

非支配排序粒子群遗传算法解决 车辆位置路由问题

刘琼昕^{1,2}, 王甜甜², 王亚男²

(1. 北京理工大学 北京市海量语言信息处理与云计算应用工程技术研究中心, 北京 100081; 2. 北京理工大学 计算机学院, 北京 100081)

摘 要:提出一种混合全局局部搜索的非支配排序粒子群遗传算法,该算法能够有效解决车辆位置路由问题。全局搜索使用粒子群和遗传算法以提高收敛速度,使用第三代非支配排序遗传算法挑选种群下一代个体以保留种群多样性。局部搜索策略针对优质和次优个体进行优化,以提高得到更优解的概率,对种群中后 1/12 个体打乱用户顺序,提高种群质量。使用开放标准数据集将本文算法与基准算法对比,结果表明本文算法在种群质量、多样性上均更优,能够为车辆位置路由问题提供有效的解决方案。

关键词:计算机应用;车辆位置路由问题;第三代非支配排序遗传算法;粒子群算法;遗传算法

中图分类号:TP391 **文献标志码:**A **文章编号:**1671-5497(2025)07-2464-11

DOI:10.13229/j.cnki.jdxbgxb.20231086

Non-dominated sorted particle swarm genetic algorithm to solve vehicle location routing problems

LIU Qiong-xin^{1,2}, WANG Tian-tian², WANG Ya-nan²

(1. Beijing Engineering Applications Research Center on High Volume Language Information Processing and Cloud Computing, Beijing Institute of Technology, Beijing 100081, China; 2. School of Computer Science and Technology, Beijing Institute of Technology, Beijing 100081, China)

Abstract: A non-dominated sorted particle swarm genetic algorithm with hybrid global-local search is proposed, by which the vehicle location routing problem can be effectively solved. Both particle swarm optimization and genetic algorithm operators are utilized in the global search to improve convergence speed. The non-dominated sorting genetic algorithm III is employed so that population diversity is maintained. The local search strategy is applied separately to superior and inferior individuals, by which the probability of obtaining better solutions is increased. Additionally, the user orders of the last 1/12 individuals in the population are shuffled so that the overall population quality is enhanced. The proposed algorithm is compared with benchmark algorithms by using the open standard dataset, and it is demonstrated that population quality and diversity are better provided, and an effective solution to the vehicle location routing problem can be supplied.

收稿日期:2023-12-26.

基金项目:国家自然科学基金项目(62072039).

作者简介:刘琼昕(1972-),女,副教授,博士.研究方向:人工智能.E-mail: summer@bit.edu.cn

Key words: computer application; vehicle location routing problem; non-dominated sorting genetic algorithm III; particle swarm optimization algorithm; genetic algorithm

0 引言

车辆位置路由问题(Location-routing problem, LRP)是车辆任务调度领域的经典问题,可以拆分为位置分配问题(Location-allocation problem, LAP)和车辆路径问题(Vehicles routing problem, VRP)^[1]。Laporte等^[2]在1986年描述了经典的LRP,该问题需要从潜在设施位置中选择最优位置,并规划从仓库出发对用户服务的车辆路线。

针对LRP,Barreto等^[3]对LAP和VRP分开进行解决。然而,Salhi等^[4]指出,同时考虑LAP和VRP通常会得到更优解。Perl等^[5]在此基础上同时分析LAP和VRP,使用启发式算法来解决LRP。元启发式算法是解决LRP的途径之一。Huang等^[6]使用蚁群优化算法解决无人机配送服务的VRP。Ma等^[7]使用遗传算法解决自动驾驶汽车的LRP。胡圣邦等^[8]考虑安全目标和时间目标双重属性,使用改进的蚁群算法选择运输路线。杜玫谛等^[9]构建车辆路径优化模型,利用改进蚁群算法进行模型求解。

进化多目标优化(Evolutionary multi-objective-optimization, EMO)也是解决LRP的有效途径之一。目前,多目标优化在方法和应用领域均取得了大量的成果,如第二代强度Pareto进化算法(Strength Pareto evolutionary algorithm 2, SPEA2)^[10]、第二代非支配排序遗传算法(Non-dominated sorting genetic algorithm II, NSGA-II)^[11]和第三代非支配排序遗传算法(Non-dominated sorting genetic algorithm III, NSGA-III)^[12]。Rabbani等^[13]首先将NSGA-II应用于双目标的LRP中。杨红波等^[14]提出了齿轮副宏观参数多目标优化方法,利用NSGA-II解决多个优化目标的冲突。Hernandez等^[15]将NSGA-III应用于求解多目标优化问题,并取得了不错的效果。该类算法通过参考点将目标空间分解为子空间进行并行搜索,该类算法典型代表有TC-SEA^[16]、I-DBEA^[17]和RVEA^[18]。

虽然已有很多方法用于求解LRP,但很少有学者将两种元启发式算法和EMO算法结合来解

决LRP。考虑到粒子群算法(Particle swarm optimization, PSO)能够提高收敛效率,遗传算法(Genetic algorithm, GA)能够避免陷入局部最优^[19],而NSGA-III能够找到收敛良好且多样化的解^[20],本文将PSO、GA和NSGA-III三者结合,提出混合全局局部搜索的非支配排序粒子群遗传算法(Hybrid global-local search non-dominated sorting particle swarm optimization-genetic algorithm, HSNS-PSOGA)。针对种群多样性和收敛性差的问题,全局搜索策略主要采用PSO,选择机制为NSGA-III;针对种群多样性差和解的分布不均匀问题,对优质个体和次优个体分别进行反转、交换和插入操作,同时对种群中后1/12个体打乱用户顺序。实验结果表明,本文提出的HSNS-PSOGA在大多数情况下优于基准算法。

1 车辆位置路由问题分析

1.1 问题描述

针对本文研究的LRP,描述如下:给定候选仓库位置,每个仓库可以派遣车辆服务用户,其中车辆容量有限,车辆数目无限,另外已知用户位置和需求。要求从候选仓库中选择一定数量的仓库,该问题被称为LAP;确定从仓库出发的车辆数目及行驶路线,该问题被称为VRP。LRP中,减少车辆使用数量,可能需要将车辆分配给较远的用户,从而增加路由距离;而减小路由距离,可能会使车辆行驶成本较高的路径。因此,LRP是一个多目标优化问题,需要最优化的3个目标是车辆数量,车辆路由距离和车辆路由成本。

