

王菱, 潘大志. 近邻域搜索算法求解带容量约束的选址-路径问题[J]. 智能计算机与应用, 2024, 14(6): 102-109. DOI: 10.20169/j.issn.2095-2163.240614

近邻域搜索算法求解带容量约束的选址-路径问题

王菱^{1,2}, 潘大志^{1,2}

(1 西华师范大学 数学与信息学院, 四川 南充 637009;

2 西华师范大学 最优化理论与应用四川省高校重点实验室, 四川 南充 637009)

摘要: 针对遗传算法求解选址-路径问题存在收敛速度慢、易早熟等缺点, 提出了一种近邻域搜索算法 (Nearest Neighbor Search algorithm, NNS)。首先, 在构造初始解阶段利用贪婪策略产生初始种群, 提高初始解的质量; 其次, 在交叉阶段采用复制交叉的方式, 增强算法的全局搜索; 最后, 在突变阶段使用近邻域局部搜索过程。此外, 还设计了种群管理方案, 以促进遗传算法种群多样性。在2个基准集上进行了试验, 结果表明, 与 GRASP、MAPM、LRGTS 等算法相比, 该算法不仅显著提高了收敛速度, 而且具有较强的寻优能力。

关键词: 选址-路径; 贪婪算法; 遗传算法; 近邻域搜索

中图分类号: TP301.6

文献标志码: A

文章编号: 2095-2163(2024)06-0102-08

Solutions of the Location-Routing Problem with capacity constraints using Nearest Neighbor Search algorithm

WANG Ling^{1,2}, PAN Dazhi^{1,2}

(1 School of Mathematics and Information, China West Normal University, Nanchong 637009, Sichuan, China;

2 Key Laboratory of Optimization Theory and Applications at China West Normal University of Sichuan Province, China West Normal University, Nanchong 637009, Sichuan, China)

Abstract: The genetic algorithm has the shortcomings of slow convergence speed and easy precocious maturity to solve the location-routing problem, the Nearest Neighbor Search algorithm is proposed. Firstly, the greedy strategy is used to generate the initial population in the stage of constructing the initial solution to improve the quality of the initial solution. Secondly, in the intersection stage, the hybrid crossover mode of copying intersection is adopted to enhance the global search of the algorithm. Finally, the near-domain local search process is used in the mutation stage. In addition, population management programmes have been designed to promote genetic algorithm population diversity. Experiments are carried out on two benchmark sets, and the results show that compared with GRASP, MAPM, LRGTS and other algorithms, the proposed algorithm not only significantly improves the convergence speed, but also has strong optimization ability when solving the location-routing problem.

Key words: location-routing; greedy algorithm; genetic algorithm; neighborhood local search

0 引言

在选址-路径 (Location-Routing Problem, LRP) 问题中, 仓库的选址决策与车辆的路径决策是联合进行的, 相互独立地做出这些类型的决策可能会导致次优解^[1]。LRP 的研究已走过了几十年, 近年来也出现了许多关于选址-路径的变体问题, 其中带

容量选址-路径问题 (CLRP) 是目前研究最多的 LRP 变体。给定一组潜在的有容量的仓库位置和一组已知需求的客户, 目标是以最小化总成本的方式确定要开放的仓库和从开放的仓库为客户提供服务路线, 总成本由仓库和车辆的固定成本和可变路线成本组成。CLRP 是经典的 NP-hard 问题, 该问题在交通、物流等实际应用场景中广泛存在, 有效的

