

面向复杂物流配送场景的车辆路径规划多任务辅助进化算法

李坚强^{1,2} 蔡俊创¹ 孙涛¹ 朱庆灵¹ 林秋镇¹

摘要 在现代社会中,复杂物流配送场景的车辆路径规划问题 (Vehicle routing problem, VRP) 一般带有时间窗约束且需要提供同时取送货的服务. 这种复杂物流配送场景的车辆路径规划问题是 NP-难问题. 当其规模逐渐增大时,一般的数学规划方法难以求解,通常使用启发式方法在限定时间内求得较优解. 然而,传统的启发式方法从原大规模问题直接开始搜索,无法利用先前相关的优化知识,导致收敛速度较慢. 因此,提出面向复杂物流配送场景的车辆路径规划多任务辅助进化算法 (Multitask-based assisted evolutionary algorithm, MBEA), 通过使用迁移优化方法加快算法收敛速度,其主要思想是通过构造多个简单且相似的子任务用于辅助优化原大规模问题. 首先从原大规模问题中随机选择一部分客户订单用于构建多个不同的相似优化子任务,然后使用进化多任务 (Evolutional multitasking, EMT) 方法用于生成原大规模问题和优化子任务的候选解. 由于优化子任务相对简单且与原大规模问题相似,其搜索得到的路径特征可以通过任务之间的知识迁移辅助优化原大规模问题,从而加快其求解速度. 最后,提出的算法在京东物流公司快递取送货数据集上进行验证,其路径规划效果优于当前最新提出的路径规划算法.

关键词 车辆路径规划问题, 时间窗约束, 同时取送货, 进化算法, 迁移优化

引用格式 李坚强, 蔡俊创, 孙涛, 朱庆灵, 林秋镇. 面向复杂物流配送场景的车辆路径规划多任务辅助进化算法. 自动化学报, 2024, 50(3): 544-559

DOI 10.16383/j.aas.c230043

Multitask-based Assisted Evolutionary Algorithm for Vehicle Routing Problems in Complex Logistics Distribution Scenarios

LI Jian-Qiang^{1,2} CAI Jun-Chuang¹ SUN Tao¹ ZHU Qing-Ling¹ LIN Qiu-Zhen¹

Abstract In complex logistics, addressing the vehicle routing problem (VRP) with simultaneous pickup and delivery and time windows, an NP-hard problem, becomes increasingly challenging as the scale expands. Traditional heuristic methods, often unable to leverage prior optimization knowledge, result in slow convergence. To address this, we introduce a multitask-based evolutionary algorithm (MBEA), which assists the optimization of the original large-scale problem by constructing multiple simple and similar subtasks and utilizing transfer learning to accelerate convergence speed. First, a subset of orders is randomly selected from the original problem to construct various subtasks, and then a multitask evolutionary approach is applied to generate candidate solutions for the original problem and subtasks. Given that the subtasks are simpler yet similar to the original problem, useful routing traits can be shared through knowledge transfer among the tasks, thereby speeding up its evolutionary search. To validate MBEA's efficacy, empirical studies were conducted on a large-scale express dataset from Jingdong, and the results demonstrate that MBEA outperforms recently proposed vehicle routing algorithms.

Key words Vehicle routing problem (VRP), time windows constraint, simultaneous pickup and delivery, evolutionary algorithm, transfer optimization

Citation Li Jian-Qiang, Cai Jun-Chuang, Sun Tao, Zhu Qing-Ling, Lin Qiu-Zhen. Multitask-based assisted evolutionary algorithm for vehicle routing problems in complex logistics distribution scenarios. *Acta Automatica Sinica*, 2024, 50(3): 544-559

收稿日期 2023-02-10 录用日期 2023-08-07

Manuscript received February 10, 2023; accepted August 7, 2023

国家自然科学基金 (62325307, 62073225, 62203134, 62376163, 62203308), 广东省自然科学基金 (2023B1515120038, 2019B151502018), 深圳市科技计划项目 (20220809141216003), 深圳大学科学仪器开发项目 (2023YQ019) 资助

