

本文引用格式：曾俊海,廉胤东,彭雄峰,等.面向矩阵式制造车间 AGV 调度的改进模拟退火算法[J].自动化与信息工程,2024,45(5):32-39.

ZENG Junhai, LIAN Yindong, PENG Xiongfeng, et al. Improved simulated annealing algorithm for AGV scheduling in matrix manufacturing workshops[J]. Automation & Information Engineering, 2024,45(5):32-39.

面向矩阵式制造车间 AGV 调度的改进模拟退火算法

曾俊海¹ 廉胤东² 彭雄峰¹ 余锦伟³

(1.华南理工大学自动化科学与工程学院, 广东 广州 510641

2.南方电网供应链科技(广东)有限公司, 广东 广州 510630

3.中国电信股份有限公司广东研究院, 广东 广州 510660)

摘要: 随着客制化和大规模生产需求的不断增长, 矩阵制造车间的重要性日益凸显。同时, 多台自动导引车 (AGV) 的有效调度可提高矩阵制造车间的运行效率。以矩阵制造车间内的 AGV 运输过程为研究对象, 提出一种改进的模拟退火 (ISA) 算法, 旨在找到降低运输成本的最佳调度方案。ISA 算法通过 Metropolis 准则来提高算法跳出局部最优解的能力; 利用顺序交叉算子和前置交叉算子更新种群, 以提高 ISA 算法寻找全局最优解的收敛速度和精度; 提出一种种群重生机制, 避免 ISA 算法陷入局部最优解。为了评估 ISA 算法的有效性, 在某工厂 100 个真实实例的数据集上进行了仿真测试, 并与 FCFS 算法、HFOA、IHS 算法进行对比实验。实验结果表明, ISA 算法更适合解决矩阵制造车间的 AGV 调度问题。

关键词: 矩阵式制造车间; 自动导引车; 模拟退火算法; 调度问题

中图分类号: TP242.6

文献标志码: A

文章编号: 1674-2605(2024)05-0005-08

DOI: 10.3969/j.issn.1674-2605.2024.05.005

开放获取

Improved Simulated Annealing Algorithm for AGV Scheduling in Matrix Manufacturing Workshops

ZENG Junhai¹ LIAN Yindong² PENG Xiongfeng¹ YU Jinwei³

(1.School of Automation Science and Engineering, South China University of Technology, Guangzhou 510641,

China 2.Southern Power Grid Supply Chain Technology (Guangdong) Co., Ltd., Guangzhou 510630, China

3.Guangdong Research Institute of China Telecom Corporation Limited, Guangzhou 510660, China)

Abstract: With the increasing demand for customization and large-scale production, the importance of matrix manufacturing workshops is becoming increasingly prominent. Meanwhile, the effective scheduling of multiple Automated Guided Vehicle (AGV) can improve the operating efficiency of matrix manufacturing workshops. Taking the AGV transportation process in the matrix manufacturing workshop as the research object, an improved simulated annealing (ISA) algorithm is proposed to find the optimal scheduling scheme to reduce transportation costs. The ISA algorithm improves its ability to escape from local optima through the Metropolis criterion; Using sequential crossover operators and pre crossover operators to update the population, in order to improve the convergence speed and accuracy of the ISA algorithm in finding the global optimal solution; Propose a population regeneration mechanism to prevent the ISA algorithm from getting stuck in local optima. In order to evaluate the effectiveness of the ISA algorithm, simulation tests were conducted on a dataset of 100 real instances in a certain factory, and comparative experiments were conducted with the FCFS algorithm, HFOA, and IHS algorithm. The experimental results indicate that the ISA algorithm is more suitable for solving AGV scheduling problems in matrix manufacturing workshops.

Keywords: matrix manufacturing workshops; automated guided vehicle; simulated annealing algorithm; scheduling problem

0 引言

自动导引车 (automated guided vehicle, AGV) 能够自主导航, 执行多种物料的搬运和分配任务, 广泛应用于工厂和仓库等场合^[1-2], 已成为众多制造企业提高效率 and 降低成本不可或缺的工具。AGV 调度问题 (AGV scheduling problem, AGVSP) 是当前制造业备受关注的一项挑战^[3-4], 其高效地规划和管理 AGV 的任务和路径, 确保物料和产品在按时、按需运输的同时, 最大限度地减少等待时间和资源浪费^[5]。AGVSP 包括 AGV 分配问题 (合理分配任务给 AGV, 以设定每台 AGV 的任务服务顺序) 和 AGV 路径规划问题 (为给定的任务服务顺序寻找最佳路径)。AGVSP 的复杂性源于生产设备状态、任务优先级、工厂布局、车辆容量等因素。有效的 AGV 调度可以提升生产线效率和灵活性、减小误差、降低能源消耗, 从而增强制造企业的竞争力^[6-7]。

