吉林大学学报(工学版)
Journal of Jilin University(Engineering and Technology Edition)
ISSN 1671-5497,CN 22-1341/T

《吉林大学学报(工学版)》网络首发论文

题目：非支配排序粒子群遗传算法解决车辆位置路由问题
作者：刘琼昕，王甜甜，王亚男
DOI：10.13229/j.cnki.jdxbgxb.20231086
收稿日期：2023-12-26
网络首发日期：2024-02-04
引用格式：刘琼昕，王甜甜，王亚男．非支配排序粒子群遗传算法解决车辆位置路由问题[J/OL]．吉林大学学报(工学版).
https://doi.org/10.13229/j.cnki.jdxbgxb.20231086

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： 本文提出了混合全局局部搜索的非支配排序粒子群遗传算法，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，提高算法的收敛速度，使用第三代非支配排序遗传算法挑选种群下一代个体，保留种群多样性。局部搜索策略针对优质和次优个体进行，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提出算法与基准算法对比，本文算法从种群质量、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： 计算机应用技术；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

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-tian[2]，WANG Ya-nan[2]

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

Abstract: In this paper, we propose non-dominated sorted particle swarm genetic algorithm with hybrid global-local search, which can effectively solve the vehicle location routing problem. The global search utilizes both particle swarm optimization and genetic algorithm operators to improve convergence speed. The non-dominated sorting genetic algorithm III is employed to maintain population diversity. The local search strategy is applied separately to superior and inferior individuals, increasing the probability of obtaining better solutions. Additionally, the last 1/12 individuals' user orders in the population are shuffled to enhance overall population quality. We compare our proposed algorithm with benchmark algorithms using the open standard dataset. Our algorithm is better in terms of population quality and diversity and can provide an effective solution to the vehicle location routing problem.

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

收稿日期： 2023-12-26.
基金项目： 国家自然科学基金项目资助（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.

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）和车辆路线选择问题（Vehicles routing problem, VRP）[1]。Laporte[2]在 1986 年描述了经典的 LRP 问题，该问题需要从潜在设施位置中选择最优位置，并确定从仓库开始对用户服务的车辆路线。

针对 LRP，Barreto[3]等人对 LAP 和 VRP 分开进行解决。Salhi 和 Rand[4]指出，同时考虑 LAP 和 VRP 通常会得到更好的效果。Perl[5]在文中同时分析 LAP 和 VRP，使用启发式算法来解决 LRP。

使用元启发式算法是解决 LRP 的途径之一。Huang[6]等人使用蚁群优化算法解决无人机配送服务的车辆路径问题（VRPD）。Ma[7]等人使用遗传算法解决自动驾驶汽车的位置路由问题。Hu[8]等人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，Du[9]等人构建了车辆路径优化模型，利用改进蚁群算法进行模型求解。

使用进化多目标优化（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都有了大量的成果，如第二代强度 Pareto 进化算法（Strength Pareto Evolutionary Algorithm 2，SPEA2）[10]，第二代（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[13]等人首先使用 NSGA-II 应用于二目标的 LRP 中。Yang[14]等人提出了齿轮副宏观参数多目标优化方法，利用 NSGA-II 解决多个优化目标之间的冲突。Hernandez[15]等人将 NSGA-III 应用于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，TC-SEA[16]、I-DBEA[17]和 RVEA[18]都是该类代表性的算法。

虽然很多方法都 LRP 进行了解决，但很少有学者将两种元启发算法和进化多目标优化算法结合来解决 LRP。考虑到粒子群算法（Particle Swarm Optimization, PSO）和遗传算法（Genetic Algorithm, GA）能够分别提高收敛效率和避免陷入局部最优[19]，而 NSGA-III 能够找到收敛良好且多样化的点[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，减小车辆使用数量，可能需要将车辆分配给较远的用户，从而增加路由距离，减小路由距离，可能会使车辆行驶成本较高的路，因此 LRP 是一个多目标优化问题，需要最优化的三个目标是车辆数量，车辆路由距离和车辆路由成本。