Supported by National Natural Science Foundation of China (62325307, 62073225, 62203134, 62376163, 62203308), Natural Science Foundation of Guangdong Province (2023B1515120038, 2019B151502018), Shenzhen Science and Technology Program (20220809141216003), and the Scientific Instrument Developing Project of Shenzhen University (2023YQ019)

本文责任编辑 李力

车辆路径规划问题 (Vehicle routing problem, VRP) 作为一个经典的组合优化问题, 在车辆辅助多无人机监控^[1]、机场接送服务^[2]、无人驾驶车辆路径规划^[3-5]、物流^[6-8]等领域广泛应用, 因此近年来引

Recommended by Associate Editor LI Li

1. 深圳大学计算机与软件学院 深圳 518060 2. 深圳大学大数据系统计算技术国家工程实验室 深圳 518060

1. College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060 2. National Engineering Laboratory for Big Data System Computing Technology, Shenzhen University, Shenzhen 518060

起研究者的广泛关注. 一般来说, VRP 的优化目标是最小化车辆从仓库出发并为多个客户派送货物的路径总成本. VRP 已被证明是 NP-难问题^[9-10], 因此是一个极具挑战性的组合优化问题. 在过去的几十年中, 学者们开展了大量求解 VRP 相关问题的研究工作. 例如, 为更好地优化大规模 VRP, 文献 [11] 提出一种进化多目标路径分组方法. 该方法采用多目标进化算法进行路径分组, 同时优化 3 个目标, 即组内距离、组间距离和组间大小平衡, 然后使用局部搜索方法提高组内路径的质量.

近年来, 复杂物流配送场景的 VRP 引起研究人员的广泛关注. 在实际应用中, 绿色制造业和物流在现代供应链管理中扮演着重要角色^[12]. 制造工厂需要从客户处收集废弃产品, 以供再次使用或适当处置, 这被称为逆向物流^[13]. 一般来说, 逆向物流与货物的双向流动有关, 即交货和取货. 前者指向客户交付货物, 后者指从客户处收取货物. 由于逆向物流在降低能源消耗和减少环境污染方面的显著作用, 其已应用于各个应用场景的配送系统, 如图书馆图书配送^[14]、杂货配送^[15]和包裹配送^[16]等. 此外, 为提高配送效率, 取送货服务需要在预定的时间窗口内完成. 这类问题称为带时间窗和同时取送货的车辆路径问题 (Vehicle routing problem with simultaneous pickup-delivery and time windows, VRPPDT)^[17].

VRPPDT 是一个更具挑战性的组合优化问题, 包含经典 VRP 中不存在的一些复杂约束, 因此也是一个 NP-难问题^[18-20]. 在这些约束条件中, 时间窗口定义了车辆到达客户并开始服务的最早和最晚时间, 具有软时间窗和硬时间窗约束. 配送服务违反软时间窗约束将会受到惩罚^[21], 但不能违反硬时间窗约束, 即如果车辆在时间窗口之前到达, 则必须等到服务开始时间, 而且不允许在时间窗口之后到达^[22]. 本文考虑硬时间窗约束, 规划一队同质或异质车辆从仓库出发, 为分布在不同地区的客户提供服务. 车辆不仅需要将货物从仓库交付给客户, 还需要同时在客户处收取货物带回仓库, 且不能违反车辆装载容量和客户指定的时间窗约束^[23]. 一般的数学规划方法难以求解上述 VRPPDT^[17], 因此研究人员通常设计启发式算法进行求解, 期望在合理计算时间内找到高质量的候选解. 目前求解 VRPPDT 的启发式算法包括差分进化^[24]、遗传算法^[17]、模拟退火^[13]、群体智能^[25]、可变邻域搜索^[26]、自适应大邻域搜索^[27]等.