矩阵制造车间是现代制造企业广泛采用的智能车间之一^[8], 通常被划分为多个生产区域, 每个生产区域负责不同产品或零部件的制造。每台 AGV 配备多个隔间, 每个隔间仅能装载一种物料或工具, 且有装载量限制。矩阵制造车间的 AGVSP 可以看作是车辆路径问题的一个复杂变种, 但因其具有多个隔间、可变需求和多个约束等特征, 更为复杂。车辆路径问题是 NP-hard 问题, 这意味着 AGVSP 也是 NP-hard 问题。针对这种问题, 精确的求解方法在有限的时间内难以获得较优的调度方案。因此, 通常采用基于规则的启发式算法或智能优化算法来求解^[9]。迄今为止, 矩阵制造车间的 AGVSP 只吸引了少数研究者的关注。ZOU 等^[8]提出了矩阵制造车间的 AGVSP, 其基于离散人工蜂群算法结合多种启发式策略实现了 AGV 的有效调度。ZHANG 等^[10]引入一种改进的迭代贪婪算法, 采用 AGV 路线合并和车间分区及修复策略, 防止算法收敛到局部最优解, 最大限度地降低 AGV 的运输成本。ZOU 等^[11]提出一种迭代贪婪算法, 通过解构和重构过程的迭代来增强 AGVSP 的解决方案; 引入加速定理来提高解评估的效率; 采用 Metropolis 准

则来选择解; 最后通过真实案例的比较实验, 证明了所提算法生成的解优于现有算法。LI 等^[12]在矩阵制造车间 AGVSP 中引入离散入侵杂草优化算法, 通过两个有效的邻域算子来提高算法性能; 基于插入的局部搜索方法来增强局部搜索能力; 最后通过综合仿真实验, 验证了所提算法的有效性。尽管以上研究在矩阵制造车间的 AGVSP 上取得了一定进展, 但所采用的方法仍可能陷入局部最优解, 限制了调度方案的质量, 且在大规模或复杂场景下, 计算时间较长, 无法满足实时调度的需求, 影响了实际应用。因此, 探索更高效的调度方案以应对矩阵制造车间 AGVSP 的复杂性, 具有重要的研究意义。

本文构建了一个混合整数线性模型, 并提出一种改进的模拟退火 (improved simulated annealing, ISA) 算法来解决矩阵制造车间的 AGVSP。首先, 将顺序交叉算子和前置交叉算子作为种群更新机制, 提高 ISA 算法寻找全局最优解的收敛速度和精度; 然后, 基于 Metropolis 准则, 以一定的概率接受较差解, 使 ISA 算法在优化前期具有较广的搜索范围; 接着, 提出一种种群重生机制, 以避免 ISA 算法在优化后期陷入局部最优解; 最后, 通过在 100 个真实实例的数据集上进行仿真测试, 验证 ISA 算法解决 AGVSP 的有效性和可靠性。

1 矩阵制造车间 AGVSP 建模

1.1 环境描述

本文研究的矩阵制造车间 AGVSP 描述如下: 在矩阵制造车间中, 设有多台 AGV、一个仓库和 m 个生产区域, 其布局图如图 1 所示。

仓库用于储存各生产区域所需的物料。根据物料的数量, 将矩阵制造车间划分为 m 个生产区域, 每个生产区域拥有若干个工作站。每个工作站均设有缓冲区和数控机床。当缓冲区的物料数量低于设定的阈值时, 叫料工作站向控制中心发出物料请求信号, 要求 AGV 进行补给。为了避免 AGV 资源浪费和运输成本增加, 将生产时间划分为若干个周期, 每个周期包括规划阶段和调度阶段。在规划阶段, 控制中心为 AGV

分配任务；在调度阶段，控制中心派遣 AGV 将物料运输到指定的工作站，并返回仓库。第 x 个生产周期内收到的物料请求将在第 $x+1$ 个生产周期内处理。

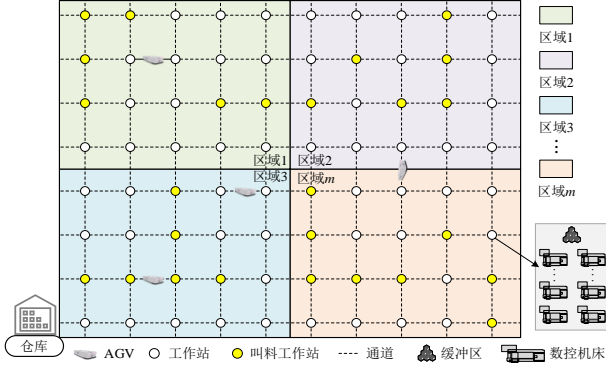


图 1 矩阵制造车间布局图

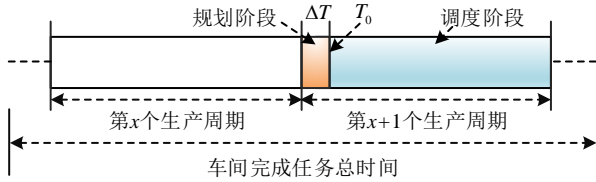


图 2 生产周期划分图

假设所需物料都储存在仓库中；所有 AGV 的质量相同，匀速行驶，且不受装载量的影响；每台 AGV 拥有多个隔间，每个隔间用于存放特定的物料；隔间数量等于物料和刀具种类的总数；AGV 在矩阵制造车间内独立行驶，不发生停机、故障或碰撞等事故；AGV 从仓库出发，最终返回仓库；每条 AGV 路线只能由一台 AGV 执行；每个客户只能被服务一次；一条路线上所有客户的物料需求总量不超过 AGV 的总载重量；AGV 在客户要求的时间窗口内提供服务。

