doi:10.16018/j.cnki.cn32-1650/n.202504009

基于改进NSGA-Ⅲ算法的多AGV任务分配

陈永祥,王建平,邱子麒,代周杰,孙 杰
(安徽工程大学 机械与汽车工程学院,安徽 芜湖 241000)

摘要:针对仓库多AGV协同调度的拓扑图法无法考虑行驶冲突,栅格图法需对各AGV行驶路径单独规划,从而浪费算力的问题,提出了一种在栅格地图中采用双通道的建模方法和以冲突路口建立拓扑点映射的方法进行路径规划,无需使用单车寻路算法。在算法方面,针对迭代算法过程中广度和深度不能在前后期合理地兼顾,优化目标单一的问题,提出了一种基于NSGA-Ⅲ的多目标优化算法;将算法中的遗传和交叉概率调整为自适应,交叉改为分段交叉,并保留优秀父代片段,从而提高算法的收敛速度;设置考虑AGV系统的能耗、运行时间、负载标准差3个目标函数,并在栅格地图中多AGV调度问题上进行了仿真实验。结果表明,提出的NSGA-Ⅲ改进算法具有更好的优化结果,收敛代数分布更小;3个目标函数平均收敛速度相比改进前分别提升了26.7%、33.3%、23.5%。

关键词:AGV调度;NSGA-Ⅲ;路径规划;自适应交叉变异;分段交叉;多目标优化
中图分类号:TP391.4 **文献标志码**:A **文章编号**:1671-5322(2025)04-0060-10

随着工业生产规模的扩大和电商行业的快速崛起,现代物流和仓储行业快速发展,仓库货物转运、入库、出库的效率要求日益提高。自动导引车(automated guided vehicle,AGV)作为一种重要的自动化运输设备,在提高工作效率和降低人力成本方面的作用日益明显[1-3]。因此,在复杂的仓储环境中,如何高效地为多自动导引车分配任务和规划路径,确保AGV系统能够快速、准确地完成各类运输任务,是当前研究的一个热点问题。该问题需同时优化多个目标如任务完成时间、能耗、负载均衡等,属于典型的NP(non-deterministic polynomial)难问题[4-5]。传统算法如遗传算法、粒子群优化在处理高维目标空间时易陷入局部最优且解集分布不均,而非支配排序遗传算法(non-dominated sorting genetic algorithm,NSGA)系列因其高效的帕累托前沿搜索能力成为研究热点,其中第三代非支配排序遗传算法(NSGA-III)通过引入参考点机制和动态种群管理策略,显著提升了高维目标优化问题的收敛性和解集多样性。

虽然NSGA-III算法在解决多目标优化方面表现出色,但其性能在很大程度上依赖于遗传算子的效率。标准的遗传操作如模拟二进制交叉SBX和多项式变异通常采用固定的参数,难以在收敛速度和种群多样性之间实现动态平衡,从而影响了算法在复杂问题上的求解效果。为了克服这一局限性,本文对NSGA-III的遗传算子进行了2项关键改进:

1)提出了一种自适应交叉与变异机制,该机制能够根据种群进化过程中的分布特性动态调整算子参数,从而在进化早期促进全局探索,在进化后期加速局部收敛;

2)设计了一种分段交叉策略,该策略让决策变量根据自身特性进行分段,并在不同段上应用不同的交叉方法,以更精细地挖掘解空间的结构信息,提升生成后代的质量。

这些改进显著增强了算法维持多样性和逼近帕累托前沿的能力。通过上述改进,本文提出

收稿日期:2025-09-24
基金项目:安徽省重点实验室开放基金项目(QCKJJ202505),芜湖市重点研发计划项目(2023ly05)。
作者简介:陈永祥(1999—),男,安徽蚌埠人,硕士生,主要研究方向为智能网联汽车。
通信作者:王建平(1970—),男,甘肃天水人,正高级工程师,博士,主要研究方向为智能网联汽车。

了自适应分段交叉与变异的NSGA-III算法，通过引入先进的优化算法和策略，实现对AGV系统的智能化调度和管控。

1　路径规划模型

1.1　建模假设

在利用自动导引车对仓储货物运输时，为节约成本、降低能耗，需要对小车行驶路径进行规划。为研究小车最优行驶路径，需要对行驶路径进行规划建模。为此，特作出如下假设：

1）仓库内不存在人为或其他障碍物干扰，AGV可以按照计算的最优路线行驶；

2）任务点之间存在不少于一条通路，所有任务点之间的距离都是已知量；

3）每台任务车都是由起始点出发，依次到达各任务点后再返回起始点；

4）所有任务点的作业时间均相同，所有AGV均是相同型号并具有相同性能。

根据上述假设，本文利用栅格法将物流仓库抽象成一个50×50的栅格地图[6]，每个栅格的边长为栅格最小单位1，栅格中心点坐标为(x,y)；仓库货架与货架之间设计成双通道，所有的AGV在行驶过程中均采用靠右行驶的规则，不存在因狭窄路段而发生对向行驶冲突的问题；利用拓扑图表示AGV行驶的路径段，省去了使用单车寻路算法的时间，提高了协同调度算法的运行效率。物流仓库栅格图和拓扑图如图1所示。

1.2　目标函数构造

在以往的多AGV协同调度中，为研究能耗方便，多默认AGV处于匀速直线和停止2种状态。本文考虑AGV待机状态下各种控制系统及散热系统仍在工作，且AGV在加速、转弯、减速、匀速状态的能耗不同，推导出更符合AGV实际工作状态的能耗公式。

