

6-23-2021

Pathfinder Algorithm for Green Pipeline Scheduling with Limited Buffers

Hu Rong

Kunming University of Science and Technology, Faculty of Information Engineering and Automation, Kunming 650500, China;

Yuming Dong

Kunming University of Science and Technology, Faculty of Information Engineering and Automation, Kunming 650500, China;

Bin Qian

Kunming University of Science and Technology, Faculty of Information Engineering and Automation, Kunming 650500, China;

Follow this and additional works at: <https://dc-china-simulation.researchcommons.org/journal>



Part of the Artificial Intelligence and Robotics Commons, Computer Engineering Commons, Numerical Analysis and Scientific Computing Commons, Operations Research, Systems Engineering and Industrial Engineering Commons, and the Systems Science Commons

This Paper is brought to you for free and open access by Journal of System Simulation. It has been accepted for inclusion in Journal of System Simulation by an authorized editor of Journal of System Simulation.

Pathfinder Algorithm for Green Pipeline Scheduling with Limited Buffers

Abstract

Abstract: A Hybrid Pathfinder Algorithm (HPFA) is proposed for solving the green flow shop scheduling problem with limited buffers and energy threshold constraints (GFSSP_LBET). The optimization criteria are to minimize the total energy consumption and the makespan. *In order to enhance the global search ability of HPFA, a distance-based selection scheme is designed to determine each pathfinder's followers to ensure that the near regions of any pathfinder can get a certain search. A self-learning search strategy integrating multiple operations is designed to perform multi-neighborhood search on the updated pathfinders, which can improve the local exploitation ability of HPFA.* Simulation experiments show that the presented improvement strategies can effectively enhance the performance of HPFA and also verify the effectiveness and robustness of HPFA for dealing with the GFSSP_LBET.

Keywords

hybrid pathfinder algorithm, limited buffers, flow shop scheduling problem, total energy consumption, energy threshold constraints

Recommended Citation

Hu Rong, Dong Yuming, Qian Bin. Pathfinder Algorithm for Green Pipeline Scheduling with Limited Buffers[J]. Journal of System Simulation, 2021, 33(6): 1384-1396.

基于探路者算法的绿色有限缓冲区流水线调度

胡蓉, 董钰明, 钱斌

(昆明理工大学 信息工程与自动化学院, 云南 昆明 650500)

摘要: 针对带能耗阈值约束和有限缓冲区的绿色流水车间调度问题(Green Flow Shop Scheduling Problem with Limited Buffers and Energy Thresholds, GFSSP_LBET), 提出一种混合探路者算法(Hybrid Pathfinder Algorithm, HPFA)进行求解, 优化目标为总能量消耗和最大完工时间。为增强算法的全局搜索能力, 设计基于距离的选择机制确定每个探路者对应的跟随者, 以确保各探路者的附近区域均能得到一定的搜索; 设计融合多种邻域操作的自学习搜索策略, 用于对更新后的探路者进行多邻域搜索, 从而提升算法的局部开采能力。通过仿真实验表明所提改进措施能有效增强算法性能, 也验证了HPFA求解GFSSP_LBET的有效性和鲁棒性。

关键词: 混合探路者算法; 有限缓冲区; 流水车间调度问题; 总能量消耗; 能耗阈值约束

中图分类号: TP391.9

文献标志码: A

文章编号: 1004-731X (2021) 06-1384-13

DOI: 10.16182/j.issn1004731x.joss.20-0077

Pathfinder Algorithm for Green Pipeline Scheduling with Limited Buffers

Hu Rong, Dong Yuming, Qian Bin

(Kunming University of Science and Technology, Faculty of Information Engineering and Automation, Kunming 650500, China)

Abstract: A Hybrid Pathfinder Algorithm (HPFA) is proposed for solving the green flow shop scheduling problem with limited buffers and energy threshold constraints (GFSSP_LBET). The optimization criteria are to minimize the total energy consumption and the makespan. *In order to enhance the global search ability of HPFA, a distance-based selection scheme is designed to determine each pathfinder's followers to ensure that the near regions of any pathfinder can get a certain search. A self-learning search strategy integrating multiple operations is designed to perform multi-neighborhood search on the updated pathfinders, which can improve the local exploitation ability of HPFA.* Simulation experiments show that the presented improvement strategies can effectively enhance the performance of HPFA and also verify the effectiveness and robustness of HPFA for dealing with the GFSSP_LBET.

Keywords: hybrid pathfinder algorithm; limited buffers; flow shop scheduling problem; total energy consumption; energy threshold constraints

引言

随着智能制造的发展和推进, 各企业在追求生产设备智能化的同时也在深入研究制造过程的智能化, 智能算法在企业生产调度中应用已成为工厂自动化的智能决策引擎。特别是化工制药、半导体制造和钢铁业等行业“智慧工厂”战略的提出与建

设, 中间存储容量受限的生产调度再次成为学者们研究的热点, 即有限缓冲区流水车间调度问题(Flow Shop Scheduling Problem with Limited Buffer, FSSP_LB)。已有文献证明, 即使是仅有两台机器的FSSP_LB也是NP-hard问题^[1]。

目前, 各国家在获得经济高速增长后愈来愈重

收稿日期: 2020-02-18

修回日期: 2020-04-22

基金项目: 国家自然科学基金资助项目(61963022, 51665025)

第一作者: 胡蓉(1974-), 女, 博士, 副教授, 研究方向为智能优化调度、物流优化。E-mail: ronghu@vip.163.com

视环境保护和能源节制,考虑节能减排的绿色制造浪潮正在全球范围内兴起^[2]。近年来国际工程学会、期刊已发表了许多关于此方面的研究论文,主要分为以下 3 部分:①以生产能耗为优化目标的低碳调度研究^[3-5];②设置开、关机约束,缩短机器空转时间^[6-7];③及时调整机器转速,在不影响生产进度的前提下实现节能减排^[8]。但是,学术的研究与企业的实际生产仍存在着一定的差距,部分企业仍旧会追求经济利益而不采取真正的节能减排。所以,政府或企业在生产时设定阈值,如,污染阈值(允许企业在造成的污染不超过阈值的情况下正常生产)、损耗阈值(批次生产造成的机器损耗不能超过阈值),且两者都与机器的能量损耗有关。因此,针对上述情况,本文提出一种带能耗阈值约束和有限缓冲区的绿色流水车间问题(Green Flow Shop Scheduling Problem with Limited Buffers and Energy Thresholds, GFSSP_LBET),在满足每台机器的能耗阈值约束的同时考虑企业生产的经济指标和绿色指标,具有较强的理论依据和研究价值。

对于 FSSP_LB,于艳辉等^[9]将以 Makespan 为目标函数转化成以最小化机器空闲时间为目标函数,分析空闲时间与目标函数之间的关系,提出一种启发式规则求解。杜田田等^[10]以最小化最大完工时间为优化目标,引入基于 NEH 的初始化方法和基于 Pairwise 的邻域搜索,提出一种混合蝙蝠算法,避免了算法陷入局部最优,加强算法的搜索性能。徐震浩等^[11]针对缓冲区空间和时间同时受限的流水车间调度问题,以最小化完工时间为优化目标,提出一种改进的离散帝国竞争算法求解。Jiang S L 等^[12]在之前的基础上研究了面向能源的 FSSP_LB 问题。

对于带能耗阈值(Energy Threshold, ET)的绿色生产调度问题, Fang 等^[13]针对置换流水车间调度问题,以完工时间和峰值功耗为目标,利用数学规划方法提出组合多项式时间算法求解。Liu 等^[14]针对 seru 生产系统,以最小化碳排放总量和最大完工时间为优化目标,提出基于非支配排序的遗传算法