基金项目: 国家自然科学基金 (11871059); 四川省教育厅自然科学基金项目 (18ZA0469); 西华师范大学英才科研基金项目 (17YC385)。

作者简介: 王菱 (1999-), 女, 硕士研究生, 主要研究方向: 智能计算, 组合优化。

通讯作者: 潘大志 (1974-), 男, 博士, 教授, 硕士生导师, 主要研究方向: 智能计算, 算法设计。Email: pdzjz@126.com

收稿日期: 2023-11-14

求解将带来巨大的经济价值。

针对 CLRP 问题,提出了精确算法和启发式算法两种求解方法。其中,Akca 等学者^[2]提出了一种分支-价格算法。Belenguer 等学者^[3]提出了一种基于 0-1 线性模型的分支切断算法,并通过新的有效不等式族加以强化。最后,Contardo 等学者^[4]提出了一种分支-切割-列生成算法。当采用精确算法求解大规模组合优化 NP-hard 问题时,通常难以得到最优解,且求解效率不高,而智能算法能够在短时间内对大规模优化问题提供近似最优解。因此针对此类问题,Barreto 等学者^[5]提出了一种基于聚类的启发式算法解决无车辆成本的 CLRP 问题,采用多种聚类方法获取路由数据,利用折叠后的路由解决设施定位问题。Prins 等学者^[6]提出了一种基于贪婪随机自适应搜索过程的算法 (GRASP) 用于求解 CLRP 问题。Marinakis 等学者^[7]利用双层遗传算法分 2 个层次解决 CLRP 问题:一是解决有容量设施选址问题;二是为每个选取设施点解决车辆路径问题 (CVRP)。Xue 等学者^[8]针对有容量约束的情况,提出了遗传算法,既能搜索可行解空间,又能搜索不可行解空间。Hemmelmayr 等学者^[9]提出了一种自适应大邻域搜索的启发式算法求解两级车辆配送问题。Ting 等学者^[10]将 CLRP 分解为设施位置问题 (FLP) 和多仓库车辆路径问题 (MDVRP),应用分层蚁群结构,旨在优化不同的子问题。Pirkwieser 等学者^[11]、Ghaffari-Nasab 等学者^[12]、Escobar 等学者^[13]通过将变邻域搜索与其他邻域搜索结合,用于解决大规模 CLRP。

本文提出了一种近邻域搜索算法 (Nearest Neighbor Search algorithm, NNS) 求解 CLRP,算法遵循标准的遗传算法框架,在交叉阶段实行仓库-路线混合交叉的策略,并在变异阶段进行近邻域的局部搜索,改善了遗传算法的求解效率。通过比较 GRASP、MAPM、LRGTS 等算法在标准测试集上的实验结果后可知,所提算法对问题的规模适应性强,并取得较满意的求解结果。

1 选址-路径问题描述及数学模型

1.1 问题描述

带容量约束的选址路径问题 (CLRP) 定义为一个完全无向图 $G = (V, E)$, 这里 V 表示顶点集, E 表示边集。 V 由包含 n 个客户的子集 I 和包含 m 个潜在仓库的子集 J ($J = V - I$) 组成。每一个客户 $i \in I$ 都有一个非负的需求量 d_i , 被分配到一个容量为

w_j ($j \in J$) 的仓库中,其中分配给仓库的客户的总需求量不得超过仓库的容量。配送阶段由容量为 Q 的无限数量的同质车辆 k ($k \in K$) 完成,且每辆车的装载量不得超过 Q 。当仓库启用时产生固定成本 f_j , 每辆车的启动费用为 C_0 , 每次产生的旅行费用为 C_{ij} , 其中 $(i, j) \in E$, 目标是使得仓库的成本、车辆的费用以及旅行费用之和最少。

1.2 数学模型

目标函数:

$$\min \sum_{j \in J} f_j y_j + \sum_{i \in I} \sum_{j \in J} \sum_{k \in K} C_0 x_{ijk} + \sum_{i \in V} \sum_{j \in V} \sum_{k \in K} C_{ij} x_{ijk} \quad (1)$$

约束条件:

$$\sum_{i \in I} z_{ij} d_i \leq w_j y_j, \quad \forall j \in J \quad (2)$$

$$\sum_{j \in V} \sum_{i \in I} x_{ijk} d_i \leq Q, \quad \forall k \in K \quad (3)$$

$$\sum_{k \in K} \sum_{j \in V} x_{ijk} = 1, \quad \forall i \in I \quad (4)$$

$$\sum_{j \in J} \sum_{i \in I} x_{ijk} \leq 1, \quad \forall k \in K \quad (5)$$

$$\sum_{j \in J} z_{ij} = 1, \quad \forall i \in I \quad (6)$$

$$x_{ijk} = 0, \quad \forall i, j \in J, \quad k \in K \quad (7)$$

$$\sum_{i \in I} z_{ij} \geq y_j, \quad \forall j \in J \quad (8)$$

$$z_{ij} \leq y_j, \quad \forall i \in I, \quad \forall j \in J \quad (9)$$

$$\sum_{k \in K} \sum_{i \in I} x_{ijk} \geq y_j, \quad \forall j \in J \quad (10)$$

$$\sum_{i \in I} x_{ijk} \leq y_j, \quad \forall i \in I, \quad \forall k \in K \quad (11)$$

