

混合分散搜索算法求解带容量约束车辆路径问题

张晓楠, 范厚明

(大连海事大学 a. 交通运输管理学院, b. 战略管理与系统规划研究所, 辽宁 大连 116026)

摘要: 设计一种解决带容量约束车辆路径问题的混合分散搜索算法. 在基本分散搜索的基础上, 保留参考集更新策略和组合策略的全局搜索能力. 采用随机插入法作为解的多样性产生方法, 以扩大搜索空间, 避免陷入局部最优. 应用简化的变邻域搜索作为改进策略进行局部开发, 引入邻域半径减少策略提高开发效率. 对改进后的新种群实施精英保留策略, 保证算法收敛. 实验结果分析表明, 混合分散搜索算法优于所对比的算法, 寻优能力可靠.

关键词: 带容量约束车辆路径问题; 随机插入法; 分散搜索; 变邻域搜索

中图分类号: TP18

文献标志码: A

Hybrid scatter search algorithm for capacitated vehicle routing problem

ZHANG Xiao-nan, FAN Hou-ming

(a. School of Transportation Management, b. Institute of Strategy Management and System Planning, Dalian Maritime University, Dalian 116026, China. Correspondent: FAN Hou-ming, E-mail: fhm468@163.com)

Abstract: A hybrid scatter search algorithm(HSSA) for solving the capacitated vehicle routing problem is proposed. Based on the basic scatter search, the reference set update method(RSUM) and the solution combination method(SCM) are applied to search the global-space. In order to expand the search space and prevent the local minimum, a random insertion method is used as the diversification generation method(DGM). A simplified variable neighborhood search(VNS), as the solution improvement method(SIM), is developed to search the local-space. To improve the efficiency of the SIM, a neighborhood size reduction scheme(NERS) is applied. The Elitism strategy to ensure convergence is introduced. The computational results show that the HSSA can find the global optimal solution with high performance, better than comparison algorithms.

Keywords: capacitated vehicle routing problem; random insertion method; scatter search; variable neighborhood search

0 引言

带容量约束的车辆路径问题(CVRP)是车辆路径问题(VRP)的扩展,属于经典组合优化的NP难题,具有较为广泛的工程应用和现实生活背景.

当前,CVRP并未得到完全解决,如何快速、有效地求解具有很高的实际应用价值,智能优化算法成为研究的重要方向.文献[1]提出利用双种群遗传算法求解,两个种群执行不同进化,并交换种群间的精英个体信息;文献[2]将CVRP转化成准连续优化问题,提出了改进微粒群算法;文献[3]提出了捕食搜索算法;文献[4]提出了量子进化算法.然而,算法的单独使用往往表现出自身的缺陷,如文献[1]的算法收敛速度慢、易陷入局部最优;文献[4]的算法虽不易陷入局部最优,但搜索能力较弱.算法的混合使

用能实现优化性能的互补,获得较为理想的求解效果.文献[5]提出了结合2-Opt子路径优化的遗传算法;文献[6]提出了改进的蚁群算法,以路径节约值为启发式信息,动态更新信息素,2-Opt、交换和插入进行局部搜索;文献[7]提出了混合粒子群和变邻域搜索的PSO-VNS算法,变邻域包括2-Opt和3-Opt;文献[8]提出了混合节约算法和集合覆盖的SC-ESA算法.

分散搜索(SS)是一种种群进化算法^[9],与量子进化算法相同,该算法也具有不易陷入局部最优的特点,且灵活性好、易与其他算法结合,近年来在VRP中有很好的应用^[10-11].变邻域搜索(VNS)是文献[12]提出的一种元启发式算法,也在VRP中展现了良好的搜索性能^[13],近年来多与禁忌搜索^[14]、人工蜂群算法^[15]等混合应用,可加快寻优速度,增强逃离局部最

收稿日期: 2014-11-14; 修回日期: 2015-04-03.

基金项目: 国家自然科学基金项目(70801007, 61473053); 辽宁省软科学指导计划项目(2012401005); 辽宁省教育厅科学技术研究一般项目(L2014196); 大连市科学技术计划项目(2010A16GX084).

作者简介: 张晓楠(1988-),女,博士生,从事物流系统优化设计和智能优化算法的研究; 范厚明(1962-),男,教授,博士生导师,从事战略管理与系统规划等研究.

优的能力.

本文设计由分散搜索和变邻域搜索组成的混合分散搜索算法(HSSA)求解CVRP问题. 实验结果分析表明, 混合分散搜索算法优于所对比的算法, 寻优能力可靠.

1 基本问题描述

1.1 CVRP问题

CVRP可描述为: 1个配送中心最多可用 m 辆车访问 n 个客户, 每个客户必须访问且只能访问一次, 目标是寻找起止配送中心且满足车辆容量限制的服务路径, 以最小化行驶距离. 0表示配送中心, 可用车辆集合 $K = \{1, 2, \dots, m\}$, 客户集合 $J = \{1, 2, \dots, n\}$, 所有点集合 $V = \{0, 1, \dots, n\}$, 边集合 $E = \{(i, j) | i, j \in V\}$, 客户需求为 d_j , 车辆容量限制为 CV , 两点间配送距离为 c_{ij} . 决策变量为

$$x_{ijk} = \begin{cases} 1, & \text{边}(i, j) \text{有直接路径且服务车辆为} k; \\ 0, & \text{otherwise.} \end{cases}$$

$$y_{jk} = \begin{cases} 1, & \text{客户} j \text{由车辆} k \text{服务}; \\ 0, & \text{otherwise.} \end{cases}$$