令 I 为候选仓库集合, J 为用户顶点集合, V 为顶点集合且 $V=I \cup J$, K 为车辆序列, $u_k=1$ 表示第 k 辆车被使用, $x_{i,j,k}=1$ 表示第 k 辆车的行驶路由覆盖点 i 到 j 的边, $d_{i,j}$ 为顶点 i 到 j 的路由距离, $c_{i,j}$ 为顶点 i 到 j 的路由代价。3个目标函数的数学化描述为:

$$f_1(x) = \sum_{k \in K} u_k \quad (1)$$

$$f_2(x) = \sum_{i \in I, j \in V, k \in K} d_{i,j} x_{i,j,k} \quad (2)$$

$$f_3(x) = \sum_{i \in I, j \in V, k \in K} c_{i,j} x_{i,j,k} \quad (3)$$

式中: $f_1(x)$ 为最小化车辆使用数量目标函数; $f_2(x)$ 为最小化车辆派送路线的总距离目标函数; $f_3(x)$ 为最小化车辆派送路线产生的总代价成本目标函数。

3 个优化目标存在竞争关系且重要性相当, 因此需要根据染色体计算个体在 3 个目标上的取值, 再考虑 Pareto 支配关系来判断解的优劣。

令 K_a 为被分配任务的车辆序列, $z_{j,k}=1$ 表示第 k 辆车被分配服务用户 j , q_i 为用户顶点需求, q 为车辆的货物负载量。

本文研究的车辆 LRP 有以下 4 个约束条件: 式(4)确保每个用户只被一辆车服务, 式(5)确保一辆车服务用户的总需求不超过其负载能力, 式(6)确保每辆车最多执行一次派送, 式(7)确保每条完整路由的起点和终点为同一仓库。

$$\sum_{k \in K_a} z_{j,k} = 1, \forall j \in J \quad (4)$$

$$\sum_{j \in J} q_j z_{j,k} \leq q, \forall k \in K_a \quad (5)$$

$$\sum_{i \in I} \sum_{j \in J} x_{i,j,k} = 1, \forall k \in K_a \quad (6)$$

$$\sum_{i \in I} \sum_{j \in J} x_{i,j,k} = \sum_{i' \in I} \sum_{j \in J} x_{i',j,k} = 1, \forall k \in K_a, i = i' \quad (7)$$

约束条件在 1.2 节染色体编码与解码中体现。

1.2 染色体编码与解码

针对染色体无法对应可行解的问题, 本文使用以下编码方式: 假设用户个数为 $|J|$, 仓库个数为 $|I|$, 令解向量的长度为 $|J|+|I|-1$, 解向量的每一维取值均为 1 到 $|J|+|I|-1$ 的整数。其中, 仓库从 0 开始编号, 用户从 $|I|$ 开始编号, 染色体第一维度默认为 0。

这种编码方式的含义为: 相邻两个仓库之间的用户由前一个仓库服务。若车辆服务用户的总需求超出车辆容量, 为了满足式(5)对应约束, 由当前仓库派遣新的车辆。由于每个仓库车辆数量没有限制, 可以保证仓库每次派遣车辆不同, 满足式(6)对应约束。

如图 1 所示, 假设共有 3 个仓库、8 个用户, 则仓库编号为 0~2, 用户编号为 3~10, 编号为 0 的仓库(0 号仓库)服务第 3、8 号用户, 1 号仓库服务第 7、4、9 号用户, 2 号仓库服务第 5、6、10 号用户。图 1 所示编码方案对应的解决方案如图 2 所示。

图 2 中, 由于第 7、4 号用户总需求超出了车

辆容量, 则派遣新的车辆进行服务。

解向量每一维取值不重复出现(详细方法见 2.1 节染色体归一化), 保证每个用户只被一个仓库服务, 满足式(4)对应约束条件。确定仓库服务的用户后, 根据车辆容量和用户需求进行派遣, 对应式(5)(6)。相邻两个仓库之间的用户由前一个仓库服务, 对应式(7)。

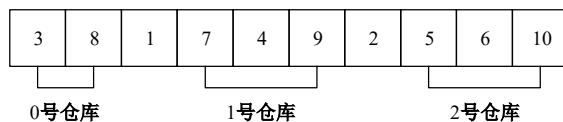


图 1 解向量示意图

Fig. 1 Solution vector schematic

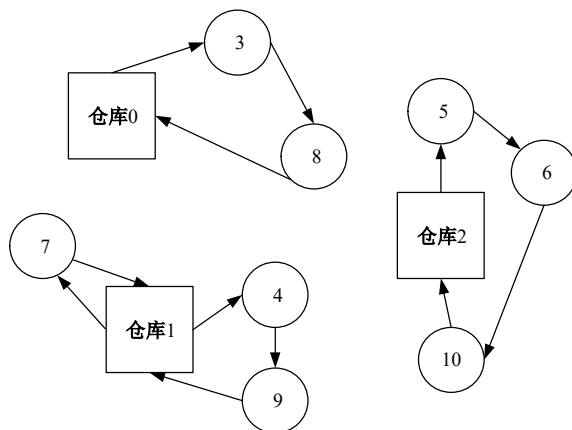


图 2 LRP 解决方案

Fig. 2 LRP solution

2 算法描述

PSO 算法能够提高收敛效率, 而 GA 有助于避免陷入局部最优^[19]。为了平衡种群多样性和收敛性, 本文算法流程见算法 1。

算法 1 HSNS-PSOGA

Input: 优化问题 MOP, 最大迭代次数 max_gen, 种群规模 N

Output: 结果集 EP

- 1: 初始化参考点 init_refpointsND();
- 2: 初始化粒子位置与速度 init_population();
- 3: while(gen < max_gen) do
- 4: Global Search();
- 5: Local Search();
- 6: gen = gen + 1;
- 7: end while

首先初始化参考点和种群, 接着进行全局搜索: 全局搜索中使用 PSO 算法, 同时结合 GA 的交叉变异算子对个体进行优化, 之后使用 NS-

GA-III选择个体,产生子代种群 P_{t+1} 。随后进行局部搜索,对 P_{t+1} 中的优质、次优个体进一步改进,再更新后1/12个体。

2.1 全局搜索策略

针对种群收敛性差的问题,全局搜索策略采用PSO算法^[21],首先初始化父代种群 P_t ,之后对种群中第 i 个粒子使用粒子群变化算子:

$$\begin{aligned} v_i(t+1) &= \omega v_i(t) + c_1 r_1 (x_{pbest_i} - x_i(t)) + \\ &\quad c_2 r_2 (x_{gbest} - x_i(t)) \end{aligned} \quad (8)$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (9)$$