1.2 数学建模

基于上述假设，AGVSP 可以定义为无向图 $G = \{V, E\}$ ，其中 $V = \{0, 1, \dots, n\}$ 表示顶点集合，顶点 0 表示仓库； $E = \{(i, j) | i, j \in V, i \neq j\}$ 表示运输成本。需要运输的物料种类集合为 $M = \{0, 1, 2, \dots, m\}$ ，其中 0 表示刀具，其余编号表示不同的物料。AGV 路线集合定义为 $K = \{1, 2, \dots, k\}$ 。每台 AGV 从仓库出发，拥有多个隔间用于搭载物料，总载重量为 Q ，每个隔间

Q_m 负责搭载对应的物料，并满足 $\sum_0^m Q_m = Q$ 。

在矩阵制造车间的 AGVSP 过程中，AGV 需在客户 i 的时间窗口 $[T_i^c, T_i^d]$ 内到达，其中， T_i^c 为叫料时间， T_i^d 为交货时间。AGV 到达后，需花费固定的时间 \bar{t} 卸载物料。

假设客户 j 的物料请求包括生产物料和刀具。

当客户 j 请求物料 m 时，AGV 从仓库中装载相应的物料，运输给客户 j ；在 AGV 运输过程中，客户 j 同时也在消耗物料。因此，AGV 到达客户 j 时的装载量 q_j^m 的计算公式为

$$q_j^m = \left[(S_m - S_j^c) + \left[(T_j^a - T_j^c) / t_m \right] \right] \omega_m, \quad (1)$$

$$\forall j \in V \setminus \{0\}, m \in M \setminus \{0\}$$

式中： S_m 为缓冲区物料 m 的总库存量， S_j^c 为客户 j 请求物料时缓冲区物料 m 的库存量， T_j^a 为 AGV 到达客户 j 的时间， t_m 为客户 j 消耗每片物料所需的时间， ω_m 为每片物料的质量。

AGV 到达客户 j 的时间与其行驶距离紧密相关。假设一台 AGV 从客户 i 运输物料给客户 j ，则其行驶距离 d_{ij} 和行驶时间 T_{ij} 的计算公式分别为

$$d_{ij} = |x_i - x_j| + |y_i - y_j| \quad (2)$$

$$T_{ij} = d_{ij} / v \quad (3)$$

式中： v 为 AGV 的行驶速度。

AGV 到达客户 j 的时间 T_j^a 的计算公式为

$$T_j^a = \begin{cases} T_0 + \bar{t} + T_{ij}, & i = 0 \\ T_i^a + \bar{t} + T_{ij}, & i \neq 0 \end{cases} \quad (4)$$

式中： T_0 为 AGV 离开仓库的时间。

当客户 j 请求刀具时，刀具质量 q_j^0 的计算公式为

$$q_j^0 = n_j^{\text{cut}} \omega_0, \forall j \in V \setminus \{0\} \quad (5)$$

式中： n_j^{cut} 为客户 j 请求的刀具数量， ω_0 为每片刀具的质量。

本文定义的运输成本主要包括 AGV 的行驶成本、

服务成本和使用成本。

行驶成本 C'_{ij} 与行驶距离 d_{ij} 相关，计算公式为

$$C'_{ij} = u' d_{ij} \quad (6)$$

式中： u' 为行驶距离的单位成本。

服务成本与交货时间相关。AGV 到达客户的时间越准确，意味着客户服务的顺序越合理。客户 j 的服务成本 C^s_j 的计算公式为

$$C^s_j = u^s (T_j^d - T_j^a) \quad (7)$$

式中： u^s 为客户服务的单位成本。

使用成本与运行的 AGV 数量相关，计算公式为

$$C^v = u^v n_v$$

$$n_v = \sum_{k=1}^K \sum_{j=1}^n L_{ijk}, i=0 \quad (8)$$

式中： u^v 为每台 AGV 的使用成本； $L_{ijk} = 1$ 为第 k 台 AGV 服务客户 i 和客户 j ，反之， $L_{ijk} = 0$ 。

根据 AGV 的行驶成本、服务成本和使用成本，构建矩阵制造车间 AGVSP 的混合整数线性方程为

$$\min f(i, j, k) = \sum_{k=1}^{\bar{k}} \sum_{i=0}^n \sum_{j=0}^n L_{ijk} C'_{ij} + \sum_{k=1}^{\bar{k}} \sum_{i=0}^n \sum_{j=0}^n L_{ijk} C^s_j n + C^v \quad (9)$$

s.t.

