解进行竞争型启发式局部搜索, 加速了群体的演化过程与算法收敛速度. 此处, 群体递进过程包含两层含义: 通过从外部伴随群体中按照稀疏优先的原则选择一定比例的精英个体进入下一层进化群体, 以保证进化的继承性和延续性; 通过随机初始化其他进化个体, 保证群体进化的多样性, 避免算法早熟.

上述递进进化算法的流程图如图 1 所示.

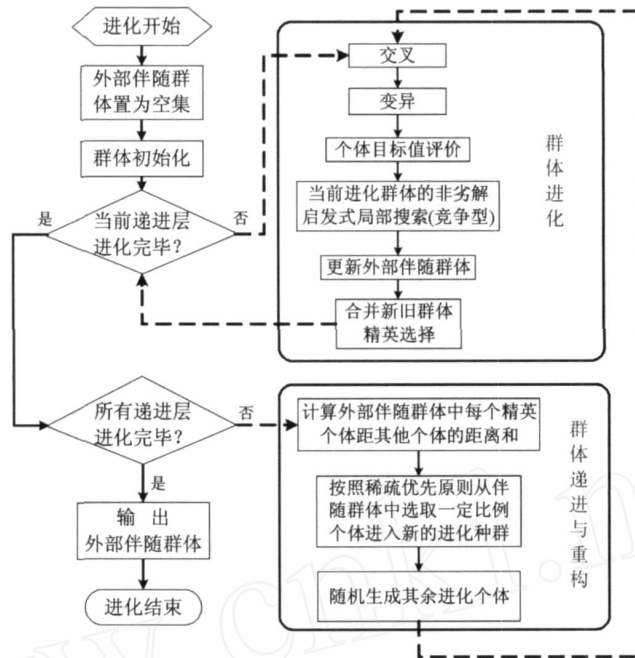


图 1 基于递进进化结构的进化算法流程

Fig 1 Algorithm flow chart for escalating based evolutionary structure

2.4 竞争型启发式可变邻域局部搜索策略

以遗传算法为代表的进化算法在优化时常常难以快速有效地对已发掘的“潜在最优解”做有效的局部搜索,本文针对已有算法的这种不足,提出了一种基于启发式搜索的改进邻域搜索算法。

可变邻域搜索算法由 Hansen、Nenald 等人最早提出,它是一种利用局部搜索过程中改变邻域结构及邻域范围的搜索策略^[23,24]。本文提出一种将 flow shop 排序启发式算法引入到可变邻域搜索的改进算法:竞争型可变邻域启发式搜索算法 (Tournament Variable Neighborhood Meta-heuristic Search, TVNMS)。假设进化算法的外部保留非劣解集为 A。则本文 TVNMS 算法过程描述为:

对当前进化群体非劣解集中的每个个体,执行下面的局部搜索及局部微搜索:

分别以每种求解 low shop 排序问题的启发式算法 (如 Palmer, Gupta, CDS, RA, NEH 等,设有 q 种) 为启发式策略,按照下面方式生成当前非劣解 x_i 的邻域新解 $\{ (x_i)^p \} (p = 1, 2, \dots, q)$ (设 x_i 的编码为 $[s_1 \ s_2 \ \dots \ s_n]$):

以 Palmer 算法为例构造邻域解的过程为:在 x_i 编码中随机产生两个截断点 $0 < C_1, C_2 < n$ (n 为 flow shop 问题总的工件数),在 C_1, C_2 两个截断点之间,按公式 (10) 计算截断部分各工件对应的 j 指标

$$j = \sum_{i=1}^m \left(i - \frac{m+1}{2} \right) \times p_{ij};$$

$$j = C_1, C_1 + 1, \dots, C_2 \quad (10)$$

其中, p_{ij} 是工件 j 在机器 i 上的加工时间, m 是 flow shop 问题总机器数;

对 C_1, C_2 截断点之间的工件序号按照 j 降序排列, $i_{[C_1]} > i_{[C_1+1]} > \dots > i_{[C_2]}$, 得到新的部分序 $s_{[C_1]} \ s_{[C_1+1]} \ \dots \ s_{[C_2]}$, 将排序后的编码与 C_1 之前、 C_2 之后的编码连接起来得到新解 $(x_i)^1$ 。

$$(x_i)^1 \leftarrow [s_{[1]} \ s_{[2]} \ \dots \ s_{[C_1]} \ s_{[C_1+1]} \ \dots \ s_{[C_2]} \ \dots \ s_{[n]}]$$

