

控制与决策

Control and Decision

基于路径切割和自适应检测的大规模限量弧路由问题求解

方伟, 朱建阳

引用本文:

方伟,朱建阳. 基于路径切割和自适应检测的大规模限量弧路由问题求解[J]. *控制与决策*, 2023, 38(12): 3571–3577.

在线阅读 View online: <https://doi.org/10.13195/j.kzyjc.2022.0300>

您可能感兴趣的其他文章

Articles you may be interested in

[基于平衡鲸鱼优化算法的无人车路径规划](#)

Path planning of unmanned ground vehicle based on balanced whale optimization algorithm

控制与决策. 2021, 36(11): 2647–2655 <https://doi.org/10.13195/j.kzyjc.2020.0416>

[基于改进蚁群算法的水面无人艇路径规划](#)

Path planning for unmanned surface vehicle based on improved ant colony algorithm

控制与决策. 2021, 36(4): 847–856 <https://doi.org/10.13195/j.kzyjc.2019.0839>

[基于粒子群算法的满载需求可拆分车辆路径规划](#)

Split vehicle route planning with full load demand based on particle swarm optimization

控制与决策. 2021, 36(6): 1397–1406 <https://doi.org/10.13195/j.kzyjc.2019.1323>

[基于 \$\mu\$ 正态概率区间分族遗传蚁群算法的机器人路径规划](#)

Path planning of mobile robot based on μ normal probability interval population division using genetic ant-colony algorithm

控制与决策. 2021, 36(12): 2861–2870 <https://doi.org/10.13195/j.kzyjc.2020.0745>

[基于改进烟花算法的并联冷机负荷分配优化](#)

Load distribution optimization of parallel chillers based on improved firework algorithm

控制与决策. 2021, 36(11): 2618–2626 <https://doi.org/10.13195/j.kzyjc.2020.0823>

基于路径切割和自适应检测的大规模限量弧路由问题求解

方 伟[†], 朱建阳

- (1. 江南大学 江苏人工智能国际合作联合实验室, 江苏 无锡 214122;
2. 江南大学 江苏省模式识别与计算智能工程实验室, 江苏 无锡 214122)

摘要: 大规模限量弧路由问题 (large scale capacitated arc routing problem, LSCARP) 是一个组合优化问题, 应用广泛, 采用分治策略是解决 LSCARP 的有效方法之一. 首先, 为了利用分治策略取得更优的分解结果, 提出改进路径切割算子来求解 LSCARP, 其能够自动识别路径集合中形态较差的路径并对其进行针对性地进行切割, 从而在迭代中通过将切割后的路径进行重组以获得更优的分解, 有利于算法跳出局部最优取得更小的最终费用; 然后, 针对 LSCARP 的结构会影响算法最终效果的问题, 设计一种自适应数据集检测算子, 其能够根据 LSCARP 中任务边与非任务边的关系来进行参数分配从而提高分解质量; 最后, 将以上 2 个算子应用于 SHAiD 算法, 并与当前主流相关算法进行对比. 实验结果表明了所提出算法的有效性.

关键词: 限量弧路由问题; 组合优化; 分治策略; 大规模优化; 路径切割; 自适应

中图分类号: TP301 文献标志码: A

DOI: 10.13195/j.kzyjc.2022.0300

引用格式: 方伟, 朱建阳. 基于路径切割和自适应检测的大规模限量弧路由问题求解[J]. 控制与决策, 2023, 38(12): 3571-3577.

Solving large scale capacitated arc routing problem based on route cutting off decomposition and adaptive detection

FANG Wei[†], ZHU Jian-yang

- (1. Jiangsu Artificial Intelligence International Cooperation Joint Laboratory, Jiangnan University, Wuxi 214122, China;
2. Jiangsu Provincial Engineering Laboratory of Pattern Recognition and Computational Intelligence, Jiangnan University, Wuxi 214122, China)

Abstract: The large scale capacitated arc routing problem (LSCARP) is a combinatorial optimization problem and has a wide range of applications. The divide and conquer strategy is one of the effective methods to solve the LSCARP. In order to use the divide and conquer strategy to obtain better decomposition results, an improved route cutting operator is proposed to solve the LSCARP. The proposed operator can automatically identify the path with poor shape in the path set and carry out targeted processing on it. In order to achieve better decomposition by reorganizing the divided paths in the iteration, it is beneficial for the algorithm to jump out of the local optimum and obtain a smaller final cost. In addition, since the structure of the LSCARP may affect the final effect of the algorithm, an adaptive dataset detection operator is designed, which can allocate parameters according to the relationship between the task edge and non-task edge in order to improve the decomposition quality. Finally, the above two operators are used in the SHAiD algorithm. The effectiveness of the proposed algorithm is evaluated by compared with the state-of-the-art algorithms.