$$\sum_{k=1}^{\bar{k}} \sum_{j=0}^n L_{ijk} = 1, \forall j \in V \setminus \{0\} \quad (10)$$

$$\sum_{k=1}^{\bar{k}} \sum_{j=0}^n L_{ijk} = 1, \forall i \in V \setminus \{0\} \quad (11)$$

$$\sum_{i=0}^n L_{ijk} = \sum_{i=0}^n L_{jik}, \forall j \in V \setminus \{0\}, \forall k \in K \quad (12)$$

$$\sum_{i=1}^n L_{i0k} = \sum_{j=1}^n L_{0jk} = 1, \forall k \in K \quad (13)$$

$$L_{ijk} (T_i^a + t_{ij} + \bar{t} - T_j^a) = 0, \forall i \in V, \forall j \in V \setminus \{0\} \quad (14)$$

$$\sum_{j=1}^n \sum_{i=0}^n L_{ijk} \cdot q_j^m \leq Q^m, \forall k \in K, m \in M \quad (15)$$

$$T_i^c \leq T_i^a \leq T_i^d, \forall i \in V \setminus \{0\} \quad (16)$$

$$L_{ijk} \in \{0, 1\}, \forall i, j \in V, \forall k \in K \quad (17)$$

$$L_{ijk} = 0, \forall i, j \in V, i = j \quad (18)$$

$$T_i^a = T_0, i = 0 \quad (19)$$

2 ISA 算法

模拟退火 (simulated annealing, SA) 算法是一种随机优化算法，其灵感源自金属的退火过程。该算法主要分为 Metropolis 准则和退火过程两部分。其中，Metropolis 准则允许 SA 算法以一定的概率 P 接受较差解，以避免算法陷入局部最优解，计算公式为

$$P = \begin{cases} 1, f(t+1) < f(t) \\ e^{-\frac{f(t+1)-f(t)}{Te(t)}}, f(t+1) > f(t) \end{cases} \quad (20)$$

式中： $f(t)$ 为第 t 代的个体目标值； $Te(t)$ 为第 t 代的温度，该温度随着迭代次数的增加而不断下降，这个过程即为退火过程，计算公式为

$$Te(t+1) = \alpha Te(t) \quad (21)$$

式中： α 为退火速率，一般取值为 $[0.8, 0.99]$ 。

尽管 SA 算法接受较差解，以跳出局部最优解，但在某些情况下，如解空间结构复杂时，仍可能陷入局部最优解。因此，本文提出一种 ISA 算法，通过引入顺序交叉算子和前置交叉算子，提高算法的求解精度和收敛速度。ISA 算法流程如图 3 所示。

2.1 解的编码方式

为保持 ISA 算法的简洁性，本文采用直接解表示法。对于具有 n 个任务和 k 条路径的矩阵制造车间，采用一个长度为 $n+k-1$ 的向量 $\mathbf{R} = \{r_1, 0, r_2, 0, \dots, r_k\}$ 来表示调度方案。其中， $r_k = \{r_{k,1}, r_{k,2}, \dots, r_{k,n_k}\}$ 为第 k 条路线， n_k 为在路线 k 的任务数量；0 为不同路线之间的分隔符。

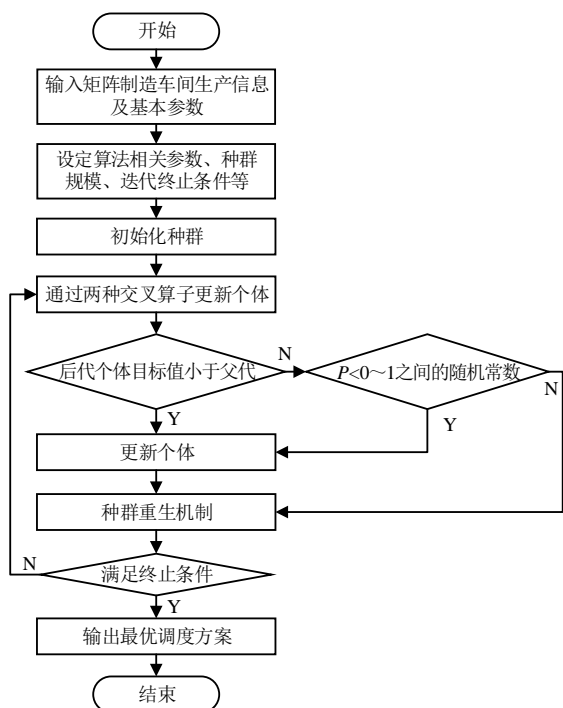


图3 ISA 算法流程图

2.2 种群更新机制

本文采用顺序交叉算子和前置交叉算子更新种群。其中，顺序交叉算子有效地保留了父代个体的相对顺序，增加了后代个体的多样性，有助于进一步探索解空间；前置交叉算子可以在特定的位置插入客户，更灵活地适应不同任务的调度需求，提高了ISA算法的适应性。通过这两种交叉算子的结合使用，ISA算法在保持优良特性的同时，促进了种群的演化与优化。

2.2.1 顺序交叉算子

顺序交叉算子的具体实现过程如下：

1) 从每个父代中随机选择连续的基因构成基因段，基因段中基因数量 N 的计算公式为

$$N = \left\lceil \frac{m}{2} \cdot \frac{Te(t)}{Te_{\max}} \right\rceil \quad (22)$$