式中: v_i 粒子 i 的速度; x_{pbest_i} 为粒子 i 的历史最优解; x_{gbest} 为整个种群的全局最优解; ω 为动态惯性权重; c_1 和 c_2 为加速因子; r_1 和 r_2 为 $[0,1]$ 的随机数; x_i 为粒子 i 的位置; t 为迭代轮数。

为了确保每个用户只被一辆车服务(满足式(4)约束),需要保证解向量每一维取值为唯一整数。因此,需要对更新后的粒子坐标进行处理,下面给出了染色体归一化过程:

假设粒子坐标 $X=(x_1, x_2, \dots, x_n)$,其中 $n=|I|+|J|-1$ 。

(1)设元组集合 $V=\{(x_i, i) | 1 \leq i \leq n\}$,每个元组包含维度值 x_i 和维度 i 。

(2)对集合 V 中元组按照 x_i 升序排列,得到 $V'=\{(x_{i_1}, i_1), (x_{i_2}, i_2), \dots, (x_{i_n}, i_n)\}$,其中 $x_{i_1} \leq x_{i_2} \leq \dots \leq x_{i_n}$ 。

(3)令 $D=\{(i_j, j) | 1 \leq j \leq n\}$,其中 j 表示排序后的序号。

(4)将第 j 小的维度 i_j 赋值为 j ,即 $x_{i_j}=j$ 。

根据染色体计算粒子在3个目标函数上的取值,若存在粒子 i 的解Pareto支配 x_{pbest_i} 或 x_{gbest} ,则进行更新。

对种群中个体使用PSO算法之后,使用GA的交叉和变异算子对临时子代种群 E_t 进行交叉变异操作,形成子代种群 Q_t 。对于交叉操作,生成随机数 p ,获取染色体 A 在位置 p 的值 a ,即 $A[p]=a$,在染色体 B 中找到值为 a 的位置 q ,并获取 A 在位置 q 的值 b ,即 $A[q]=b$,将 A 和 B 各自的 a 、 b 交换。对于变异操作,生成随机下标 p 、 q ,交换染色体 C 在 p 、 q 位置的元素。交叉和变异操作见图3。

为了解决种群多样性差的问题,使用NSGA-III对父子代合并后的 $2N$ 个体进行选择,具体选择过程见算法2的4~16行。NSGA-III通过

精英策略,能够保证种群中优质父代被保留;其基于均匀分布的参考点选择个体,能够保证解的多样性。全局搜索算法流程见算法2。

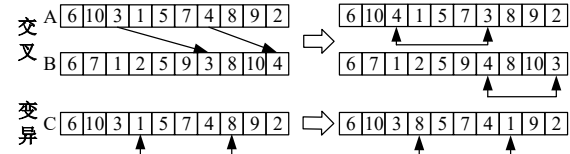


图3 交叉和变异操作

Fig. 3 Operation of crossing and compiling

算法2 GLOBAL SEARCH

Input:结构化参考点序列 B ,父代种群 P_t

Output:子代种群 P_{t+1}

- 1:对 P_t 使用PSO算法变化算子产生 E_t
- 2:对 E_t 使用GA变化算子产生 Q_t
- 3:将 Q_t 中每个新个体与原个体对比,如果新个体Pareto支配原个体,替换原个体
- 4: $R_t = P_t \cup Q_t$
- 5:对 R_t 进行非支配排序生成多个非支配层: $F_1, F_2, \dots$
- 6: $S_t = \emptyset, i=1$
- 7:repeat
- 8: $S_t = S_t \cup F_i, i=i+1$
- 9:until $|S_t| \geq N$
- 10:if $|S_t| = N$ then
- 11: $P_{t+1} = S_t$
- 12:else $P_{t+1} = \sum_{j=1}^{l-1} F_j$
- 13:对 S_t 的所有解在各个目标函数维度上自适应归一化
- 14:根据自适应归一化后的 S_t ,将每个个体与在超平面上距离最近的参考点相关联
- 15:根据参考点相关联的个体数量,挑选 $K=N-|P_{t+1}|$ 个个体加入 P_{t+1} 中
- 16:选择关联个体数量最少的参考点,从参考点相关联个体集合中选择属于 F_1 支配层个体,与当前全局最优个体对比,挑选更优解作为新的全局最优解
- 17:end if

2.2 局部搜索策略

本文对优势个体进行局部搜索,以提高获得更优解的概率;对种群中次优个体进行局部搜索,以减少Pareto支配解的个数,提高整个种群的质量并加快收敛;对后1/12个体打乱用户顺序,以增加种群多样性。局部搜索算法流程见算法3。

算法 3 LOCAL SEARCH

Input: 经过全局搜索得到的子代种群 P_{t+1} Output: 局部搜索过后的子代种群 P_{t+1} 1: 随机选择参考平面 B 中的一个参考点作为 λ 值2: 计算 P_{t+1} 中每个个体的聚合函数值3: $i=1$

4: repeat

5: 以无放回锦标赛方式选择一个个体 x , 使用三种局部搜索变化算子得到新个体 x' , $i=i+1$ 6: 判断新个体 x' 是否能够 Pareto 支配任意一个 P_{t+1} 中的个体, 如果是则替换该个体进入 P_{t+1} , 否则放弃该新个体7: 对 P_{t+1} 中的所有 Pareto 支配解 $\Sigma_{j=2}^l P_{t+1}$ 进行局部搜索, 使用 3 种变化算子得到新个体 x'' 8: 如果 x'' 属于非支配 Pareto 解, x'' 替换 x 进入 P_{t+1} , 否则放弃该新个体9: until $i=T_{\text{local}}$ 10: 对 P_{t+1} 中后 1/12 的个体进行用户顺序的打乱

11: 如果新解 Pareto 支配原解, 替换

文献[22]表明, 在局部搜索中, 基于聚合函数选择优质个体比基于 Pareto 排序更有效, 本文使用目标函数的加权和作为聚合函数, 公式为:

$$f^*(x) = \lambda_1 f_1^i(x) + \lambda_2 f_2^i(x) + \lambda_3 f_3^i(x),$$

$$\lambda_1 + \lambda_2 + \lambda_3 = 1 \quad (10)$$

式中: $\lambda_1, \lambda_2, \lambda_3$ 为参考平面上随机点在 3 个维度的取值; $f_1^i(x), f_2^i(x), f_3^i(x)$ 为个体 i 对应的 3 个目标函数值。