Keywords: capacitated arc routing problem; combinatorial optimization; divide and conquer; large scale optimization; route cutting off decomposition; adaptive

0 引言

限量弧路由问题 (capacitated arc routing problem, CARP) 是一个典型的 NP-hard 组合优化问题, 其模型被广泛应用于城市垃圾清理^[1-4]、邮件投递^[5]、道路洒

水撒盐规划^[6]等问题. CARP 是旅行商问题的变体^[7-8], 由 Golden 等^[9]提出, 该问题是在特定的约束条件下, 由一组服务于特定任务边的车辆, 在带权连通图中寻找一组具有最小费用的路线. 学者们已提出

收稿日期: 2022-02-27; 录用日期: 2022-09-20.

基金项目: 国家自然科学基金项目 (62073155, 62002137, 62106088, 62206113).

责任编辑: 刘士新.

[†]通讯作者. E-mail: fangwei@jiangnan.edu.cn.

*本文附带电子附录文件, 可登录本刊官网该文“资源附件”区自行下载阅览.

了许多求解算法,如 Ulusoy-Split^[10]、Memetic Algorithm^[11]和 MAENS^[12]等。其中:Ulusoy-Split为很多元启发式算法生成初始解的一个经典算法;Memetic Algorithm为一种局部搜索能力较强的基因算法,但是全局搜索能力较弱;MAENS提出了一种扩展步长的搜索算子,提高了Memetic Algorithm的全局搜索能力。但是CARP的实际应用一般是大规模的,如在一个城市道路洒水问题中,通常需要处理几千条道路。而面对大规模CARP的应用场景,经典CARP算法通常难以获得理想的结果,或因为运行时间的指数级增长而失效^[4,12]。因此,现阶段对大规模限量弧路由问题(large scale capacitated arc routing problem, LSCARP)的求解进行研究具有更大的价值,一般认为超过200个顶点和300条边的CARP属于LSCARP的范畴^[13]。分治策略是一种求解LSCARP行之有效的方法^[12,14-15],其思想是将大规模问题通过特定的方法分解为一定数量的子问题,对每个子问题的求解结果合并得到最终结果。近些年来,基于分治策略的方法求解LSCARP大多基于协同进化框架^[16-17],如RDG-MAENS^[12]、RRG^[4]、VND^[18]、SHAiD^[15]等。其中:RDG-MAENS在协同进化思想下提出了路由距离分组的概念,但是对初始解的要求较高;RRG提出了一种简单有效的随机路由分组方案,优化子问题的求解;VND通过研究解的邻域空间,提高了算法跳出局部最优的能力,但是未考虑到不可行解;SHAiD在协同进化框架下提高了生成LSCARP优秀解的速度,全局优化能力较高。以上算法均可在一定程度上对LSCARP进行有效的分解,但是距离子问题完全独立的完备分解仍然有提升空间。具体而言,在LSCARP中,完备分解中任务边间的距离通常较近,且不同子路径中任务边间的耦合度为可达到的最低水平。

在LSCARP中,容量约束等因素的影响,路径之间关系复杂,很难通过简单的分治策略在较大的解空间挖掘到全部优秀解,因此,分治策略应针对某一突出的问题更加细化。路径切割算子(route cutting off decomposition operator, RCO)^[14]的提出正是对分治策略的应用,但其分治策略可进一步细化。此外,作为当前解决LSCARP先进算法之一的SHAiD算法在不同应用所产生的效果也有差距。如在同一城市中,车辆需要服务的道路数量不同,该算法的效果也会不同,该特点会影响SHAiD算法通过分治所得子路径的质量。所以在同一LSCARP应用中,不同结构所对应的最优参数存在一定的差异。

综合现有LSCARP方法的不足之处,本文提出改进的路径切割算子(improved route cutting off decomposition operator, IRCO)和自适应数据集检测算子(adaptive dataset detection operator, ADDO),主要内容如下。

1)在路径集合中,根据路径的任务数与花费的关系定义差形态路径,并根据任务边与非任务边的关系定义任务稀疏度(task sparsity, TS)。

2)IRCO在原RCO的基础上增加对差形态路径的处理机制,从而获得更优的目标路径。

3)ADDO根据任务稀疏度TS,在SHAiD的基础上自适应地控制虚拟路径^[15]合并的速率和路径基础切割概率,从而获得更优的分解。

1 问题定义

CARP定义在一个带权无向连通图 $G(V, E)$ 中,其中 V 和 E 分别为点集合和边集合。每条属于集合 E 的边 e 有3个属性,分别为需求量 $de(e)$ 、服务费用 $sc(e)$ 、经过费用 $ce(e)$ 。边的集合中分为任务边和非任务边,由需求量 de 来决定: $de(e) > 0$ 为任务边, $de(e) = 0$ 为非任务边。所有 $de(e) > 0$ 的边组成任务集合,所有任务的集合记为 $T = e \in E | de(e) > 0 \subseteq E$ 。点的集合 V 中分为普通点和仓库点(depot),仓库点即车辆的起始和终止点。仓库点中有一组最大装载量为 Q 的车辆集合为所有任务边服务。CARP的目标是在满足容量约束的前提下,寻找一条最终费用最小的路线,并满足以下条件。