式中： Te_{\max} 为最大温度。

2) 对于随机选择的基因段 a (来自父代1) 和 b (来自父代2)，父代1的后代1保留基因段 a ，其余位置暂时留空；

3) 从左到右平移父代2中的基因序列，直到与基因段 b 序列后的第一个基因匹配；

4) 删除平移后父代2中与基因段 a 匹配的基因，并将剩余的基因按顺序插入到父代1的基因段 a 之后，从而生成新的后代1，父代2也采用相同的方法生成后代2；

5) 从所有后代中选择表现最优秀的个体作为新的解。顺序交叉算子示意图如图4所示。



图4 顺序交叉算子示意图

2.2.2 前置交叉算子

前置交叉算子的具体实现过程如下：

1) 选择父代中相同连续位置的基因段，基因段的基因数量由 N 决定；

2) 将父代2所选的基因段放在后代1的开始位置，后代1的其他未确定位置用父代1中暂时没有在后代1中出现，且离后代1中出现的最后一个基因最近的基因作为后续基因；

3) 从左到右将父代1中未出现的基因依次移至后代1中，生成后代1，父代2同理。

前置交叉算子的示意图如图5所示。

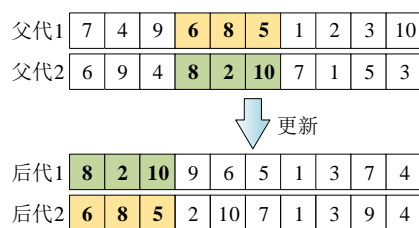


图5 前置交叉算子示意图

在矩阵制造车间的AGVSP过程中，前置交叉算子将距离作为判断依据，选择最短行驶距离的任务作为后续基因。为了使ISA算法能够选择更加符合矩阵制造车间调度指标的后续基因，本文提出一种基于最短行驶距离和交货时间的衡量标准，计算公式为

$$H_{ij} = \beta w^i d_{ij} + (1 - \beta) u^s T_i^d \quad (23)$$

式中: β 为行驶距离和交货时间的权重因子。

2.3 种群重生机制

为了避免 ISA 算法陷入局部最优解, 本文提出一种种群重生机制。当 ISA 算法在第 l 次迭代过程中仍然无法找到更优解时, 则认为算法陷入局部最优解, 此时采用种群重生机制重新生成个体并进一步探索解空间, 从而摆脱局部最优解。种群重生机制的具体实现过程如下:

1) 选择解 R 中的一条路线 r_k , 并从该条路线中随机选择 $[0, n_k]$ 数量的客户放入到档案库 π 中;

2) 提取 π 中的第一个客户, 插入到 R 剩余路线所有能够插入的位置, 并选择最优位置作为该客户的最终位置;

3) 若 R 的剩余路线都无法插入该客户, 则重新生成一条新路线。以此类推, 直至 π 中的客户全部被插入为止。

3 实验与分析

本文采用 ZOU 等提供的 100 个矩阵制造车间的真实实例进行仿真测试^[8]。该测试实例的客户数量包括 10、20、30、40 和 50 个, 每个大小类别分别含有 20 个实例, 并利用 T10I1 表示任务为 10 的第一个实例。为了进一步评估本文 ISA 算法求解矩阵制造车间 AGVSP 的有效性。将本文 ISA 算法与先来先服务

(first come first serve, FCFS) 算法、混合果蝇优化算法 (hybrid fruit-fly optimization algorithm, HFOA)、改进的和谐搜索 (improved harmony search, IHS) 算法进行对比实验。设置最大迭代时间为 5 s, 种群数量为 150 个, 其他参数设置如表 1 所示。4 种算法均在 Windows 10 操作系统上 MATLAB R2020a 的 .m 文件实现, 并在一台 CPU 主频为 2.60 GHz、处理器内存为 16 GB 的计算机上运行。

表 1 4 种算法参数设置

算法	参数
FCFS	-
HFOA	$\alpha = 1, \beta = 2, q_0 = 0.9, \rho = 0.9$
IHS	$HMS = 4, HMCR_{\min} = 0.2, HMCR_{\max} = 0.8$
ISA	$Te_{\max} = 10\ 000, \alpha = 0.98, \beta = 0.5, l = 3$

为了更准确地量化 ISA 算法与最优调度方案之间的差异, 采用相对百分比偏差 (relative percentage deviation, RPD) 来评估实验结果。RPD 的计算公式为

$$RPD = \frac{f - f_{\text{best}}}{f_{\text{best}}} \times 100\% \quad (24)$$

式中: f 为所用算法获得的运输成本, f_{best} 为 4 种算法对同一实例获得的最优运输成本。

为确保对比实验的公平性, FCFS 算法、HFOA、HIS 算法和 ISA 算法先在所有测试实例中各自独立运行 30 次, 再对平均 RPD、最优 RPD 进行比较, 结果如表 2 所示。