分别以其他的启发式算法为核心,构造 x_i 的另外 $q - 1$ 个邻域解 $(x_i)^2, \dots, (x_i)^q$, 选择其中的非劣解进入局部微搜索 (设此非劣解子集为 S);

对 $\forall (x_i)^p \in S$, 采用相邻结构的置换变换, 得到新解 $(x_i)^p$ 。若 $(x_i)^p > (x_i)^p$, 则 $(x_i)^p \rightarrow (x_i)^p$; 否则继续进行邻域置换变换, 直至所有邻域置换遍历完毕, 转 ;

令 $A = A \cup S$, 对 $\forall y_i \in A$, 若 $\exists y_k \in A$, $k \neq i$ 且满足 $y_i > y_k$, 则 $A = A \setminus y_k$;

若当前非劣解局部搜索过程尚未完成, 转 ;

若所有非劣解完成局部搜索, 则可变换邻域局部搜索完毕; 否则转 .

采用竞争型可变邻域局部搜索策略, 既有效利用了现有求解 flow shop 启发式算法快速寻优

(可高效地获得次优或满意解)的特点,将局部搜索策略应用于改进潜力较大的非劣解个体上,避免了盲目的局部搜索;同时,借鉴可变邻域搜索思想,使算法根据当前非劣解集信息自动调整搜索邻域,提高了局部微搜索效率.

2.5 递进多目标进化算法流程

设置外部伴随非劣解集为空集;

随机生成初始进化群体;

判别当前递进层进化是否完成,若未完成,转;

判别所有递进层进化是否完成,若未完成,转;

输出外部伴随非劣解集,算法结束;

执行群体进化过程:通过交叉和变异,生成新群体,评价新群体中各个体的目标值;对当前进化群体中的非劣解个体进行竞争型启发式可变邻域搜索;更新外部伴随群体;合并新、旧群体,在目标空间内进行非劣解分级,按照精英复制策略选择下一代进化群体;转;

执行群体递进与重构:从外部伴随群体中选取一定比例最为稀疏的精英个体进入下一层进化群体;随机生成其他进化个体;转.

2.6 算法复杂性分析

假设总的递进层数为 E_{\max} , 每层进化代数数为 G_{\max} , 群体规模 N , 外部伴随群体规模为 N_A , 评价单个个体目标值的时间花费为 f_k , 计算个体之间距离的时间花费为 f_d , 启发式搜索策略种类为 q , flow shop 问题的工件数为 n , 则本文递进多目标进化算法的复杂度为:

第 步时间花费为 $N + N \cdot f_k$; 第 、 、 、 、 花费的时间开销可忽略;主要的算法时间开销在第 步,其中:

交叉和变异及评价目标值: $G_{\max} \cdot E_{\max} \cdot N + G_{\max} \cdot E_{\max} \cdot N \cdot f_k$;

可变邻域搜索的最大时间花费为: $G_{\max} E_{\max} \cdot (qn) \cdot f_k$;

更新外部伴随群体的最大时间花费为: $G_{\max} \cdot E_{\max} \cdot N_A^2$;

合并群体后,非劣解分级最坏情况是混合群体中的非劣解等级只分一级的情况,它所对应的分级时间为 $G_{\max} \cdot E_{\max} \cdot (2N)^2$, 这种情况下,精英选择过程的时间花费也最大,计算混合群体内两两个体之间距离的时间花费为 $G_{\max} \cdot E_{\max} \cdot (2N)^2 \cdot f_d$;

综上可得,本文求解 flow shop 问题的递进多目标进化算法的复杂性可估算为

$$o(EMRA) = G_{\max} \cdot E_{\max} (N + (N + qn) f_k + (2N)^2 (1 + f_d) + N_A^2) \quad (11)$$

由于评价个体目标值 f_k 和计算同一非劣解等级内两个体间距离 f_d 过程的时间花费数量级高于其他过程的时间花费,因此式 (11) 中算法复杂性主要成分可进一步归约为式 (12)

$$\begin{aligned} o(EMRA) &\cong G_{\max} \cdot E_{\max} (N + qn) f_k + (2N)^2 (1 + f_d) \\ &\cong G_{\max} \cdot E_{\max} (N + qn) f_k + (N)^2 f_d \\ &\cong G_{\max} \cdot E_{\max} (2N)^2 f_d \\ &\cong G_{\max} \cdot E_{\max} (4N^2) \end{aligned} \quad (12)$$

从式 (12) 可以看出,本文算法在最坏求解情况下 (去除高阶小量后) 的计算复杂度为 $o(4N^2)$, 这一复杂度主要由算法第 步产生.