1)车辆从仓库点出发,服务结束后返回仓库点。

2)任务边只能被服务一次,但是可被遍历无数次。

3)每辆车所服务的任务边总需求量不能超过其最大装载量 Q 。

CARP的解可用一组表示服务顺序的顶点序列来表示,将每对连续顶点间的最短路径相加,可得到最小费用^[19]。CARP的解由路径 $S = (R_1, \dots, R_k, \dots, R_n)$ 组成。其中: R_k 为第 k 条路径, k 为路径 R_k 的下角标, n 为解中路径的总数。每条路径 R_k 由一组任务边组成,表示为 $R_k = (t_{k_1}, t_{k_2}, \dots, t_{k_n})$ 。每个任务 t 均可使用2个代表方向的ID表示,每个ID分别对应一个头节点 $hv(t)$ 、一个尾节点 $tv(t)$ 和一个逆向ID $inv(t)$ 。如对于某个任务 t 的2个ID t_1 和 t_2 ,有 $hv(t_1) = tv(t_2) = u, tv(t_1) = hv(t_2) = v, t_2 = inv(t_1)$ 和 $t_1 = inv(t_2)$ 。因此,CARP的优化求解可建模为

$\min \text{ Total Cost} =$

$$\sum_{n=1}^{|S|} \sum_{m=1}^{|R_n|} [\text{sc}(R_n[m]) + \Sigma(\text{tv}(R_n[m]), \text{hv}(R_n[m+1]))]. \quad (1)$$

s.t. $R_n[m] \neq R_{n'}[m'], \forall 1 \leq n, n' \leq |S|, 2 \leq m,$

$$m' \leq |R_n - 1|, \forall (n, m) \neq (n', m'); \quad (2)$$

$$R_n[m] \neq \text{inv}(R_{n'}[m']), \forall 1 \leq n, n' \leq |S|, 2 \leq m,$$

$$m' \leq |R_n| - 1, \forall (n, m) \neq (n', m'); \quad (3)$$

$$\sum_{n=1}^{|S|} (|R_n| - 2) = |T|; \quad (4)$$

$$\sum_{m=2}^{|R_n|-1} \text{de}(R_n[m]) \leq Q, \forall 1 \leq n \leq |S|; \quad (5)$$

$$R_n[m] \in T, \forall m = 2, 3, \dots, |R_n| - 1, \\ n = 1, 2, \dots, |S|. \quad (6)$$

其中: 式(1)为目标函数, 表示总回路的最小费用; $\Sigma(a, b)$ 为顶点 a 到顶点 b 的最小距离, 可使用 Dijkstra 算法求得; 式(2)~(6)为约束条件; 式(2)和(3)为在路径不同位置服务的任意2个任务ID属于不同的任务, $|S|$ 为路径总数; 式(4)表示被服务的任务ID总数等于任务总数, 除去头节点和尾节点为仓库点, $|T|$ 为任务总数. 式(2)~(4)保证了所有任务只能被执行一次; 式(5)保证了每条路径的总需求不超过单个车辆的容量 Q ; 式(6)定义了每条路径中元素的域, 即除第1个和最后1个元素外, 其他所有元素必须属于该任务集.

2 基于IRCO和ADDO的LSCARP求解

在LSCARP中, 主问题的求解对象为所有任务边组成的大型任务序列, 分解后得到子问题的求解对象为每个子任务序列, 其中每个子问题的总需求不大于车辆的容量约束 Q . 一般倾向于将这样的2个任务分配到同一个子集: 1) 2个距离接近的任务; 2) 2个在当前最优解中同一个路径中的任务.

RCO与RDG^[20]的作用类似, 均为结合CARP的领域知识进行分解以得到更优的解序列, 其思路是对迭代过程中迄今为止最优任务序列进行启发式切割, 设计了任务秩矩阵作为切割优连接(good links)和差连接(poor links)的标准, 其中good links和poor links分别为彼此距离较近的和彼此距离较远的任务对. RCO在每次迭代的最优解中, 以设定的概率来切断good links和poor links, 所得到的新的路径集合通过合并得到新的任务序列. RCO通过这种方式, 打破路径内部较差的形态来获得新的路径, 以此有更大

可能获得更优秀的任务组合模式, 提高了算法的全局搜索能力, 能够在有限的时间内取得相对更优的结果^[14].

RCO验证了对路径内部连接模式的探索价值, 但是, 对路径形态与路径间的连接模式可进一步深入, 更合理地切断更多poor links要优于在所有路径中以相同概率切断poor links, 通过更多切断不良形态路径中任务间的联系也能够创造出更多潜在的优秀任务组合. 此外, 在同一LSCARP的应用场景中, 任务边在所有边中的占比也会影响算法的最终效果^[15], 这表明根据LSCARP的具体规模和结构自适应设计算法运行模式能够有针对性地提升算法的性能.

为了进一步提高分解的有效性, 本文研究了LSCARP路径的形态和内部连接模式, 提出了算子IRCO和ADDO, 引入差形态路径的概念, 并将其嵌入SHAiD中构成新算法用于求解LSCARP.