$$x_{ijk} \in \{0, 1\} \quad (12)$$

$$y_j \in \{0, 1\} \quad (13)$$

$$z_{ij} \in \{0, 1\} \quad (14)$$

目标函数 (1) 表示最小化成本,包括仓库建设成本、车辆启用成本、车辆运输成本;式 (2) 表示仓库分配客户的总需求量不得超过仓库的容量;式 (3) 表示每一辆车服务客户的总需求量要小于等于车辆的最大装载量;式 (4) 表示每个客户有且只有一辆车为其服务;式 (5) 表示每辆车最多只能启用一次;式 (6) 表示一个客户只能分配到一个仓库;式 (7) 表示每一个仓库独立运行,即任意 2 个仓库之间没有路线连接;式 (8)、式 (9) 表示只有当仓库打开时,客户才能被其服务;式 (10)、式 (11) 表示只有仓库开放时,才有车辆从仓库出发;式 (12) 表示车辆经过弧 (i, j) 为 1, 否则为 0;式 (13) 表示仓库 j 打开为 1, 否则为 0;式 (14) 表示客户 i 由仓库 j 服务为 1, 否则为 0。

2 近邻域搜索算法

遗传算法起源于对生物系统所进行的计算机模拟,从任一初始种群出发,通过随机选择、交叉和变异操作产生一群更适合环境的个体,是一种随机全局搜索优化的方法。根据 CLRP 问题的具体特征,本文将基础的遗传算法与贪心思想和近邻域局部搜索相结合,设计了一种求解 CLRP 的混合遗传算法。首先,将初始解的生成方式运用贪心算法替换,生成质量相对较高的初始解;然后,对客户所属仓库以及相关配送路线进行交叉操作。不同于将 CLRP 分成 CLP 和 CVRP 两个问题进行交叉,本文的交叉操作同时考虑客户分配阶段与路线配送阶段;最后,由近邻域搜索操作



图1 编码示意图

Fig. 1 Schematic diagram of encoding

2.2 初始种群生成

初始解的质量对于算法收敛起着至关重要的作用,本文根据贪心算法思想^[14]生成初始解,具体步骤如下:

步骤1 将潜在的候选设施点随机排列,从一个仓库开始,依次分配客户。设 C 为未被分配的客户的集合, sub_C 为 C 的一个子集,表示未被分配客户中需求量不超过仓库剩余容量的集合;

步骤2 选取集合 sub_C 中离仓库最近的客户 i 开启一条新的路线;

步骤3 在 sub_C 中选取离客户 i 最近的客户加入该路线,如果选取客户的需求量大于车辆剩余容量,则继续在 sub_C 中选取离 i 最近且满足需求量不超过车辆剩余容量的客户;若 sub_C 中所有客户都不满足条件,转至步骤2;否则将当前选择的客户设置为新的 i ,重复步骤3;

步骤4 判断 sub_C 是否为空,若不为空,则直

改进原始算法,针对邻域搜索操作有较大的改动。

2.1 编码

遗传算法中的染色体表示一个完整的解,即路径集合。基因长度和染色体长度都是可变的,取决于服务的客户端数量的解决方案中的路径数量。例如,给定一个具有4个可选仓库{1,2,3,4}和11个客户端{5,6,7,8,9,10,11,12,13,14,15},解决方案的染色体表示如图1所示。图1中的解决方案表示选取的仓库为1和3,4条路线分别为{1-8-5-12-1}, {1-10-7-1}, {3-11-15-6-3}, {3-13-14-9-3},车辆从仓库出发,受车辆容量的限制,最终回到仓库。通过此种编码方式,仓库开放、客户归属、基于每个仓库构建的配送路线清晰可见。

接返回步骤2,否则,打开下一个仓库,返回步骤2。

2.3 交叉算子

在交叉阶段,父代的选择极大程度影响着子代的质量,本文采用 K-tournament^[8]和二元锦标赛两种方式。其中, K-tournament 方法能够产生高质量子代个体,把种群中第 K 个最优个体作为父本 $P1$,由二元锦标赛选择得到父本 $P2$,在此过程中,最多产生 K 个后代,选择最优个体加入种群。交叉操作的具体流程如下。

将 $P1$ 的 p 条路线复制给子代,未访问的客户按照 $P2$ 的原始顺序插入到子代,在新的路径中,优先使用当前已打开的仓库,如果当前打开的仓库不能装下所有客户,则随机打开一个新的仓库。交叉方式如图2所示。图2中,假设 $P1$ 的路线1和路线3直接遗传给子代,其余未包含在子代中客户在遵循仓库容量和车辆容量的条件下,依次插入子代路线。