然而, 在现实应用中, 很少有问题是孤立存在的, 正确使用从相关问题中学到的知识, 可以提高解决新问题的能力^[28-29]. 因此, 相关文献中提出迁移

优化方法, 通过从已优化的相似任务中迁移知识到新任务来提高其优化效果^[30]. 这种进化多任务 (Evolutional multitasking, EMT) 算法已成为进化计算领域的研究热点, 其目的是通过多个相似任务之间的知识迁移, 加快全局最优解的搜索速度^[31-34]. 与传统进化算法求解单个优化任务相比, EMT 算法可以同时求解多个优化任务, 且每个任务对应一个特定的优化问题. 通过利用优化问题之间潜在的协同作用, EMT 算法在解质量和搜索速度方面的优秀性能已在组合优化问题上得到验证^[35-37]. 例如, 随着众包和共享经济的出现, 文献 [35] 研究一种具有临时驾驶员的广义车辆路径问题变体, 称为具有异质容量、时间窗和临时驾驶员的车辆路径问题 (Vehicle routing problem with heterogeneous capacity, time window, and occasional driver, VRPHTO), 同时提出一种新的进化多任务算法, 使用单个种群同时优化多个 VRPHTO. 与大多数现有的通过交叉实现跨任务隐式知识迁移的 EMT 算法不同, 文献 [36] 提出一种显式 EMT 算法 (Explicit EMT algorithm, EEMTA) 用于求解 VRP 等组合优化问题. EEMTA 包含用于捕获迁移映射的加权 L_1 范数正则化学习过程, 以及基于解的跨 VRP 知识迁移过程, 其性能在仿真实验中被证明是有效的. 为加快车辆路径优化的速度, 文献 [38] 建议通过学习新的客户表示来捕获之前优化的路径规划解中隐藏的有用特征. 这些客户表示可以作为先验知识在 VRP 之间进行传递, 从而优化目标 VRP. 文献 [39] 求解只带有容量约束的一般 VRP 时采用 K-means 方法, 将原 VRP 分解为多个只包含单条路径的简单 VRP 作为子任务, 并使用模因搜索同时优化子任务和原始任务. 最后, 所有子任务的解直接叠加组成原始任务的解, 以实现子任务优化原始 VRP 任务. 然而, 其知识迁移方式比较简单, 只是使用简单的分解和合并方法来生成原始 VRP 任务的候选解.

受此启发, 为提高 VRP 算法的性能, EMT 算法可以在多种形式的 VRP 上执行进化搜索, 而不仅仅是在单一形式 VRP 上. 因此, 在不同形式的 VRP 上搜索得到的有用路径轨迹可以在多任务之间进行传递, 以加速车辆路径规划的搜索过程. 因此, 为更好地求解复杂物流配送场景的大规模 VRPPDT, 本文首先介绍使用 EMT 算法求解 VRPPDT 的想法, 并提出一种面向复杂物流配送场景的车辆路径规划多任务辅助进化算法 (Multi-task-based assisted evolutionary algorithm, MBEA) 用于求解大规模 VRPPDT. MBEA 首先将原大规模 VRPPDT 分解为多个低维子任务, 并

利用这些子任务使用 EMT 算法求解原 VRPPDT. MBEA 主要包括 3 个操作: 1) 子任务生成; 2) 基于多任务的知识迁移; 3) 环境选择. 在子任务生成中, MBEA 通过从原任务中随机选择一些客户订单来创建多个不同的子任务. 因为子任务的客户规模比原任务小, 所以更容易求解这些子任务. 在基于多任务的知识迁移中, MBEA 使用进化多任务方法用于生成原任务和优化子任务的候选解. 在环境选择中, MBEA 从父代种群和子代种群的混合种群中选择 N 个 (N 是种群大小) 最好的个体存活到下一代. 经过不断迭代进化, MBEA 可以快速得到原大规模 VRPPDT 的高质量候选解. 本文的主要贡献总结如下.

1) 针对复杂物流配送场景的大规模 VRPPDT, 提出一种多任务辅助进化算法 MBEA. MBEA 包括子任务生成、基于多任务的知识迁移和环境选择等操作, 可以在子任务和迁移优化方法的帮助下更快地求解大规模 VRPPDT. 这是首次应用进化多任务方法用于求解大规模 VRPPDT, 具有显著的实际应用与研究价值.