令 I 表示候选仓库集合，J 表示用户顶点集合，V 表示顶点集合，其中 $V=I\cup J$，K 表示车辆序列，u_k 为 1 表示第 k 辆车被使用，$x_{i,j,k}$ 为 1 表示第 k 辆车的行驶路由覆盖点 i 到 j 的边，$d_{i,j}$ 表示顶点 i 到 j 的路由距离，$c_{i,j}$ 表示顶点 i 到 j 的路由代价。三个目标函数的数学化描述如下。

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

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

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

其中，第一个目标是最小化车辆使用数量，第二个目标是最小化车辆派送路线的总距离，第三个目标是最小化车辆派送路线产生的总代价成本。

三个优化目标存在着竞争关系，重要性相当，因此需要根据染色体计算个体在三个目标上的取值，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，$z_{j,k}$ 为 1 表示第 k 辆车被分配去服务用户 j，q_i 表示用户顶点需

求，q 表示车辆的货物负载量。

本文研究的车辆位置路由问题有以下四个约束条件，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与其负载能力相称，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一个仓库。

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

$$\sum_{j\in J} q_j z_{j,k} \le q,\ \forall k \in K_a \tag{5}$$

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

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

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 之间的整数。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：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，为了确保式 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，确保式 6 对应约束。

如图 1 所示，假设共有三个仓库，八个用户，则仓库编号为 0-2，用户编号为 3-10，编号为 0 的仓库(1 号仓库)服务第 3、8 号用户，2 号仓库服务第 7、4、9 用户，3 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。

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。确定了仓库服务的用户，根据车辆容量和用户需求进行派遣，对应式(5)、(6)。两个仓库之间的用户由前一个仓库服务，对应式(7)。

2 算法描述

PSO 和 GA 能够分别提高收敛效率和避免陷入局部最优[19]，为了平衡种群多样性和收敛性，本文提出了 HSNS-PSOGA，算法流程见算法 1。

图 1 解向量示意图

Fig.1 Solution vector schematic

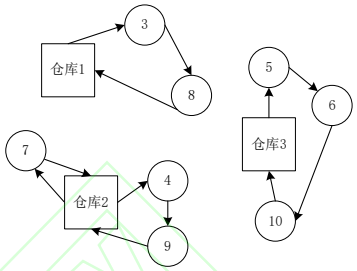

图 2 LRP 解决方案

Fig.2 The LRP solution

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，同时结合 GA 的变化算子对个体进行交叉变异，之后使用 NSGA-III 选择个体，全局搜索产生子代个体 P_{t+1} 后进行局部搜索，对 P_{t+1} 中优质、次优个体进一步改进，再更新后 1/12 个体。

Algorithm1 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**

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[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：

$$\begin{aligned}\boldsymbol{v}_i(t+1) = \omega \boldsymbol{v}_i(t) + c_1 r_1\left(\boldsymbol{x}_{pbest_i} - \boldsymbol{x}_i(t)\right) + \\ c_2 r_2\left(\boldsymbol{x}_{gbest} - \boldsymbol{x}_i(t)\right)\end{aligned} \tag{8}$$

$$\boldsymbol{x}_i(t+1) = \boldsymbol{x}_i(t) + \boldsymbol{v}(t+1) \tag{9}$$

其中，粒子 i 的速度记作 $\boldsymbol{v}_i$，粒子 i 的历史最优解记作 $\boldsymbol{x}_{pbest_i}$，整个种群的全局最优解记作 x_{gbest}，ω 为动态惯性权重，c_1 和 c_2 是加速因子，r_1 和 r_2 是[0,1]之间的随机数，$\boldsymbol{x}_i$ 表示粒子 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（式 4），需要保证解向量每一维取值为整数且不重复出现，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 $\boldsymbol{X}=(x_1,x_2,...,x_n)$，$n=|J|+|I|-1$。

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

2.对 V 中元组按照 x_i 升序排列，得到$V'=\{(x_{i_1},i_1),(x_{i_2},i_2),\ldots,(x_{i_n},i_n)\}$，其中$x_{i_1}\le x_{i_2}\le\ldots\le x_{i_n}$。