求解,并验证算法的有效性。Keisuke 等^[15]针对流水车间调度问题,以时间和峰值功耗为目标,研究最佳空闲时间(工件输入延迟时间)和最佳插入点,提出一种随机处理时间和峰值功率的生产调度模型。Kemmo 等^[16]针对可变能耗阈值的作业车间调度问题,以完工时间和能耗峰值为目标,引入生产能量消耗峰值和峰值过后动态耗能评价策略,并应用 CPLEX 12.4 求解器验证了模型的可行性。Kemmo 等^[17]在已有的研究基础上,以最小化最大完工时间为主要优化目标,进一步改进了局部搜索机制,提出一种贪婪随机自适应搜索算法,并通过实验对比验证了所提方法的求解结果优于当前已有的求解同类问题的算法。根据上述文献调研,目前尚无智能优化算法求解 GFSSP_LBET 的相关研究。

探路者算法(Pathfinder Algorithm, PFA)是 2019 年由 Yapici 提出的一种新的元启发式算法^[18]。该算法受群体动物的狩猎行为启发,将种群中的个体分为探路者(领导者)和跟随者;算法的寻优过程模拟了种群寻找食物的探索过程,通过探路者、跟随者二种不同种群角色间的交流来实现优化,同时在进化过程中也加强了种群三代间优良信息的保留;PFA 具有容易理解、实现简单、优化性能较好等优点,其搜索性能也优于目前已在组合优化问题有较多应用的粒子群算法和教与学算法^[18]。

标准 PFA 求解多目标问题时将 Pareto 前沿中离散度最大的个体选为探路者,但多目标问题的最优解(非受支配解)往往不止一个,仅有一个探路者会限制算法的搜索广度。因此,本文将当代种群中所有的非受支配解都作为算法寻优的探路者,并将“首次改进跳出”思想用于探路者位置的更新,使探路者在邻域内进行多次探索;同时,跟随者的更新方式中也加入探路者选择机制,跟随者可以根据自身位置选择距离自己较近的探路者跟随;此外,跟随者在更新过程中可能存在“掉队”的情况,将采用随机生成的方式替换掉队的跟随者,从而丰富种群的多样性。最后,通过仿真实验验证所提改进措施

能够有效地增强算法的搜索性能。

本文建立以最大完工时间(C_{\max})和总能量消耗(TEC)为优化目标的 GFSSP_LBET 问题模型,并提出一种混合探路者算法(Hybrid Pathfinder Algorithm, HPFA)进行求解。在 HPFA 中,首先采用基于随机键编码的 LOV(Largest-order-value)规则将 PFA 的位置信息(即实数向量)离散化,使算法可对问题的离散解空间进行搜索;其次,在标准 PFA 的进化机制上加入了探路者选择机制,跟随者根据所处区域选择不同的探路者;然后,将算法的全局搜索与具备自我学习能力的局部搜索策略相结合,增加算法的局部探索能力;最后,通过仿真实验和算法比较,验证了所提 HPFA 的有效性。

1 GFSSP_LBET 问题描述

本节将对带能耗阈值约束且优化目标为最大完工时间(C_{\max})和总能量消耗(TEC)的 GFSSP_LBET 进行介绍。

1.1 FSSP_LB 的问题描述

FSSP_LB 通常可以描述为: n 个工件 $J = \{1, 2, \dots, n\}$ 在 m 台机器 $M = \{1, 2, \dots, m\}$ 上的加工,并存在以下约束:①所有工件在 m 台机器上加工顺序不变,且每个操作的加工时间已知;②在任一时刻,每个工件至多在一台机器上进行加工,且每台机器至多加工一个工件;③在每台机器上,所有工件在该机器上的加工次序相同;④工件完成当前操作时但下一操作所在机器处于加工状态,则该工件需要进入缓冲区等待,若缓冲区已满,则被阻塞在当前机器上直到缓冲区有位置空出为止。此外,约定所有工件在缓冲区中均服从先入先出(First In First Out, FIFO)规则。

1.2 GFSSP_LB 的能耗模型

在企业的生产过程中,通常将机器分为加工状态(Processing)和待机状态(Standby)。由于机器设备的启停瞬间能量消耗较大,且在实际生产中具有不

停机的特点,因此本文不考虑开关机能耗。此外,由于 FSSP_LB 模型中存在阻塞状态,此时工件未被加工,所以工件的阻塞状态也属于待机状态。令 TEC^P 表示加工状态下的机器能耗, TEC^S 表示待机状态下的机器能耗,因此, FSSP_LB 整个过程的总能量消耗可以描述为

$$TEC = TEC^P + TEC^S \quad (1)$$

根据 1.1 节中 FSSP_LB 的问题描述,以最大完工时间(C_{\max})和总能量消耗(TEC)为优化目标的 GFSSP_LBET 数学模型可以通过式(2)~(9)来描述。其中, $\pi = \{\pi(1), \pi(2), \dots, \pi(n)\}$ 表示工件的加工排列, B_j 代表机器 j 和机器 $j+1$ 之间缓冲区的大小, $D_{\pi(i),j}$ 表示工件 $\pi(i)$ 在机器 j 上的脱机时间, $P_{\pi(i),j}$ 表示工件 $\pi(i)$ 在机器 j 上的加工时间, C_{\max} 表示该加工排列 π 的最大完工时间。此外,机器的加工状态也有转速的区别,机器转速的大小通过档位(转速)来设定,令 $V_{\pi(i),j}$ 表示工件 $\pi(i)$ 在机器 j 上的加工档位且 $V_{\pi(i),j} \in S, S = \{V_1, V_2, \dots, V_s\}$, 工件的实际加工时间与机器的档位成反比。

$$D_{\pi(1),1} = P_{\pi(1),1} / V_{\pi(1),1} \quad (2)$$

$$D_{\pi(1),j} = D_{\pi(1),j-1} + P_{\pi(1),j} / V_{\pi(1),j}, j = 2, 3, \dots, m \quad (3)$$

$$D_{\pi(i),1} = D_{\pi(i-1),1} + P_{\pi(i),1} / V_{\pi(i),1}, \\ i = 2, 3, \dots, B_1 + 1 \quad (4)$$

$$D_{\pi(i),j} = \max\{D_{\pi(i-1),j}, D_{\pi(i),j-1}\} + P_{\pi(i),j} / V_{\pi(i),j}, \\ i = 2, 3, \dots, B_j + 1, j = 2, 3, \dots, m - 1 \quad (5)$$

$$D_{\pi(i),1} = \max\{D_{\pi(i-1),1} + P_{\pi(i),1} / V_{\pi(i),1}, D_{\pi(i-B_1-1),2}\}, \\ i > B_1 + 1 \quad (6)$$

$$D_{\pi(i),j} = \max\{D_{\pi(i-1),j} + P_{\pi(i),j} / V_{\pi(i),j}, \\ D_{\pi(i),j-1} + P_{\pi(i),j} / V_{\pi(i),j}, D_{\pi(i-B_j-1),j+1}\}, \\ i > B_j + 1, j = 2, 3, \dots, m - 1 \quad (7)$$

$$D_{\pi(i),m} = \max\{D_{\pi(i-1),m}, D_{\pi(i),m-1}\} + P_{\pi(i),m} / V_{\pi(i),m}, \\ i = 2, 3, \dots, n \quad (8)$$

$$C_{\max} = D_{\pi(n),m} \quad (9)$$

虽然机器档位的提高会缩短相应操作的加工时间,但同时也会增加对应加工操作的能量消耗^[19],即:

$$\forall V_p > V_q \quad p, q \in \{1, 2, \dots, s\} \quad (10)$$

$$P_{\pi(i),j} / V_p < P_{\pi(i),j} / V_q \quad (11)$$

$$PP_{j,V_p} \cdot P_{\pi(i),j} / V_p > PP_{j,V_q} \cdot P_{\pi(i),j} / V_q \quad (12)$$