在推导公式之前，需对模型参数及变量进行定义。为此，定义模型参数如下：(x_0,y_0)为AGV的起点坐标；m_0为AGV自重，m为AGV数量，AGV集合为$C=\{m_1,m_2,\cdots,m_m\}$；n为任务数量，任务集合为$T=\{n_1,n_2,\cdots,n_n\}$，每个任务n_p都是一个包含任务序号p、任务点坐标(x^p,y^p)、货物质量m^p的3个元素的集合；第k辆AGV的任务数量为l，则第k辆AGV的任务集合为$C_k=\{n_1^k,n_2^k,\cdots,n_l^k\}$；AGV运行速度为$v_k$、最大载重为$m_{\max}$。同时规定单个AGV运输任务总质量不大于小车最大载重，即

$$\sum_{i=1}^{l} m_i^k \leqslant m_{\max} \tag{1}$$

则AGV系统内部额定能耗E_0为

$$E_0=PT_k \tag{2}$$

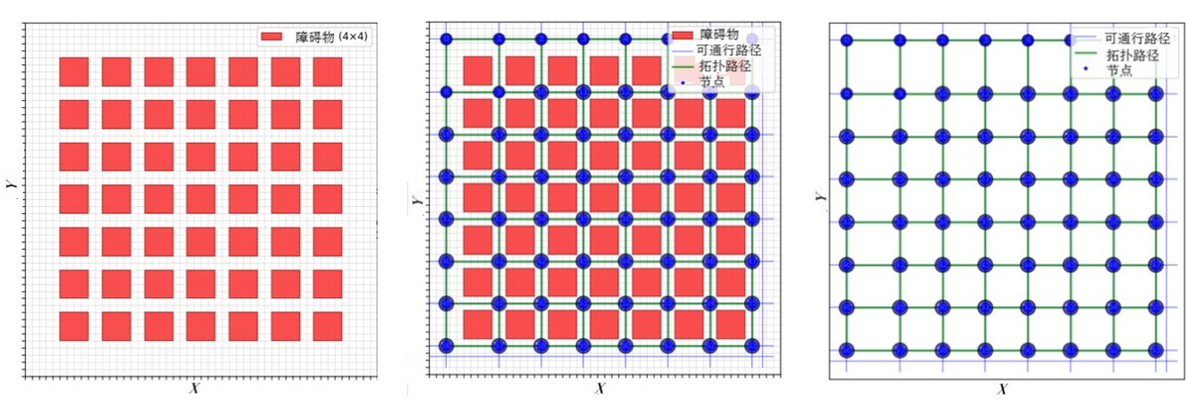
式中：P为AGV内部各系统的额定效率；T_k为第k辆AGV的运行时长。

在转弯过程中，由于本文研究的AGV是潜伏式AGV，要先减速至转弯速度，再加速至原速度且转弯半径为0，因此整个转弯过程消耗的能量可以理想化为匀速直线运动消耗的能量，加上加速运动额外消耗的能量，减去减速过程少消耗的能量，再加上全过程内部系统的额定能耗。

AGV在匀速直线运动状态下的驱动力方程为[7]：

$$F = C_r Mg \tag{3}$$

式中：F为AGV的驱动力；C_r表示滚动阻力系数；M表示AGV和运输货物的质量之和；g表示重力加速度。



a 栅格地图　　b 融合地图　　c 拓扑图

图1　仓库环境栅格地图

Fig. 1　Warehouse environmental grid map

AGV在加速状态下的驱动力方程为[7]:

$$F = C_r Mg + Ma \tag{4}$$

式中:a表示小车的加速度。

结合AGV行驶过程中的做功$W=Fd$(d为AGV行驶的位移),可得到AGV从起点到第一个任务点的匀速能耗为:

$$E_{k1} = \frac{\left(\left|x_1^k - x_0\right| + \left|y_1^k - y_0\right|\right)\left(\sum_{i=1}^{l} m_i^k + m_0\right)g}{\eta} \tag{5}$$

式中:E_{k1}表示第k辆AGV从起点到第一个任务点的匀速能耗;(x_1^k, y_1^k)表示第k辆AGV第一个任务点的坐标;η表示驱动电机功率因数。

AGV在任务之间匀速运动的能耗为:

$$E_{k2} = \frac{\sum_{i=1}^{l-1}\left(\left(\begin{matrix}\left|x_{i+1}^k - x_i^k\right| + \\ \left|y_{i+1}^k - y_i^k\right|\end{matrix}\right)\left(\sum_{j=1}^{l} m_j^k - \sum_{j=1}^{i} m_j^k\right)g\right)}{\eta} \tag{6}$$

式中:E_{k2}表示第k辆AGV在任务之间匀速运动的能耗;i表示第k辆AGV的第i个任务。

AGV完成最后一个任务返回起点的匀速能耗为:

$$E_{k3} = \frac{\left(\left|x_0 - x_l^k\right| + \left|y_0 - y_l^k\right|\right)m_0 g}{\eta} \tag{7}$$

式中:E_{k3}表示第k辆AGV完成最后一个任务返回起点的匀速能耗;(x_l^k, y_l^k)表示第k辆AGV最后一个任务点的坐标。

假设第k辆AGV在起点和第一个任务点之间、每两个任务点之间、最后一个任务点到起点之间均存在u_1、u_2、…、u_{l+1}个加减速点和v_1、v_2、…、v_{l+1}个转弯处,则AGV小车的加速距离d_1为:

$$d_1 = \frac{v^2}{2a} \tag{8}$$

式中:v为AGC匀速直线运动的速度。

AGV小车的减速距离d_2为:

$$d_2 = vt + \frac{1}{2}a_1 t^2 \tag{9}$$

式中:a_1代表AGV小车减速过程中的加速度。

则第k辆AGV因加速而增加的能耗E_e为:

$$E_e = \frac{v^2\left[\begin{matrix}\left(m_0 + \sum_{i=1}^{l} m_i^k\right)\left(u_1 + v_1\right) + \\ \sum_{i=1}^{l}\left(m_0 + \sum_{j=1}^{l} m_j^k - \sum_{j=1}^{i} m_j^k\right)\left(u_{i+1} + v_{i+1}\right)\end{matrix}\right]}{2\eta} \tag{10}$$