2.1 差形态路径

首先给出长路径和短路径的定义: 长路径指在每次迭代的路径集合中, 路径内部的任务数大于路径集合中任务数平均值的路径; 反之, 定义为短路径. 本文沿用了RCO中2个任务边间最短距离的定义, 有

$$\Sigma(t_1, t_2) = \frac{1}{4} \delta(\text{hv}(t_1), \text{hv}(t_2)) + \frac{1}{4} \delta(\text{hv}(t_1), \text{tv}(t_2)) + \frac{1}{4} \delta(\text{tv}(t_1), \text{hv}(t_2)) + \frac{1}{4} \delta(\text{tv}(t_1), \text{tv}(t_2)). \quad (7)$$

其中: t_1 和 t_2 分别为2个任务边, $\delta(V_1, V_2)$ 为点 V_1 到 V_2 的最短直线距离.

所提出差形态路径是指同时满足以下2个条件的路径: 1) 该路径内所有任务间最短距离和大于其他所有路径的平均值; 2) 该路径为短路径. 图1为一个CARP算例中的3条路径Route-1、Route-2和Route-3. 其中: Route-1任务数为4, 所有任务间最短距离和为 $1 + 2 + 0 = 3$; Route-2任务数为3, 所有任务间最短距离和为 $3 + 4 = 7$; Route-3任务数为4, 所有任

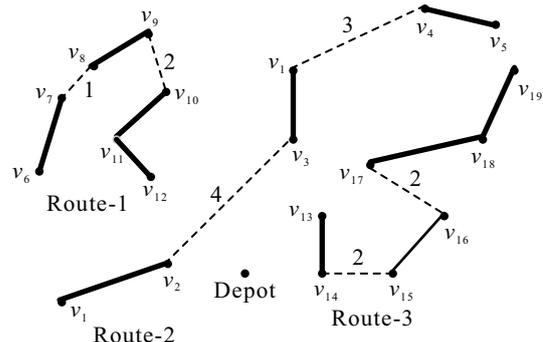


图1 CARP算例中的一个路径实例

务间最短距离和为 $2+2+0=4$ 。所以Route-2同时满足以上2个条件,符合差形态路径的定义,应被赋予更大的切割概率。

如图1所示,差形态路径因其任务数少且任务间距离大的特点,通常会造成任务间耦合度低的问题,影响路径的聚类效果。所以应合理减少差形态路径,对其进行启发式的切割,产生的新路径经合并后有更大概率会生成优秀的任务组合,也会增加跳出局部最优的可能性。

2.2 改进的路径切割算子(IRCO)

基于差形态路径的定义,本文提出了改进的路径切割算子。该算子在RCO的基础上,增加了对差形态路径的操作,即在每次迭代中自动检测当前解中的差形态路径,为其分配额外的切割概率。给定每次迭代的最优解,求出所有路径的任务总间距的平均值,根据此平均值划分差形态路径与常规路径。之后基于已知条件可得到以下4种类型的连接:1)差形态路径的good links;2)差形态路径中的poor links;3)常规路径中的good links;4)常规路径中的poor links。最后根据以上4种不同类型的连接,为其分配不同的切割概率。算法IRCO的具体步骤如下。

step 1: 输入当前最优解 S ,任务等级矩阵 τ ,基础切割概率 α 、 β ,额外切割概率 θ 。

step 2: 置当前子路径集合 ω 为空。

step 3: 计算平均任务连接等级 γ 。

step 4: 计算全部路径任务总间距的平均值RA。

step 5: 遍历解集中每条路径:

step 5.1: 置集合good-links、poor-links为空。

step 5.2: 计算该路径内部的任务总间距ra。

step 5.3: 遍历路径中的每个连接 $(S_i[i], S_i[i+1])$: 若 $\tau_{i,i+1} < \gamma$,则将 $(S_i[i], S_i[i+1])$ 加入集合good-links,否则将其加入集合poor-links。

step 5.4: 置待切割集合cut-gl和cut-pl为空。

step 5.5: 随机在区间 $[0, 1]$ 生成一个随机数 r_1 。

step 5.6: 当满足 $r_1 < \alpha$ 时,随机选择一条集合good-links中的连接加入集合cut-gl。

step 5.7: 随机在区间 $[0, 1]$ 生成一个随机数 r_2 。

step 5.8: 当满足 $ra > RA$ 时,若 $\beta < r_2 < \beta + \theta$,则随机选择一条集合poor-links中的连接加入cut-pl;当不满足 $ra > RA$ 时,若 $r_2 < \beta$,则随机选择一条集合poor-links中的连接加入cut-pl。

step 5.9: 将集合cut-gl和集合cut-pl中的连接进行切割生成子路径集合Ssub,将其加入集合 ω 中。

step 6: 返回子路径集合 ω 。

算法首先输入当前最优解 S 、基础good links切割概率 α 、基础poor links切割概率 β 、额外切割概率 θ ;然后,根据任务等级矩阵 τ 计算出平均任务连接等级 γ ^[14]求出任务总间距的平均值RA;接着,根据每条路径中各连接的质量划分good links与poor links;最后,根据该路径是否为差形态路径对其进行切割概率分配,并重点对差形态路径中的poor links额外分配更高的切割概率。