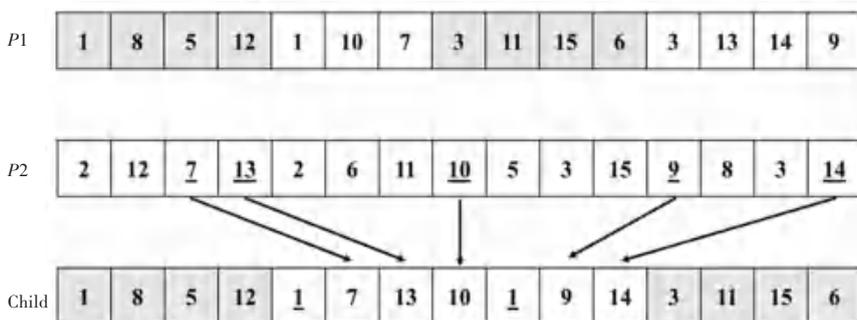


图2 交叉示意图

Fig. 2 Schematic diagram of crossover

2.4 变异算子

遗传算法的主要组成部分之一是突变算子, 本文的变异策略为随机增加一个仓库。由于本文的交叉操作容易使得所求解快速收敛到具有少量仓库的解决方案, 所以在仓库容量允许的范围内, 通过随机

分配 q 条路线到新仓库来增加仓库的多样性。变异方式如图 3 所示。图 3 中, 选取第 2 条和第 3 条路线, 随机打开的仓库序号为 2, 在满足容量约束的情况下, 将这 2 条线路的客户放入 2 号仓库中。

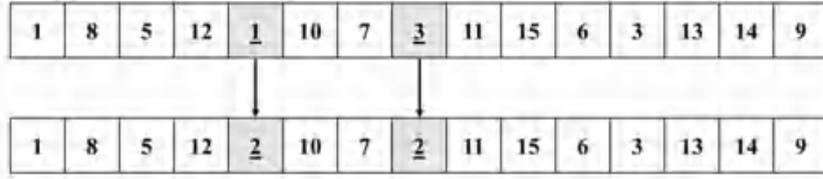


图 3 变异示意图

Fig. 3 Schematic diagram of the mutation

2.5 近邻域搜索

在解决 CLRP 问题时, 往往需要使用局部搜索来改进路径。针对 CLRP 的特性, 本文采用近邻域局部搜索的方式进一步改进客户分配以及路径规划。每个顾客 u 的邻域 $N(u)$ 定义为离 u 距离最近的客户的集合, 其中 $nsize$ 为邻域大小。邻域局部搜索主要使用 3 种经典的邻域搜索算子: Swap^[15]、Relocate^[16] 以及 2-opt^[17]。为了实现快速的路线改进, 邻域局部搜索总是以迭代的方式对每个客户进行最优移动。对于每一个客户 u , 邻域搜索在 u 和其邻域 v ($v \in N(u)$) 上进行移动, 如果可以改进路线, 则对 u 执行最佳移动。邻域搜索算法步骤具体如下:

步骤 1 将个体 P 中的客户随机排序, 设置邻域大小 $nsize$ 以及每次迭代路线优化的最低次数 $lsize$;

步骤 2 初始化计数器 $count = 0$, 遍历 P 中的客户 u ;

步骤 3 找寻 u ($u \in P$) 的邻域客户 v ($v \in N(u)$);

步骤 4 进行 Swap 操作, 交换 u 和 v 的位置, 重新计算相关路线的需求总量, 若不满足容量约束, 则返回步骤 3, $v = v + 1$, 反之, 则继续;

步骤 5 进行 Relocate 操作, 在满足容量约束的情况下将客户重新插入新的路线;

步骤 6 进行 2-opt 操作, 并计算路径优化的次数 $count$, 若 $count > lsize$, $u = u + 1$, 返回步骤 2, 否则终止循环。

2.6 种群多样性

为了避免种群陷入局部最优, 引入种群多样化

策略。当前最优解没有达到全局最优解时, 按照随机排列客户顺序, 采用轮盘赌选择方法依次为每个客户选择一个仓库, 从而完成分配。通过随机生成新的个体取代较差的个体, 增加种群的多样性, 同时保存适应度较好的个体。对于每一个客户 i , 每个仓库 d 的概率如下:

$$Prob(d, i) = \frac{Prob_d}{C_{d,i} + C_{i,d}} \quad (15)$$

其中, $Prob_d$ 表示仓库 d 的全局概率, 计算公式如下:

$$Prob_d = \frac{Q_d}{C_d} + \frac{Q_d}{TC_d} + \frac{n_d}{n} \quad (16)$$

其中, Q_d 和 C_d 分别表示仓库的容量和开设成本; n_d 的值基于贪婪序列构造, 对于每一个仓库 d , 从离 d 最近的客户开始, 反复选择离 d 最近但还没有被选中的顾客追加到序列末尾, 构造一个贪婪序列 Π_d , 直到仓库容量用完; TC_d 表示在不考虑车辆约束的情况下, 依次计算 Π_d 内客户的出行成本。