2) MBEA 的优化性能已在实际大规模 VRPPDT 数据集上得到验证. 该数据集来自于京东公司的物流配送系统. 与最近提出的 5 种 VRPPDT 算法相比, MBEA 在大多数测试问题上都能取得更好的优化性能, 证明了迁移优化对大规模 VRPPDT 的有效性.

本文的其余部分组织如下. 第 1 节给出本文研究的 VRPPDT 定义, 并回顾现有的相关研究. 第 2 节详细介绍 MBEA 算法, 并在第 3 节给出其与最新提出的 VRPPDT 算法在大规模京东数据集的仿真比较结果. 最后, 第 4 节给出本文的结论并讨论未来的研究工作.

1 背景

1.1 带时间窗和同时取送货的车辆路径问题

在复杂物流配送场景里, 带时间窗和同时取送货的车辆路径问题 VRPPDT 需要在特定时间窗口内提供同时取送货服务. VRPPDT 的目标是为多辆车规划路径, 并在满足约束条件的同时以最低的总成本为客户提供同时取送货服务. 一般来说, VRPPDT 可以由完全图 $G = (V, E)$ 来表示, 其中 $V = \{0, 1, 2, \dots, M\}$ 是仓库和 M 个客户节点的集合, $E = \{(i, j) | i, j \in V, i \neq j\}$ 是每个节点之间的弧集. VRPPDT 模型如图 1 所示. 为方便表示, 仓库始终表示为 0, 客户表示为 $1, 2, \dots, M$. 每条弧 $(i, j) \in E$ 与行驶距离 $dist(i, j)$ 和行驶时间 $time(i, j)$ 相关. 每

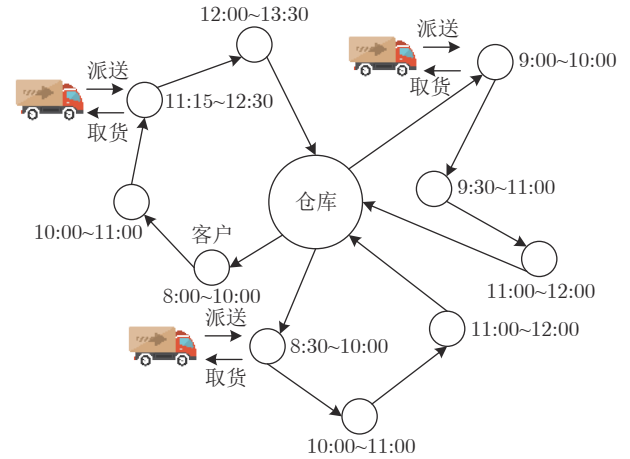


图 1 VRPPDT 模型

Fig. 1 The model of the VRPPDT

个节点 $i \in V$ 有 5 个属性, 即送货需求 d_i 、提货需求 p_i 、时间窗口 $[a_i, b_i]$ 和服务时间 s_i . d_i 是从仓库交付给客户 i 的货物数量. p_i 是从客户 i 处取走的必须交付到仓库的货物数量. a_i 和 b_i 分别是客户 i 接受取货服务和送货服务的开始时间和结束时间. 当车辆在开始时间 a_i 之前到达客户 i 时, 车辆必须等待; 车辆不能在结束时间 b_i 之后到达客户 i . 最后, s_i 是车辆在客户 i 卸载和装载货物所需的时间. a_0 和 b_0 分别是车辆可以离开仓库 0 的最早时间和车辆可以返回仓库 0 的最晚时间, 且 $d_0 = p_0 = s_0 = 0$.