3.令 $D=\{(i_j,j)|1\le j\le n\}$，$j$ 表示排序后的顺序。

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三个目标上的取值，若存在粒子 i Pareto 支配$\boldsymbol{x}_{pbest_i}$或 $\boldsymbol{x}_{gbest}$，进行更新。

对种群中个体使用粒子群优化算法之后，使用遗传算法的变化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，染色体 A 在位置 p 的值为 a，即 $A[p]=a$，在染色体 B 中找到值为 a 的位置 q，A 在位置 q 的值为 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。

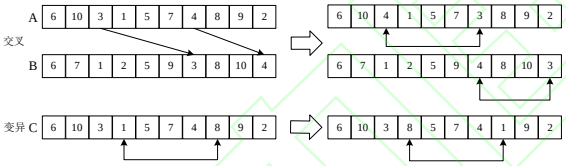

图 3 交叉和变异操作

Fig.3 Operation of crossing and compiling

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 使用精英策略，能够保证种群中优质父代被保留，NSGA-III 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。

Algorithm2 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 使用粒子群变化算子产生 E_t
2: 对 E_t 使用遗传算法变化算子产生 Q_t
3: 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：$F_1, F_2, ...$
6: $S_t=\emptyset$, $i=1$
7: **repeat**
8: $S_t=S_t\cup F_i$, $i=i+1$
9: **until** $|S_t|\ge N$
10: **if** $|S_t|=N$ **then**
11: $P_{t+1}=S_t$
12: **else** $P_{t+1}=\Sigma_{j=1}^{i-1}F_j$
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 $K=N-|P_{t+1}|$个个体加入到 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，从而以更高的概率获得更优解；对种群中次优个体进行局部搜索，减少 Pareto 支配解的个数，提高整个种群的质量，进一步加快收敛；对后 1/12 个体打乱用户顺序，增加种群多样性。局部搜索算法流程见算法 3。

有文献[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，见式(10)。λ_1、λ_2、λ_3 为参考平面上的随机点在三个维度的取值，$f_1^i(x)$、$f_2^i(x)$、$f_3^i(x)$ 是个体 i 对应的三个目标函数值。

$$f^*(x)=\lambda_1 f_1^i(x)+\lambda_2 f_2^i(x)+\lambda_3 f_3^i(x),\ \lambda_1+\lambda_2+\lambda_3=1 \tag{10}$$

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，每次选择两个个体进行比较，$f^*(x)$取值更小的个体进入优质种群。对优质种群采用三种变化算子：反转、交换和插入。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到首位。上述过程重复 $N*P$ 次，P 是对 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 x'代替 x"。通过对优质个体进行局部搜索，可以以一定概率寻找到更加优质的个体，解决了种群收敛性差的问题。

除对优质个体进行局部搜索，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，进行反转、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 y'替换 y。对次优个体进行的局部搜索由于引入了遗传算法的变化算子，可以在一定程度上增加种群的多样性，同时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，从而提高种群多样性，具体做法为：仓库编号保持不变，将两个仓库编号之间的用

户编号打乱顺序，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。

Algorithm3 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，使用三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** $i = T_{local}$
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

3 实验结果与分析

为证明 HSNS-PSOGA 的有效性，设置了对比实验，分别与 NSGA-II[11]、NSGA-II+PSO[24]、NSGA-III[12]、NSGA-III+PSO[25]进行对比。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 是进化多目标优化算法和元启发算法的结合。另外，本文设计了消融实验，以证明对优质、次优个体局部搜索、使用粒子群优化算法和对后 1/12 个体打乱用户顺序的合理性。另外，本文也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、YLBSN、YLPAM、YLPSN 八组数据集，其中，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 数据集中提供的 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_NGaskell67_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，本文的对比指标除三个目标函数：车辆使用数量、车辆路由距离成本、车辆路由代价之外，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，SM1 值越小代表更好的均匀性，dis_i 是第 i 个解与 Pareto 最优解集中最接近它的解之间的欧氏距离，$\overline{dis}$ 是所有 dis_i 求和取平均，N 为种群中个体总数。

$$\mathrm{SM1} = \sqrt{\frac{1}{N-1} * \sum_{i=1}^{N} \left(dis_i - \overline{dis}\right)^2} \tag{11}$$