因此,本文算法是一种计算复杂度为 $o(4N^2)$ 的多项式算法. 这一计算复杂度与现有代表性算法 NSGA-II, MOGLS 及 SPEA 等算法的计算复杂度相当 (这些算法的计算复杂性均为 $o(MN^2)^{[25]}$, 其中, M 为优化目标个数, 本文 $M = 2$, 因此, 本文 EMEA 最坏情况下的计算复杂性约为 NSGA-II 的两倍).

3 算例分析

3.1 算例描述

为检验本文算法对双目标 flow shop 排序问题的有效性, 选取 OR_library 发布的 31 个 flow shop 算例 (包括 8 个小规模 CAR 算例、21 个中规模及大规模的 REC 系列算例和 2 个大规模 HEL 系列算例) 对本文算法进行验证. 算例优化模型如式 (1) ~ (5) 所述, 并根据求解问题规模大小不同, 将算例分为小规模问题 (工件数 14)、中等规模问题 (14 < 工件数 < 50) 和大规模问题 (工件数 50) 三种规模的算例.

由于 OR_library 公布的算例未包含交货期 (due date) 信息, 因此本文参照文献 [26, 27] 中的 TWK 思想, 利用公式 (13) 计算出各算例中的

工件交货期

$$d_j = k \prod_{i=1}^m p_{ji}, \quad j = 1, 2, \dots, n \quad (13)$$

其中, k 为 TWK方法中确定误工比例的常数. 此处取 $k = 1.5$ (即, 工件平均延误比例约为 50%).

权重 w_j 的选取原则为^[28]: 20% 的工件总延误权重取 4 (非常重要客户订单), 60% 的工件总延误权重取 2 (较重要客户订单); 其余 20% 的工件总延误权重取 1 (一般客户订单).

Deb等人提出的 NSGA-II算法, 是迄今为止公认最为有效的多目标进化算法之一^[29], 本文采用 NSGA-II作为参照算法, 验证 EMEA 算法求解双目标 flow shop问题的性能.

3.2 进化参数实验设计

为使验证算法 NSGA-II与本文 EMEA 算法的计算结果的比较具有公平性, 本文参照文献 [13] 求解 flow shop 排序问题的进化参数实验设计方法: 分别采用两种算法在不同参数设定条件下对几个代表不同规模算例的 flow shop 问题进行多次优化, 根据优化统计结果的优劣确定出多次运行中性能稳定且结果较优的参数对其他算例进行优化.

参数实验的变化范围:

交叉概率从 0.4 ~ 0.9 按照 0.1 递增方式变化;

变异概率从 0.1 ~ 0.3 按照 0.05 递增方式变化;

局部搜索步长从 5 ~ 50 按照 5 递增方式变化;

进化代数分别从 200 ~ 500 (小规模算例)、

200 ~ 1000 (中等规模算例)、500 ~ 2000 (大规模算例) 按照 100 方式递增变化.

每个规模算例选取两个算例进行参数实验, 根据最后统计结果, 确定出两种算法具有结果可比性的一组参数对所有算例进行优化.

3.3 进化参数设定

经过参数实验分析, 确定出最终进化参数:

采用 EMEA 与 NSGA-II 优化不同规模 flow shop 算例时, 不随问题规模变化的优化参数设置如下:

交叉概率 0.9; 变异概率 0.1; EMEA 选取 7 种启发式策略作为可变邻域局部搜索的新解构建策略. 在不同规模算例优化中, 随问题规模增长变化的优化参数设置为:

对小规模算例, 进化群体为 100; NSGA-II 的进化代数为 400, EMEA 的进化代数为 100 × 4,

即总共递进 4 层, 每层进化 100 代.

对中等规模算例, 进化群体为 200; NSGA-II 的进化代数为 400 或 500, EMEA 的进化代数为 100 × 4 或 100 × 5, 即总共递进 4 层或 5 层, 每层进化 100 代.

对大规模算例, 进化群体为 200; NSGA-II 的进化代数为 1000, EMEA 的进化代数为 200 × 5, 即总共递进 5 层, 每层进化 200 代.

3.4 结果分析

分别采用 EMEA 和 NSGA-II 算法对上述 31 个算例各自独立优化 20 次, 将各次独立运行所得的非劣解集合合并后, 筛选获得两种算法求解各算例的最终非劣解集, 图 2 ~ 5 给出了几个典型算例的非劣解集.