建立如下数学模型:

$$\min \sum_{k \in K} \sum_{i \in V} \sum_{j \in V} c_{ij} x_{ijk}. \quad (1)$$

$$\text{s.t.} \sum_{i \in V} \sum_{j \in J} x_{ijk} d_j \leq CV, \forall k \in K; \quad (2)$$

$$\sum_{k \in K} \sum_{i \in V} x_{ijk} \doteq 1, \forall j \in J; \quad (3)$$

$$x_{iik} \doteq 0, \forall i \in V, \forall k \in K; \quad (4)$$

$$\sum_{i \in V} x_{ijk} - \sum_{i \in V} x_{jik} \doteq 0, \forall j \in V, \forall k \in K; \quad (5)$$

$$\sum_{j \in J} x_{0jk} \leq 1, \forall k \in K; \quad (6)$$

$$\sum_{i \in S} \sum_{j \in S} x_{ijk} \leq |S| - 1, \forall S \subseteq J, \forall k \in K; \quad (7)$$

$$\sum_{k \in K} y_{jk} \doteq 1, \forall j \in J; \quad (8)$$

$$\sum_{i \in V} x_{ijk} \doteq y_{jk}, \forall j \in J, \forall k \in K; \quad (9)$$

$$\sum_{j \in V} x_{ijk} \doteq y_{ik}, \forall i \in J, \forall k \in K; \quad (10)$$

$$y_{jk} \in \{0, 1\}, \forall j \in J, \forall k \in K; \quad (11)$$

$$x_{ijk} \in \{0, 1\}, \forall i, j \in V, \forall k \in K. \quad (12)$$

其中: 式(1)为目标函数; 式(2)为车辆容量约束; 式(3)~(7)为服务路径相关约束; 式(8)保证每个客户有且仅由一辆车服务; 式(9)和(10)为决策变量间关系式; 式(11)和(12)为决策变量属性.

1.2 分散搜索算法(SS)

SS与遗传算法GA类似, 也是一种全局搜索算法. 传统GA往往存在收敛速度慢、易陷入局部最优和优化精度低的缺点, 如何在提高收敛速度的同时确保种群多样性, 使寻优结果接近最优解是其不断改进的目标^[16]. SS和GA的本质区别在于参考集中多样性解的引入和组合策略中子集对的选择组合方式不同. 算法不再基于随机性, 而是运用“分散-收敛集聚”的迭代机制、解的多样性产生方法、参考集更新策略和组合策略以保证种群多样性(全局搜索能力), 改进策略保证收敛速度(局部开发能力). 在很多文献中已证明SS较GA更好^[10-11], SS具体流程如图1所示.

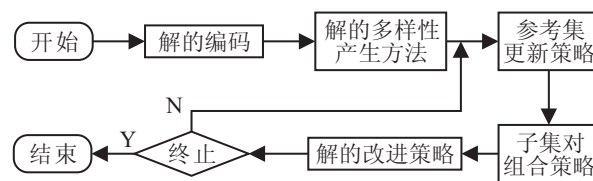


图1 SS流程

1) 解的多样性产生方法, 类似GA的种群初始化, 因参考集有多样性要求, 这里多采用兼顾解质量和多样性的方法.

2) 参考集更新策略, 类似GA的精英种群. 因其不仅要求质量高, 还要兼有多样化, 故由精英解集和多样性解集构成.

3) 子集对组合策略, 类似GA的选择和交叉. 子集对类似GA的父代, 新解类似子代. 不同的是SS的子集对两两随机组成, 组合无概率影响, GA的父代需轮盘赌选择后两两分组而成, 交叉需依据交叉概率.

4) 解的改进策略, 类似GA的变异, 也可看作改进GA的局部搜索策略.

1.3 变邻域搜索算法(VNS)

VNS在局部搜索过程中有规律地改变邻域结构以增强算法搜索能力, 由局域搜索过程和改变邻域过程两部分构成. 局域搜索过程在同一个邻域结构内寻求局部最优解; 改变邻域过程在局部最优解的基础上改变邻域结构, 使接下来的求解跳出局部最优.

2 求解CVRP的混合分散搜索算法

2.1 解的编码

采用自然数编码方法, 如1个配送中心、3个可用车辆和8个客户的编码例子为58013054820760, 客户排列顺序表示车辆服务顺序, 不同车辆用0隔开.

2.2 解的多样性产生方法

采用插入法构建初始解^[17], 随机参数控制解的多样性, 多个解组成初始种群. 插入位置可以是已派

表 1 w_1 和 w_2 的参数影响

w_1	w_2		
	[0, 1]	[0, 2]	[0, 3]
[0,1]	1 113 ~ 1 276 (24/300)	1 071 ~ 1 276 (71/300)	995 ~ 2 008 (85/300)
[0, 2]	1 096 ~ 1 255 (32/300)	1 055 ~ 1 957 (79/300)	1 055 ~ 2 008 (93/300)
[0, 3]	1 096 ~ 1 337 (45/300)	1 093 ~ 2 008 (96/300)	1 096 ~ 2 008 (105/300)

出车辆的已有路径弧段, 也可以是新派出车辆. 客户 j 插入弧段 g, h 的插入费用为 $\text{cost}_{ghj} = c_{gj} + c_{jh} - w_1 c_{gh} + w_2 |c_{gj} - c_{jh}|$, w_1 和 w_2 为随机数.

表 1 以算例 A-n32-k5 为例展示了 w_1 和 w_2 对初始种群 (种群规模 $P_{\text{size}} = 300$) 的影响, 当 $w_1, w_2 \in [0, 1]$ 时, 种群解波动范围 1 113 ~ 1 276, 多样性解 24 个, 随着 w_1, w_2 取值范围增大, 解波动范围增大, 搜索空间增大, 多样性解增多. 综合考虑多样性和高质量要求, 取 $w_1 \in [0, 2], w_2 \in [0, 1]$.