2.7 算法流程

算法流程如图 4 所示。

3 实验结果以及分析

3.1 测试算例

本文选取 Prodhon^[6] 和 Barreto^[18] 两组标准算例进行测试, 并将其测试结果与解决该问题的几种代表性算法 GRASP^[8]、MAPM^[19]、LRGTS^[20] 进行对比, 验证 NNS 算法的寻优性。相关实验均在同一实验环境下完成: 英特尔 i5 处理器, 2.50 GHz 主频, 8 G 的 RAM, 编程环境为 Matlab2021a。

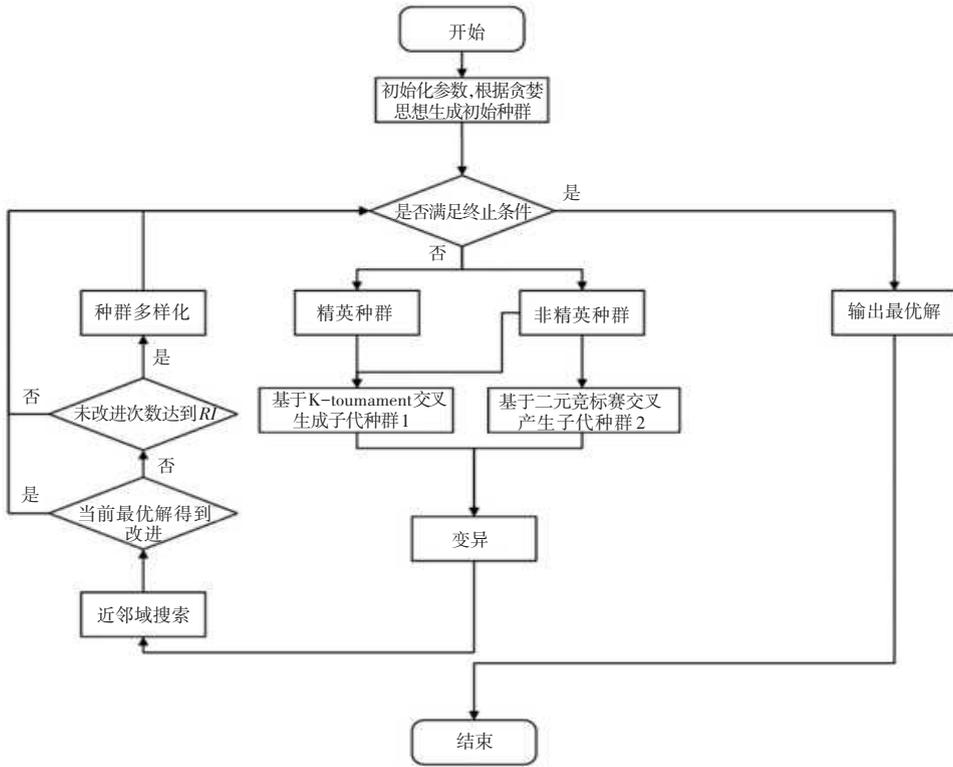


图4 算法流程图

Fig. 4 Flow chart of the algorithm

3.2 参数设置

为了获取参数的最佳组合情况,各参数的取值使用控制变量法进行调整。交叉概率设置为0.8,变异概率为0.2,种群多样化未改进迭代次数 $RI = 5$, K-tournament 子代 K 为10, p, q 为 $\frac{N}{3} \sim \frac{2N}{3}$ 之间的随机整数 (N 表示个体中路线条数); 近邻域大小 $nsize$ 和最低优化次数 $Isize$ 分别设置为 $\frac{n}{4}$ 和 $\frac{n}{10}$ (n 表示客户总数)。

3.3 实验结果

Prodhon 和 Barreto 的基准测试集分别由12个实例和10个实例组成,客户数量从20到88,仓库数量为5到8。为验证算法的性能,实验中每个算法均独立运行20次,实验结果见表1、表2。表1、表2中, BKS 为每个实例目前最优解决方案成本。