其中 PP_{j,V_p} 表示机器 j 加工档位为 V_p 时单位时间内的能量消耗。为了更好地描述该性质, 如图 1 所示, 为 3 台机器、3 个工件实例所对应的实时能量消耗情况; 在模型中加入档位选择, 如式(14)所示, 机器的档位 $S = \{1, 1.5, 2\}$, 则工件的加工时间由 $P_{\pi(i),j}$ 变为 $P'_{\pi(i),j}$, 此时该实例所对应实时能量消耗图如图 2 所示。

$$P_{\pi(i),j} = \begin{bmatrix} 2 & 4 & 1 \\ 1 & 3 & 3 \\ 2 & 1 & 1 \end{bmatrix} \quad (13)$$

$$V_{\pi(i),j} = \begin{bmatrix} 1 & 2 & 1 \\ 1 & 1.5 & 1.5 \\ 2 & 1 & 1 \end{bmatrix} \quad (14)$$

$$P'_{\pi(i),j} = \begin{bmatrix} 2 & 2 & 1 \\ 1 & 2 & 2 \\ 1 & 1 & 1 \end{bmatrix} \quad (15)$$

由图 2 中可以看出, $0 \sim t_1$ 时段只有机器 M_1 处于加工状态且档位(转速)不变, 此时的能量消耗未发生变化; $t_1 \sim t_2$ 时段机器 M_1, M_2 为加工状态且 M_2 为加速状态(2 档), 此时能量消耗大于原有的状态; $t_2 \sim t_3$ 时段机器 M_1, M_2 均为加速状态, 此时机器的能量消耗远高于未加速时的能量消耗。

令 $X_{j,V}(t)$ 表示机器 j 在时刻 t 以档位 V 运行, $Y_j(t)$ 表示机器 j 在时刻 t 为待机状态。 $X_{j,V}(t)$ 和 $Y_j(t)$ 可以表示为

$$X_{j,V}(t) = \begin{cases} 1, & \text{机器 } j \text{ 在时刻 } t \text{ 以档位 } V \text{ 运行} \\ 0, & \text{其他} \end{cases} \quad (16)$$

$$Y_j(t) = \begin{cases} 1, & \text{机器 } j \text{ 在时刻 } t \text{ 为待机状态} \\ 0, & \text{其他} \end{cases} \quad (17)$$

综上可得, GFSSP_LB 在整个生产过程中的总能量消耗仍可描述为式(18)所示, 式中, SP_j 为机器处于待机时单位时间内的能量消耗。

$$TEC = \int_0^{C_{\max}} \left(\sum_{j=1}^m \left(\sum_{V=1}^s PP_{j,V} \cdot X_{j,V}(t) + SP_j \cdot Y_j(t) \right) \right) dt \quad (18)$$

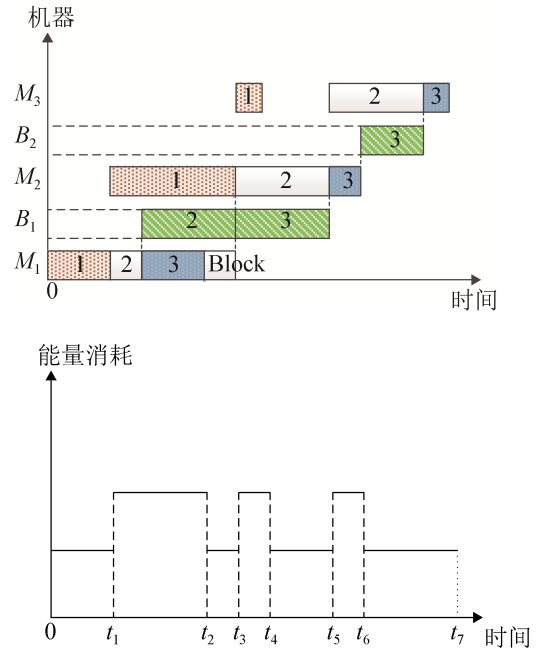


图 1 GFSSP_LB 实时能量消耗图
Fig. 1 Real-time energy consumption graph of GFSSP_LBET

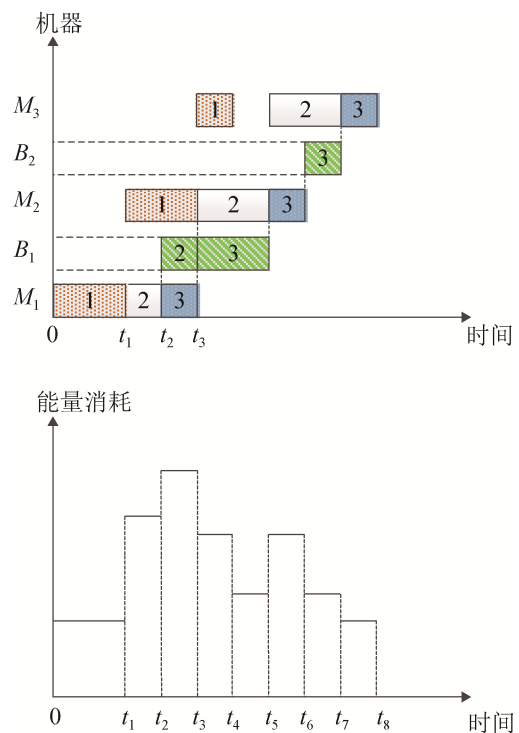


图 2 带档位选择的 GFSSP_LB 实时能量消耗图
Fig. 2 Real-time energy consumption graph of GFSSP_LBET with gear selection

此外, 在 GFSSP_LB 的求解过程中应满足以下约束条件:

$$\int_0^{C_{\max}} \left(\sum_{j=1}^S PP_{j,V} \cdot X_{j,V}(t) + SP_j \cdot Y_j(t) \right) dt \leq Q_k \quad (19)$$

约束中 Q_k 为机器的能量约束，即整个生产过程中机器 M_k 的总能量消耗不能超过该阈值，且由于机器的功能不同，所以各机器能耗阈值也有所不同。

1.3 GFSSP_LBET 的优化目标

综上所述，本文提出 GFSSP_LBET 调度模型的优化目标如下：

$$(1) \text{ 最小化最大完工时间 } C_{\max} \\ f_1 = \min C_{\max}(\pi) \quad (20)$$

$$(2) \text{ 最小化总能量消耗 } TEC \\ f_2 = \min TEC(\pi) \quad (21)$$

求解的目标就是在所有排列的集合 Ω 中，寻找在满足阈值约束的前提下最小化最大完工时间 C_{\max} 和总能量消耗 TEC 的解，即：

$$\text{Min } f_1(\pi), f_2(\pi), \pi \in \Omega \quad (22)$$

从图 1 和图 2 实时能量消耗的对比可以发现，机器转速的提高虽使工件的加工时间缩短，但也会导致该时间段的能量消耗剧增。因此，优化目标最小化 C_{\max} 和最小化 TEC 之间存在着一定的冲突和制约，同时所求的排列应满足阈值约束，否则为不可行解。

2 HFPA 求解 GFSSP_LBET

本节将对 HFPA 的编码与解码、种群初始化、种群的更新、局部搜索进行介绍。

2.1 编码与解码

标准 PFA 中个体的位置向量为连续域上的实数，因而标准的 PFA 不能直接应用于 GFSSP_LBET 的求解。因此，需要将实数的位置向量映射为工件的排列，本文利用基于随机键编码的 LOV (Largest-order-value) 规则，将探路者个体的位置向量 $X = [x_1, x_2, \dots, x_n]$ 成功转化为工件的排列 $\pi = [\pi_1, \pi_2, \dots, \pi_n]$ 。