图 2 以算例 A-n48-k7 为例 ($P_{\text{size}} = 500$) 展示了当 $w_1 \in [0, 2], w_2 \in [0, 1]$ 时随机插入法与随机贪婪法的初始种群解统计分布图, 贪婪法的多样性解数量始终小于客户规模, 且相同规模下, 随机插入法构造的初始种群多样性解和高质量解比例较高, 有助于加快寻优速度, 同时避免陷入局部最优. 较多的多样性解也可避免后续出现相同子集对的重复组合.

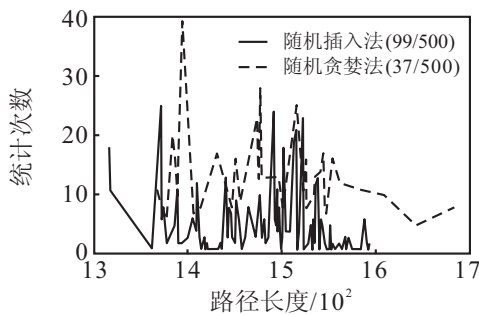


图 2 初始种群解统计分布

2.3 参考集更新策略

设参考集为 B , 精英解集和多样性解集为 B_1 和 B_2 . B_1 由目标函数最好的 $|B_1|$ 个解组成, B_2 由非精英解集中多样性距离最大的 $|B_2|$ 个解组成, $B = B_1 \cup B_2$. 某一非高质量解的多样性距离用该解与所有高质量解的多样性距离的最小值表示^[11], 两解间的多样性距离等于两解弧数最大值减去两解共有弧数.

2.4 子集对组合策略

在参考集 B 中, 随机选择两解组成子集对, 对每个子集对采用组合策略产生新解. 本文的组合策略将每个子集对产生两个新解: 以子集对 X_1 和 X_2 为例, 随机产生交叉点 θ , 保留 X_1 中 θ 之前的基因位, 其余基因按 X_2 的客户排列顺序删除已服务客户点, 再依次复制到相应位置, 形成新解 Y_1 ; 新解 Y_2 类似. 以解 013054820760 和 016034720580 为例, 客户排列顺序

分别为 13548276 和 16347258. 设交叉点 $\theta = 5$, 新解 Y_1 和 Y_2 分别为 013056470280 和 016035480270. 组合后新解可能是不可行解, 不可行解添加罚函数 P_e 修正.

表 2 以算例 A-n32-k5 的某一种群为例展示了 SS “参考集 + 子集对组合” 策略和 GA “精英个体 + 父代交叉” 策略产生的新解 10 次实验平均值 ($P_{\text{size}} = 20, B_1 = 10, B_2 = 5$, 精英个体数 15), SS 进化机制解搜索空间更大, 多样性解更多, 全局搜索能力更强. 较多的多样性新解可避免后续出现相同解的重复改进.

表 2 不同策略的新解

操作策略	新解空间 (种群规模 20)	新解最小值
参考集 + 子集对组合	1 112 ~ 1 992 (约 18/20)	1 112
精英个体 + 父代交叉	1 123 ~ 1 930 (约 16/20)	1 123

图 3 是算例 A-n48-k7 的某次实验结果统计分布图 ($P_{\text{size}} = 60, B_1 = 15, B_2 = 5$, 精英个体数 20), SS 的 “参考集 + 子集对组合” 的新解整体多样性为 51/60, 其中出现次数最多的是路径长度为 1 537 的解, 出现次数为 3 次; GA 的 “精英个体 + 父代交叉” 的子代整体多样性为 30/60, 多样性解较少, 且其中出现次数最多的是路径长度为 1 427 的解, 出现次数为 27 次, 这也意味着后续将对 1 427 的解重复改进 27 次.

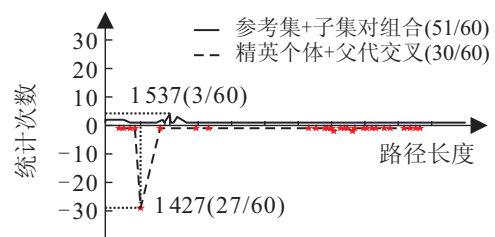


图 3 结果统计分布

2.5 解的改进策略

采用简化的变邻域搜索 (VNS), 即在变更邻域时严格按照变更顺序依次变更, 不重复回到最初邻域结构再次搜索. 这里采用 2-Opt、1-1 Exchange、0-1 Exchange 三种邻域结构, 每种邻域的终止条件均为连续 MAXCount 次搜索到的最好解不变. 为提高搜索效率, 引入邻域半径减少策略 (NERS)^[17], 具体如下:

1) 弧交换 (2-Opt). 任意选中两个不相邻弧组成弧对, 若属于同一路线, 则删除它们创建两条新弧并逆序之间路径; 若属于不同路线, 则删除它们将所在路线分为开始和结尾部分, 交叉相连形成新路线, 新

路线属于原配送中心. 引入 NERS 后的 2-Opt 以客户为对象, 设任意选中客户 j , avg_j 为 j 与其他客户的距离平均值, 选择满足 $c_{ji} < avg_j$ 的所有 i 点 ($i \in J$) 组成集合 NV_j . 弧对组合缩减为路线上 j 的两条附近弧与 NV_j 中客户的两条附近弧, 逆序和交叉相连均可以使 j 与 NV_j 中客户能直接相连的方式为主. 考虑所有可能弧对, 以最好解作为客户 j 的 2-Opt 邻域解.

2) 客户交换 (1-1 Exchange). 任意选中两客户组成客户对, 将其在原位置移除之后交换插入对方的位置. 引入 NERS 后, 选中客户 j 的组合对象为 j 的两附近客户和 NV_j 中客户点的两附近客户. 这里的 avg_j 和 NV_j 与 2-Opt 相同, 考虑所有可能客户对, 以最好解作为客户 j 的 1-1 Exchange 邻域解.