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，其中 $\bar{d}$ 是所有d_i求和取平均，f_1^i、f_2^i、f_3^i是第i个个体的三个目标函数值，f_1^j、f_2^j、f_3^j是第j个个体的三个目标函数值。

$$\mathrm{SM2} = \sqrt{\frac{1}{N-1} * \sum \left(d_i - \bar{d}\right)^2} \tag{12}$$

$$d_i = \min_j \begin{pmatrix} |f_1^i(x) - f_1^j(x)| + \\ |f_2^i(x) - f_2^j(x)| + |f_3^i(x) - f_3^j(x)| \end{pmatrix} i, j = 1, 2, ..., N \tag{13}$$

DM 为 Pareto 最优解集中个体分布多样性，取值越大越好，其中 $\max\left(\| x_t^i - y_t^i \|\right)$ 为 Pareto 最优解 x_t^i 和 y_t^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。

$$\mathrm{DM} = \sqrt{\sum_{i=1}^{N} \max\left(\| x_t^i - y_t^i \|\right)} \tag{14}$$

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 进行了对比，实验结果如表 1 所示，表中加粗的字体表示上述五种算法的最优情况。

通过对比可以看出，HSNS-PSOGA 取得了更好的效果，在 10 个实例中的三个目标函数表现方面，本文使用算法多数情况优于 NSGA-II、NSGA-II+PSO、NSGA-III、NSGA-III+PSO，SM1 和 SM2 在第 1、2、5、9 四个数据集上表现更好，针对 DM 指标，本文提出算法在 7 个实例中表现更优，说明

HSNS-PSOGA 分布多样性较好，这在一定程度上也说明了 HSNS-PSOGA 能够尽量避免局部最优。

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 $O(TMN^2)$，PSO 的计算复杂度为 $O(TMN)$，NSGA-II+PSO 、 NSGA-III+PSO 的计算复杂度为 $O(TMN^2)=O(TMN^2)+O(TMN)$。全局搜索中，GA 计算复杂度为 $O(TMN)$，局部搜索计算复杂度为 $O(TMN^2)$，因此 HSNS-PSOGA 计算复杂度为 $O(TMN^2)$。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。

表 1 KBSN 实例中实验结果对比

Table 1 Comparison of experimental results in KBSN examples

实例	车辆数量	距离成本	路线代价	SM1	SM2	DM
混合全局局部搜索的非支配排序粒子群遗传算法						
Bar_8_2_X_NSrivastava86_8x2	**3**	**406.709**	**28.5459**	**7.942**	**8.054**	122.19
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	**2587.95**	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**	**1008.59**	2.628	3.588	**136.783**
Bar_88_8_X_NDaskin95_88x8	**3**	**981.619**	2211.79	**2.839**	**3.666**	**202.410**
Bar_100_10_X_NCh69_100x10	5.8	**2720.40**	2838.28	3.962	5.031	**230.094**
NSGA-II						
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	4769.868	**199.782**	9.150	9.361	385.335
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**	1424.860	691.934	6.385	7.177	76.333
Bar_55_15_X_NPerl83_55x15	8.445	1093.083	1095.700	3.512	3.999	69.795
Bar_88_8_X_NDaskin95_88x8	3.442	1473.301	2210.251	4.825	5.292	84.102
Bar_100_10_X_NCh69_100x10	**5.730**	3161.142	2876.133	3.599	4.723	113.101
NSGA-II+PSO						
Bar_8_2_X_NSrivastava86_8x2	**3**	520.677	31.128	14.840	14.818	**690.584**
Bar_12_2_X_NPerl83_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**
Bar_27_5_X_NMin92_27x5	1.002	4551.351	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	1421.524	688.783	4.326	4.976	**412.546**
Bar_55_15_X_NPerl83_55x15	8.067	1058.433	1078.535	**1.979**	**2.621**	46.320
Bar_88_8_X_NDaskin95_88x8	**3**	1318.619	2189.086	8.793	8.866	53.580
Bar_100_10_X_NCh69_100x10	6.015	3092.161	2868.368	6.603	7.118	91.082
NSGA-III						
Bar_8_2_X_NSrivastava86_8x2	**3**	523.657	30.673	10.853	10.664	467.230
Bar_12_2_X_NPerl83_12x2	**2**	144.113	40.070	1.327	1.402	16.4359
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	5133.62	214.711	**4.588**	**5.514**	**1426.19**
Bar_32_5_X_NGaskell67_32x5_1	3.2	912.4	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	1342.91	689.454	1.983	2.556	196.267
Bar_55_15_X_NPerl83_55x15	8.235	1074.41	1034.88	2.348	3.099	105.374
Bar_88_8_X_NDaskin95_88x8	3.6	1482.23	2235.74	2.956	3.864	142.481