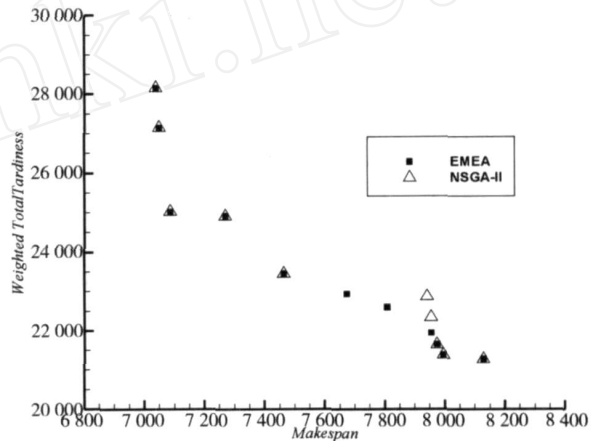


图 2 两种算法优化 CAR1 获得的非劣解集

Fig 2 CAR1's Pareto frontiers obtained by the two algorithms

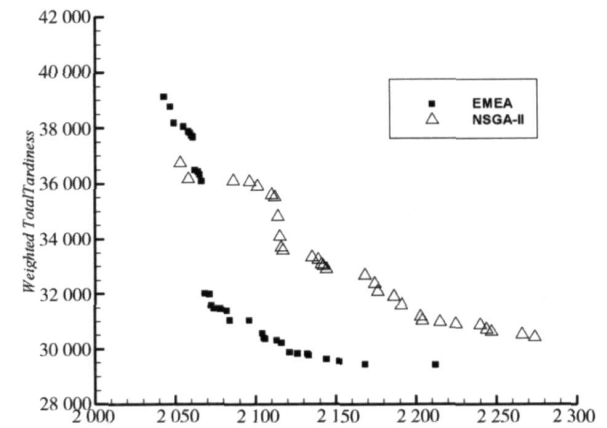


图 3 两种算法优化 REC19 获得的非劣解集

Fig 3 REC19's Pareto frontiers obtained by the two algorithms

本文 NSGA-II 分析程序的原型来源于 <http://delta.cs.cinvestav.mx/~ccoellb/EMOO/EMOOsoftware.html>, 这个程序是由算法提出者 Deb K 博士编写的求解函数优化的实数编码 NSGA-II 程序, 本文在此程序基础上将编码方式修改为求解 flow shop 排序的基于工件号的编码方式, 进化操作算子与本文 EMEA 采用的进化操作算子完全相同.

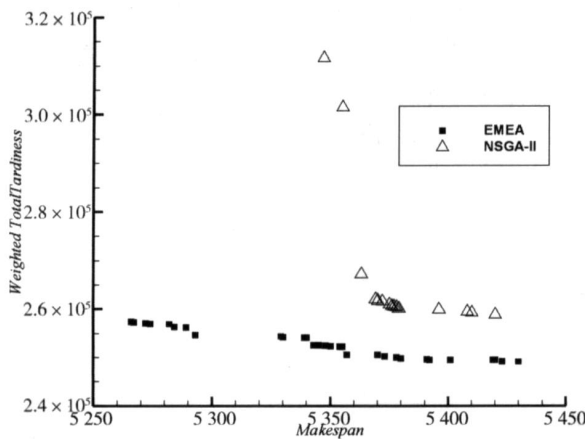


图 4 两种算法优化 REC37 获得的非劣解集

Fig 4 REC37's Pareto frontiers obtained by the two algorithms

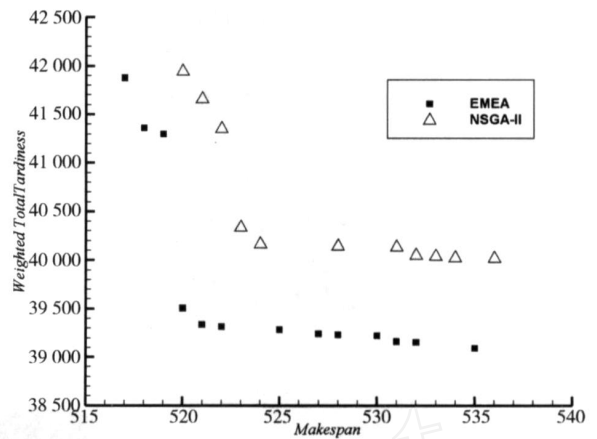


图 5 两种算法优化 HEL1 获得的非劣解集

Fig 5 HEL1's Pareto frontiers obtained by the two algorithms

表 1 采用 EMEA、NSGA-II 优化标准双目标 flow shop 算例的统计结果

Table 1 Statistical results of standard biobjective flow shop case studies obtained by EMEA and NSGA-II