减速状态下,由于驱动电机的输出功率会迅速降低到接近于0,且可能会存在制动力,因此本文假设减速过程的耗能和匀速时相同,则第k辆AGV原地转弯过程中旋转消耗的能量为:

$$E_s = e_s\left[\left(m_0 + \sum_{i=1}^{l} m_i^k\right)v_1 + \sum_{i=1}^{l}\left(m_0 + \sum_{j=1}^{l} m_j^k - \sum_{j=1}^{i} m_j^k\right)v_{i+1}\right] \tag{11}$$

式中:E_s表示第k辆AGV原地旋转消耗的能量;e_s为旋转单位质量货物θ角度时消耗的能量;v_1表示起点到第一个任务点的转弯数量;v_{i+1}表示第i个任务点到第i+1个任务点的转弯数量,当$i=l$时表示最后一个任务点到起点的转弯数量。

假设AGV每次到达任务点作业时长为t_1、减速时长为t_2、转弯时长为t_3、加速时长为t_4,则第k辆AGV的匀速运动时长t_k为

$$t_k = \frac{\left(\begin{matrix}\left|x_0 - x_1\right| + \left|y_0 - y_1\right| + \\ \sum_{i=1}^{l-1}\left(\left|x_{i-1} - x_i\right| + \left|y_{i-1} - y_i\right|\right) - \\ \left(d_1 + d_2\right)\sum_{i=1}^{l-1}\left(u_i + v_i\right)\end{matrix}\right)}{v} \tag{12}$$

第k辆AGV运动总时长T_k为:

$$T_k = l \times t_1 + t_2\sum_{i=1}^{l+1}\left(u_i + v_i\right) + t_3\sum_{i=1}^{l+1} v_i + t_4\sum_{i=1}^{l+1}\left(u_i + v_i\right) + t_k \tag{13}$$

第k辆AGV的总能耗E_k为:

$$E_k = E_{k1} + E_{k2} + E_{k3} + E_e + E_s + E_0 \tag{14}$$

所有小车的总能耗为:

$$E_{sum} = \sum_{k=1}^{m} E_k \tag{15}$$

1.3 路径规划

AGV在交叉路口经常碰到路径冲突问题,本文使用了一种基于时间窗(time window)的调度优化方法,通过为每辆AGV分配路径占用时间片,并结合全局优化算法,最小化所有AGV的总通行时间,确保无路径冲突,避免死锁和碰撞,其流程图如图2[8]所示。

图2中的算法设计主要分为3个层面,第一是将路径资源进行划分,将交叉路口区域划分为路径资源R_j;第二是对时间窗进行分配,采用启发式算法为每辆AGV分配时间窗,确保无冲突;第三是全局优化,使用遗传算法(GA)调整时间窗分配,最小化总通行时间T。

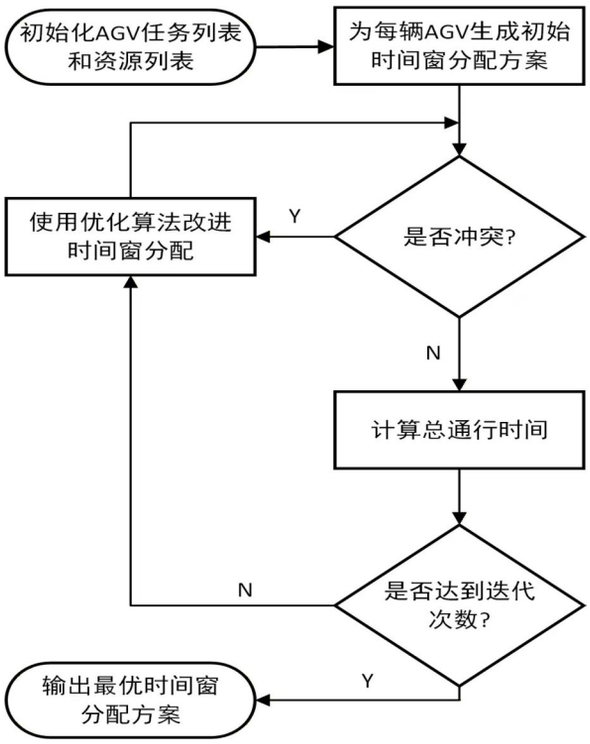


图2　时间窗算法流程

Fig. 2　Flow of the time window algorithm

根据图2的流程图，将交叉路口定义为路径资源R_j，每个路径资源的时间窗为$W_j=[t_{j,\text{start}},t_{j,\text{end}}]$，AGV小车$V_i$在路径资源$R_j$上占用时间窗为$W_{i,j}=[t_{i,j,\text{start}},t_{i,j,\text{end}}]$。为保证AGV小车顺利通过各交叉路口，路径资源需同时满足以下3个约束条件：

1）无冲突约束，对于同一路径资源R_j，任意两辆AGV的时间窗$W_{i,j}$和$W_{k,j}$不得重叠；

2）顺序约束，AGV必须按照任务顺序依次通过路径资源；

3）时间连续性约束，AGV在相邻路径资源上的时间窗满足时间连续性。

根据上述规则建立数学模型为[9]：

$$\min T = \max\{t_1,t_2,\cdots,t_m\} \tag{16}$$

约束条件为：

$$\begin{cases} t_{i,j,\text{start}} \geqslant t_{i,j-1,\text{start}} \\ t_{i,j,\text{start}} \geqslant t_{k,j,\text{end}} \ \text{ or } \ t_{k,j,\text{start}} \geqslant t_{i,j,\text{end}} \\ t_i = \sum_j^K (t_{i,j,\text{end}} - t_{i,j,\text{start}}) \end{cases}$$

式中：K为全局路口总数。

为了验证时间窗调度算法的效果，本文选取3组不同的AGV数量和路口数量，通过随机生成各AGV的起点和终点坐标，对时间窗法和固定优先级法进行了对照实验，结果如表1所示。

表1　时间窗对照实验结果

Table 1　Experimental results of time window comparison

使用车辆/部	路口	平均等待时长/s	
		固定优先级	时间窗
10	4	15.2	5.7
50	20	25.6	8.7
100	30	45.3	12.1

表1中的实验初始参数为AGV运行速度1格/s，时间步长1 s。由表1可知，时间窗法的3组对照试验平均等待时长相比较固定优先级的分别减少了62.5%、66.0%、73.3%。实验结果表明采用时间窗法可以大幅度减少路口等待时长，提升路口通行效率。