Bar_100_10_X_NCh69_100x10	5.807	2978	2914.27	6.614	6.776	93.457
NSGA-III+PSO						
Bar_8_2_X_NSrivastava86_8x2	**3**	517.544	30.552	19.966	20.031	328.193
Bar_12_2_X_NPerl83_12x2	**2**	**96.852**	42.472	0.665	0.710	11.827
Bar_21_5_X_NGaskell67_21x5	3.03	423.3	152.914	1.479	1.933	106.731
Bar_27_5_X_NMin92_27x5	**1**	3026.06	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**	1221.68	689.97	**1.952**	**2.531**	171.18
Bar_55_15_X_NPerl83_55x15	8.762	1034.769	1076.832	2.763	3.652	76.873
Bar_88_8_X_NDaskin95_88x8	3.5	1577.867	**2001.672**	4.732	4.745	65.432
Bar_100_10_X_NCh69_100x10	7.003	3040.71	**2831.79**	**2.125**	**3.277**	100.654

表 2 消融实验的实验结果

Table 2 Experimental results of ablation experiments

实例	车辆数量	距离成本	路线代价	SM1	SM2	DM	收敛迭代轮数
混合全局局部搜索的非支配排序粒子群遗传算法							
Bar_8_2_X_NSrivastava86_8x2	**3**	**406.709**	28.5459	**7.942**	**8.054**	122.19	**440.2**
Bar_12_2_X_NPerl83_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**
Bar_55_15_X_NPerl83_55x15	7.667	814.943	1008.59	**2.628**	**3.588**	**136.783**	648.2
Bar_88_8_X_NDaskin95_88x8	3	**981.619**	2211.79	2.839	3.666	202.41	961.4
Bar_100_10_X_NCh69_100x10	5.8	**2720.40**	2838.28	**3.962**	**5.031**	230.094	**723.6**
本文算法-PSO							
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	444.42	152.287	1.905	2.552	137.681	805.8
Bar_27_5_X_NMin92_27x5	1.1	4767.98	238.801	**6.195**	26.466	529.6	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	3.09	4.076	**319.632**	875.4
Bar_50_5_X_NCh69_50x5	4.2	1249.31	690.01	3.505	4.725	196.947	991.3
Bar_55_15_X_NPerl83_55x15	7.5	865.433	1071.002	4.055	5.332	130.369	805.3
Bar_88_8_X_NDaskin95_88x8	**2.8**	1273.02	**2170.12**	5.616	7.062	165.368	1075.2
Bar_100_10_X_NCh69_100x10	5.8	2814.23	2840.56	6.548	7.957	205.378	883.5
本文算法-GA							
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	162.03	620.6
Bar_27_5_X_NMin92_27x5	2.2	2767.62	220.468	12.165	12.979	1037.52	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	1174.53	697.699	4.625	5.699	36.472	860.4
Bar_55_15_X_NPerl83_55x15	8.1	1193.84	1060.51	3.250	4.240	94.1642	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	3209.465	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	1017.267	705.384	2.518	3.278	210.372	830.6
Bar_55_15_X_NPerl83_55x15	**7**	**791.689**	1096.257	3.382	4.368	120.783	857.3
Bar_88_8_X_NDaskin95_88x8	3.2	1093.835	2218.392	3.628	4.473	158.367	952.6
Bar_100_10_X_NCh69_100x10	6.6	2890.457	2867.463	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	3434.716	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	1315.367	680.523	2.693	3.396	174.719	849.5
Bar_55_15_X_NPerl83_55x15	**7**	970.392	1010.67	2.631	4.736	79.230	862.9
Bar_88_8_X_NDaskin95_88x8	4.3	1525.381	2220.42	2.465	**3.482**	200.183	**928.4**
Bar_100_10_X_NCh69_100x10	7.4	2956.491	2926.38	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**	**28.5459**	7.983	8.182	122.19	442.8
Bar_12_2_X_NPerl83_12x2	**2**	96.413	42.1748	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	3113.85	219.381	10.116	10.844	635.685	444.9
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	1293.36	2208.9	6.230	7.875	**209.23**	986.1
Bar_100_10_X_NCh69_100x10	**5.25**	2744.04	**2776.48**	8.974	10.039	**325.476**	877.2