3) 插入 (0-1 Exchange). 任意选中某一客户, 在原

位置移除后插入到任意弧. 引入 NERS 后, 选中客户 j 的插入位置为 NV_j 中点的两条附近弧. 这里的 NV_j 为满足 $c_{ji} < avg_j$ 的所有 i 点集合 ($i \in V$), avg_j 为 j 与配送中心和其他客户的距离平均值. 若 NV_j 中的某一点 $i = 0$, 则表示 j 可插入配送中心 0 的附近弧, 也可单独作为新路径插入. 考虑所有可能插入, 以最好解作为客户 j 的 0-1 Exchange 邻域解.

表 3 以算例 A-n32-k5 的某一种群 ($P_{size} = 20$) 为例展示了不同改进策略的改进率和求解时间, 表明了 VNS 和 NERS 的有效性, 最终取 VNS 按“2-Opt \rightarrow 1-1 Exchange \rightarrow 0-1 Exchange”顺序变更. 表 4 是不同 MAXCount 取值下的 VNS 改进率和求解时间, 最终取 MAXCount = 10. 图 4 为算例 A-n48-k7 某一可行解 VNS 改进前后的路线图, (a, b) 为位置坐标.

表 3 不同改进策略的改进率和求解时间

策略	单独作用		共同作用(VNS)	
	2-Opt	1-1 Exchange	0-1 Exchange	整体(正序) / 整体(逆序)
传统	28.85% (1.48 s)	20.45% (0.67 s)	25.06% (0.73 s)	34.81% (2.78 s) / 35.32% (2.82 s)
引入 NERS	32.71% (1.41 s)	18.53% (0.39 s)	22.74% (0.63 s)	37.76% (2.23 s) / 36.51% (2.32 s)

表 4 不同 MAXCount 取值下的 VNS 改进率和求解时间

MAXCount 取值	5	10	15	20
改进率/%	33.90	37.76	36.39	35.89
求解时间/s	1.63	2.35	2.52	2.54

3 算例验证和结果分析

以文献 [1-8,18-19] 中采用的公共标准算例进行实验, 实验参数选取如下: 种群规模 $P_{size} = 20 \sim 80$, 精英解集 $|B_1| = 10 \sim 15$, 多样性解集 $|B_2| = 5$, VNS 迭代次数 MAXCount = 10, 总迭代次数 MAXN = 20 \sim 100, 罚函数 $P_e = 500 \sim 1000$. P_{size} , $|B_1|$ 和 MAXN 由算例规模决定, P_e 由算例的成本波动范围决定. 采用 Matlab 10.0 编译算法在 PC 机上运行 (CPU: 双核奔腾 2.9 GHz; 内存: 4 G; 操作系统: Window 764 位).

实验 1 取文献 [1-5] 采用的包含 8 个客户、1 个仓库和 2 辆车, 每辆车载重量 8 吨的简单算例. 算例源自文献 [18], 已知最优解 67.5. 表 5 为 20 次实验并与文献 [1-5] 的比较结果. 结果显示: 本文算法和文献 [1-5] 的算法均能找到最优解, 但本文算法的每次实验都能找到最优解, 表明本文算法性能较文献 [1-5] 的算法稳定且更优. 在求解时间上, 本文算法耗时 1.6 s, 求解效率高.

表 5 实验 1 的比较结果

算例 (1 \times 8)	最优值	最差值	平均值
双种群遗传算法 ^[1]	67.5*	72	69.575
微粒群算法 ^[2]	67.5*	69.5	68.325
捕食搜索算法 ^[3]	67.5*	69	67.65
量子进化算法 QEA ^[4]	67.5*	71	68.358
混合遗传算法 ^[5]	67.5*	69	67.875
本文算法	67.5*	67.5*	67.5*

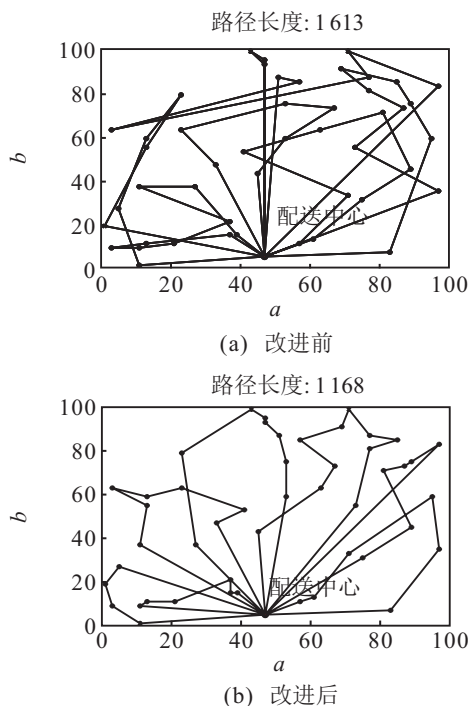


图 4 VNS 改进前后的路线图

2.6 精英保留策略和终止条件

用当前最好解替代改进后的新种群最差解, 保证算法收敛, 终止条件为最大迭代次数 MAXN 次.

实验 2 取文献 [5] 采用的标准算例 E-n22-k4 和

E-n33-k4, 算例来源于 <http://neo.lcc.uma.es/vrp/vrp-instances/capacitated-vrp-instances/>. 表 6 是 20 次实验并与文献 [5] 比较的结果, BK 为已知最优解. 结果显示: 本文算法误差均为 0%, 且每次实验都能找到最优解, 进一步表明了本文算法优于文献 [5] 的算法. 在求解时间上, 本文算法耗时均小于 60 s, 耗时较少.