2　NSGA-Ⅲ算法调度方案

2.1　算法流程

NSGA-Ⅲ算法作为一种改进的非支配排序遗传算法，在解决多目标优化问题方面具有显著优势。通过结合参考点机制，该算法能够更有效地引导种群进化，从而在复杂的决策空间中寻找到多个优秀的非支配解[10]。

本研究将NSGA-Ⅲ算法应用于多AGV任务分配与路径规划问题，充分考虑了AGV系统的能耗、运行时间和路径冲突等多个目标，通过构建相应的数学模型（式(5)~(14)）和目标函数（式(15)），使得算法能够更准确地评估不同路径和任务分配方案下的能耗情况。此外，本研究还针对AGV任务分配问题的特殊性，对于标准遗传算法的二进制编码不适用于AGV的调度问题，设计了一种能够自然表达任务分配方案且便于遗传操作的结构化编码方式。该方式采用双层混合编码：第一层采用整数编码，表示任务到AGV的分配关系（每个基因位代表任务，值代表分配的AGV编号）；第二层采用排列编码，表示每台AGV内部的任务执行顺序。这种编码方式清晰分离了“分配决策”与“排序决策”，符合调度问题的决策逻辑。另外，通过引入基于时间窗的调度策略，有效解决了AGV在交叉路口的路径冲突问题；同时，为减少迭代次数，在NSGA-Ⅲ算法中引入自适应交叉和变异概率。

改进后的NSGA-III算法流程如图3所示，具体步骤如下：

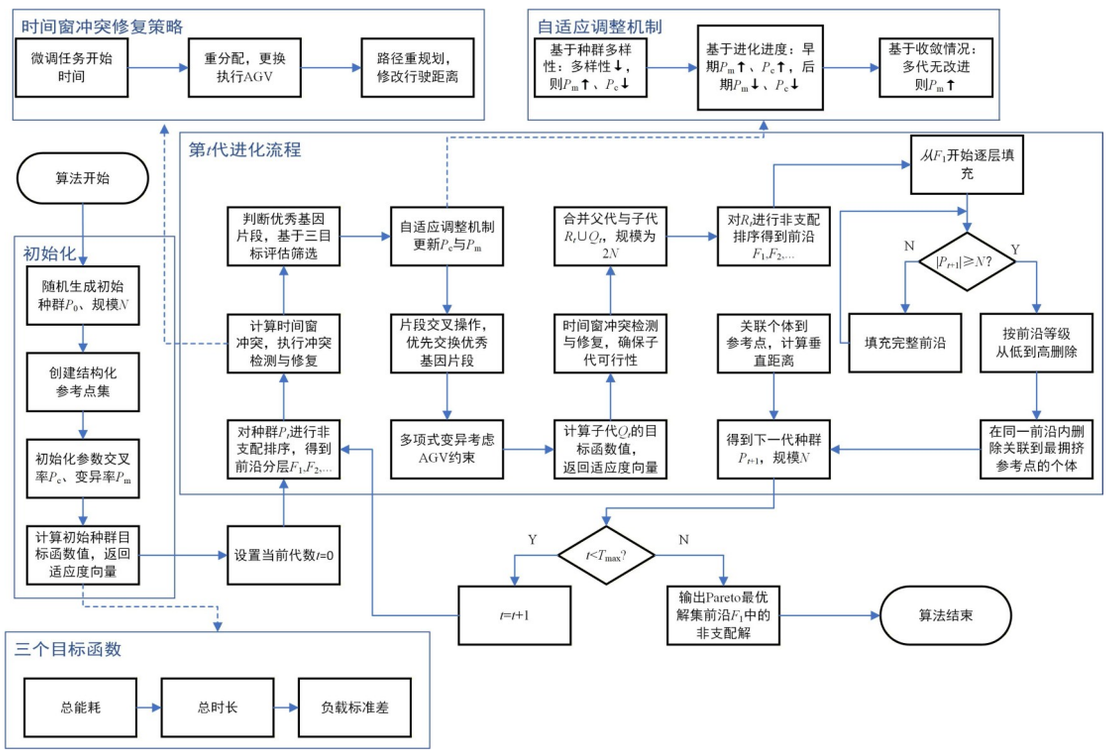


图 3　NSGA-Ⅲ算法流程图

Fig 3　Flowchart of the NSGA-III algorithm

1) 初始化种群,采用拉丁超立方采样生成初始种群P_0,确保解空间的均匀分布。

2) 进行混合整数编码(路径规划为整数序列,速度参数为实数)。

3) 非支配排序,对种群P_t(第t代种群,为初始种群P_0迭代t次所得)进行Pareto分层,计算个体间的支配关系,生成非支配前沿F_1、F_2、……。

4) 按前沿层级逐层填充下一代候选池,直至个体数超过种群容量N。

5) 采用Das and Dennis系统化方法,在归一化超平面上生成结构化参考点集R,再对目标值进行归一化。归一化公式为[11]:

$$f_i' = \frac{f_i - z_i^{\min}}{z_i^{\max} - z_i^{\min}}, i \in \{1,2,3\} \tag{17}$$

式中:$z_i^{\max}$、$z_i^{\min}$分别为当前种群各目标的最大值与最小值;f_i代表某个个体第i个目标函数,分别为总能耗、总时长和负载标准差

6) 将个体映射至参考点,计算个体(即每个候选解)与参考线的垂直距离,并选择最近的参考点。

7) 统计每个参考点关联的个体数,优先选择关联较少的参考点区域的解。

8) 若候选解数量$|P_{t+1}|<N$,按低层级前沿个体(优先F_1)、参考点关联密度低的区域个体进行填充,然后使用自适应交叉变异算子(SBX、多项式变异)生成的子代种群Q_t;若候选解数量$|P_{t+1}|\geq N$,按前沿等级从低到高删除,即先保留F_1,再从最低等级前沿开始,在同一前沿内删除关联到最拥挤参考点的个体(参考点关联个体数最多的区域);