表 3 不同比例参数实验结果

Table 3 Experimental results of different proportional parameters

实例	车辆数量	距离成本	路线代价	SM1	SM2	DM
混合全局局部搜索的非支配排序粒子群遗传算法（1/12）						
Bar_8_2_X_NSrivastava86_8x2	**3**	**406.709**	**28.5459**	7.942	8.054	122.19
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**	**2587.95**	215.346	12.113	7.149	**1321.41**
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**	**1008.59**	3.863	**3.588**	**136.783**
Bar_88_8_X_NDaskin95_88x8	**3**	**981.619**	2211.79	2.839	3.666	212.046
Bar_100_10_X_NCh69_100x10	**5.8**	**2720.40**	**2838.28**	**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	2710.487	**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	1109.468	3.947	3.833	129.672
Bar_88_8_X_NDaskin95_88x8	3.5	1007.365	2389.754	**1.745**	**2.478**	201.467
Bar_100_10_X_NCh69_100x10	7.1	3002.384	3005.478	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	25.0624	1.850	2.438	82.934
Bar_12_2_X_NPerl83_12x2	**2**	103.15	**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	3303.97	207.18	12.755	**6.801**	1262.67
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	1101.31	**3.801**	4.673	120.781
Bar_88_8_X_NDaskin95_88x8	4	1144.64	**2207.71**	4.152	5.149	**411.74**
Bar_100_10_X_NCh69_100x10	7.5	3010.99	2847.83	6.659	8.208	232.554

3.4 消融实验

为了更加有效地说明 HSNS-PSOGA 的可行性，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，消融实验的实验结果如表 2 所示。

由消融实验结果可知，本文算法-PSO 的收敛迭代次数更大，因此通过 PSO 能够提高收敛速度。本文算法-GA 的 DM 值更小，说明解的多样性差，因此通过 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 的值，能够增加解的均匀性。通过次优个体局部搜索， SM2 的值有明显的减小，说明 Pareto 最优解在目标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的方法，能够提升种群的质量。通过对种群中后 1/12 的个体打乱用户顺序，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，以探究比例的选取对算法性能的影响。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，在三个目标函数上的取值更优，在第 2、3、6、7、10 五个数据集上 SM1 指标更好，在第 2、5、8、10 四个数据集上 SM2 指标更好，说明解的分布更均匀，在第 2、4、5、8、10 五个数据集上 DM 指标更好，说明种群多样性更好。分析原因为：比例选取为 1/6 时，种群中更新比例大，可能陷入局部最优。比例选取为 1/24 时，更新比例较小，对算法作用不大，未能找到最优解。

4 结论

本文提出的混合全局局部搜索的非支配排序粒子群遗传算法，混合了进化多目标优化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粒子群和遗传算法，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且增加了 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了验证，在大多数情况下更优。本文在较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这个问题。

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.

[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: 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] 杜玟谛, 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 Min-shi, 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 nondominated 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 Edition), 2023, 53(4): 1007-1018.

[15] Hernandez C, Lara J, Arjona M A, et al. Electromagnetic 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]//Evolutionary Multi-Criterion Optimization: 7th International Conference, 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: 124344.

[20] Deb K, Jain H. An evolutionary many-objective optimization algorithm using reference-point-based nondominated 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]//2017 International Conference on Networks & Advances in Computational Technologies (NetACT). IEEE, 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]//Artificial Life and Computational Intelligence: Third Australasian Conference, Australia, 2017: 326-338.