从表1、表2不难看出,对于小规模问题,大多数算法能达到最优,但对于中规模和大规模问题来说, NNS 算法有更强的寻优能力。

表1 Prodhon 算例结果对比

Table 1 Comparison of the results of the Prodhon study

序号	instance	BKS	GRASP	MAPM	LRGTS	NNS
1	20-5-1a	54 793	55 021	54 793	55 131	54 793
2	20-5-1b	39 104	39 104	39 104	39 104	39 104
3	20-5-2a	48 908	48 908	48 908	48 908	48 908
4	20-5-2b	37 542	37 542	37 542	37 542	37 542
5	50-5-1a	90 111	90 632	90 160	90 160	90 111
6	50-5-1b	63 242	64 761	63 242	63 256	63 242
7	50-5-2a	88 298	88 786	88 298	88 715	88 298
8	50-5-2b	67 340	68 042	67 893	67 698	67 340
9	50-5-3a	86 203	87 380	86 203	86 203	86 203
10	50-5-3b	61 830	61 890	61 830	61 830	61 830

表 2 Barreto 算例结果对比

Table 2 Comparison of the results of the Barreto study

序号	instance	BKS	GRASP	MAPM	LRGTS	NNS
1	Gaskell67-21x5	424.9	429.6	424.9	424.9	424.9
2	Gaskell67-22x5	585.1	585.1	611.8	587.4	585.1
3	Gaskell67-29x5	512.1	515.1	512.1	512.1	512.1
4	Gaskell67-32x5	562.2	571.9	571.9	584.6	562.2
5	Gaskell67-32x5	504.3	504.3	534.7	504.8	504.3
6	Gaskell67-36x5	460.4	460.4	485.4	476.5	460.4
7	Min92-27x5	3 062.0	3 062.0	3 062.0	3 065.2	3 062.0
8	Christofides69-50x5	565.6	599.1	565.6	586.4	561.7
9	Christofides69-75x10	861.6	861.6	866.1	863.5	861.6
10	Daskin95-88x8	355.8	356.9	355.8	368.7	355.8

表 3 为不采用近邻域搜索和采用近邻域搜索的结果对比,图 5 是算例 coord20-5-1b 和 coord50-5-2b 采用近邻域搜索和普通局部搜索的迭代进化曲线。从图 5 中可以看出,对于小规模算例来说,普通

邻域搜索虽然能够达到最优值,但是容易陷入局部最优,而对于中、大规模算例而言,普通邻域搜索不仅会陷入局部最优,而且达不到最优解,但近邻域搜索能更快收敛到最优值。

表 3 普通邻域搜索与近邻域搜索结果对比

Table 3 Comparison of normal neighborhood search results with near neighborhood searches

算例	普通邻域搜索	近邻域搜索	算例	普通邻域搜索	近邻域搜索
20-5-1a	55 021	54 793	Gaskell67-21x5	424.9	424.9
20-5-1b	39 104	39 104	Gaskell67-22x5	585.1	585.1
20-5-2a	48 908	48 908	Gaskell67-29x5	512.1	512.1
20-5-2b	37 542	37 542	Gaskell67-32x5	567.1	562.2
50-5-1a	91 450	90 111	Gaskell67-32x5	505.8	504.3
50-5-1b	63 939	63 242	Gaskell67-36x5	476.1	460.4
50-5-2a	89 428	88 298	Min92-27x5	3 062.0	3 062.0
50-5-2b	68 665	67 340	Christofides69-50x5	597.0	561.7
50-5-3a	86 748	86 203	Christofides69-75x10	877.7	861.6
50-5-3b	63 065	61 830	Daskin95-88x8	361.5	355.8