9) 循环执行非支配排序、选择、交叉变异等操作,直至满足最大迭代次数T_{max}[12]。

2.2　交叉变异改进

假设存在10个任务,任务序列如图4所示。为适应本问题的特点,通过图5中的映射表(4↔7,5↔2,6↔1,7↔4)改进了遗传算法的交叉算子,修复非交叉段的冲突元素,使得父代个体$P1$=$\{1,2,3,4,5,6,7,8,9,10\}$和图4中的任务序列$P2$=$\{9,10,5,7,2,1,4,6,3,8\}$经交叉段交换后,形成子代$C1$=$\{9,10,2,4,5,6,7,1,3,8\}$、$C2$=$\{6,5,3,7,2,1,4,8,9,10\}$。这种通过建立映射关系并按照映射表改变任务编号,使得交叉之后生成的子代不会出现任务重复完成或任务遗漏的问题,既避免了原本就存在的任务编号与交换来的任务编号相冲突的问题,又确保任务调度序列的完整性和唯一性,从而保留了优秀的任务序列



图4　任务基因序列

Fig. 4　Task gene sequences

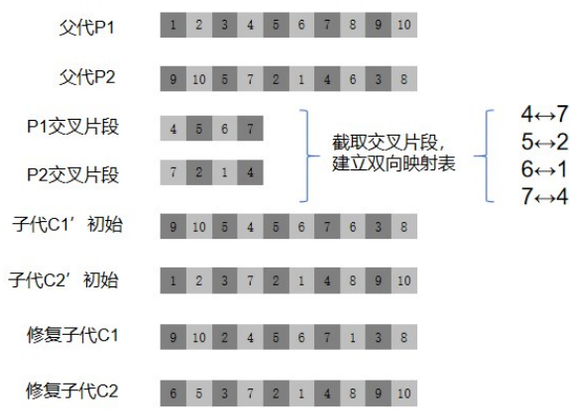


图5　交叉算子

Fig. 5　Cross operator

片段。

为平衡多目标优化过程中全局探索与局部开发的矛盾，本文提出了一种基于参考点分布的自适应交叉与变异概率调整机制。该机制通过动态感知种群多样性状态，自适应调整遗传操作，从而在算法前期增强全局搜索能力，在后期聚焦局部开发效率。自适应调整机制的有关公式定义如下：

1）平均拥挤度：

$$C_{\text{avg}}(t)=\frac{1}{W}\sum_{i=1}^{W}C_i \tag{18}$$

式中：$C_{\text{avg}}(t)$表示平均拥挤度；W为参考点总数；C_i为第i个参考点关联的个体数量。

2）历史最大拥挤度：

$$C_{\max}(t)=\max\left(C_{\max}(t-1),C_{\text{avg}}(t)\right) \tag{19}$$

式中：$C_{\max}(t)$、$C_{\max}(t-1)$分别表示第t代、第$t-1$代历史最大拥挤度。

3）多样性比率：

$$d(t)=\frac{C_{\text{avg}}(t)}{C_{\max}(t)} \tag{20}$$

式中：$d(t)$表示第t代多样性比率。

4）交叉概率：

$$P_{\text{c}}(t)=P_{\text{c max}}-\left(P_{\text{c max}}-P_{\text{c min}}\right)\times\left(1-d(t)\right) \tag{21}$$

式中：$P_{\text{c}}(t)$为交叉概率；$P_{\text{c max}}$为最大交叉概率；$P_{\text{c min}}$为最小交叉概率。

5）变异概率：

$$P_{\text{m}}(t)=P_{\text{m min}}+\left(P_{\text{m max}}-P_{\text{m min}}\right)\times\left(1-d(t)\right) \tag{22}$$

式中：$P_{\text{m}}(t)$为变异概率；$P_{\text{m max}}$为最大变异概率；$P_{\text{m min}}$为最小变异概率。

具体的实现步骤为：首先通过统计每个参考点关联的个体数量来反映种群的分布情况；其次，当某些参考点关联的个体过多时，说明种群在这些区域过于集中，需要增强变异概率以增强多样性，当参考点关联的个体数量分布均匀，说明种群多样性好，可以增加交叉概率以促进收敛；最后，根据具体的需要调整概率范围和计算方法。实现步骤的伪代码如下：

Algorithm1：NSGA-Ⅲ

Input：种群大小N，参考点集R，最大进化代数T

Output：近似Pareto最优解集

1：初始化种群P_0，生成参考点集R

2：计算P_0的参考点关联，得到$C_{\text{avg}}(0)$，$C_{\max}=C_{\text{avg}}(0)$

3：for t = 1 to T，do

4：选择父代个体，基于$P_c(t)$和$P_m(t)$通过分段交叉和多项式变异生成子代Q

5：合并种群$R_t=P_t\cup Q_t$

6：对R_t进行非支配排序，归一化目标空间，关联参考点$C_{\text{avg}}(t)$

7：计算参考点拥挤度$C_{\text{avg}}(t)$，$C_{\max}=\max(C_{\max},C_{\text{avg}}(t))$

8：多样性比率计算$d(t)=C_{\text{avg}}(t)/C_{\max}$

9：调整概率：

$P_c(t)=P_{cmax}-(P_{cmax}-P_{cmin})*(1-d(t))$

$P_m(t)=P_{mmin}+(P_{mmax}-P_{mmin})*(1-d(t))$

10：选择新一代P_{t+1}

11：end for

3　仿真实验与分析

3.1　交叉和变异概率仿真实验

为了验证本文提出的基于改进的NSGA-Ⅲ算法在多AGV协同调度上的效果，利用传统的NSGA-III算法及其改进算法，以及NSGA-II算法进行了2组不同的交叉和变异概率仿真实验，实验条件如下：AGV在50×50的栅格地图上运输货物，并随机生成30个任务点，每个任务点货物质量为100~200 kg，AGV自重215 kg、最大负载600 kg、额定运行速度2 m/s、旋转速度150°/s。3种算法交叉与变异参数如表2所示。