与实验 1 的 c_{ij} 不同, 客户 i 的位置坐标为 (a_i, b_i) , 实验 1 中取

$$c_{ij} = \sqrt{(a_i - a_j)^2 + (b_i - b_j)^2},$$

这里取

$$c_{ij} = \lfloor \sqrt{(a_i - a_j)^2 + (b_i - b_j)^2} + 0.5 \rfloor.$$

实验 3 取文献 [6] 采用的 8 个算例, 算例来源与实验 2 相同, c_{ij} 与实验 1 相同. 表 7 是 20 次实验并与文献 [6] 比较的结果. 结果显示: 本文算法平均误差 0.12%, 小于文献 [6] 的 0.49%, 并在其中 7 个算例找到

问题的最优解, 在其中 6 个算例中最优值、最差值和平均值相同, 在算例 E-n51-k5 中最差值、最好值的偏差仅 2.64%, 表明本文算法性能较文献 [6] 的算法更好. 在求解时间上, 本文算法耗时虽始终多于文献 [6], 并在客户规模高于 50 的算例上显著增加, 但仍在可接受范围内.

实验 4 取文献 [7] 采用的 15 个算例, 算例来源和 c_{ij} 与实验 2 相同. 表 8 是 20 次实验并与文献 [7-8, 19] 比较的结果. 结果显示: 本文算法在所有算例上均能找到问题的最优解, 误差 0%, 小于文献 [19] 的 8.09%, 文献 [7] 的 3.39%, 文献 [8] 的 0.07%, 表明本文算法性能较文献 [7,19] 的算法优势明显, 较文献 [8] 的算法差异较小、性能相当. 在求解时间上, 文献 [7,19] 均未给出求解时间, 文献 [8] 平均耗时 336 s, 且均在 200 s 以上, 本文算法平均耗时 158 s, 大多在 200 s 以下, 可见本文算法虽耗时较多, 但仍在可接受范围内.

表 6 实验 2 的比较结果

算例	BK	混合遗传算法 ^[5]				本文算法				
		最优值	最差值	平均值	误差/%	最优值	最差值	平均值	误差/%	CPU/s
E-n22-k4	375*	375*	382	375.7	0.00	375*	375*	375*	0.00	14
E-n33-k4	835*	837	862	846.6	0.24	835*	835*	835*	0.00	55

表 7 实验 3 的比较结果

算例	BK	改进蚁群算法 ^[6]				本文算法				
		最优值	平均值	误差/%	CPU/s	最优值	最差值	平均值	误差/%	CPU/s
P-n16-k8	451.33*	451.33*	457.34	0.00	2	451.33*	451.33*	451.33*	0.00	22
P-n19-k2	212.65*	212.65*	215.92	0.00	3	212.65*	212.65*	212.65*	0.00	14
P-n20-k2	217.41*	217.41*	220.80	0.00	2	217.41*	217.41*	217.41*	0.00	14
P-n21-k2	212.71*	217.48	222.46	2.24	5	212.71*	212.71*	212.71*	0.00	15
P-n23-k8	531.17*	531.17*	532.70	0.00	4	536.34	536.34	536.34	0.97	41
P-n40-k5	461.72*	470.31	472.39	1.68	13	461.72*	461.72*	461.72*	0.00	72
E-n22-k4	375.28*	375.28*	381.62	0.00	6	375.28*	375.28*	375.28*	0.00	15
E-n51-k5	524.61*	524.61*	524.92	0.00	18	524.61*	538.49	527.59	0.00	266
平均值				0.49					0.12	

表 8 实验 4 的比较结果

算例	BK	CON(2012) ^[19]		PSO-VNS(2014) ^[7]		SC-ESA(2013) ^[8]		本文算法				
		最优值	误差/%	最优值	误差/%	误差/%	CPU/s	最优值	最差值	平均值	误差/%	CPU/s
A-n32-k5	784*	853.6	8.88	807.31	2.97	0.00	311	784*	784*	784*	0.00	26
A-n33-k5	661*	705.91	6.79	685.3	3.68	0.00	257	661*	661*	661*	0.00	43
A-n33-k6	742*	793.95	7.00	762.79	2.8	0.00	278	742*	742*	742*	0.00	54
A-n34-k5	778*	827.26	6.33	794.66	2.14	0.00	264	778*	778*	778*	0.00	79
A-n36-k5	799*	859.48	7.57	811.96	1.62	0.00	290	799*	815	808.6	0.00	97
A-n37-k5	699*	709.67	6.08	691.01	3.29	0.00	308	669*	669*	669*	0.00	61
A-n37-k6	949*	1031.74	8.72	984.22	3.71	0.00	271	949*	962	952.1	0.00	178
A-n38-k5	730*	775.25	6.20	768.98	5.34	0.00	403	730*	731	730.6	0.00	154
A-n39-k5	822*	869.58	5.79	846.12	2.93	0.00	202	822*	822*	822*	0.00	138
A-n39-k6	831*	883.44	6.31	859.47	3.43	0.00	333	831*	833	832.6	0.00	151
A-n44-k6	937*	1029.72	9.90	988.7	5.52	0.00	390	937*	949	940.5	0.00	256
A-n45-k6	944*	1027.92	8.89	977.4	3.54	0.00	611	944*	953	950	0.00	152
A-n45-k7	1146*	1288.7	12.45	1197.22	4.47	0.00	353	1146*	1148	1147	0.00	505
A-n46-k7	914*	1003.19	9.75	950.3	3.97	0.00	401	914*	917	915.4	0.00	200
A-n48-k7	1073*	1188.06	10.72	1088.9	1.48	1.03	370	1073*	1091	1076.6	0.00	277
平均值			8.09		3.39	0.07	336				0.00	158