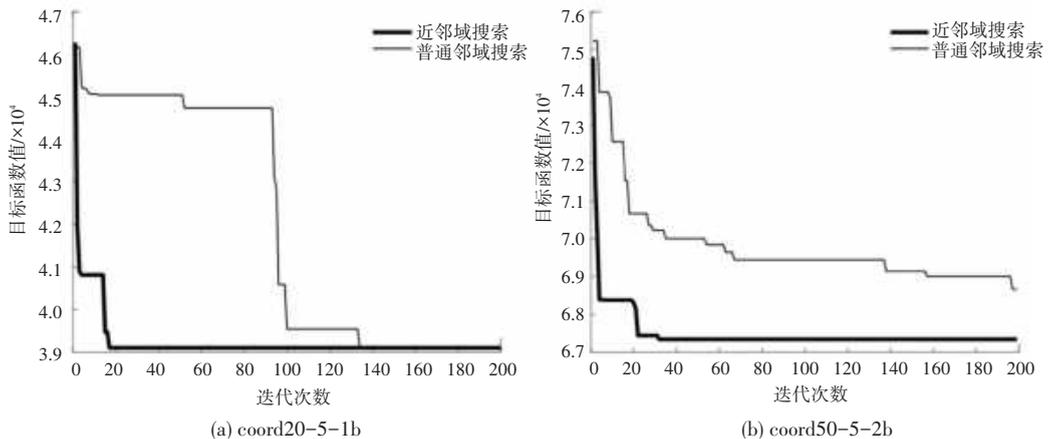


图 5 算例迭代图

Fig. 5 Iterative diagram of the study

为了测试算法的稳定性,将 NNS 应用于不同规模的问题算例研究。这里,小规模算例:coord20-5-2b;中规模算例:coord50-5-1b、Christofides69;大规

模算例:coordDas88。实验结果如图 6 所示。从图 6 中可以看出,NNS 无论是小规模、中规模、还是大规模算例,都能取得较好的结果。

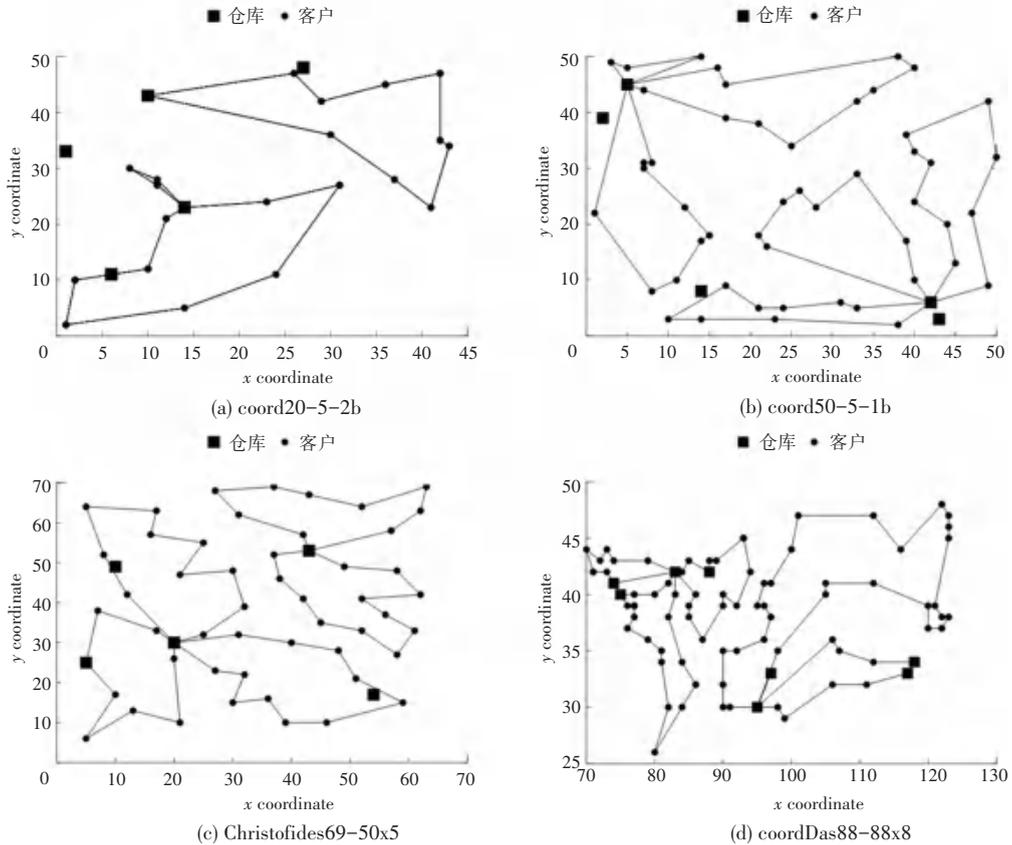


图 6 最优路径图

Fig. 6 Optimal path diagram

4 结束语

本文针对有容量选址-路径问题进行了研究,在满足客户需求、仓库容量以及车辆容量的条件下,通过合理地调派车辆、规划行车路径,进而节约配送成本。近邻域搜索算法利用与已知客户相近邻域进行邻域搜索,既提高了解的质量,又保证了种群的多样性,避免陷入局部最优,相较于其他算法,该算法在求解质量和时间效率两方面都更具优势。新算法的求解方案可以充分降低成本,提高仓库以及车辆容量的利用率,减少资源浪费,对于交通、物流等行业有较强的借鉴意义。