按表2的实验参数和前面的实验条件，当种群规模为100，随机初始化30个任务时，在MATLAB软件中编写程序代码，并使用orgin软件

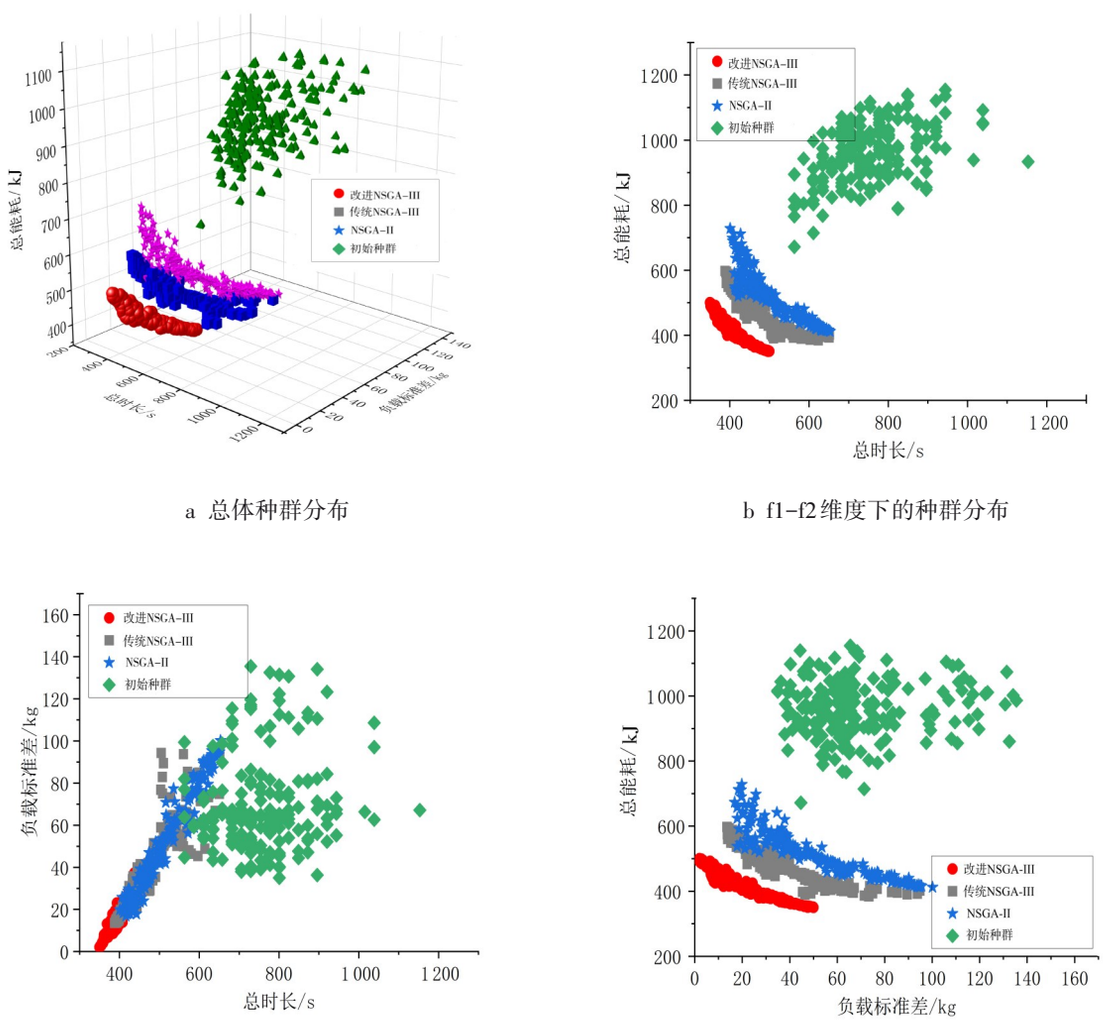
绘制初始种群与各算法最终种群的分布,以及不同目标函数维度下的种群映射分布,结果如图 6 所示。

由图 6a 可知,与初始种群相比,3 种算法的最终种群在三维目标空间中显著聚集于左下角并勾勒出前沿边界,这一空间分布特征印证了该解集的非支配性,即任意一个 Pareto 解均未被其他解所全面超越(即无法在保持其他目标不恶化的前提下,于任一目标上获得进一步改善),该前沿面为决策者提供了一组处于最佳权衡关系的候选方案集合;相比于 NSGA-Ⅱ和传统 NSGA-Ⅲ算法,改进后的 NSGA-Ⅲ算法所求得的 Pareto 最优解集(球形散点)的位置更靠近左下方,更具有竞争力。由图 6b 可知,总能耗与总时长呈现反比的

表 2　3 种算法实验参数

Table 2　Experimental parameters of the three algorithms

算法类型	交叉概率		变异概率	
	最大交叉概率	最小交叉概率	最大变异概率	最小变异概率
改进 NSGA-Ⅲ算法	0. 90	0. 60	0. 10	0. 01
传统 NSGA-Ⅲ算法	0. 85		0. 05	
NSGA-Ⅱ算法	0. 80		0. 10	



a 总体种群分布　　b f1-f2 维度下的种群分布

c f1-f3 维度下的种群分布　　d f2-f3 维度下的种群分布

图 6　总体种群分布与不同维度下的种群分布

Fig. 6　Overall population distribution and population distribution in different dimensions

趋势，这是因为当AGV任务分配较均匀时，最后完成任务的AGV所用时间相较于任务分配不均匀的最后完成任务的AGV所用时间较长；由图6c可知，负载标准差与总时长呈现正比趋势，这是因为当任务分配均匀时，各AGV完成任务时间比较接近，最后完成任务的AGV所用时间较短；由图6d可知，总能耗与负载标准差呈现反比的趋势，这是因为当AGV采用“就近执行”策略时，在它执行完一个任务后会就近执行距离较近的任务，从而节约能耗，但这会导致与任务较少的AGV的负载质量差值增大。