以式(10)作为衡量标准, 并使用无放回的锦标赛方法^[23]选择优质个体进行局部搜索。每次随机选择两个个体进行比较, 选择 $f^*(x)$ 值更小的个体进入优质种群。对优质种群采用反转、交换和插入 3 种变异操作: 反转是指反转个体向量中的一段; 交换是对个体向量中的两位编码进行位置交换; 插入是从个体向量中截取一段插入首位。上述过程重复 $N \times P$ 次, 其中 P 为对 N 个个体进行局部搜索的概率。如果新解 x' 能够 Pareto 支配种群中的任一个体 x'' , 则用 x' 代替 x'' 。通过对优质个体进行局部搜索, 能够以一定概率寻找到更加优质的个体, 解决种群收敛性差的问题。

除对优质个体进行局部搜索外, 本文还对除 F_1 层中的次优个体 (P_{t+1}/F_1) 进行局部搜索, 以提

高种群整体解的质量。对每个次优个体 y , 通过反转、交换和插入操作得到 y' 。若 P_{t+1}/F_1 中没有个体 Pareto 支配 y' , 则以 y' 替换 y 。对次优个体进行的局部搜索由于引入了 GA 的变化算子, 既能在一定程度上增加种群的多样性, 又可提高次优个体的质量。

对次优个体局部搜索后, 对种群后 1/12 个体进行用户顺序打乱, 具体做法为: 保持仓库编号不变, 将两个仓库编号之间的用户编号随机打乱顺序; 若更新后的解 Pareto 支配原解, 则进行替换。这种做法考虑到仓库选取和其服务的用户已经最优, 但车辆的行驶路线可能因为服务的用户顺序没有达到最优。通过打乱后 1/12 个体的用户顺序, 提高了种群多样性。

3 实验结果与分析

为证明 HSNS-PSOGA 的有效性, 设置了对比实验, 对比算法分别为 NSGA-II^[11]、NSGA-II+PSO^[24]、NSGA-III^[12]、NSGA-III+PSO^[25]。NSGA-II 和 NSGA-III 是多目标优化领域的经典算法, NSGA-II+PSO 和 NSGA-III+PSO 是将 EMO 算法和元启发式算法相结合的改进算法。同时, 本文设计了消融实验, 以证明对优质、次优个体局部搜索, 使用 PSO 算法和对后 1/12 个体打乱用户顺序的合理性。另外, 本文对用户顺序打乱比例进行了实验分析。实验在所有测试问题上均独立运行 30 次。

3.1 数据集

在本次实验中, 使用的数据集来源于 KBAM、KBSN、KPAM、KPSN、YLBAM、YLB-SN、YLPAM、YLPSN 八组数据集。每组数据集的每组数据均提供了用户的位置坐标、用户需求、待选择仓库的位置坐标、车辆容量, 另外有距离矩阵提供路由距离成本。本次实验从 KBSN 数据集中的 30 组数据中随机选择 10 组数据进行测试, 这 10 组数据中, 每个实例的用户个数取值为 8~100, 仓库个数取值为 2~10。通过实例的名称能够得到用户和仓库的数量, 如 Bar_32_5_X_NGas-kell67_32x5_1, 包含 32 个用户和 5 个仓库。本文使用 KBSN 提供的 10 个实例并结合随机生成的道路路由成本进行实验。

3.2 性能指标

为了更好地说明 HSNS-PSOGA 的有效性, 本文的对比指标除车辆使用数量、车辆路由距离

成本、车辆路由代价 3 个目标函数之外,还设计了 SM1(Spacing metrics 1)、SM2(Spacing metrics 2) 和 DM(Diversification metrics)指标。

SM1 能够衡量解的分布均匀性,其值越小代表均匀性越好,其计算公式如下:

$$SM1 = \sqrt{\frac{1}{N-1} \times \sum_{i=1}^N (\text{dis}_i - \overline{\text{dis}})^2} \quad (11)$$

式中: dis_i 为第 i 个解与 Pareto 最优解集中最接近它的解之间的欧氏距离; $\overline{\text{dis}}$ 为所有 dis_i 求和取平均; N 为种群中个体总数。

SM2 能够有效表示 Pareto 最优解在目标空间中的分布情况,取值越小越好,其计算公式见式(12)。

$$SM2 = \sqrt{\frac{1}{N-1} \times \sum (d_i - \bar{d})^2} \quad (12)$$

$$d_i = \min (|f_1^i(x) - f_1^j(x)| + |f_2^i(x) - f_2^j(x)| + |f_3^i(x) - f_3^j(x)|) i$$

$$(13)$$

式中: $i, j = 1, 2, \dots, N$; $\bar{d}$ 为所有 d_i 的平均值; f_1^i 、 f_2^i 、 f_3^i 为第 i 个个体的 3 个目标函数值; f_1^j 、 f_2^j 、 f_3^j 为第 j 个个体的 3 个目标函数值。

DM 为 Pareto 最优解集中个体分布多样性,取值越大越好,其计算公式如下:

$$DM = \sqrt{\sum_{i=1}^N \max(\|x_i^i - y_i^i\|)} \quad (14)$$

式中: $\max(\|x_i^i - y_i^i\|)$ 为 Pareto 最优解 x_i^i 和 y_i^i 最大

的欧氏距离, t 为 Pareto 最优解不再发生变化时的迭代轮数。

3.3 对比实验

本文设置了对比实验,将 HSNS-PSOGA 与 NSGA-II、NSGA-II+PSO、NSGA-III、NSGA-III+PSO 进行对比,实验结果如表 1 所示。

通过对比可以看出,HSNS-PSOGA 取得了更好的效果。在 10 个实例中的 3 个目标函数表现方面,本文算法多数情况优于 NSGA-II、NSGA-II+PSO、NSGA-III、NSGA-III+PSO; SM1 和 SM2 在第 1、2、5、9 个数据集上的表现更好;针对 DM 指标,本文算法在 7 个实例中的表现更优,说明 HSNS-PSOGA 分布多样性较好,这在一定程度上也说明 HSNS-PSOGA 能够尽可能避免局部最优。

在目标数为 M 、个体数为 N 、迭代次数为 T 的情况下,NSGA-II 和 NSGA-III 的计算复杂度为 $O(TM^2N^2)$,PSO 的计算复杂度为 $O(TM^2N)$,NSGA-II+PSO、NSGA-III+PSO 的计算复杂度为 $O(TM^2N^2) = O(TM^2N^2) + O(TM^2N)$ 。全局搜索中,GA 计算复杂度为 $O(TM^2N)$,局部搜索计算复杂度为 $O(TM^2N^2)$,因此 HSNS-PSOGA 计算复杂度为 $O(TM^2N^2)$ 。通过对比可以看出,本文提出的 HSNS-PSOGA 在不增加计算复杂度的同时能够取得更优的结果。