通过调用 LOV 规则可将种群个体的位置信息

转化为 GFSSP_LBET 模型中的加工排序，实现实数向量与工件排列的一一对应。同时，当工件的排列发生变化时，也可以通过 RLOV (Reverse Largest-Order-Value) 调整个体的位置信息。

2.2 标准 PFA 的位置更新

探路者算法中每个个体的位置代表 GFSSP_LBET 问题的一个解，种群位置的更新即是算法种群的更新，是算法迭代搜索的核心机制。合适的位置更新机制可以加快算法的收敛速度，有效提高算法的性能，对算法的寻优有着至关重要的作用。

在探路者算法中，种群中的个体分为探路者 (Pathfinder) 和跟随者 (Follower Members)，它们共同组成种群团队。探路者个体是团队的领导者，指引算法的全局搜索方向，而种群中的跟随者沿着探路者的方向进行移动。种群中个体的移动使其位置向量发生变化，其对应的工件排列也随之改变，即种群的更新。

在种群的更新过程中，探路者是该种群运动方向的探索者，先于跟随者移动，其更新方式如式 (23)，(24) 所示。

$$X_p^{K+1} = X_p^K + 2r_1 \cdot (X_p^K - X_p^{K-1}) + A \quad (23)$$

$$A = u_1 \cdot e^{\frac{-2K}{K_{\max}}} \quad (24)$$

式中： K 表示算法当前的迭代代数， K_{\max} 为算法的最大迭代数； X_p^K 表示当代探路者的位置， X_p^{K-1} 表示上一代探路者的位置，则 X_p^{K+1} 表示探路者更新后的位置； r_1 为探路者移动的步长因子，在范围 [0,1] 内服从均匀分布； A 表示探路者移动的多向性和随机性：多向性由 u_1 的取值决定， u_1 为 [-1,1] 范围内的随机数；步长大小的随机性由 $-2K/K_{\max}$ 决定，即与算法迭代的次数有关。

此外，由于探索过程存在多向性和随机性，探路者探索到的位置可能不如原位置。因此，在探路者完成更新后加入保优操作，即更新后位置不如原位置，则探路者回到原位置。

探路者更新完成后，种群中跟随者根据探路者

位置进行更新:

$$X_i^{K+1} = X_i^K + R_1 \cdot (X_j^K - X_i^K) + R_2 \cdot (X_p^K - X_i^K) + \varepsilon, i \geq 2 \quad (25)$$

$$R_1 = \alpha \cdot r_2 \quad (26)$$

$$R_2 = \beta \cdot r_3 \quad (27)$$

$$\varepsilon = \left(1 - \frac{K}{K_{\max}}\right) \cdot u_2 \cdot D_{ij} \quad (28)$$

$$D_{ij} = \|X_i - X_j\| \quad (29)$$

式中: X_i^K 为跟随者的当前位置, 则 X_i^{K+1} 为其更新后的位置; 跟随者的移动不仅与探路者 X_p^K 有关, 而且受到其他跟随者 X_j^K 的影响; α 为跟随者之间的相互作用系数, β 为探路者对跟随者的吸引系数, 均在 $[1, 2]$ 服从均匀分布; r_2, r_3 分别为与其他跟随者和探路者移动的步长因子, 均为 $[0, 1]$ 范围内的随机数; 同理, 与探路者一样, ε 为跟随者移动的随机性; u_1, D_{ij} 决定随机运动的方向和步长; D_{ij} 为当前跟随者与其他跟随者之间的距离。

由上述可知, 无论是探路者还是跟随者, 其随机步长的大小都与算法迭代的次数有关, 且代数越大步长越小。这是由于算法搜索初期需要在庞大的解空间中探索优质解区域, 而在后期对优质解区域进行更细致的搜索, 该操作有效地平衡了 PFA 全局搜索与局部搜索的关系。

2.3 改进 PFA 的位置更新

GFSSP_LBET 为多目标优化问题, 该问题不存在单独的最优解(近似最优解), 其最优解(近似最优解)由多个非受支配解组成并分散于解空间的不同区域, 显然, 算法在求解该类问题时需尽量进行分散搜索。现有的探路者算法在求解多目标问题时, 大多只采用非受支配解集中分散度最大的非受支配解作为探路者^[18]。这实际上难以实现较广范围的搜索, 从而限制了算法的性能。因此, HPFA 将算法进化过程中的当代非受支配解集的每个个体均作为探路者执行搜索, 以驱动算法到达较广的解空间区域。同时, 为确保各探路者的附近区域均能得到一定的搜索, 在 HPFA 中采用一种基于距离的

选择机制来确定各探路者的跟随者, 进而对这些跟随者进行更新。

$$\|X_i - X_p\| \leq d \quad (30)$$

如图 3 所示, 非受支配解集中有 3 个解, 它们均为种群的探路者。根据式(30), 以探路者的位置为圆心划分区域, 跟随者根据自己所处区域选择相应的探路者进行跟随, 如 1, 3, 8 号跟随者选择探路者 1; 若跟随者所处位置位于两个甚至多个区域的重叠处, 则会判断该个体与 3 个探路者之间的距离, 优先选择与自己较近的探路者, 如 10 号探路者离探路者 2 更近所以跟随探路者 2 移动; 此外, 搜索过程中也会出现“掉队”情况, 如 5, 7 号探路者均不在任何探路者的影响范围内, 此时将抛弃该部分跟随者, 重新随机生成新个体加入到种群中, 在保证种群质量的同时丰富个体的多样性。

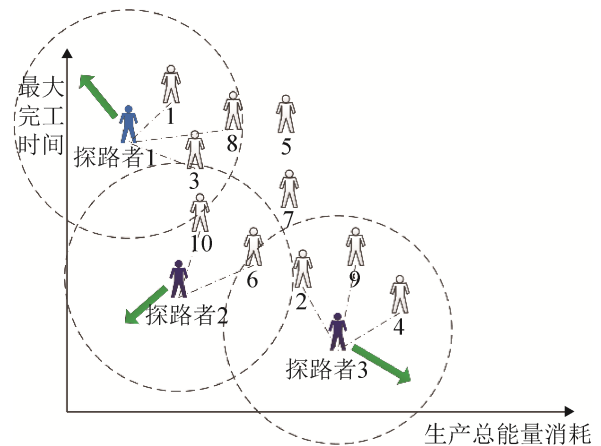


图 3 基于距离的探路者选择示意图

Fig. 3 Schematic of distance-based pathfinder selection

2.4 局部搜索

本文将 Insert, Swap, Inverse 3 种局部搜索操作进行组合, 形成具有自我学习能力的局部搜索策略, 分为训练阶段和工作阶段, 其具体步骤为:

训练阶段: 令 P_Insert , P_Swap , $P_Inverse$ 分别代表 Insert 操作、Swap 操作、Inverse 操作的选择概率。此外, N_Insert , N_Swap , $N_Inverse$ 分别代表其有效操作的次数。

step 1: 随机从种群中选择某个个体 X ，通过 LOV 规则将其转换为排序 π 。

step 2: 令 $Loop=1$, $N_Insert=1$, $N_Swap=1$, $N_Inverse=1$;

step 3: 随机选择位置 u 和 v ，对 π 执行 *Insert* 操作得到 π' ，若 $\pi' < \pi$ ，则 $N_Insert = N_Insert + 1$ 且 $\pi = \pi'$;

step 4: 随机选择位置 u 和 v ，对 π 执行 *Swap* 操作得到 π' ，若 $\pi' < \pi$ ，则 $N_Swap = N_Swap + 1$ 且 $\pi = \pi'$;

step 5: 随机选择位置 u 和 v ，对 π 执行 *Inverse* 操作得到 π' ，若 $\pi' < \pi$ ，则 $N_Inverse = N_Inverse + 1$ 且 $\pi = \pi'$;

step 6: 令 $Loop=Loop+1$ ，若 $Loop \leq n(m-1)$ ，则跳转至 step3;

step 7: 计算各操作的选择概率

$$P_Insert = N_Insert / (N_Insert + N_Swap + N_Inverse)$$

$$P_Swap = N_Swap / (N_Insert + N_Swap + N_Inverse)$$

$$P_Inverse = N_Inverse / (N_Insert + N_Swap + N_Inverse)$$

$$P_Inverse = N_Inverse / (N_Insert + N_Swap + N_Inverse)$$

工作阶段: 位置更新后的探路者无论是否优于原探路者，都对其执行如下的多邻域搜索，从而对探路者所发现的区域进行较深入的局部开采。这有利于提高算法获取优质解的能力。

step 1: 探路者位置更新后，将其位置向量 X 通过 LOV 规则将其转换为排序 π ，令 $Loop=1$;

step 2: 根据各操作的概率进行轮盘赌选择对 π 执行的操作;

step 3: 若选中 *Insert* 操作，则执行后得到 π' ，若 $\pi' < \pi$ ，则 $N_Insert = N_Insert + 1$ 且 $\pi = \pi'$;

若选中的是 *Swap* 操作，则执行后得到 π' ，若 $\pi' < \pi$ ，则 $N_Swap = N_Swap + 1$ 且 $\pi = \pi'$;

若选中的是 *Inverse* 操作，则执行后得到 π' ，若 $\pi' < \pi$ ，则 $N_Inverse = N_Inverse + 1$ 且 $\pi = \pi'$;

step 4: 对各操作的概率值进行更新

$$P_Insert = N_Insert / (N_Insert + N_Swap + N_Inverse)$$

$$P_Swap = N_Swap / (N_Insert + N_Swap + N_Inverse)$$

$$P_Inverse = N_Inverse / (N_Insert + N_Swap + N_Inverse)$$

step 5: 令 $Loop=Loop+1$ ，若 $Loop \leq n$ ，则跳转至 step 2;

step 6: 探路者的排序发生变化时，将其排序 π' 通过 RLOV 规则更新探路者的位置得到位置向量 X' 。

3 仿真实验与对比分析

本文仿真实验中所有的算法和测试均由 Delphi 2010 编程实现，并在操作系统为 Window10、CPU 为 2.2 GHz、内存为 8 GB 的电脑上进行实验。实验数据均来自国际上流水车间调度问题的标准测试数据集，共采用 6 组不同规模、18 种问题类型的数据来测试各算法的性能。

本文的机器速度档位设定为 $S = \{1, 1.1, 1.2, 1.3, 1.4\}$ ， $PP_{j,V_p} = 4V_p^2$ ， $SP_j = 1$ ^[20]。

3.1 评价指标

本文将采用 3 种评价指标算法的性能进行评价，具体评价依据如下:

(1) \bar{N} 指标

对非受支配解集进行筛选，统计剔除被其他算法的非受支配解支配后的非受支配解个数^[21]。

$$N(S_j) = |S_j - \{x \in S_j | \exists y \in S_{sum} : y < x\}| \quad (31)$$

式(31)中 S_{sum} 表示所有算法找到的非受支配解集合， S_j 表示某个算法的非受支配解集，若 S_j 中可能存在被其他算法的非受支配解支配的解，则该解应被剔除。一般认为能寻到非受支配解数目较多且不被其他算法支配的算法性能较好。

(2) R 指标

R 指标评价的是各算法非受支配解集的占优情况^[22]。

$$R(S_j) = |S_j - \{x \in S_j | \exists y \in S_{sum} : y < x\}| / |S_j| \quad (32)$$

式(32)中 $R(S_j)$ 表示 S_j 的非受支配解集不被其他算法的非受支配解集支配占其总数的比例, 其值介于 0~1 之间, 该值越大则反映该算法寻到的解的质量和算法的收敛性越好。

(2) IGD 指标

计算各算法非受支配解集与最优解集的逼近程度, 该指标可反映算法所得非受支配解在目标空间的分布情况^[23]。

$$IGD(S_j) = \sum_{i=1}^{|S_j|} \text{Min } d(x, S^*) / |S_j|, x \in S_j \quad (33)$$

式(33)中 $\text{Min } d(x, S^*)$ 表示算法非受支配解集 S_j 解 x 到最优解集 S^* 中最近点的欧式距离。由于问题的解空间非常庞大, 无法获得真实的最优解集, 因此本文将 3 种算法分别运行 3 倍对比时间后得到的非受支配解集合并, 并剔除被支配的解作为最优解集 S^* 。显然, IGD 越小代表该算法的逼近性越好, 非受支配解集分布越均匀。

3.2 改进 PFA 与标准 PFA 的对比

为了验证所提改进位置更新操作的有效性, 采用 Buffer=2 的测试集对其与标准的 PFA 进行对比。算法对比中各算法的种群规模、算法参数设置均相同, 且均未加入其他局部搜索操作。由于比较的是改进后算法的性能, 因此该对比中未加入阈值约束, 对比结果采用 \bar{N} 指标和 R 指标进行评价, 这样更能直观的看出两个算法的占优情况。

表 1 为 2 种算法分别运行 20 代、50 代和 100 代且各算法独立运行 20 次后 \bar{N} 指标和 R 指标平均值。从表 1 中可以看出, 无论是 \bar{N} 指标还是 R 指标, 改进 PFA 在与标准 PFA 的对比中占据绝对优势, 在绝大多数的规模两种评价指标均优于标准 PFA。此外, 根据结果分析可以看出, 随着迭代次数的增加, 改进 PFA 的算法性能会得到进一步提升, 几乎不存在标准 PFA 占优的情况。因此, 本文所提的改进位置更新操作在进行多目标问题求解时, 能够有效地提升 PFA 的性能。

3.3 HPFA 与其他多目标算法的对比

为了验证所提 HPFA 求解 GFSSP_LBET 的有效性, 本文将其与其他著名的多目标算法 IMMOGLS^[23] 和 SFLA^[20] 进行对比。其中 IMMOGLS 采用随机权重的方法将多目标问题转化为单目标问题求解, 其在车间调度问题方面的求解性能优于经典的多目标算法 SPEA 和 NSGAI。此外, SFLA 由雷等提出并用于求解绿色柔性作业车间调度问题的多目标算法, 文献[20]中已证明其算法具有很强的搜索性能。

智能算法的优化效果与其参数选择密切相关, 因此本文采用实验设计方法(Design of Experiment, DOE)进行实验分析。针对本文所提的 HPFA, 对算法性能影响较大的分别是种群的规模 $Popsiz$ 、区域半径 d 和局部搜索次数 \max_L 。实验结果表明本文所提算法在求解 GFSSP_LBET 时的最佳参数组合为种群的规模 $Popsiz=60$ 、区域半径 $d=3\ 000$ 和局部搜索次数 $\max_L=5n$ 。