2.3 任务稀疏度(TS)

本文所研究的LSCARP内部结构为任务边与非任务边间的关系,主要体现在任务边与非任务边数量的比例,该比例定义为任务稀疏度,其计算方式为

$$TS = \frac{Nt}{Nnt} \quad (8)$$

其中: Nt 为任务边数量, Nnt 为非任务边数量。在同一个数据集中,TS的值由小至大分别为算例中任务边在所有边中稀疏度由小至大的关系。图2为低任务稀疏度的算例与高任务稀疏度的算例间的对比。图2中:实线为任务边,虚线为非任务边。路径(单个任务边也算作路径)间的相互作用非常复杂^[14],且路径间的相对距离也会影响分治的最终结果。

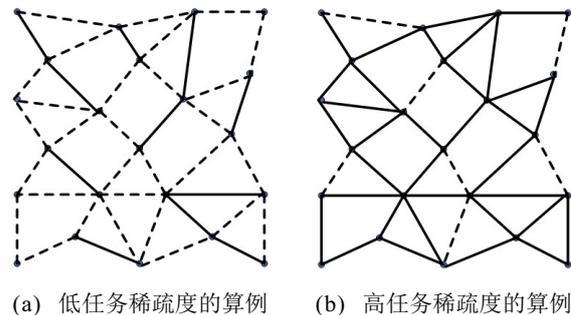


图2 同数据集中不同任务稀疏度的2个算例

2.4 自适应数据集检测算子(ADDO)

通过SHAiD^[15]中的实验可表明,简单将虚拟路径合并速度与基础差连接切割概率设置为固定值,在不同规模算例中的提升率差距较大,所以拥有不同结构的算例所对应的最优参数也是差异较大的,这会一定程度上影响分解的质量。

在LSCARP的数据集中,对于任务稀疏度低的算例,由于任务数量相对较少,需要进行更多路径合并来挖掘优秀的任务组合,需要更慢地合并速度。而对于任务稀疏度高的算例,路径合并速度过慢或过多路径的产生,会导致算法执行效率过低,在限定时间内难以取得优秀的结果。所以基于对任务稀疏度的研究,本文提出了ADDO。该算子在SHAiD算法框架之上,增加了对同LSCARP应用中不同结构的检测,通过 Nt 和 Nnt 计算出TS并以其为标注,合理控制其在

层次分解中路径的合并速度^[15].

2.5 嵌入IRCO和ADDO的SHAiD

为了验证以上IRCO和ADDO两算子的有效性, 本文将其与算法SHAiD相结合, 名为IRCO-ADDO-SHAiD, 其流程如下.

- step 1: 输入数据集算例 D , IRCO 的参数 α, β, θ .
- step 2: 将 ADDO 应用于算例 D 中, 得到 R_m 和 R_s .
- step 3: 通过 SHAiD^[15] 得到初始解 S .
- step 4: 将局部搜索应用于初始解 S 中.
- step 5: 初始化最优值 $S_{best} = S$.
- step 6: 在不超过算法最大执行时间且未达到最大迭代次数时, 重复执行 step 5:
 - step 6.1: $\omega = \text{IRCO}(S, \tau, \alpha, \beta, \theta)$;
 - step 6.2: 基于 ω 中的子路径, 构造出虚拟路径集合;
 - step 6.3: 将自适应参数 R_m 和 R_s 应用于 HDU 算法中得到解 S' ;
 - step 6.4: 将局部搜索应用于 S' 中;
 - step 6.5: 当 $\text{TotalCost}(S') \leq \text{TotalCost}(S_{best})$ 时, 将 S' 的值更新至 S_{best} 中.
 - step 7: 返回最优解 S_{best} .

2.6 时间复杂度分析

设算例中任务边总数为 T_{num} , 点的数量为 N_{num} , 最大迭代次数为 G_{max} . ADDO 仅需计算一次即可得到参数信息, 故时间复杂度为 $O(1)$. IRCO 切割路径集合的复杂度为 $O(m \times R_{t_{num}})$. 其中: m 为当前路径的总数, $R_{t_{num}}$ 为路径平均包含任务边的数量. SHAiD 层次分解阶段的复杂度为 $O\left(T_{num} \times R_m^{i_{max}} \times \sum_i^{i_{max}} Vt_i^3\right)$. 其中: i_{max} 为该阶段最大迭代次数, Vt_i 为每次迭代的虚拟任务数量^[15], R_m 的值介于 $[0, 1]$. Ulsysplt^[10] 的复杂度为 $O(T_{num}^2)$. 路径反转算法的时间复杂度为 $O(2m \times R_{t_{num}})$. MergeSplit^[21] 局部搜索的时间复杂度为 $O\left(\frac{5}{2}m^2 \times R_{t_{num}}^2\right)$. 故算法的时间复杂度约为

$$O\left(\frac{5}{2}m^2 \times R_{t_{num}}^2 + T_{num} \times R_m^{i_{max}} \times \sum_i^{i_{max}} Vt_i^3 + T_{num}^2\right).$$

为了使得算法间的时间复杂度作统一化对比, 本文将决定时间复杂度高低的最关键变量, 即任务边数量 T_{num} 作为算法时间复杂度的对比标准. IRCO-ADDO-SHAiD 的时间复杂度进行统一化简后约为 $O((m^2 + \alpha)T_{num}^2)$, 其中 α 为介于 $[1, m]$ 的常数.