表 2 4 种算法 RPD 结果

实例	FCFS 算法		HFOA		HIS 算法		ISA 算法		实例	FCFS 算法		HFOA		HIS 算法		ISA 算法	
	平均	最优	平均	最优	平均	最优	平均	最优		平均	最优	平均	最优	平均	最优	平均	最优
T10I01	27.3	27.3	6.2	4.1	0.1	0.0	0.1	0.0	T20I01	25.4	25.4	15.2	12.6	5.7	4.3	5.1	0.0
T10I02	40.2	40.2	10.2	7.4	1.4	0.0	1.1	0.0	T20I02	28.7	28.7	16.1	11.8	4.9	2.0	4.6	0.0
T10I03	34.2	34.2	6.3	1.8	0.2	0.0	1.8	1.0	T20I03	33.9	33.9	9.6	7.9	2.1	0.0	2.4	1.5
T10I04	34.1	34.1	4.6	2.6	0.1	0.0	0.7	0.5	T20I04	16.1	16.1	9.9	7.5	2.2	0.0	4.1	1.6
T10I05	40.4	40.4	4.7	1.5	0.0	0.0	0.2	0.0	T20I05	41.9	41.9	11.8	10.9	2.4	1.0	2.2	0.0
T10I06	38.7	38.7	6.6	2.1	0.5	0.0	0.6	0.0	T20I06	23.5	23.5	14.0	10.3	3.2	1.4	3.1	0.0
T10I07	33.5	33.5	7.5	2.4	0.2	0.0	1.2	0.0	T20I07	13.8	13.8	12.2	8.4	2.1	0.0	3.7	1.3
T10I08	39.8	39.8	7.1	2.6	0.3	0.0	0.6	0.2	T20I08	40.7	40.7	12.3	6.5	2.7	0.0	2.6	0.7
T10I09	42.6	42.6	6.7	3.3	0.0	0.0	0.0	0.0	T20I09	30.3	30.3	13.6	7.4	4.1	1.4	3.5	0.0
T10I10	25.7	25.7	5.9	3.7	0.9	0.0	0.1	0.0	T20I10	21.6	21.6	12.8	10.9	2.6	0.0	3.7	1.0