问题	独立优化各算例所需平均时间 (单位: S)		C 指标统计 (20 次独立运行的统计值)			
	EMEA	NSGA-II	C (EMEA, NSGA-II)		C (NSGA-II, EMEA)	
			平均值	标准差	平均值	标准差
CAR1	1.254	1.711	0.995	0.022	0.079	0.200
CAR2	1.544	1.650	0.931	0.147	0.298	0.317
CAR3	1.492	1.680	0.975	0.112	0.396	0.397
CAR4	1.418	1.736	0.959	0.149	0.116	0.256
CAR5	1.310	1.700	0.979	0.096	0.400	0.366
CAR6	1.272	1.762	1.000	0.000	0.000	0.000
CAR7	1.004	1.712	1.000	0.000	0.552	0.470
CAR8	1.168	1.801	1.000	0.000	0.737	0.153
REC01	7.294	7.107	0.985	0.060	0.032	0.059
REC03	8.784	7.179	0.991	0.041	0.005	0.020
REC05	7.766	6.800	0.951	0.152	0.162	0.280
REC07	10.137	7.945	0.984	0.050	0.095	0.248
REC09	9.814	8.532	0.928	0.162	0.105	0.207
REC11	9.298	8.493	0.993	0.032	0.016	0.053
REC13	10.991	10.180	0.958	0.103	0.083	0.155
REC15	11.397	9.884	0.931	0.131	0.135	0.199
REC17	10.966	10.079	0.981	0.062	0.039	0.104
REC19	16.409	11.712	1.000	0.000	0.042	0.086
REC21	15.522	12.193	1.000	0.000	0.003	0.015
REC23	15.395	12.301	0.986	0.036	0.117	0.263
REC25	24.892	17.505	0.940	0.193	0.125	0.309
REC27	23.237	17.888	1.000	0.000	0.012	0.045
REC29	25.243	17.099	0.998	0.010	0.088	0.245
REC31	36.925	23.720	0.983	0.078	0.040	0.179
REC33	34.944	23.344	1.000	0.000	0.206	0.310
REC35	37.174	23.307	0.980	0.066	0.232	0.338
REC37	74.322	59.000	0.994	0.027	0.024	0.090
REC39	78.243	57.242	1.000	0.000	0.017	0.075
REC41	75.599	59.561	1.000	0.000	0.004	0.016
HEL1	81.227	43.097	0.984	0.074	0.094	0.193
HEL2	11.970	8.462	0.961	0.150	0.105	0.270

表 1 给出了 31 个算例各次独立优化的平均 运行时间 (机器配置为 P-IV 3.2G, 512M RAM)

及 C 指标统计结果 (包括反映 EMEA 优于 NSGA - I 程度的 $C(EMEA, NSGA-I)$ 与反映 NSGA-II 优于 EMEA 的 $C(NSGA-II, EMEA)$), 图 6、7 给出 C 指标统计结果图。

评价两种算法获得非劣解集 A、B 相对优劣的 C 指标由 Zitzler 提出^[30], 其定义为: 设为 A、B

$\subseteq X$ 两种算法优化获得的非劣解集, C 指标是一种值域定义在 (0, 1) 上、用来刻画 (A, B) 之间偏序能性的指标

$$C(A, B) = \frac{| \{ b \in B \mid \exists a \in A, a \succ b \} |}{| B |} \quad (14)$$

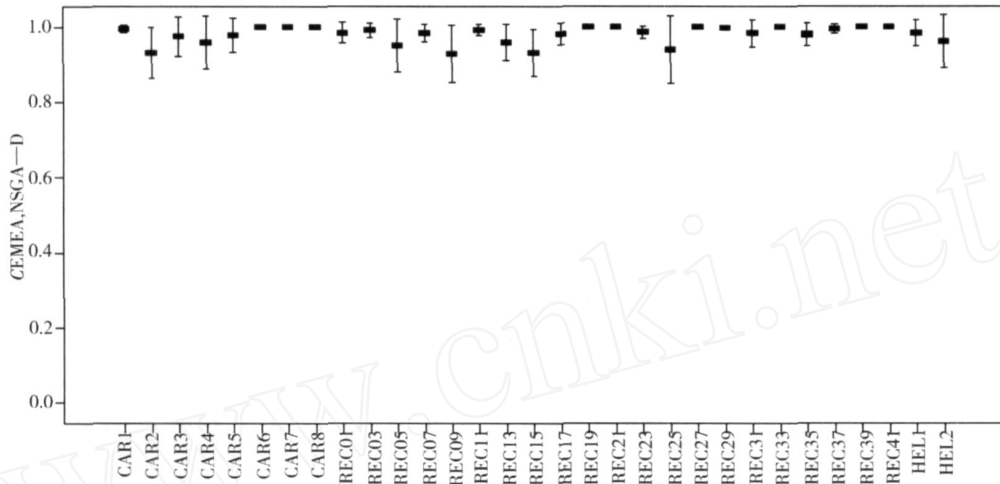


图 6 测试算例的 $C(EMEA, NSGA-II)$ 统计结果 (EMEA 优于 NSGA-II 的程度)