表 9 实验 5 的比较结果

算例	BK	SC-ESA (2013) ^[8]		本文算法				
		误差/%	CPU/s	最优值	最差值	平均值	误差/%	CPU/s
B-n38-k6	805*	1.24	470	805*	806	805.30	0.00	146
B-n41-k6	829*	4.46	276	829*	836	831.90	0.00	153
B-n44-k7	909*	1.32	572	909*	943	924.40	0.00	143
B-n45-k6	678*	1.18	275	680	715	708.40	0.29	248
B-n50-k8	1312*	1.30	386	1330	1349	1337.80	1.37	422
B-n63-k10	1496*	2.81	367	1545	1558	1554.60	3.28	340
B-n66-k9	1316*	1.90	273	1331	1358	1344.75	1.14	320
B-n67-k10	1032*	1.74	763	1044	1075	1058.00	1.16	340
B-n68-k9	1272*	1.57	344	1290	1292	1290.75	1.42	406
B-n78-k10	1221*	2.05	439	1257	1300	1271.33	2.95	340
E-n76-k7	682*	2.05	873	693	696	694.67	1.61	808
E-n76-k8	735*	1.09	908	752	787	768.50	2.31	983
P-n19-k2	212*	3.30	2	212*	212*	212*	0.00	19
P-n55-k7	568*	1.06	429	576	583	578.83	1.41	397
P-n76-k4	593*	2.36	553	594	611	600.00	0.17	842
P-n76-k5	627*	3.03	615	633	655	639.90	0.96	806
平均值		2.03	471				1.13	419

表 10 实验 6 的结果比较

算例	BK	SC-ESA (2013) ^[8]		本文算法				
		误差/%	CPU/s	最优值	最差值	平均值	误差/%	CPU/s
CMT1-n101-k8	817*	1.84	821	821	839	828.4	0.49	2143
CMT1-n101-k10	820*	0.00	18565	820*	829	824.8	0.00	691
CMT1-n121-k7	1034*	0.19	1719	1038	1089	1064.1	0.39	1760
CMT1-n151-k12	1052*	0.86	1126	1052*	1098	1075.9	0.00	5134
CMT1-n200-k17	1291*	3.33	7016	1345	1446	1442	4.18	7435
平均值		1.24	5849				1.01	3432

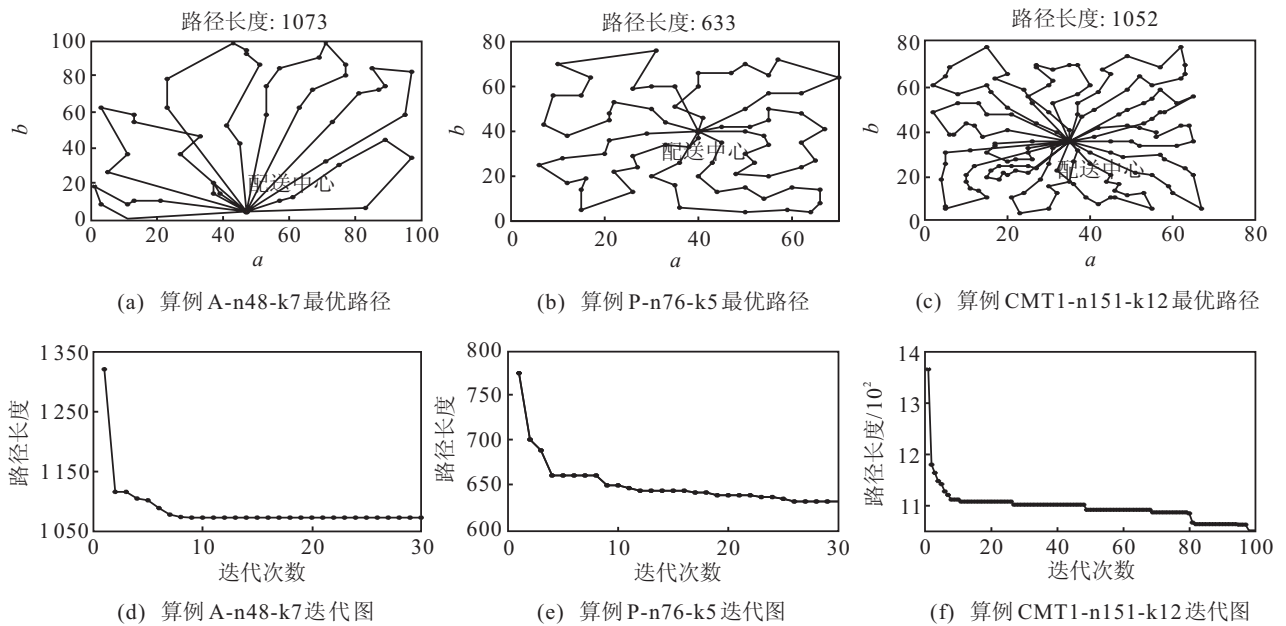


图 5 部分算例的最优路径和收敛迭代

实验 5 为进一步比较本文算法与文献 [8] SC-ESA 算法的性能差异, 取文献 [8] 中结果误差高于 1.0 的 16 个算例, 算例来源和 c_{ij} 与实验 2 相同. 表 9 是 20 次实验并与文献 [8] 比较的结果. 结果显示: 对于 SC-

ESA 求解结果误差高于 1.0 的 16 个测试算例而言, 本文算法平均误差为 1.13%, 低于 SC-ESA 的 2.03%, 且其中 11 个算例的求解结果优于 SC-ESA, 在其中 4 个算例找到最优解, 进一步表明了本文算法优于文献

[8] 的 SC-ESA 算法. 在求解时间上, 本文算法平均耗时 419 s, 小于文献 [8] 的 471 s.