3.4 消融实验

如图 2 所示,为验证 HSNS-PSOGA 的可行

表 1 KBSN 实例中实验结果对比

Table 1 Comparison of experimental results in KBSN examples

算法	实例	车辆数量	距离成本	路线代价	SM1	SM2	DM
HSNS-PSOGA	Bar_8_2_X_NSrivastava86_8x2	3	406.709	28.545 9	7.942	8.054	122.190
	Bar_12_2_X_NPerl83_12x2	2	102.081	41.503	0.337	0.421	36.589
	Bar_21_5_X_NGaskell67_21x5	3	354.279	144.429	1.513	1.930	187.626
	Bar_27_5_X_NMin92_27x5	1.6	2 587.950	215.346	7.520	8.449	983.118
	Bar_32_5_X_NGaskell67_32x5_1	3	598.918	326.554	1.688	2.191	326.516
	Bar_36_5_X_NGaskell67_36x5	3	646.198	408.339	1.590	2.159	132.189
	Bar_50_5_X_NCh69_50x5	4	891.776	680.538	2.047	2.583	201.946
	Bar_55_15_X_NPerl83_55x15	7.667	814.943	1 008.590	2.628	3.588	136.783
	Bar_88_8_X_NDaskin95_88x8	3	981.619	2 211.790	2.839	3.666	202.410
NSGA-II	Bar_100_10_X_NCh69_100x10	5.8	2 720.400	2 838.280	3.962	5.031	230.094
	Bar_8_2_X_NSrivastava86_8x2	3	549.315	31.493	9.044	8.071	603.429
	Bar_12_2_X_NPerl83_12x2	2	168.886	39.931	3.653	3.690	208.700
	Bar_21_5_X_NGaskell67_21x5	3.083	504.445	151.643	1.860	2.016	178.333
	Bar_27_5_X_NMin92_27x5	1.002	4 769.868	199.782	9.150	9.361	385.335

续表

算法	实例	车辆数量	距离成本	路线代价	SM1	SM2	DM
NSGA-II+ PSO	Bar_32_5_X_NGaskell67_32x5_1	3.087	964.129	327.394	2.369	2.505	101.066
	Bar_36_5_X_NGaskell67_36x5	3.062	890.955	388.732	1.657	2.034	103.409
	Bar_50_5_X_NCh69_50x5	3.723	1 424.860	691.934	6.385	7.177	76.333
	Bar_55_15_X_NPer183_55x15	8.445	1 093.083	1 095.700	3.512	3.999	69.795
	Bar_88_8_X_NDaskin95_88x8	3.442	1 473.301	2 210.251	4.825	5.292	84.102
	Bar_100_10_X_NCh69_100x10	5.730	3 161.142	2 876.133	3.599	4.723	113.101
	Bar_8_2_X_NSrivastava86_8x2	3	520.677	31.128	14.840	14.818	690.584
	Bar_12_2_X_NPer183_12x2	2	153.566	44.322	1.067	1.082	254.661
	Bar_21_5_X_NGaskell67_21x5	3.968	435.416	157.698	77.506	78.316	340.150
NSGA-III	Bar_27_5_X_NMin92_27x5	1.002	4 551.351	204.491	8.036	8.103	260.896
	Bar_32_5_X_NGaskell67_32x5_1	3	784.149	327.703	2.835	2.982	78.992
	Bar_36_5_X_NGaskell67_36x5	3.008	831.141	392.585	1.285	1.497	153.438
	Bar_50_5_X_NCh69_50x5	3.963	1 421.524	688.783	4.326	4.976	412.546
	Bar_55_15_X_NPer183_55x15	8.067	1 058.433	1 078.535	1.979	2.621	46.320
	Bar_88_8_X_NDaskin95_88x8	3	1 318.619	2 189.086	8.793	8.866	53.580
	Bar_100_10_X_NCh69_100x10	6.015	3 092.161	2 868.368	6.603	7.118	91.082
	Bar_8_2_X_NSrivastava86_8x2	3	523.657	30.673	10.853	10.664	467.230
	Bar_12_2_X_NPer183_12x2	2	144.113	40.070	1.327	1.402	16.435 9
NSGA-III+ PSO	Bar_21_5_X_NGaskell67_21x5	3.1	481.374	158.128	1.304	1.767	155.288
	Bar_27_5_X_NMin92_27x5	1.8	5 133.620	214.711	4.588	5.514	1 426.190
	Bar_32_5_X_NGaskell67_32x5_1	3.2	912.400	327.739	2.420	1.937	111.288
	Bar_36_5_X_NGaskell67_36x5	3.3	862.456	396.162	1.437	1.926	143.357
	Bar_50_5_X_NCh69_50x5	3.9	1 342.910	689.454	1.983	2.556	196.267
	Bar_55_15_X_NPer183_55x15	8.235	1 074.410	1 034.880	2.348	3.099	105.374
	Bar_88_8_X_NDaskin95_88x8	3.6	1 482.230	2 235.740	2.956	3.864	142.481
	Bar_100_10_X_NCh69_100x10	5.807	2 978.000	2 914.270	6.614	6.776	93.457
	Bar_8_2_X_NSrivastava86_8x2	3	517.544	30.552	19.966	20.031	328.193
HSNS- PSOGA	Bar_12_2_X_NPer183_12x2	2	96.852	42.472	0.665	0.710	11.827
	Bar_21_5_X_NGaskell67_21x5	3.03	423.300	152.914	1.479	1.933	106.731
	Bar_27_5_X_NMin92_27x5	1	3 026.060	207.487	8.226	7.794	881.259
	Bar_32_5_X_NGaskell67_32x5_1	3.5	790.263	308.185	4.742	4.293	144.994
	Bar_36_5_X_NGaskell67_36x5	3	555.626	387.822	1.667	2.196	126.4
	Bar_50_5_X_NCh69_50x5	3.9	1 221.680	689.970	1.952	2.531	171.18
	Bar_55_15_X_NPer183_55x15	8.762	1 034.769	1 076.832	2.763	3.652	76.873
	Bar_88_8_X_NDaskin95_88x8	3.5	1 577.867	2 001.672	4.732	4.745	65.432
	Bar_100_10_X_NCh69_100x10	7.003	3 040.710	2 831.790	2.125	3.277	100.654