考虑到各对比算法求解的问题模型不同, 其相应的算法最优参数组合也会有所不同, 为了显示 3 种算法对比的公平性, 本文也对 IMMOGLS 和 SFLA 采用相同的实验方法进行参数整定。IMMOGLS 的最佳参数组合为种群的规模 $P_Ga=60$ 、交叉概率 $P_c=0.8$ 、变异概率 $P_m=0.1$ 以及局部搜索概率 $P_{LS}=0.3$ 。SFLA 在求解 GFSSP_LBET 时的最佳参数组合为种群的规模 $N=40$ 、分组数 $s=5$ 以及局部搜索次数 $\mu=120$ 。此外, 由于阈值限制为虚拟值, 本文的阈值标准设置为: 先求得 3 种算法运行 5 倍对比时间后所有非受支配解的集合, 然后取所有非受支配解对应机器能耗平均值的 1.1 倍为本文的阈值标准。

表 2, 3 为各算法分别运行 20 次后输出的非受支配解集的 R , IGD 性能指标的平均值, 且对两个指标分开进行比较, 其中 GLS 代表随机权重遗传算法 IMMOGLS, 表 2 中最后一列 AVG_T 代表该算法求解对应规模问题的实际平均运行时间。图 4 为 3 种算法分别运行 20 次所有非受支配解集(剔除

被支配的解)的对比图。以上实验中每种算法的独立运行时间都为 100 nm ms。

从表 2 可以看出,本文所提的 HPFA 在对比结果中占优明显,HPFA 的 R 指标值为 1 的情况出现较多,此时表示该算法输出的非受支配解集不被其他算法的支配,另外两个比较算法均未出现为 1 的情况;SFLA 的 R 指标数值均为 0~1 之间,此情况代表输出的非受支配解集部分被其他算法支配;而 IMMOGLS 在缓冲区较小时其值为 0 或几乎为 0,此时代表其非受支配解均被其他算法的解支配。从表中数据还可看出,随着缓冲区大小增加对 IMMOGLS 影响较大,在缓冲区为 4 时,其性能明显有所提升,但仍不如其他两种算法的求解性能。

虽然 SFLA 的 R 指标在部分问题中与 HPFA 相近甚至略优,但在绝大多数的问题类型中,HPFA 的非受支配解集的质量占到绝对优势。

从表 3 的 IGD 指标可以看出,在大部分问题类型中 HPFA 的非受支配解集与对应问题的最优解集的逼近性最好,均明显优于 SFLA 和 IMMOGLS,而在大多数情况下 SFLA 大都明显优于 IMMOGLS。在少部分问题类型中,SFLA 的 IGD 指标虽小于 HPFA,但两者差值较小,甚至几乎相等。从 IGD 指标的数值变化中也可以发现,IMMOGLS 的算法性能会随着缓冲区的增大而有所提高,而 HPFA 和 SFLA 的搜索性能受缓冲区变化的影响较小。

表 1 改进 PFA 与标准 PFA 的对比结果
Tab. 1 Improved PFA comparison with standard PFA

		运行代数											
问题	规模	20 代				50 代				100 代			
		改进 PFA		标准 PFA		改进 PFA		标准 PFA		改进 PFA		标准 PFA	
		\bar{N}	R	\bar{N}	R	\bar{N}	R	\bar{N}	R	\bar{N}	R	\bar{N}	R
rec01	20_5	3.60	0.64	2.75	0.59	3.65	0.67	2.50	0.44	4.75	0.69	3.10	0.37
rec03	20_5	4.80	0.74	3.10	0.49	5.50	0.74	2.45	0.44	4.20	0.54	3.50	0.58
rec05	20_5	3.85	0.68	1.90	0.42	3.55	0.64	2.70	0.47	3.85	0.57	2.75	0.50
rec07	20_10	2.55	0.70	2.40	0.51	2.50	0.74	1.95	0.50	3.00	0.65	2.40	0.51
rec09	20_10	2.75	0.80	1.65	0.43	2.90	0.66	2.30	0.55	2.75	0.63	2.25	0.48
rec11	20_10	2.10	0.55	2.65	0.67	2.35	0.54	2.35	0.61	3.45	0.76	2.05	0.44
rec13	20_15	1.85	0.66	1.80	0.73	1.55	0.62	2.00	0.61	2.30	0.68	2.10	0.64
rec15	20_15	1.85	0.70	1.80	0.58	2.15	0.62	1.75	0.59	2.60	0.67	2.30	0.56
rec17	20_15	2.25	0.62	2.20	0.71	2.60	0.63	2.20	0.59	2.95	0.72	2.35	0.59
rec19	30_10	3.50	0.69	2.00	0.51	3.15	0.68	1.95	0.38	3.15	0.61	2.70	0.51
rec21	30_10	2.30	0.60	2.35	0.58	2.95	0.65	2.10	0.46	2.45	0.71	2.15	0.50
rec23	30_10	3.10	0.75	1.75	0.47	3.40	0.84	2.05	0.40	3.10	0.59	3.15	0.62
rec25	30_15	1.90	0.62	1.70	0.54	2.50	0.67	2.15	0.51	2.30	0.62	2.00	0.47
rec27	30_15	2.10	0.67	1.40	0.44	1.85	0.59	1.85	0.57	2.30	0.71	1.50	0.45
rec29	30_15	2.20	0.63	1.80	0.63	2.65	0.70	1.50	0.45	2.90	0.72	1.80	0.53
rec31	50_10	3.15	0.61	2.70	0.63	2.15	0.62	2.55	0.52	3.70	0.63	3.20	0.60
rec33	50_10	2.25	0.56	2.40	0.69	2.80	0.61	2.00	0.56	3.85	0.77	2.05	0.40
rec35	50_10	3.20	0.73	1.95	0.51	2.95	0.70	2.05	0.39	3.05	0.69	2.10	0.45

表 2 HPFA 与 SFLA、IMMOGLS 在不同缓冲区下的 R 指标
Tab. 2 R indicators in different buffers of SFLA、IMMOGLS and HPFA

Problem	n,m	缓冲区大小												AVG_T(ms)		
		Buffer=0			Buffer=1			Buffer=2			Buffer=4					
		HPFA	SFLA	GLS	HPFA	SFLA	GLS	HPFA	SFLA	GLS	HPFA	SFLA	GLS	HPFA	SFLA	GLS
rec01	20,5	0.95	0.20	0.00	0.78	0.37	0.05	0.94	0.20	0.04	0.88	0.49	0.20			
rec03	20,5	0.86	0.27	0.01	0.92	0.22	0.01	0.95	0.12	0.03	0.93	0.41	0.07	10 193	10 449	10 606
rec05	20,5	0.90	0.17	0.00	0.90	0.36	0.01	0.93	0.25	0.02	0.90	0.27	0.39			
rec07	20,10	1.00	0.14	0.00	1.00	0.15	0.00	1.00	0.22	0.00	1.00	0.30	0.29			
rec09	20,10	0.96	0.21	0.02	0.90	0.31	0.00	1.00	0.22	0.03	0.90	0.57	0.13	20 016	20 037	20 156
rec11	20,10	0.93	0.33	0.00	0.62	0.49	0.00	0.90	0.37	0.05	0.93	0.38	0.27			
rec13	20,15	0.62	0.65	0.00	0.89	0.56	0.00	0.70	0.64	0.00	0.53	0.75	0.06			
rec15	20,15	0.55	0.68	0.00	0.76	0.57	0.00	0.75	0.60	0.00	0.60	0.60	0.00	30 016	30 034	30 233
rec17	20,15	0.85	0.30	0.00	0.83	0.44	0.00	0.95	0.47	0.00	0.63	0.54	0.17			
rec19	30,10	0.95	0.40	0.00	0.90	0.28	0.00	1.00	0.31	0.00	0.88	0.27	0.11			
rec21	30,10	0.65	0.48	0.00	0.62	0.67	0.00	0.89	0.58	0.00	0.72	0.63	0.04	30 024	30 030	30 155
rec23	30,10	0.94	0.32	0.00	0.82	0.36	0.00	1.00	0.12	0.00	0.84	0.38	0.03			
rec25	30,15	0.75	0.46	0.00	0.91	0.54	0.00	1.00	0.25	0.00	1.00	0.30	0.02			
rec27	30,15	0.75	0.67	0.00	0.73	0.56	0.00	1.00	0.30	0.00	0.95	0.42	0.00	45 027	45 052	45 159
rec29	30,15	0.70	0.53	0.00	0.64	0.70	0.00	0.85	0.57	0.00	0.95	0.49	0.02			
rec31	50,10	0.70	0.47	0.00	0.86	0.31	0.00	0.77	0.52	0.00	0.96	0.33	0.02			
rec33	50,10	0.87	0.35	0.00	0.80	0.48	0.00	0.64	0.54	0.00	0.95	0.42	0.00	50 053	50 100	50 093
rec35	50,10	0.68	0.44	0.00	0.78	0.38	0.00	0.82	0.30	0.00	0.95	0.37	0.00			