Fig 6 Statistical results of $C(EMEA, NSGA-II)$ with case studies (the optimality degree of EMEA by contrast to NSGA-II)

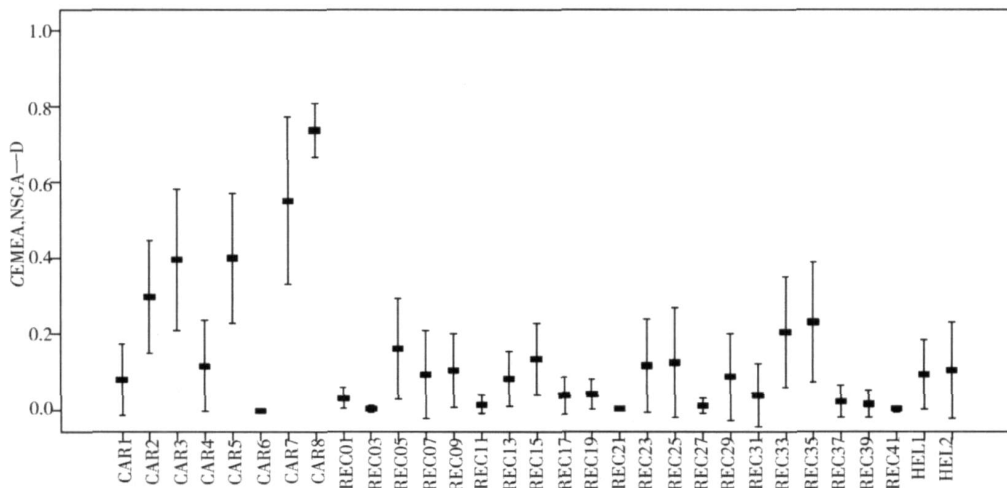


图 7 测试算例的 $C(NSGA-II, EMEA)$ 统计结果 (NSGA-II 优于 EMEA 的程度)

Fig 7 Statistical results of $C(NSGA-II, EMEA)$ with case studies (the optimality degree of NSGA-II by contrast to EMEA)

若 $C(A, B) = 1$, 说明对于 B 集中的任一个非劣解个体, A 集中总存在“优于”它的解个体; $C(A, B) = 0$, 说明对于 B 集中任一个解个体, A 集中都不存在“优于”它的解个体。

从图 2 ~ 5 可以看出: 两种算法优化各算例获

得的非劣解集中, EMEA 在小规模 CAR 系列算例中获得了略优于 NSGA-II 的非劣解集, 而在中大规模算例优化中获得了明显优于 NSGA-II 的非劣解集。

从表 1 的时间性能比较可以看出: EMEA 对

小规模算例优化的时间略少于 NSGA-II(CAR 系列算例的时间花费为 NSGA-II 的 0.59 - 0.94 倍),而中大规模的算例优化中 EMEA 的平均时间花费为 NSGA-II 的 1.03 ~ 1.88 倍,且花费时间的比例随问题规模增大而增加,这主要是由于 EMEA 在优化小规模问题时,其邻域搜索的时间花费较少,而随着问题规模的增大非劣解局部搜索的计算时间显著增加的缘故(邻域置换的时间花费),从而导致中大规模问题求解的算法时间性能下降,从 2.6 节的计算复杂性分析不难看出,EMEA 在最坏情况下的时间花费为 NSGA-II 的 2 倍。

从表 1 和图 6、7 的 C 指标统计结果可以看出:CAR 系列的 8 个小规模算例中,EMEA 取得了略优于 NSGA-II 的优化结果;而在中大规模算例优化中,EMEA 取得了明显优于 NSGA-II 的优化结果,这反映出 EMEA 较 NSGA-II 更适合处理中等规模、大规模的双目标 flow shop 问题。

总体而言,尽管本文 EMEA 在中大规模算例优化时的时间性能略逊于 NSGA-II,但这种差异可通过并行策略予以弥补(两种算法的计算时间

差异小于 2 倍,且仅随问题规模增长呈小幅变化);而本文方法在所有算例(尤其是中大规模算例)优化中均获得了较目前公认性能较优的 NSGA-II 更好的非劣解集,从而验证了本文 EMEA 优化双目标 flow shop 的有效性。