表 2 消融实验的实验结果

Table 2 Experimental results of ablation experiments

算法	实例	车辆数量	距离成本	路线代价	SM1	SM2	DM	收敛迭代轮数
HSNS- PSOGA	Bar_8_2_X_NSrivastava86_8x2	3	406.709	28.5459	7.942	8.054	122.19	440.2
	Bar_12_2_X_NPer183_12x2	2	102.081	41.503	0.337	0.421	36.589	685.3
	Bar_21_5_X_NGaskell67_21x5	3	354.279	144.429	1.513	1.930	187.626	622.7
	Bar_27_5_X_NMin92_27x5	1.6	2587.95	215.346	7.520	8.449	983.118	403.7
	Bar_32_5_X_NGaskell67_32x5_1	3	598.918	326.554	1.688	2.191	326.516	820.4
	Bar_36_5_X_NGaskell67_36x5	3	646.198	408.339	1.590	2.159	132.189	780.3
	Bar_50_5_X_NCh69_50x5	4	891.776	680.538	2.047	2.583	201.946	810.4

续表

算法	实例	车辆数量	距离成本	路线代价	SM1	SM2	DM	收敛迭代轮数
本文算法-PSO	Bar_55_15_X_NPerl83_55x15	7.667	814.943	1 008.59	2.628	3.588	136.783	648.2
	Bar_88_8_X_NDaskin95_88x8	3	981.619	2 211.79	2.839	3.666	202.41	961.4
	Bar_100_10_X_NCh69_100x10	5.8	2 720.40	2 838.28	3.962	5.031	230.094	723.6
	Bar_8_2_X_NSrivastava86_8x2	3	424.673	28.699	3.882	4.11	128.052	693.6
	Bar_12_2_X_NPerl83_12x2	2	134.434	40.072	1.212	1.404	131.233	782.4
	Bar_21_5_X_NGaskell67_21x5	3.3	444.420	152.287	1.905	2.552	137.681	805.8
	Bar_27_5_X_NMin92_27x5	1.1	4 767.980	238.801	6.195	26.466	529.600	598.3
	Bar_32_5_X_NGaskell67_32x5_1	3.1	856.442	318.987	3.894	4.497	189.016	1065.36
	Bar_36_5_X_NGaskell67_36x5	3	785.541	417.356	3.090	4.076	319.632	875.4
	Bar_50_5_X_NCh69_50x5	4.2	1 249.310	690.010	3.505	4.725	196.947	991.3
	Bar_55_15_X_NPerl83_55x15	7.5	865.433	1 071.002	4.055	5.332	130.369	805.3
	Bar_88_8_X_NDaskin95_88x8	2.8	1 273.020	2 170.120	5.616	7.062	165.368	1 075.2
本文算法-GA	Bar_100_10_X_NCh69_100x10	5.8	2 814.230	2 840.560	6.548	7.957	205.378	883.5
	Bar_8_2_X_NSrivastava86_8x2	3	415.404	28.107	2.132	2.628	112.328	470.7
	Bar_12_2_X_NPerl83_12x2	2	113.156	40.094	0.539	0.728	45.548	709.4
	Bar_21_5_X_NGaskell67_21x5	3.5	458.437	150.533	1.785	2.358	162.030	620.6
	Bar_27_5_X_NMin92_27x5	2.2	2 767.620	220.468	12.165	12.979	1 037.520	503.4
	Bar_32_5_X_NGaskell67_32x5_1	3.4	646.538	334.773	2.471	3.096	312.291	728.4
	Bar_36_5_X_NGaskell67_36x5	3.1	682.457	417.378	2.398	2.528	138.824	794.2
	Bar_50_5_X_NCh69_50x5	4	1 174.530	697.699	4.625	5.699	36.472	860.4
	Bar_55_15_X_NPerl83_55x15	8.1	1 193.840	1 060.510	3.250	4.240	94.164 2	890.2
	Bar_88_8_X_NDaskin95_88x8	3.5	415.404	28.107	2.132	2.628	112.328	982.6
	Bar_100_10_X_NCh69_100x10	7.2	113.156	40.094	0.539	0.728	45.548	750.3
本文算法-聚合函数选择 优质个体局部搜索	Bar_8_2_X_NSrivastava86_8x2	3	418.357	29.357	1.935	2.178	150.371	450.2
	Bar_12_2_X_NPerl83_12x2	2	102.680	43.672	0.877	1.172	31.357	688.4
	Bar_21_5_X_NGaskell67_21x5	3.4	339.256	157.278	2.263	2.836	195.378	601.3
	Bar_27_5_X_NMin92_27x5	3	3 209.465	238.468	33.826	34.372	610.291	490.3
	Bar_32_5_X_NGaskell67_32x5_1	3.1	600.764	310.235	9.725	11.367	180.267	710.6
	Bar_36_5_X_NGaskell67_36x5	3	576.367	385.283	2.168	2.856	169.356	785.3
	Bar_50_5_X_NCh69_50x5	4.3	1 017.267	705.384	2.518	3.278	210.372	830.6
	Bar_55_15_X_NPerl83_55x15	7	791.689	1 096.257	3.382	4.368	120.783	857.3
	Bar_88_8_X_NDaskin95_88x8	3.2	1 093.835	2 218.392	3.628	4.473	158.367	952.6
	Bar_100_10_X_NCh69_100x10	6.6	2 890.457	2 867.463	6.720	7.798	228.493	737.9
本文算法-次优个体局部搜索	Bar_8_2_X_NSrivastava86_8x2	3.3	411.267	31.725	2.718	3.029	197.346	452.7
	Bar_12_2_X_NPerl83_12x2	2	93.250	43.710	0.562	0.774	35.782	679.5
	Bar_21_5_X_NGaskell67_21x5	3	402.679	143.591	2.162	2.676	185.618	590.6
	Bar_27_5_X_NMin92_27x5	2.3	3 434.716	315.625	25.862	13.095	929.623	458.2
	Bar_32_5_X_NGaskell67_32x5_1	3	667.418	320.634	2.371	3.205	219.512	699.3
	Bar_36_5_X_NGaskell67_36x5	3.2	671.783	401.729	2.320	2.848	147.641	778.3
	Bar_50_5_X_NCh69_50x5	4	1 315.367	680.523	2.693	3.396	174.719	849.5
	Bar_55_15_X_NPerl83_55x15	7	970.392	1 010.670	2.631	4.736	79.230	862.9
	Bar_88_8_X_NDaskin95_88x8	4.3	1 525.381	2 220.420	2.465	3.482	200.183	928.4
	Bar_100_10_X_NCh69_100x10	7.4	2 956.491	2 926.380	6.117	7.729	203.714	730.5
本文算法-对种群中后 1/12 个体进行局部搜索	Bar_8_2_X_NSrivastava86_8x2	3	406.709	28.545 9	7.983	8.182	122.190	442.8
	Bar_12_2_X_NPerl83_12x2	2	96.413	42.174 8	1.258	1.390	293.196	673.6
	Bar_21_5_X_NGaskell67_21x5	3	378.831	146.445	1.805	2.322	262.696	524.8
	Bar_27_5_X_NMin92_27x5	2	3 113.850	219.381	10.116	10.844	635.685	444.9