参考文献

[1] SALHI S, RAND G K. The effect of ignoring routes when locating depots[J]. *European Journal of Operational Research*, 1989, 39(2): 150-156.
[2] AKCA Z, BERGER R T, RALPHS T K. A branch-and-price

algorithm for combined location and routing problems under capacity restrictions[M]// CHINNECK J W, KRISTJANSSON B, SALTZMAN M J. *Operations Research and Cyber - Infrastructure. Operations Research/Computer Science Interface*. Boston, USA: Springer, 2009, 47: 309-330.
[3] BELENGUER J M, BENAVENTE E, PRINS C, et al. A branch-and-cut method for the capacitated location-routing problem[J]. *Computers & Operations Research*, 2011, 38(6): 931-941.
[4] CONTARDO C, CORDEAU J F, GENDRON B. An exact algorithm based on cut-and-column generation for the capacitated location-routing problem[J]. *Informatics Journal on Computing*, 2014, 26(1): 88-102.
[5] 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.
[6] PRINS C, PRODHON C, CALVO R W. Solving the capacitated location-routing problem by a GRASP complemented by a learning process and a path relinking[J]. *4OR*, 2006, 4(3): 221-238.
[7] MARINAKIS Y, MARINAKI M. A bilevel genetic algorithm for a real life location routing problem[J]. *International Journal of Logistics: Research and Applications*, 2008, 11(1): 49-65.
[8] XUE Yu, ZHOU Yuren, LIU Xiaofang. A novel hybrid genetic

- algorithm for the location routing problem with tight capacity constraints[J]. *Applied Soft Computing*, 2019, 85: 105760.
- [9] HEMMELMAYR V C, CORDEAU J F, CRAINIC T G. An adaptive large neighborhood search heuristic for two - echelon vehicle routing problems arising in city logistics[J]. *Computers & operations research*, 2012, 39(12): 3215-3228.
- [10] TING C J, CHEN C H. A multiple ant colony optimization algorithm for the capacitated location routing problem [J]. *International Journal of Production Economics*, 2013, 141(1): 34-44.
- [11] PIRKWIESER S, RAIDL G R. Variable neighborhood search coupled with ILP-based very large neighborhood searches for the (periodic) location-routing problem[C]// *Proceedings of Hybrid Metaheuristics: 7th International Workshop*. Berlin/Heidelberg: Springer, 2010: 174-189.
- [12] GHAFFARI-NASAB N, JABALAMELI M S, Aryanezhad M B, et al. Modeling and solving the bi-objective capacitated location-routing problem with probabilistic travel times [J]. *The International Journal of Advanced Manufacturing Technology*, 2013, 67: 2007-2019.
- [13] ESCOBAR J W, LINFATI R, BALDOQUIN M G, et al. A Granular Variable Tabu Neighborhood Search for the capacitated location-routing problem [J]. *Transportation Research Part B: Methodological*, 2014, 67: 344-356.
- [14] 周迅, 张惠珍. 求解开放式选址路径问题的离散烟花算法 [J]. *软件导刊*, 2021, 20(3): 43-50.
- [15] OSMAN I H. Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem [J]. *Annals of Operations Research*, 1993, 41: 421-451.
- [16] PITAKASO R, SETHANAN K, THEERAVIRIYA C. Variable neighborhood strategy adaptive search for solving green 2-echelon location routing problem [J]. *Computers and Electronics in Agriculture*, 2020, 173: 105406.
- [17] NGUYEN V P, PRINS C, PRODHON C. Solving the two-echelon location routing problem by a GRASP reinforced by a learning process and path relinking [J]. *European Journal of Operational Research*, 2012, 216(1): 113-126.
- [18] 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.
- [19] PRINS C, PRODHON C, CALVO R W. A memetic algorithm with population management (MA | PM) for the capacitated location - routing problem [C]// *European Conference on Evolutionary Computation in Combinatorial Optimization*. Berlin/Heidelberg: Springer, 2006: 183-194.
- [20] PRINS C, PRODHON C, RUIZ A, et al. Solving the capacitated location-routing problem by a cooperative Lagrangean relaxation-granular tabu search heuristic [J]. *Transportation Science*, 2007, 41(4): 470-483.