4 结束语

本文提出了一种求解双目标 flow shop 排序的递进多目标进化算法。算法采用改进的精英复制策略、递进进化策略及竞争型可变邻域启发式搜索策略,获得了良好的算法收敛性能。采用本文算法(EMEA)与现有的代表算法 NSGA-II 对 31 个典型的双目标 flow shop 算例进行优化,研究结果表明,本文算法在所有算例优化中均获得了优于 NSGA-II 的非劣解集,验证了算法有效性。如何将求解其他排序问题的启发式算法与本文的竞争型启发式邻域搜索策略相结合,解决更多类型的多目标排序问题,是进一步研究的重点。

参考文献:

- [1] Dudek R A, Panwalkar S S, Smith M L. The lessons of flow shop scheduling research[J]. *Operational Research*, 1992, 40(1): 7—13.
- [2] Pinedo M. *Scheduling—Theory, Algorithms, and Systems*[M]. New Jersey: Prentice Hall, 1995.
- [3] 唐恒永, 赵传立. 排序引论[M]. 北京: 科学出版社, 2002.
Tang Hengyong, Zhao Chuanli. *Introduction to Scheduling*[M]. Beijing: Science Press, 2002 (in Chinese)
- [4] Nagar A, Haddock J, Heragu S. Multiple and bicriteria scheduling: A literature survey[J]. *European Journal of Operational Research*, 1995, 88—104.
- [5] Tkindt V, Billaut J C. *Multicriteria Scheduling: Theory, Models and Algorithms*[M]. Berlin: Springer, 2002.
- [6] 谢涛, 陈火旺, 康立山. 多目标优化的演化算法[J]. *计算机学报*, 2003, 26(8): 997—1003.
Xie Tao, Chen Huowang, Kang Lishan. Evolutionary algorithms of multi-objective optimization problems[J]. *Chinese Journal of Computers*, 2003, 26(8): 997—1003. (in Chinese)
- [7] Coello C A C. An comprehensive survey of evolutionary-based multiobjective optimization techniques[J]. *Knowledge and Information System*, 1999, 11(3): 269—308.
- [8] 王凌. 车间调度及其遗传算法[M]. 北京: 清华大学出版社, 2003.
Wang Ling. *Shop Scheduling with Genetic Algorithms*[M]. Beijing: Tsinghua University Press, 2003. (in Chinese)
- [9] 周国华. 生产作业调度问题的软计算方法研究[D]. 成都: 西南交通大学, 2003.
Zhou Guohua. Study on Soft Computing of Production Scheduling Problem[D]. Chengdu: Xi nan Jiaotong University, 2003. (in Chinese)