续表

算法	实例	车辆数量	距离成本	路线代价	SM1	SM2	DM	收敛迭代轮数
	Bar_32_5_X_NGaskell67_32x5_1	3	741.388	310.122	10.320	11.116	145.673	844.4
	Bar_36_5_X_NGaskell67_36x5	3	653.843	390.472	2.142	2.705	201.434	695.5
	Bar_50_5_X_NCh69_50x5	3.5	965.022	702.648	4.999	5.924	357.133	862.6
	Bar_55_15_X_NPerl83_55x15	7.667	934.004	993.711	4.656	5.882	111.279	643.7
	Bar_88_8_X_NDaskin95_88x8	3	1 293.360	2 208.900	6.230	7.875	209.230	986.1
	Bar_100_10_X_NCh69_100x10	5.25	2 744.040	2 776.480	8.974	10.039	325.476	877.2

性,本文设计了消融实验,分别将本文算法与本文算法-PSO、本文算法-GA、本文算法-聚合函数选择优质个体局部搜索、本文算法-次优个体局部搜索以及本文算法-对种群中后 1/12 个体进行局部搜索进行了对比。

由消融实验结果可知,本文算法-PSO 的收

敛迭代次数更少,表明通过 PSO 能够提高收敛速率。本文算法-GA 的 DM 值更小,说明解的多样性差,证明通过 GA 能够增加多样性,避免陷入局部最优。通过优质个体局部搜索能够减小 SM1 值,增加解的均匀性。通过次优个体局部搜索,SM2 值有明显的减小,说明 Pareto 最优解在目标

表 3 不同比例参数实验结果

Table 3 Experimental results of different proportional parameters

打乱比例	实例	车辆数量	距离成本	路线代价	SM1	SM2	DM
HSNS-PSOGA (1/12)	Bar_8_2_X_NSrivastava86_8x2	3	406.709	28.5459	7.942	8.054	122.190
	Bar_12_2_X_NPerl83_12x2	2	102.081	41.503	0.337	0.421	36.589
	Bar_21_5_X_NGaskell67_21x5	3	354.279	144.429	1.303	1.930	187.626
	Bar_27_5_X_NMin92_27x5	1.6	2 587.950	215.346	12.113	7.149	1 321.410
	Bar_32_5_X_NGaskell67_32x5_1	3	598.918	326.554	1.688	2.077	326.516
	Bar_36_5_X_NGaskell67_36x5	3	646.198	385.745	1.590	2.159	132.189
	Bar_50_5_X_NCh69_50x5	4	852.713	680.538	2.047	2.583	201.946
	Bar_55_15_X_NPerl83_55x15	7.667	814.943	1 008.590	3.863	3.588	136.783
	Bar_88_8_X_NDaskin95_88x8	3	981.619	2 211.790	2.839	3.666	212.046
	Bar_100_10_X_NCh69_100x10	5.8	2 720.40	2 838.280	3.962	5.031	235.119
对后 1/6 个体进行 用户顺序 打乱	Bar_8_2_X_NSrivastava86_8x2	3	410.577	30.674	1.713	2.071	126.398
	Bar_12_2_X_NPerl83_12x2	2	105.761	44.127	0.566	0.771	18.861
	Bar_21_5_X_NGaskell67_21x5	3	349.005	152.029	2.802	3.436	221.036
	Bar_27_5_X_NMin92_27x5	2.1	2 710.487	162.783	11.725	8.763	207.452
	Bar_32_5_X_NGaskell67_32x5_1	3.7	576.728	316.927	1.628	2.925	154.925
	Bar_36_5_X_NGaskell67_36x5	3.1	701.673	390.756	1.783	2.578	139.634
	Bar_50_5_X_NCh69_50x5	4	847.735	698.247	2.852	2.477	251.706
	Bar_55_15_X_NPerl83_55x15	7.8	881.743	1 109.468	3.947	3.833	129.672
	Bar_88_8_X_NDaskin95_88x8	3.5	1 007.365	2 389.754	1.745	2.478	201.467
	Bar_100_10_X_NCh69_100x10	7.1	3 002.384	3 005.478	4.825	5.732	169.567
对后 1/24 个体进行 用户顺序 打乱	Bar_8_2_X_NSrivastava86_8x2	3	445.361	25.062 4	1.850	2.438	82.934
	Bar_12_2_X_NPerl83_12x2	2	103.150	41.094	0.619	0.827	30.531
	Bar_21_5_X_NGaskell67_21x5	3	375.711	154.232	2.032	1.738	117.057
	Bar_27_5_X_NMin92_27x5	1.8	3 303.970	207.180	12.755	6.801	1 262.670
	Bar_32_5_X_NGaskell67_32x5_1	3	546.247	316.384	1.771	2.237	166.391
	Bar_36_5_X_NGaskell67_36x5	3.6	659.582	420.745	1.726	2.064	189.634
	Bar_50_5_X_NCh69_50x5	4	901.182	691.012	2.602	3.249	190.523
	Bar_55_15_X_NPerl83_55x15	7.3	900.426	1 101.310	3.801	4.673	120.781
	Bar_88_8_X_NDaskin95_88x8	4	1 144.640	2 207.710	4.152	5.149	411.740
	Bar_100_10_X_NCh69_100x10	7.5	3 010.990	2 847.830	6.659	8.208	232.554

空间中的分布情况更好;DM的取值更大,说明解的多样性得到了提升,因此,通过次优个体局部搜索能够提升种群的质量。对种群中后1/12的个体打乱用户顺序后,DM值更大,说明多样性更好。

3.5 参数选择

在局部搜索策略中,选择对部分个体打乱用户顺序,本文设计打乱比例为1/6、1/12、1/24,以探究打乱比例对算法性能的影响。实验结果如表3。

由表3结果可知,比例选取为1/12时,整体效果最好,在3个目标函数上的取值更优,在第2、3、6、7、10个数据集上SM1指标更好,在第2、5、8、10个数据集上SM2指标更好,说明解的分布更均匀,在第2、4、5、8、10个数据集上DM指标更好,说明种群多样性更好。分析原因为:打乱比例选取为1/6时,种群中更新比例大,可能陷入局部最优;打乱比例选取为1/24时,更新比例较小,对算法优化效果有限,难以找到全局最优解。