表 1 为所提出算法与对比算法的时间复杂度. RDG-MAENS 中的 $iter$ 为局部搜索阶段的总迭代次数. VNS 中的 r 为子路径的数量. UHGS 中的 ε 为介

于 $[1, T_{num}]$ 的常数. 通过对比可得出, 除 TSA-1 的时间复杂度稍高外, 所提出算法与其他对比算法的时间复杂度均处于 T_{num}^2 , 具有相近的复杂度.

表 1 IRCO-ADDO-SHAiD 与对比算法时间复杂度的对比

算法名称	时间复杂度
RDG-MAENS	$O(iter \times T_{num}^2)$
VNS	$O(r^2 \times T_{num}^2)$
TSA-1	$O(T_{num}^3)$
UHGS	$O(\varepsilon \times T_{num}^2)$
RCO-SHAiD	$O((m^2 + \alpha)T_{num}^2)$
IRCO-ADDO-SHAiD	$O((m^2 + \alpha)T_{num}^2)$

3 实验结果和分析

为了验证所提出 IRCO 和 ADDO 在大规模限量弧路径数据集中的有效性, 本文设计了算法 IRCO-ADDO-SHAiD 在数据集 Hefei、Beijing 下与 RDG-MAENS^[12]、VNS^[22]、TSA-1^[23]、UHGS^[24]、SHAiD^[15]、RCO-SHAiD^[14] 下的相关对比实验.

为了进行公平对比, 所提出算法与对比算法均在相同数据集 Hefei、Beijing^[12] 进行实验, 对比算法参数设置与原论文保持一致, 对比标准为限定运行时间内算法的最小花费. 本文采用根据 CPU 频率对不同算法的运行时间进行统一化对比的方法. 所提出算法和 UHGS 均运行在 Intel(R) Core(TM) i7-10750 H CPU, 其频率为 2.60 GHz. 其他比较算法运行在 Intel(R) Core(TM) i7-4790 CPU, 频率为 3.6 GHz. 由于所用 LSCARP 算例的最优解和最小费用未知, 且对于大规模优化问题, 在有限时间预算中是否能够快速得到一个较为优秀的解也是算法比较的一个重要标准, 本文沿用了 SHAiD 论文中以 1800 s 为基准的最大运行时间的设置. 以 CPU 频率为时间缩放标准^[16], 所提出算法和 UHGS 的最大运行时间设置为 $3.6/2.6 \times 30 \times 60 = 2492$ s, 其他对比算法的最大运行时间设置为 1800 s.

3.1 参数设置

IRCO 有 1 个自定义参数, 即额外切割概率 θ ; ADDO 的 2 个参数虚拟路径合并速率 R_m 和基础差连接切割概率 R_s 在 SHAiD^[15] 中已被定义, 本文主要对其数值进行自适应分配. 经过相关参数分析实验的验证后, IRCO 最终的参数设置为 $\theta = 0.1$, ADDO 的参数设置以 TS 为分界线, 当 $TS < 1$ 时, $(R_m, R_s) = (0.9, 0.9)$; 当 $TS > 1$ 时, $(R_m, R_s) = (0.1, 0.1)$.

3.2 IRCO-ADDO-SHAiD 与当前主流 LSCARP 算法对比的实验结果

为了验证 IRCO 和 ADDO 同时作用于 SHAiD 算法是否能够取得更优的结果, 本节将 IRCO 和 ADDO

同时应用于 SHAiD 算法中,并将其与当前 LSCARP 领域内的主流算法 RCO-SHAiD^[15]、UHGS^[24]、RDG-MAENS^[12]、VNS^[18]以及 TSA-1^[23]在 Hefei 和 Beijing 数据集上进行了对比实验. IRCO-SHAiD 和 SHAiD 分别独立运行 20 次. 对比算法所有参数设置均与原

论文保持一致.

表 2 和表 3 为 IRCO-ADDO-SHAiD 算法与对比算法在 Hefei 和 Beijing 数据集共 20 个算例中的平均结果. 由表 2 和表 3 可见, IRCO-ADDO-SHAiD 的最终费用优于除 UHGS 算法外的所有算法.