续表

实例	FCFS 算法		HFOA		HIS 算法		ISA 算法		实例	FCFS 算法		HFOA		HIS 算法		ISA 算法	
	平均	最优	平均	最优	平均	最优	平均	最优		平均	最优	平均	最优	平均	最优	平均	最优
T10I11	30.4	30.4	8.0	5.8	0.5	0.0	0.7	0.0	T20I11	22.7	22.7	11.9	5.9	3.0	2.0	1.7	0.0
T10I12	20.9	20.9	4.6	2.9	0.3	0.0	0.3	0.0	T20I12	21.1	21.1	10.8	8.6	2.8	0.0	2.0	0.9
T10I13	28.6	28.6	6.8	0.8	0.1	0.0	0.5	0.5	T20I13	11.6	11.6	8.5	5.7	1.0	0.0	2.2	0.2
T10I14	39.3	39.3	9.2	6.5	2.1	2.1	1.2	0.0	T20I14	30.0	30.0	11.0	5.6	1.5	0.0	1.8	0.8
T10I15	19.9	19.9	5.6	1.9	0.2	0.0	0.4	0.0	T20I15	26.4	26.4	10.9	8.5	4.3	1.5	3.7	0.0
T10I16	35.9	35.9	6.1	1.0	0.4	0.0	0.5	0.2	T20I16	14.3	14.3	9.8	5.7	1.9	0.0	2.2	0.3
T10I17	40.1	40.1	8.4	5.3	0.6	0.0	2.0	1.6	T20I17	25.1	25.1	13.7	9.2	3.5	0.9	3.6	0.0
T10I18	46.3	46.3	8.6	5.4	1.2	0.0	0.8	0.0	T20I18	31.3	31.3	12.9	10.1	3.0	0.2	3.3	0.0
T10I19	34.8	34.8	6.2	3.9	0.7	0.0	0.8	0.5	T20I19	43.1	43.1	15.8	11.5	6.7	4.1	6.1	0.0
T10I20	34.3	34.3	5.6	3.8	0.0	0.0	0.2	0.0	T20I20	16.7	16.7	8.7	6.5	1.4	0.0	2.3	0.1
实例	FCFS 算法		HFOA		HIS 算法		ISA 算法		实例	FCFS 算法		HFOA		HIS 算法		ISA 算法	
	平均	最优	平均	最优	平均	最优	平均	最优		平均	最优	平均	最优	平均	最优	平均	最优
T30I01	22.8	22.8	16.0	10.4	4.6	1.4	3.5	0.0	T40I01	28.8	28.8	18.7	13.8	8.5	2.1	3.2	0.0
T30I02	31.1	31.1	17.2	14.4	3.7	1.9	2.1	0.0	T40I02	33.2	33.2	20.6	17.9	9.6	7.2	2.0	0.0
T30I03	36.8	36.8	14.3	6.7	2.7	0.4	1.1	0.0	T40I03	35.4	35.4	21.9	18.6	9.6	4.6	3.7	0.0
T30I04	29.6	29.6	14.4	9.6	3.5	0.0	2.5	1.0	T40I04	24.5	24.5	18.9	15.2	7.1	4.8	2.7	0.0
T30I05	11.1	11.1	18.5	16.2	3.3	0.0	2.1	1.9	T40I05	35.0	35.0	22.2	16.3	10.0	6.0	4.3	0.0
T30I06	25.2	25.2	16.5	10.5	5.5	4.2	3.6	0.0	T40I06	31.2	31.2	23.3	20.6	12.6	7.7	3.9	0.0
T30I07	31.7	31.7	17.8	12.7	4.7	2.6	2.6	0.0	T40I07	40.1	40.1	21.2	16.3	10.0	5.4	4.6	0.0
T30I08	29.3	29.3	14.2	6.3	2.9	0.7	0.9	0.0	T40I08	31.2	31.2	21.2	18.2	10.2	6.6	3.3	0.0
T30I09	28.7	28.7	16.3	13.1	2.4	1.6	0.9	0.0	T40I09	34.2	34.2	23.5	17.7	12.3	8.2	4.1	0.0
T30I10	30.6	30.6	19.2	14.3	4.5	0.0	4.9	3.2	T40I10	18.8	18.8	18.3	14.6	8.1	1.5	2.3	0.0
T30I11	20.1	20.1	14.6	9.3	4.8	3.3	2.9	0.0	T40I11	43.2	43.2	20.9	17.7	10.3	7.0	4.1	0.0
T30I12	28.3	28.3	14.6	7.3	3.3	1.2	2.1	0.0	T40I12	31.7	31.7	20.4	15.2	10.7	8.5	3.0	0.0
T30I13	24.8	24.8	16.1	13.7	2.9	0.0	1.7	0.3	T40I13	24.8	24.8	20.6	17.5	8.9	4.4	2.2	0.0
T30I14	30.4	30.4	19.8	15.7	5.1	1.8	2.2	0.0	T40I14	36.3	36.3	19.9	14.3	7.8	2.6	1.9	0.0
T30I15	25.0	25.0	15.3	11.8	3.4	1.1	2.6	0.0	T40I15	36.3	36.3	17.7	14.1	7.6	3.4	2.8	0.0
T30I16	39.0	39.0	14.7	11.3	2.5	0.0	1.3	0.6	T40I16	26.9	26.9	22.2	15.6	10.0	7.3	3.7	0.0
T30I17	20.1	20.1	17.0	14.1	4.0	1.7	1.2	0.0	T40I17	24.3	24.3	16.2	10.6	6.7	2.4	1.9	0.0
T30I18	40.6	40.6	16.1	12.2	2.7	0.0	1.1	0.3	T40I18	31.0	31.0	19.9	15.0	10.0	4.8	4.5	0.0
T30I19	30.1	30.1	17.6	13.3	2.9	0.0	2.4	1.3	T40I19	32.1	32.1	20.6	15.8	10.3	4.2	3.9	0.0
T30I20	22.3	22.3	14.3	10.8	2.7	0.8	2.2	0.0	T40I20	32.8	32.8	21.0	15.2	10.1	6.1	5.1	0.0
实例	FCFS 算法		HFOA		HIS 算法		ISA 算法		实例	FCFS 算法		HFOA		HIS 算法		ISA 算法	
	平均	最优	平均	最优	平均	最优	平均	最优		平均	最优	平均	最优	平均	最优	平均	最优
T50I01	17.9	17.9	14.8	11.4	5.3	3.8	2.2	0.0	T50I11	32.5	32.5	17.8	13.9	8.3	5.1	3.3	0.0
T50I02	30.0	30.0	16.5	10.1	6.4	3.7	0.8	0.0	T50I12	22.2	22.2	14.8	10.2	5.4	2.4	0.8	0.0
T50I03	27.0	27.0	16.4	12.3	6.5	1.0	1.7	0.0	T50I13	15.6	15.6	17.2	13.0	6.1	3.3	2.3	0.0
T50I04	14.2	14.2	18.7	15.2	8.5	4.3	3.1	0.0	T50I14	26.1	26.1	16.7	12.6	7.5	5.3	4.9	0.0
T50I05	36.7	36.7	19.7	16.2	10.2	6.0	4.2	0.0	T50I15	25.3	25.3	15.7	12.5	5.2	3.0	1.6	0.0
T50I06	29.2	29.2	23.2	19.5	11.2	5.7	6.2	0.0	T50I16	17.9	17.9	15.8	14.4	4.2	0.0	1.4	0.0
T50I07	27.2	27.2	18.8	14.2	10.3	6.2	4.6	0.0	T50I17	19.6	19.6	15.2	10.7	6.3	2.3	2.6	0.0
T50I08	22.2	22.2	15.7	13.0	6.3	3.2	1.1	0.0	T50I18	36.0	36.0	18.2	10.9	9.7	7.3	3.1	0.0
T50I09	27.9	27.9	16.4	11.7	8.8	4.3	2.8	0.0	T50I19	33.0	33.0	20.6	18.3	10.9	6.4	5.5	0.0
T50I10	15.5	15.5	15.7	12.2	6.3	1.1	0.8	0.0	T50I20	26.2	26.2	21.1	16.3	12.1	8.6	5.9	0.0