实验 6 为验证本文算法在大规模算例上的性能, 取文献 [8] 的 5 个算例, 算例来源和 c_{ij} 与实验 2 相同, 表 10 是 20 次实验并与文献 [8] 比较的结果. 结果显示: 本文算法误差 1.01%, 小于文献 [8] 的 1.24%, 可见本文算法在大规模算例的搜索性能也优于文献 [8] 的 SC-ESA 算法. 在求解时间上, 文献 [8] 平均耗时为 5 849 s, 最多为 CMT1-n101-k10 的 18 565 s, 本文算法平均耗时为 3 432 s, 最多为 7 435 s.

综合上述实验可以看出, 本文算法优于所对比的算法, 能在可接受的时间内获得或非常接近于问题的

已知最优解, 求解精度高且求解稳定, 寻优能力可靠. 在求解时间上, 虽耗时较多, 但仍在可接受范围内.

图 5 是算例 A-n48-k7(小规模)、P-n76-k5(中规模)、CMT-n151-k12(大规模) 的最优路径图和收敛迭代图. P-n76-k5 有两个最优路径方案, 这里仅展示一个, 由收敛迭代图可看出本文算法寻优速度快, 能在较少的迭代次数内收敛到问题最优解.

为便于算法应用, 图 6 以算例 A-n48-k7 为例展示了各参数对算法性能的影响 (20 次实验结果). 可以看出, 增加迭代次数有利于提高算法寻优能力, 但会增加求解时间. 一定迭代次数后, 算法会趋于停滞, 如 $MAXN = 30 \sim 40$. 种群规模与迭代次数类似, 增大有利于初始解和初始参考集信息量的增加, 寻优能力增强, 但扩大到一定范围后, 信息量也会趋于停滞, 图中未到达停滞状态. 精英种群 $|B_1|$ 的增加有利于更多精英解进入参考集, 对算法寻优有利, 但太大的参考集会导致子集对选中较优解的概率变小, 反而不利于寻优. 多样性种群 $|B_2|$ 与 $|B_1|$ 类似, 一定数量的多样性解有利于算法跳出局部最优, 但太大也会造成子集对选中较优解的可能变小.

4 结 论

本文设计了解决带容量约束车辆路径问题 (CVRP) 的混合分散搜索算法 (HSSA), 保留基本 SS 参考集更新策略和组合策略的全局搜索能力, 采用引入邻域半径减少策略的 VNS 进行局部开发, 采用随机插入法构造初始种群, 采用精英保留策略保证算法收敛. 实验结果分析表明, HSSA 具有较好的寻优效果, 当客户规模增大时, 受运行时间影响适用性降低, 后续研究应加以改进.

参考文献(References)

[1] 赵燕伟, 吴斌, 蒋丽, 等. 车辆路径问题的双种群遗传算法求解方法[J]. 计算机集成制造系统, 2004, 10(3): 303-306.
(Zhao Y W, Wu B, Jiang L, et al. Double populations genetic algorithm for vehicle routing problem[J]. Computer Integrated Manufacturing Systems, 2004, 10(3): 303-306.)

[2] 肖健梅, 李军军, 王锡淮. 求解车辆路径问题的改进微粒群优化算法[J]. 计算机集成制造系统, 2005, 11(4): 577-581.
(Xiao J M, Li J J, Wang X H. Modified particle swarm optimization algorithm for vehicle routing problem[J]. Computer Integrated Manufacturing Systems, 2005, 11(4): 577-581.)

[3] 蒋忠中, 汪定伟. 车辆路径问题的捕食搜索算法研究[J]. 计算机集成制造系统, 2006, 12(11): 1899-1908.
(Jiang Z Z, Wang D W. Predatory search algorithm