表 2 IRCO-ADDO-SHAiD 算法在 Hefei 数据集上运行 20 次的平均值与其他算法的对比

算例	V	E	T	VNS	RDG-MAENS	TSA-1	UHGS	RCO-SHAiD	IRCO-ADDO-SHAiD
Hefei-1	850	1 212	121	247 819	247 389	252 615	245 664	247 943	246 823
Hefei-2	850	1 212	242	449 979	441 602	456 228	433 878	437 931	436 012
Hefei-3	850	1 212	364	595 263	589 183	637 201	573 701	586 885	583 016
Hefei-4	850	1 212	485	774 323	761 387	791 790	740 463	754 116	750 123
Hefei-5	850	1 212	606	994 794	991 807	1 042 701	946 602	967 198	961 952
Hefei-6	850	1 212	727	1 128 667	1 132 123	1 162 641	1 072 899	1 099 037	1 092 167
Hefei-7	850	1 212	848	1 337 353	1 361 180	1 353 502	1 272 912	1 304 230	1 295 311
Hefei-8	850	1 212	970	1 517 151	1 550 664	1 537 169	1 436 096	1 478 098	1 464 685
Hefei-9	850	1 212	1 091	1 694 957	1 749 132	1 716 256	1 605 612	1 657 764	1 632 619
Hefei-10	850	1 212	1 212	1 852 622	1 923 290	1 901 167	1 754 904	1 810 765	1 791 315

表 3 IRCO-ADDO-SHAiD 算法在 Beijing 数据集上运行 20 次的平均值与其他算法的对比

算例	V	E	T	VNS	RDG-MAENS	TSA-1	UHGS	RCO-SHAiD	IRCO-ADDO-SHAiD
Beijing-1	2 820	3 584	358	782 415	829 443	829 132	760 595	770 294	767 149
Beijing-2	2 820	3 584	717	1 192 292	1 338 002	1 401 363	1 133 012	1 164 102	1 155 376
Beijing-3	2 820	3 584	1 075	1 618 484	1 847 973	1 709 279	1 542 551	1 577 864	1 565 213
Beijing-4	2 820	3 584	1 434	1 953 892	2 193 427	2 070 885	1 847 432	1 896 983	1 888 946
Beijing-5	2 820	3 584	1 792	2 335 915	2 639 486	2 440 319	2 210 467	2 255 982	2 230 437
Beijing-6	2 820	3 584	2 151	2 743 677	3 047 311	2 814 735	2 571 790	2 654 425	2 639 798
Beijing-7	2 820	3 584	2 509	3 063 813	3 388 275	3 186 240	2 871 994	2 953 073	2 939 864
Beijing-8	2 820	3 584	2 868	3 366 215	3 697 052	3 456 037	3 150 705	3 234 275	3 218 643
Beijing-9	2 820	3 584	3 226	3 723 830	4 061 820	3 943 883	3 485 860	3 575 771	3 553 691
Beijing-10	2 820	3 584	3 584	4 040 694	4 353 974	4 103 532	3 785 821	3 885 360	3 839 621

总而言之, IRCO 和 ADDO 同时嵌入 SHAiD 算法能够使得最终结果进一步得到提升,其平均最优费用均优于除 UHGS 外所有算法,表明 IRCO 和 ADDO 在处理 LSCARP 是有效的. 而 UHGS 性能在所有对比算法中最优主要因为其搜索能力较强而非将问题分而治之. UHGS 的主要贡献是一种启发式选择邻域搜索模式的机制,且能够在 $O(1)$ 的时间内完成邻域搜索模式的选择;且结合所提出的一种增量移动评估的最短路径计算机制,一定程度上减少了算法计算量,故在较短的时间内取得了更好的结果. 所以 UHGS 的贡献主要在局部搜索优化的领域,但是本文的主要贡献是在全局优化阶段.

4 结 论

本文针对当前分治算法解决 LSCARP 获得的分解不够完备的问题,提出了 IRCO 和 ADDO 两个算子,并将其嵌入算法 SHAiD 中,设计了求解 LSCARP 的算法 IRCO-ADDO-SHAiD 以进一步提高分解的质

量. 为了验证所提出算子和算法的有效性,本文分别将 IRCO 和 ADDO 独立嵌入 SHAiD 中,实验结果表明,2 个算子均能够在一定程度上提升原算法性能. 最后,将所提出算法与当前主流 LSCARP 算法在数据集 Hefei 和 Beijing 下进行对比实验,实验结果表明了所提出算子和算法的有效性.

参考文献(References)

- [1] Mei Y, Tang K, Yao X. A memetic algorithm for periodic capacitated arc routing problem[J]. IEEE Transactions on Systems, Man, and Cybernetics Part B, Cybernetics: A Publication of the IEEE Systems, Man, and Cybernetics Society, 2011, 41(6): 1654-1667.
- [2] Zhang Y Z, Mei Y, Tang K, et al. Memetic algorithm with route decomposing for periodic capacitated arc routing problem[J]. Applied Soft Computing, 2017, 52: 1130-1142.
- [3] Chen Y N, Hao J K. Two phased hybrid local search for the periodic capacitated arc routing problem[J]. European Journal of Operational Research, 2018, 264(1): 55-65.