3.2　稳定性比较

本文以30个任务对上面3种算法进行10次300代迭代计算，获得平均值如图7所示。由图7可知：改进后的NSGA-Ⅲ算法获得的目标函数值最低，收敛速度更快，总能耗、总时长、负载标准差分别大约在110、100、130代趋于稳定，平均收敛速度比对照组（传统NSGA-III和NSGA-II）分别提升约26.7%、33.3%、23.5%（对照组关于总能耗、总时长、负载标准差趋于稳定的最小迭代次数分别约为150、150、170）。因此，相比于其他两种算法，改进后的NSGA-Ⅲ算法收敛速度更快、稳定性更好。

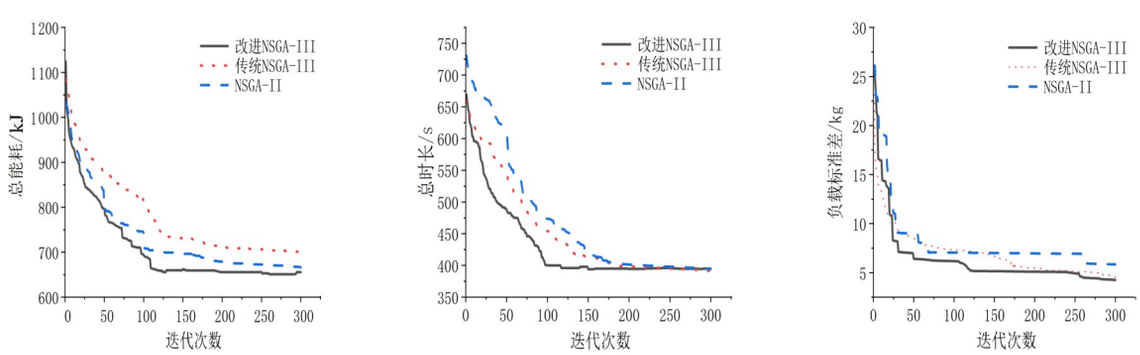
3.3　适应性比较

为验证算法的适应性，采用和上文30个任务相同的基本参数，分别验证了在10、20、40个任务时3种算法迭代300次时，总能耗、总时长、负载标准差的平均最小值如表3所示。由表3可知：在不同任务数量下，改进后的NSGA-Ⅲ算法的3个指标均为最小，说明该算法在不同任务数量下都具有良好的迭代效果。

3.4　IGD、HV指标分析

IGD和HV是多目标优化领域中用于评估算法性能的2个最重要、最常用的性能指标。在多目标问题上，IGD越小，算法的综合性越好；HV值越大，算法的综合性也越好。为验证改进的NSGA-III算法性能，本文从4~40个任务中，每间隔4个任务设计一组试验，共设计10组，结果如表4、表5所示。由表4、表5可以看出，改进后的NSGA-III算法求得的性能指标均优于NSGA-III与NSGA-II算法。显然，改进的NSGA-III算法使种群的收敛性和多样性得到了提高。

为验证性能提升的统计显著性，对表4、表5中的数据采用Wilcoxon符号秩检验，并设置显著性水平α=0.05，各指标统计结果如表6所示，检验结果如表7所示。



a 平均总能耗迭代曲线　　b 平均总时长迭代曲线　　c 负载标准差迭代曲线

图7　3种算法函数目标值平均收敛曲线

Fig. 7　Average convergence curve of function objective value of three algorithms

表3　其他任务数量函数目标值

Table 3　Number function target value of other tasks

算法	10个任务			20个任务			40个任务		
	总能耗/J	总时长/s	负载标准差/kg	总能耗/J	总时长/s	负载标准差/kg	总能耗/J	总时长/s	负载标准差/kg
NSGA-Ⅱ	273 215	413	117.78	459 820	470	7.32	989 850	436	9.28
传统NSGA-Ⅲ	269 215	357	117.85	429 465	440	6.56	958 800	397	8.15
改进NSGA-Ⅲ	255 289	342	117.78	402 979	378	5.25	905 594	378	7.45

表4　IGD测试结果
Table 4　IGD test results

组别	改进NSGA-III	传统NSGA-III	NSGA-II	组别	改进NSGA-III	传统NSGA-III	NSGA-II
mk01	10.415	21.194	61.581	mk06	59.123	103.955	350.968
mk02	18.493	32.320	69.284	mk07	76.802	137.965	332.141
mk03	19.103	47.001	65.833	mk08	66.599	121.357	620.886
mk04	24.730	85.215	76.448	mk09	71.305	237.774	772.541
mk05	44.108	116.059	268.619	mk10	186.461	336.126	1028.234

表5　HV测试结果
Table 5　HV test results

组别	改进NSGA-III	传统NSGA-III	NSGA-II	组别	改进NSGA-III	传统NSGA-III	NSGA-II
mk01	5.966×10^{-3}	3.021×10^{-3}	8.01×10^{-4}	mk06	7.57×10^{-4}	5.50×10^{-4}	2.18×10^{-4}
mk02	6.005×10^{-3}	2.310×10^{-3}	6.82×10^{-4}	mk07	8.87×10^{-4}	1.68×10^{-4}	1.00×10^{-4}
mk03	3.835×10^{-3}	1.536×10^{-3}	4.90×10^{-4}	mk08	3.93×10^{-4}	1.66×10^{-4}	5.00×10^{-5}
mk04	1.837×10^{-3}	6.870×10^{-4}	8.19×10^{-4}	mk09	2.97×10^{-4}	1.15×10^{-4}	3.90×10^{-5}
mk05	1.370×10^{-3}	5.090×10^{-4}	1.87×10^{-4}	mk10	1.25×10^{-4}	3.90×10^{-5}	1.20×10^{-5}