- [10] Brizuela C, Sannomiyana N, Zhao Y. Multi-objective Flow-shop: Preliminary Results[C]. In the Proceedings of 1st International Conference (EMO2001), Zurich, Switzerland, March 7~9, 2001. 443—457.
- [11] Brizuela C, Aceves R. Experimental genetic operators analysis for the multi-objective permutation flowshop[A]. Proceedings of 2nd International Conference (EMO2003) [C], Faro, Portugal, April 8~11, 2003. 578—592.
- [12] Murata T, Ishibuchi H. Performance of Multi-Objective Genetic Algorithms for Flowshop Scheduling Problems[C]. In the Proceedings of 14th International Conference on Production Research, Osaka, Japan, August, 1997. 498—501.
- [13] Bagchi T P. Multiobjective Scheduling by Genetic Algorithms[M]. Boston: Kluwer Academic Publisher, 1999.
- [14] Basseur M, Seynhaeve F, Talbi E G. Design of Multi-Objective Evolutionary Algorithms: Application to the Flow-shop [C]. In the Proceedings of Congress on Evolutionary Computation (CEC 2002), Piscataway, New Jersey, May, 2002, 2: 1151—1156.
- [15] Amentano V A, Claudio J E. An application of a multi-objective tabu search algorithm to a bicriteria flowshop problem [J]. Journal of Heuristics, 2004, 10(5): 463—481.
- [16] Arroyo J E C, Amentano V A. Genetic local search for multi-objective flowshop scheduling problems[J]. European Journal of Operational Research, 2005, 167(3): 717—738.
- [17] 宋华明. 混合流水生产系统的多目标协同优化研究[D]. 南京: 南京理工大学, 2003.
Song Huaming. Research on the Multiobjective Cooperation Optimization of Mixed-Model Flow Manufacturing System[D]. Nanjing: Nanjing University of Science and Technology, 2003. (in Chinese)
- [18] Gen M, Cheng R W. Genetic Algorithms and Engineering Design[M]. New York: John Wiley & Sons, Inc., 1996.
- [19] Gen M, Cheng R W. Genetic Algorithms and Engineering Optimization[M]. New York: John Wiley & Sons, Inc., 2000.
- [20] Murata T. Genetic Algorithms for Multi-Objective Optimization[D]. Japan: Osaka Prefecture University, 1997.
- [21] Ceolb C A C, Pulido G T. A Micro-Genetic Algorithm for Multiobjective Optimization[C]. In the Proceedings of 1st International Conference on Evolutionary Multi-Criteria Optimization (EMO2001), Zurich, Switzerland, March 7~9, 2001. 126—140.
- [22] 师瑞峰, 周泓, 谭小卫. 递进多目标遗传算法[J]. 系统工程理论与实践, 2005, 25(12): 48—56.
Shi Ruifeng, Zhou Hong, Tan Xiaowei. A multi-objective genetic algorithm based on escalating strategy[J]. Systems Engineering - Theory & Practices, 2005, 25(12): 48—56. (in Chinese)
- [23] Hansen P, Nenal M. Variable neighborhood search: Principles and applications[J]. European Journal of Operational Research, 2001, 130: 449—467.
- [24] Nenal M, Hansen P. Variable neighborhood search[J]. Computers Operational Research, 1997, 24(11): 1097—1100.
- [25] Deb K. Multi-Objective Optimization using Evolutionary Algorithms[M]. New York: John Wiley & Sons, 2001.
- [26] Baker K R. Sequencing rules and due-date assignments in job shop[J]. Management Science, 1984, 30(9): 1093—1104.
- [27] Sabuncuoglu I, Bayiz M. Job shop scheduling with beam search[J]. European Journal of Operational Research, 1999, (118): 390—412.
- [28] Pinedo M, Singer M. A shifting bottleneck heuristic for minimizing the total weighted tardiness in a job shop[J]. Naval Research Logistics, 1999, 46(1): 1—17.
- [29] Deb K, Pratap A, Agarwal S, et al. A fast and elitist multiobjective genetic algorithm: NSGA-II[J]. Evolutionary Computation, IEEE Transactions, 2002, 6(2): 182—197.
- [30] Zitzler E. Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications[D]. Switzerland: Swiss Federal Institute of Technology, 1999.

(下转第 27页)

Optimal model of technology adoption time with learning effect and its simulation

DAI Hong-kun, XU Jiu-ping

School of Business and Administration, Sichuan University, Chengdu 610064, China

Abstract: This paper focuses on the problem of enterprises technology adoption time decision. Constructs an optimal model of enterprise technology adoption time with learning effect, considering influence of enterprise inner learning effect and outer technology evolution on the adoption time decision, mainly describing the learning effect. Gives the decision-making rules of enterprise optimal adoption time. Analyses the relations between the optimal technology adoption time and fixed adoption cost, market volume, technology progress rate and needs elasticity. Model simulation tests the validity of the model and the rationality of adopting decision rules.

Key words: technology adoption; optimal model; learning effect; model simulation; technology strategy.

(上接第 20 页)

Escalating evolutionary algorithm with application to bi-objective flow shop scheduling problems

SHI Rui-feng^{1,2}, ZHOU Hong²

1. School of Computer Science and Engineering, Beihang University, Beijing 100083, China;

2. School of Economics and Management, Beihang University, Beijing 100083, China

Abstract: An escalating multi-objective evolutionary algorithm (EMEA), which aims at solving bi-objective flow shop scheduling problem, is proposed in this paper. The new algorithm takes a new elite duplication strategy and an innovative escalating evolutionary structure, which improved the convergence and efficiency of the algorithm and reduced its computational cost. Besides, the proposed algorithm combines those meta-heuristic algorithms, which are adept at solving specific objective optimization with flow shop scheduling problems, into a tournament variable Pareto local search strategy at the end of each generation. 31 typical bi-objective flow shop case studies have been employed for demonstration. The optimization results have shown that, EMEA has gotten outstanding Pareto frontiers in all test problems by contrast to those of a well-known algorithm NSGA-II, which revealed its efficiency and effectiveness in solving bi-objective flow shop scheduling problems.

Key words: multi-objective evolutionary algorithm; escalating evolution; tournament neighborhood search; meta-heuristic local search; flow shop