- [4] Mei Y, Li X D, Yao X. Decomposing large-scale capacitated arc routing problems using a random route grouping method[C]. IEEE Congress on Evolutionary Computation. Cancun, 2013: 1013-1020.
- [5] Ovezgeldyyev A O, Morozov A V. Developing the branch and bound method in the problem of searching for the optimal cyclic route[J]. Cybernetics and Systems Analysis, 2013, 49(5): 739-748.
- [6] Handa H, Chapman L, Yao X. Robust route optimization for gritting/salting trucks: A CERCIA experience[J]. IEEE Computational Intelligence Magazine, 2006, 1(1): 6-9.
- [7] 何庆, 吴意乐, 徐同伟. 改进遗传模拟退火算法在TSP优化中的应用[J]. 控制与决策, 2018, 33(2): 219-225. (He Q, Wu Y L, Xu T W. Application of improved genetic simulated annealing algorithm in TSP optimization[J]. Control and Decision, 2018, 33(2): 219-225.)
- [8] 张玉州, 刘晓飞, 黄师化, 等. 一种求解多车型CARP的有效memetic算法[J]. 中国科学技术大学学报, 2017, 47(7): 583-593. (Zhang Y Z, Liu X F, Huang S H, et al. An effective memetic algorithm for heterogeneous vehicle CARP[J]. Journal of University of Science and Technology of China, 2017, 47(7): 583-593.)
- [9] Golden B L, Wong R T. Capacitated arc routing problems[J]. Networks, 1981, 11(3): 305-315.
- [10] Ulusoy G. The fleet size and mix problem for capacitated arc routing[J]. European Journal of Operational Research, 1985, 22(3): 329-337.
- [11] Rui W, Ning G. Research on heterogenous fleet dynamic vehicle routing problem based on memetic algorithm[C]. The 4th Annual International Conference on Data Science and Business Analytics. Changsha, 2020: 40-44.
- [12] Mei Y, Li X D, Yao X. Cooperative coevolution with route distance grouping for large-scale capacitated arc routing problems[J]. IEEE Transactions on Evolutionary Computation, 2014, 18(3): 435-449.
- [13] Martinelli R, Poggi M, Subramanian A. Improved bounds for large scale capacitated arc routing problem[J]. Computers & Operations Research, 2013, 40(8): 2145-2160.
- [14] Zhang Y Z, Mei Y, Zhang B Z, et al. Divide-and-conquer large scale capacitated arc routing problems with route cutting off decomposition[J]. Information Sciences, 2021, 553: 208-224.
- [15] Tang K, Wang J, Li X D, et al. A scalable approach to capacitated arc routing problems based on hierarchical decomposition[J]. IEEE Transactions on Cybernetics, 2017, 47(11): 3928-3940.
- [16] Omidvar M N, Li X D, Yang Z Y, et al. Cooperative co-evolution for large scale optimization through more frequent random grouping[C]. IEEE Congress on Evolutionary Computation. Barcelona, 2010: 1-8.
- [17] Li X D, Yao X. Cooperatively coevolving particle swarms for large scale optimization[J]. IEEE Transactions on Evolutionary Computation, 2012, 16(2): 210-224.
- [18] Mei Y, Li X D, Yao X. Variable neighborhood decomposition for large scale capacitated arc routing problem[C]. IEEE Congress on Evolutionary Computation. Beijing, 2014: 1313-1320.
- [19] Lacomme P, Prins C, Ramdane-Cherif W. Competitive memetic algorithms for arc routing problems[J]. Annals of Operations Research, 2004, 131(1/2/3/4): 159-185.
- [20] Shang R H, Dai K Y, Jiao L C, et al. Improved memetic algorithm based on route distance grouping for multiobjective large scale capacitated arc routing problems[J]. IEEE Transactions on Cybernetics, 2016, 46(4): 1000-1013.
- [21] Tang K, Mei Y, Yao X. Memetic algorithm with extended neighborhood search for capacitated arc routing problems[J]. IEEE Transactions on Evolutionary Computation, 2009, 13(5): 1151-1166.
- [22] Polacek M, Doerner K F, Hartl R F, et al. A variable neighborhood search for the capacitated arc routing problem with intermediate facilities[J]. Journal of Heuristics, 2008, 14(5): 405-423.
- [23] Brandão J, Eglese R. A deterministic tabu search algorithm for the capacitated arc routing problem[J]. Computers & Operations Research, 2008, 35(4): 1112-1126.
- [24] Vidal T. Node, edge, arc routing and turn penalties: Multiple problems—One neighborhood extension[J]. Operations Research, 2017, 65(4): 992-1010.

作者简介

方伟(1980—), 男, 教授, 博士生导师, 从事计算智能、大数据智能分析等研究, E-mail: fangwei@jiangnan.edu.cn;

朱建阳(1997—), 男, 硕士生, 从事智能优化算法应用的研究, E-mail: 6201910043@stu.jiangnan.edu.cn.