表 3 HPFA 与 SFLA、IMMOGLS 在不同缓冲区下的 IGD 指标
Tab. 3 IGD indicators in different buffers of SFLA、IMMOGLS and HPFA

Problem	n,m	缓冲区大小											
		Buffer=0			Buffer=1			Buffer=2			Buffer=4		
		HPFA	SFLA	GLS	HPFA	SFLA	GLS	HPFA	SFLA	GLS	HPFA	SFLA	GLS
rec01	20,5	114.03	131.64	374.14	148.97	203.09	422.51	269.09	344.43	582.31	217.63	214.30	631.54
rec03	20,5	187.99	191.10	481.22	188.92	221.57	360.69	371.86	331.58	541.73	273.46	330.67	307.46
rec05	20,5	117.74	136.84	324.23	300.88	348.75	674.70	126.28	129.24	222.03	131.34	65.76	186.52
rec07	20,10	307.93	441.12	1 802.60	287.20	358.43	1 006.65	207.14	378.60	1 305.37	122.77	473.60	345.40
rec09	20,10	329.56	547.59	1 464.54	691.97	780.59	1 996.70	156.70	591.31	1 247.03	646.57	1 222.94	1 512.93
rec11	20,10	328.11	376.97	2 055.90	371.36	340.42	1 867.50	286.55	533.53	1 657.29	390.69	310.60	921.54
rec13	20,15	360.86	394.68	3 015.24	158.86	193.95	2 334.59	214.27	296.39	2 051.82	160.65	147.46	417.10
rec15	20,15	359.43	369.71	3 038.51	245.39	286.55	2 123.64	362.36	517.15	2 299.00	155.92	214.01	521.80
rec17	20,15	725.11	1150.42	3 252.45	277.74	480.11	2 575.86	563.84	703.71	2 196.89	222.95	322.11	381.76
rec19	30,10	868.33	748.26	2 998.11	716.29	593.64	2 640.35	177.81	231.11	738.89	367.55	801.88	2 187.09
rec21	30,10	539.55	371.25	3 171.94	981.02	903.51	3 388.68	676.09	1 088.61	3 427.41	410.82	444.65	1 033.39
rec23	30,10	1 725.09	1 868.37	3 970.75	574.87	635.85	2 262.88	572.40	580.67	2 033.48	193.59	543.49	614.18
rec25	30,15	199.82	233.40	3 100.75	2 600.62	2 834.97	6 047.18	561.85	753.60	3 598.22	210.46	443.40	2 292.12
rec27	30,15	756.33	649.96	4 746.70	582.59	544.88	4 621.28	296.10	351.39	4 052.71	160.49	234.48	1 776.94
rec29	30,15	956.11	1 109.57	3 167.36	243.35	261.45	3 179.00	395.91	294.42	3 125.20	361.69	446.74	1 400.65
rec31	50,10	1 202.60	1 518.18	4 023.83	1 161.48	1 561.91	3 374.12	582.74	697.41	2 248.59	907.20	1 574.30	2 058.80
rec33	50,10	1 257.05	1 270.31	4 696.97	7 06.39	468.13	2 532.29	699.62	1 189.38	3 509.33	355.13	708.90	3 423.77
rec35	50,10	887.72	955.69	4 215.23	1 142.05	843.91	3 469.86	1 758.14	2 075.04	4 285.48	574.59	882.29	2 422.71

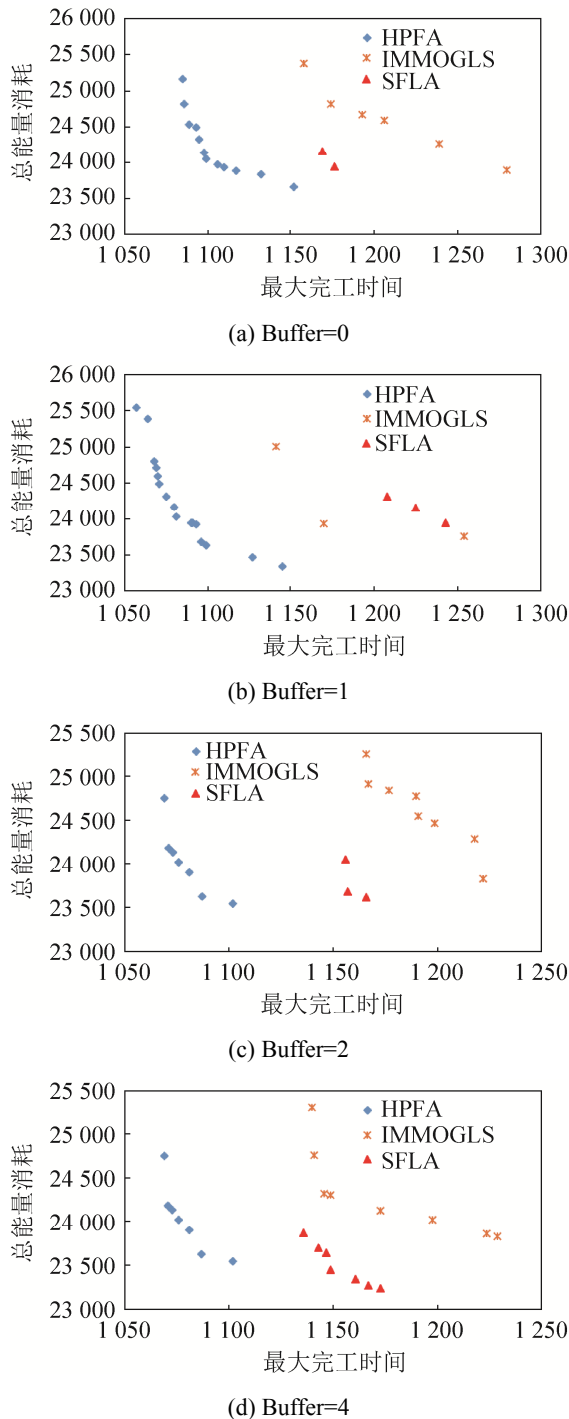


图4 3种算法在不同缓冲区下的非受支配解集对比
Fig. 4 Comparison of nondominated Solution Sets of Three Algorithms in Different Buffers

根据表2和表3的数据并结合图4可以看出, HPFA在求解GFSSP_LBET时具有较强的搜索性能, 所得非受支配解集无论是占优情况还是与最优解集的逼近性均明显优于SFLA和IMMOGLS, 并

且其性能不会随着缓冲区大小的改变而降低。此外, 从表2中的AVG_T可以看出, HPFA的实际运行时间也低于其他两种对比算法。因此可以表明, HPFA能够在较短的时间内有效求解GFSSP_LBET, 并获得较优的非受支配解集。