4 结束语

本文提出的HSNS-PSOGA结合了EMO算法与元启发式算法,能够在一定程度上平衡收敛性和多样性,同时提高解的均匀性。实验结果显示,全局搜索使用PSO算法和GA,能够加快收敛同时避免陷入局部最优;使用NSGA-III选择机制,可以在一定程度上保证种群的多样性;局部搜索分别对优质和次优个体进行搜索,提高了获得更优解的概率并增加了Pareto最优解的多样性;对种群中后1/12的个体进行局部搜索,提升了解的均匀性。在公开数据集上进行验证,本文所提算法在大多数情况下更优。但本文所提算法在大规模数据集上的表现优势并不明显,下一步的研究中将探索该问题。

参考文献:

- [1] Toth P, Vigo D. The vehicle routing problem: Society for industrial and applied mathematics[J]. Siam Monographs on Discrete Mathematics and Applications, 2001, 8(3): 135-176.
- [2] Laporte G, Nobert Y, Arpin D. An exact algorithm for solving a capacitated location-routing problem [J]. Annals of Operations Research, 1986, 6(9): 291-310.
- [3] Barreto S, Ferreira C, Paixao J, et al. Using clustering analysis in a capacitated location-routing problem [J]. European Journal of Operational Research, 2007, 179(3): 968-977.
- [4] Salhi S, Rand G K. The effect of ignoring routes when locating depots[J]. European Journal of Operational Research, 1989, 39(2): 150-156.
- [5] Perl J, Daskin M S. A warehouse location-routing problem[J]. Transportation Research Part B: Methodological, 1985, 19(5): 381-396.
- [6] Huang S H, Huang Y H, Blazquez C A, et al. Solving the vehicle routing problem with drone for delivery services using an ant colony optimization algorithm [J]. Advanced Engineering Informatics, 2022, 51: No. 101536.
- [7] Ma B, Hu D, Wu X. The location routing problem of the car-sharing system with autonomous electric vehicles[J]. KSCE Journal of Civil Engineering, 2021, 25(8): 3107-3120.
- [8] 胡圣邦,袁小芳,郭琳.改进蚁群算法的民爆物品运输路线优化[J].公路交通科技,2023,40(3): 247-253.
Hu Sheng-bang, Yuan Xiao-fang, Guo Lin. Improvement of ant colony algorithm for the optimization of transportation routes of civil explosives[J]. Highway Transportation Science and Technology, 2023, 40(3): 247-253.
- [9] 杜文帝,张虹.基于改进蚁群算法的危险化学品运输路径优化[J].科技风,2023(7): 153-156.
Du Wen-di, Zhang Hong. Optimization of hazardous chemical transportation route based on improved ant colony algorithm[J]. Science and Technology Wind, 2023(7): 153-156.
- [10] Deb K, Mohan M, Mishra S. Evaluating the ϵ -domination based multi-objective evolutionary algorithm for a quick computation of pareto optimal solutions[J]. Evolutionary Computation, 2005, 13(4): 501-525.
- [11] Deb K, Pratap A, Agarwal S, et al. A fast and elitist multi-objective genetic algorithm: NSGA-II[J]. IEEE Transactions on Evolutionary Computation, 2002, 6(2): 182-197.
- [12] Deb K, Jain H. An evolutionary many-objective optimization algorithm using reference-point-based non-dominated sorting approach, part I: solving problems with box constraints[J]. IEEE Transactions on Evolutionary Computation, 2013, 18(4): 577-601.
- [13] Rabbani M, Farrokhi-Asl H, Asgarian B. Solving a bi-objective location routing problem by a NSGA-II combined with clustering approach: Application in

- waste collection problem[J]. *Journal of Industrial Engineering International*, 2017, 13(1): 13-27.
- [14] 杨红波, 史文库, 陈志勇, 等. 基于 NSGA-II 的斜齿轮宏观参数多目标优化[J]. *吉林大学学报:工学版*, 2023, 53(4): 1007-1018.
- Yang Hong-bo, Shi Wen-ku, Chen Zhi-yong, et al. Multi-objective optimization of macro-parameters of helical gears based on NSGA-II[J]. *Journal of Jilin University(Engineering and Technology Edition)*, 2023, 53(4): 1007-1018.
- [15] Hernandez C, Lara J, Arjona M A, et al. Electro-magnetic optimal design of a PMSG considering three objectives and using NSGA-III[J]. *IEEE Transactions on Magnetics*, 2022, 58(9): 1-4.
- [16] Moen H J F, Hansen N B, Hovland H, et al. Many-objective optimization using taxi-cab surface evolutionary algorithm[C]//The 7th International Conference on Evolutionary Multi-Criterion Optimization, Sheffield, UK, 2013: 128-142.
- [17] Asafuddoula M, Ray T, Sarker R. A decomposition-based evolutionary algorithm for many objective optimization[J]. *IEEE Transactions on Evolutionary Computation*, 2014, 19(3): 445-460.
- [18] Cheng R, Jin Y, Olhofer M, et al. A reference vector guided evolutionary algorithm for many-objective optimization[J]. *IEEE Transactions on Evolutionary Computation*, 2016, 20(5): 773-791.
- [19] Shi L, Gong J, Zhai C. Application of a hybrid PSO-GA optimization algorithm in determining pyrolysis kinetics of biomass[J]. *Fuel*, 2022, 323: No. 124344.
- [20] Deb K, Jain H. An evolutionary many-objective optimization algorithm using reference-point-based non-dominated sorting approach, part I: solving problems with box constraints[J]. *IEEE Transactions on Evolutionary Computation*, 2013, 18(4): 577-601.
- [21] Poli R, Kennedy J, Blackwell T. Particle swarm optimisation[J]. *Journal of Xidian University*, 1995, 42(1): 16-22.
- [22] Ishibuchi H, Narukawa K. Some issues on the implementation of local search in evolutionary multiobjective optimization[C]//Genetic and Evolutionary Computation Conference, Berlin, Heidelberg, 2004: 1246-1258.
- [23] Yang J, Soh C K. Structural optimization by genetic algorithms with tournament selection[J]. *Journal of Computing in Civil Engineering*, 1997, 11(3): 195-200.
- [24] Sasikumar A, Muthaiah R. Operational amplifier circuit sizing based on NSGA-II and particle swarm optimization[C]//International Conference on Networks & Advances in Computational Technologies(NetACT), Thiruvanthapuram, India, 2017: 64-68.
- [25] Masood A, Mei Y, Chen G, et al. A PSO-based reference point adaption method for genetic programming hyper-heuristic in many-objective job shop scheduling[C]//Third Australasian Conference on Artificial Life and Computational Intelligence, Sydney, Australia, 2017: 326-338.