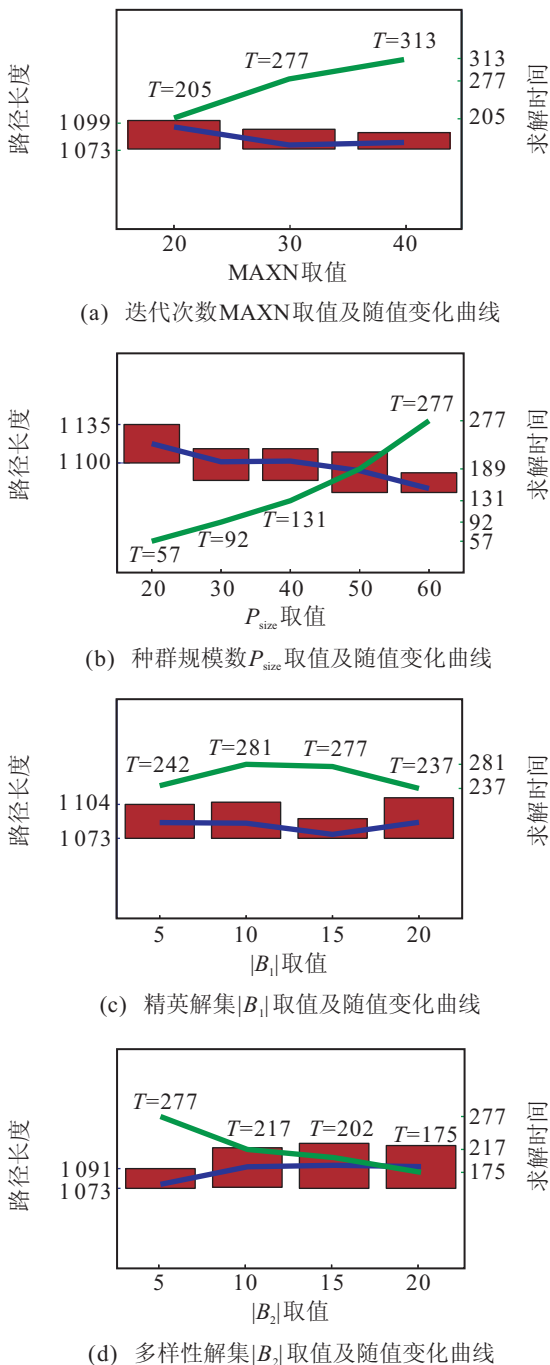


图 6 参数影响曲线

- for vehicle routing problem[J]. *Computer Integrated Manufacturing Systems*, 2006, 12(11): 1899-1908.)
- [4] 赵燕伟, 彭典军, 张景玲, 等. 有能力约束车辆路径问题的量子进化算法[J]. *系统工程理论与实践*, 2009, 29(2): 159-166.
(Zhao Y W, Peng D J, Zhang J L, et al. Quantum evolutionary algorithm for capacitated vehicle routing problem[J]. *Systems Engineering — Theory & Practice*, 2009, 29(2): 159-166.)
- [5] 姜昌华, 戴树贵, 胡幼华. 求解车辆路径问题的混合遗传算法[J]. *计算机集成制造系统*, 2007, 13(10): 2047-2052.
(Jiang C H, Dai S G, Hu Y H. Hybrid genetic algorithm for capacitated vehicle routing problem[J]. *Computer Integrated Manufacturing Systems*, 2007, 13(10): 2047-2052.)
- [6] 王沛栋, 唐功友, 李扬. 带容量约束车辆路由问题的改进蚁群算法[J]. *控制与决策*, 2012, 27(11): 1633-1638.
(Wang P D, Tang G Y, Li Y. Improved ant colony algorithm for capacitated vehicle routing problems[J]. *Control and Decision*, 2012, 27(11): 1633-1638.)
- [7] Tlili T, Faiz S, Krichen S. A hybrid metaheuristic for the distance-constrained capacitated vehicle routing problem[J]. *Procedia-Social and Behavioral Sciences*, 2014, 109: 779-783.
- [8] Stanojevic M, Stanojevic B, Vujosevic M. Enhanced savings calculation and its applications for solving capacitated vehicle routing problem[J]. *Applied Mathematics and Computation*, 2013, 219(20): 10302-10312.
- [9] Glover F. Heuristics for integer programming using surrogate constraints[J]. *Decision Sciences*, 1977, 8(1): 156-166.
- [10] Zhang T, Chaovalitwongse W A, Zhang Y J. Scatter search for the stochastic travel-time vehicle routing problem with simultaneous pick-ups and deliveries[J]. *Computers & Operations Research*, 2012, 39(10): 2277-2290.
- [11] 张军, 唐加福, 潘震东, 等. 分散搜索算法求解带货物权重的车辆路径问题[J]. *系统工程学报*, 2010, 25(1): 92-97.
(Zhang J, Tang J F, Pan Z D, et al. Scatter search algorithm for solving weighted vehicle routing problem[J]. *J of Systems Engineering*, 2010, 25(1): 92-97.)
- [12] Mladenovic N, Hansen P. Variable neighborhood search[J]. *Computers and Operations Research*, 1997, 24(11): 1097-1100.
- [13] 王征, 张俊, 王旭坪. 多车场带时间窗车辆路径问题的变邻域搜索算法[J]. *中国管理科学*, 2011, 19(2): 99-108.
(Wang Z, Zhang J, Wang X P. A modified variable neighborhood search algorithm for the multi depot vehicle routing problem with time windows[J]. *Chinese J of Management Science*, 2011, 19(2): 99-108.)
- [14] 李菊芳, 贺仁杰, 姚锋, 等. 成像卫星集成调度的变邻域禁忌搜索算法[J]. *系统工程理论与实践*, 2013, 33(12): 3040-3044.
(Li J F, He R J, Yao F, et al. Variable neighborhood tabu search algorithm for integrated imaging satellites scheduling problem[J]. *Systems Engineering — Theory & Practice*, 2013, 33(12): 3040-3044.)
- [15] 陈久梅, 曾波. 两级定位-路径问题的路径重连变邻域搜索人工蜂群算法[J]. *计算机集成制造系统*, 2014, 20(5): 1228-1236.
(Chen J M, Zeng B. Artificial bee colony algorithm with variable neighborhood search and path relinking for two-echelon location-routing problem[J]. *Computer Integrated Manufacturing Systems*, 2014, 20(5): 1228-1236.)
- [16] 于莹莹, 陈燕, 李桃迎. 改进的遗传算法求解旅行商问题[J]. *控制与决策*, 2014, 29(8): 1483-1488.
(Yu Y Y, Chen Y, Li T Y. Improved genetic algorithm for solving TSP[J]. *Control and Decision*, 2014, 29(8): 1483-1488.)
- [17] Zachariadis E E, Tarantilis C D, Kiranoudis C T. A hybrid metaheuristic algorithm for the vehicle routing problem with simultaneous delivery and pick-up service[J]. *Expert Systems with Applications*, 2009, 36(2): 1070-1081.
- [18] 姜大立, 杨西龙, 杜文, 等. 车辆路径问题的遗传算法研究[J]. *系统工程理论与实践*, 1999, 19(6): 44-45.
(Jiang D L, Yang X L, Du W, et al. A study on the genetic algorithm for vehicle routing problem[J]. *Systems Engineering — Theory & Practice*, 1999, 19(6): 44-45.)
- [19] Goodman J C, Ohlmann J W, Thomas B W. Cyclic-order neighborhoods with application to the vehicle routing problem with stochastic demand[J]. *European J of Operational Research*, 2012, 217(2): 312-323.

(责任编辑: 郑晓蕾)