4 结论

本文在综合考虑机器能耗阈值和有限缓冲区的基础上, 以最大完工时间(C_{max})和总能量消耗(TEC)为优化目标, 建立绿色流水车间调度问题模型(GFSSP_LBET), 进而提出一种混合探路者算法(Hybrid Pathfinder Algorithm, HPFA)进行求解。HPFA具有如下特点: (1) 将算法每代非受支配解集中的所有个体作为探路者, 可引导算法到达较广的解空间区域; (2) 采用基于距离的选择机制确定每个探路者对应的跟随者, 能确保各探路者的附近区域均得到较多的勘探(exploration)式搜索; (3) 采用自学习多邻域搜索对每代更新后的探路者进行较深入细致的开采(Exploitation)式搜索。由于全局勘探能力和局部开采能力均得以提升, HPFA可有效求解GFSSP_LBET。未来研究将把HPFA扩展用于求解绿色分布式装配线调度问题。

参考文献:

- [1] Soukhal A, Oulamara A, Martineau P. Complexity of Flow Shop Scheduling Problems with Transportation Constraints[J]. European Journal of Operational Research (S0377-2217), 2005, 161(1): 32-41.
- [2] 中国工程院战略咨询中心. 绿色制造[M]. 北京: 电子工业出版社, 2016: 2-5.
Strategy Consulting Centre of Chinese Academy of Engineering. Green Manufacturing[M]. Beijing: Publishing House of Electronics Industry, 2016: 2-5.
- [3] Zhang Z W, Tang R Z, Peng T, et al. A Method for Minimizing the Energy Consumption of Machining System: Integration of Process Planning and Scheduling[J]. Journal of Cleaner Production (S0959-6526), 2016, 137(1): 1647-1662.
- [4] 李明, 雷德明. 基于新型帝国竞争算法的高维多目标

- 柔性作业车间调度[J]. 控制理论与应用, 2019, 36(6): 893-901.
- Li Ming, Lei Deming. Novel Imperialist Competitive Algorithm for Many-Objective Flexible Job shop Scheduling[J]. Control Theory and Application, 2019, 36(6): 893-901.
- [5] Pach C, Berger T, Sallez Y, et al. Reactive and Energy-Aware Scheduling of Flexible Manufacturing Systems Using Potential Fields[J]. Computers in Industry (S0166-3615), 65(3): 434-448.
- [6] 马子飞, 王昭, 赵子杰. 基于开关机控制的流水车间节能调度问题研究[J]. 电子测试, 2018, 8(1): 49-51.
Ma Zifei, Wang Zhao, Zhao Zijie. Research on Energy Saving Scheduling Problem of Flow Shop Based on On-off Control[J]. Electronic test, 2018, 8(1):49-51.
- [7] 吴秀丽, 孙阳君. 机器多转速的柔性作业车间绿色调度问题[J]. 计算机集成制造系统, 2018, 24(4): 862-875.
Wu Xiuli, Sun Yangjun. Flexible Job Shop Green Scheduling Problem with Multi-speed Machine[J]. Computer Integrated Manufacturing Systems, 2018, 24(4): 862-875.
- [8] Gahm C, Denz F, Dirr M, et al. Energy-efficient Scheduling in Manufacturing Companies: A Review and Research Framework[J]. European Journal of Operational Research (S0377-2217), 2016, 248(3): 744-757.
- [9] 于艳辉, 李铁克, 王柏琳. 缓冲区有限的流水车间调度问题的启发式算法[J]. 计算机工程与应用, 2012, 48(32): 18-22.
Yu Yanhui, Li Tieke, Wang Baillin. Heuristic Algorithm for Flowshop Scheduling with Limited Buffer Constraint[J]. Computer Engineering and Applications, 2012, 48(32): 18-22.
- [10] 杜田田, 李芳, 武超然. 求解有限缓冲区流水线调度问题的混合蝙蝠算法[J]. 计算机应用与软件, 2015, 32(6): 238-241, 250.
Du Tiantian, Li Fang, Wu Chaoran. Hybrid Bat Algorithm for Solving Permutation Flow-Shop Scheduling Problem with Limited Buffers[J]. Computer Applications and Software, 2015, 32(6): 238-241, 250.
- [11] 徐震浩, 王程, 顾幸生. 基于DICA的存储受限流水车间调度[J]. 华东理工大学学报(自然科学版), 2018, 44(4): 563-572.
Xu Zhenhao, Wang Cheng, Gu Xingsheng. Flow Shop Scheduling Problem with Limited Intermediate Storage Based on Discrete Imperialist Algorithm[J]. Journal of East China University of Science and Technology (Natural Science Edition), 2018, 44(4): 563-572.
- [12] Jiang S L, Zhang L. Energy-Oriented Scheduling for Hybrid Flow Shop with Limited Buffers Through Efficient Multi-Objective Optimization[J]. IEEE Access (S2169-3536), 2019, 7(1): 34477-34487.
- [13] Fang K U, Han N A, Zhao F, et al. Flow Shop Scheduling with Peak Power Consumption Constraints[J]. Annals of Operations Research (S0254-5330), 2013, 206(1): 115-145.
- [14] Liu C G, Dang F, Li W J, et al. Production Planning of Multi-stage Multi-option Seru Production Systems with Sustainable Measures[J]. Journal of Cleaner Production (S0959-6526), 2015, 105(1): 285-299.
- [15] Nagasawa K, Ikeda Y, Irohara T. Robust Flow Shop Scheduling with Random Processing Times for Reduction of Peak Power Consumption[J]. Simulation Modelling Practice and Theory (S1569-190X), 2015, 59(1): 102-113.
- [16] Kemmoe S, Lamy D, Tchernev N. Job-Shop like Manufacturing System with Time Dependent Energy Threshold and Operations with Peak Consumption[C]// IFIP Advances in Information and Communication Technology. Berlin: Springer, 2016: 617-624.
- [17] Kemmoe S, Lamy D, Tchernev N. Job-shop like Manufacturing System with Variable Power Threshold and Operations with Power Requirements[J]. International Journal of Production Research (S0020-7543), 2017, 55(20): 6011-6032.
- [18] Yapici H, Cetinkaya N. A New Meta-heuristic Optimizer: Pathfinder Algorithm[J]. Applied Soft Computing (S1568-4946), 2019, 78(1): 545-568.
- [19] Ding J Y, Song S, Wu C. Carbon-efficient Scheduling of Flow Shops by Multi-objective Optimization[J]. European Journal of Operational Research (S0377-2217), 2016, 248(3): 758-771.

- [20] Lei D, Zheng Y, Guo X. A Shuffled Frog-leaping Algorithm for Flexible Job Shop Scheduling with the Consideration of Energy Consumption[J]. International Journal of Production Research (S0020-7543), 2017, 55(11): 3126-3140.
- [21] Qian B, Li Z H, Hu R, et al. A Hybrid Differential Evolution Algorithm for the Multi-objective Reentrant Job-shop Scheduling Problem[C]// IEEE International Conference on Control & Automation, 2013: 485-489.
- [22] 付亚平, 黄敏, 王洪峰. 面向多目标流水车间调度的多种群多目标遗传算法[J]. 控制理论与应用, 2016, 33(10): 1281-1288.
- Fu Yaping, Huang Min, Wang Hongfeng. Multipopulation Multiobjective Genetic Algorithm for Multiobjective Permutation Flow Shop Scheduling Problem[J]. Control Theory & Applications, 2016, 33(10): 1281-1288.
- [23] Ishibuchi H. Balance between Genetic Search and Local Search in Memetic Algorithms for Multiobjective Permutation Flowshop Scheduling[J]. IEEE Transactions on Evolutionary Computation (S1089-778X), 2003, 7(2): 204-223.