由表 2 可知: 在客户数量为 10 个的实例中, ISA 算法和 IHS 算法的性能不相上下; 在客户数量为 20 和 30 个的实例中, ISA 算法和 IHS 算法的性能也相差无几; 在客户数量为 40 和 50 个的实例中, ISA 算法不管是精确度还是鲁棒性, 均优于 FCFS 算法、HFOA、IHS 算法, 证明了 ISA 算法求解 AGVSP 的优越性。

4 结论

针对矩阵制造车间的多隔间、可变需求和多约束 AGVSP, 本文提出一种 ISA 算法。该算法利用顺序交叉算子和前置交叉算子来更新种群个体, 并通过 Metropolis 准则接受较差解, 以保证算法的前期勘探能力, 避免陷入局部最优解; 通过种群重生机制, 平衡算法后期的勘探能力和开发能力, 不仅增强了算法跳出局部最优解的能力, 还加强了算法进一步开发解的能力。最后, 通过与 FCFS 算法、HFOA、IHS 算法的对比实验, 验证了 ISA 算法的有效性。未来研究可以考虑以下几个方面: 1) 将 ISA 算法与其他智能优化算法相结合, 以提高其在动态环境中的适应性和灵活性; 2) 探索在更复杂的制造环境中引入多目标优化, 以平衡成本、时间和资源利用效率; 3) 集成机器学习技术, 以提升算法的预测能力和响应速度; 4) 通过对实际应用案例的深入分析, 进一步验证 ISA 算法在不同制造场景中的有效性和可扩展性。

©The author(s) 2024. This is an open access article under the CC BY-NC-ND 4.0 License (<https://creativecommons.org/licenses/by-nc-nd/4.0/>)

参考文献

[1] 李玉, 裴道方, 高银萍, 等. 基于数字孪生的自动化集装箱码头

作者简介:

曾俊海, 男, 1996 年生, 博士研究生, 主要研究方向: 智能优化算法及机器人调度。E-mail: jhaizeng@163.com

廉胤东, 男, 1992 年生, 博士研究生, 工程师, 主要研究方向: 数字供应链、机器人调度、信息物理系统。E-mail: 601292628@qq.com

彭雄峰, 男, 2000 年生, 硕士研究生, 主要研究方向: 移动机器人视觉伺服控制与协调规划。E-mail: 1241491625@qq.com

余锦伟, 男, 1995 年生, 博士研究生, 工程师, 主要研究方向: 机器视觉、图像处理、人工智能。E-mail: yujw5@chinatelecom.cn

多 AGV 动态调度[J]. 计算机集成制造系统, 2023, 29(12): 4175-4190.

[2] 马骏, 王亚东, 蔡国旗, 等. 基于智能计算方法的 AGV 制造车间调度问题研究现状[J]. 机电工程技术, 2021, 50(8): 6-10; 73.

[3] ABDERRAHIM M, BEKRAR A, TRENTESAUX D, et al. Manufacturing 4.0 operations scheduling with AGV battery management constraints[J]. Energies, 2020, 13(18): 4948.

[4] 钟建琳, 刘忠和. 制造系统中多智能体运输子系统的调度与监控[J]. 机床与液压, 2011, 39(11): 73-76; 50.

[5] HU H, JIA X, HE Q, et al. Deep reinforcement learning based AGVs real-time scheduling with mixed rule for flexible shop floor in industry 4.0[J]. Computers & Industrial Engineering, 2020, 149: 106749.

[6] 彭丽文, 沈吟东. 多隔间车辆路径问题研究综述[J]. 物流科技, 2021, 44(2): 72-77; 87.

[7] 朱伟枝, 谢娟烘, 卢子荣. 基于 SLAM 的校园配送 AGV 地图构建及路径规划研究[J]. 机电工程技术, 2023, 52(1): 107-110; 160.

[8] ZOU W Q, PAN Q K, MENG T, et al. An effective discrete artificial bee colony algorithm for multi-AGVs dispatching problem in a matrix manufacturing workshop[J]. Expert Systems with Applications, 2020, 161: 113675.

[9] 庞燕, 罗华丽, 邢立宁, 等. 车辆路径优化问题及求解方法研究综述[J]. 控制理论与应用, 2019, 36(10): 1573-1584.

[10] ZHANG X, SANG H, LI J, et al. An effective multi-AGVs dispatching method applied to matrix manufacturing workshop[J]. Computers & Industrial Engineering, 2022, 163: 107791.

[11] ZOU W Q, PAN Q K, TASGETIREN M F. An effective iterated greedy algorithm for solving a multi-compartment AGV scheduling problem in a matrix manufacturing workshop[J]. Applied Soft Computing, 2021, 99: 106945.

[12] LI Z K, SANG H Y, LI J Q, et al. Invasive weed optimization for multi-AGVs dispatching problem in a matrix manufacturing workshop[J]. Swarm and Evolutionary Computation, 2023, 77: 101227.