表6　Wilcoxon符号秩检验描述性统计量
Table 6　Wilcoxon signed-rank test descriptive statistics

性能指标	算法名称	N	平均值	标准差	最小值	最大值
IGD	改进NSGA-III	10	57.713 9	51.364 87	10.42	186.46
	传统NSGA-III	10	123.896 6	97.191 76	21.19	336.13
	NSGA-II	10	364.653 5	338.999 43	61.58	1 028.23
HV	改进NSGA-III	10	2.15×10^{-3}	2.29×10^{-3}	1.25×10^{-4}	5.966×10^{-3}
	传统NSGA-III	10	9.10×10^{-4}	1.04×10^{-3}	3.90×10^{-5}	3.900×10^{-5}
	NSGA-II	10	3.40×10^{-4}	3.26×10^{-4}	1.20×10^{-5}	8.010×10^{-4}

表7　Wilcoxon符号秩检验结果
Table 7　Wilcoxon signed-rank test results

性能指标	IGD		HV	
	改进NSGA-III-传统NSGA-III	改进NSGA-III-NSGA-II	改进NSGA-III-传统NSGA-III	改进NSGA-III-NSGA-II
Z	−2.803[a]	−2.803[a]	−2.803[a]	−2.803[a]
P(渐进显著性(双尾))	0.005	0.005	0.005	0.005

注：a表示基于正秩。

从表6可以看出，改进后的NSGA-III算法的IGD平均值、标准差、最大和最小值均小于对比算法，HV平均值、标准差、最大和最小值均大于对比算法。从表7可知，由于改进后的NSGA-III算法在10组对照实验中均取得了最优的IGD和HV值，故检测结果的Z值和P值均为−2.803和0.005，说明改进后的NSGA-III算法使种群收敛性和多样性的提高符合统计显著性。

4　结论

针对标准NSGA-III算法在复杂多目标优化中存在的不足，提出了一种改进的自适应分段交叉与变异的NSGA-III算法。通过自适应遗传算子调整机制，无需人工预先设置参数，能够根据

种群的进化状态智能地调整交叉和变异概率，有效平衡了全局探索和局部开发过程，加快了收敛速度；通过引入一种新颖的分段交叉操作，充分考虑了解的空间结构，并对决策变量分段实施差异化交叉，从而生成了更多高质量、多样化的候选解，显著提升了对帕累托前沿的表征能力；最后通过仿真实验，表明本文所提出的改进的NSGA-III算法相比于改进前的NSGA-Ⅲ算法，不仅提高了的搜索广度和收敛速度，在解决多AGV协同调度上还拥有更高的稳定性和迭代效率。但对于更合理的仓库布置以及更具迭代效率的自适应交叉与变异函数设计仍然需要深入研究。

参考文献：

[1] 张中伟，高增恩，王菁锐，等. 分布式AGV调度研究综述与发展趋势分析[J]. 制造技术与机床，2024(11)：55-61.

[2] 杨凯. AGV发展现状及关键技术综述[J]. 机电工程技术，2025，54(13)：24-30.

[3] 颜伟，黄冠鹏，高玉萍，等. 仓库多AGV路径冲突问题研究综述[J]. 科学技术与工程，2025，25(18)：7465-7474.

[4] 刘燕丽. 基于冲突的NP难问题完备算法的研究[D]. 武汉：华中科技大学，2019.

[5] 王健. 危险化学品车辆路径问题及算法研究[D]. 北京：北京化工大学，2016.

[6] 徐晗，金隼，罗磊，等. 基于拓扑栅格建模的AGV路径规划算法优化[J]. 计算机工程与设计，2022，43(1)：101-109.

[7] 王占山，陈晓华，赵瑞，等. AGV机器人越障与爬坡驱动力计算分析[J]. 装备制造技术，2022(10)：30-33.

[8] 耿焕同，戴中斌，王天雷，等. 基于参考点选择策略的改进型NSGA-III算法[J]. 模式识别与人工智能，2020，33(3)：191-201.

[9] ZARE-REISABADI E，HAMID MIRMOHAMMADI S. Site dependent vehicle routing problem with soft time window：Modeling and solution approach[J]. Computers & Industrial Engineering，2015，90：177-185.

[10] XU X H，CHENG D，WANG D，et al. An improved NSGA-III with a comprehensive adaptive penalty scheme for many-objective optimization[J]. Symmetry，2024，16(10)：1289.

[11] 陈先亮，黄元君，范勤勤. 基于多模态多目标进化算法的无人机三维路径规划[J]. 火力与指挥控制，2023，48(11)：32-39.

[12] 毕晓君，王朝. 一种基于参考点约束支配的NSGA-III算法[J]. 控制与决策，2019，34(2)：369-376.

Multi-AGV Path Planning Based on an Improved NSGA-Ⅲ Algorithm

CHEN Yongxiang，WANG Jianping，QIU Ziqi，DAI Zhoujie，SUN Jie

(School of Mechanical and Automotive Engineering，Anhui Polytechnic University，Wuhu Anhui　241000，China)

Abstract：To address the limitations of conventional topological mapping methods for multi-AGV collaborative scheduling in warehouses—particularly the inability to resolve travel conflicts and the computational overhead of separate path planning for each AGV in grid-based approaches—we propose an innovative solution. This method employs dual-channel modeling in grid maps and establishes topological point mappings at conflict intersections for path planning, eliminating the need for single-agent routing algorithms. Algorithmically, we overcome the challenges of balancing breadth and depth during iterative processes and single-objective optimization by introducing a NSGA-III-based multi-objective optimization algorithm. The algorithm incorporates adaptive genetic and crossover probabilities,adopts a segmented crossover scheme, and retains superior parent segments to enhance the algorithm′s convergence speed. Three objective functions were defined: energy consumption, operational time, and load standard deviation. Simulation experiments on multi-AGV scheduling in grid maps demonstrated that the improved NSGA-III algorithm achieves better optimization results with smaller convergence algebra distributions. The average convergence speeds of the three objective functions improved by 26.7%,33.3%, and 23.5% respectively compared to pre-improvement benchmarks.

Keywords：AGV scheduling；NSGA-III；path planning；adaptive crossover；segment crossover；multi-objective optimization

(责任编辑：李华云)