假设在当前配置中, 最初在仓库 0 中有 J 辆车可以调度, 其中每辆车的容量为 C , 调度成本为 u_1 . 每辆车从仓库出发, 为客户送货、取货, 最后返回仓库. 因此, VRPPDT 的解 S 由一组车辆路线表示, 即 $S = \{R_1, R_2, \dots, R_K\}$. 每条路线 R_i 由车辆访问的一系列节点组成, 即 $R_i = (h_{i,1}, h_{i,2}, \dots, h_{i,L_i})$, 其中, $h_{i,j}$ 表示车辆在路线 R_i 访问的第 j 个节点, L_i 表示路线 R_i 的节点长度. 为方便介绍, 这里省略了 R_i 中的下标 i , 即 $R = (h_1, h_2, \dots, h_L)$, 则 R 的总行驶距离记为 $TD(R)$, 计算式为

$$TD(R) = \sum_{j=1}^{L-1} dist(h_j, h_{j+1}) \quad (1)$$

车辆到达和离开 h_j 的时间, 分别表示为 $arr(h_j)$ 和 $dep(h_j)$, 由以下计算式计算得出

$$\begin{cases} dep(h_1) = a_0 \\ arr(h_j) = dep(h_{j-1}) + time(h_{j-1}, h_j), & j > 1 \\ dep(h_j) = \max\{arr(h_j), a_{h_j}\} + s_{h_j}, & j > 1 \end{cases} \quad (2)$$

车辆到达 h_j 时的载货量, 记为 $load(h_j)$, 计算

式为

$$\begin{cases} \text{load}(h_1) = \sum_{j=1}^L d_{h_j} \\ \text{load}(h_j) = \text{load}(h_{j-1}) - d_{h_j} + p_{h_j}, \quad j > 1 \end{cases} \quad (3)$$

解 S 的总成本记为 $TC(S)$, 由两部分组成: 车辆调度成本 $u_1 \cdot K$ 和运输成本 $u_2 \cdot TD(S)$. 因此, VRPPDT 的优化目标是找到 TC 值最小的 S , 计算式为

$$\min_S TC(S) = u_1 \cdot K + u_2 \cdot \sum_{i=1}^K TD(R_i) \quad (4)$$

$$\text{s.t. } K \leq J$$

$$h_{i,1} = h_{i,L_i} = 0, \quad 1 \leq i \leq K \quad (5)$$

$$\sum_{i=1}^K [h_{i,j} == x] = 1, \quad 1 \leq x \leq M \quad (6)$$

$$\text{load}(h_{i,j}) \leq C, \quad 1 \leq i \leq K, \quad 2 \leq j \leq L_i \quad (7)$$

$$a_{h_{i,j}} \leq \text{arr}(h_{i,j}) \leq b_{h_{i,j}}, \quad 1 \leq i \leq K, \quad 2 \leq j \leq L_i \quad (8)$$

$$\text{dep}(h_{i,j}) \geq a_0, \quad \text{arr}(h_{i,L_i}) \leq b_0, \quad 1 \leq i \leq K \quad (9)$$

在目标函数 (4) 中, u_1 表示每辆车的调度成本, u_2 表示单位距离的运输成本, 使用的车辆数 K 不能超过可用车辆数 J . 约束 (5) 表示每条路线必须从仓库出发并返回仓库. 约束 (6) 规定每个顾客只能服务一次. 约束 (7) 保证每辆车在运输过程中不能超载. 约束 (8) 表示必须在规定时间窗口内为客户服务. 约束 (9) 规定车辆只能在开始时间 a_0 之后离开仓库, 并且必须在结束时间 b_0 之前返回仓库.

VRPPDT 的主要难点是容量和服务时间受到限制. 前者是客户可以同时有送货和取货的需求. 由于这两种需求对车辆负载的影响不同 (一个增加负载, 另一个减少负载), 这样的特性使得在容量限制下很难确定分配多少车辆服务客户. 后者是每个客户都与一个硬时间窗口相关联, 进一步增加了为特定车辆规划订单服务顺序的难度. 除上述约束外, 还需要考虑 VRPPDT 搜索空间的索引. 因此, 充分平衡候选解的多样性和收敛性对于算法设计也很重要.

1.2 相关工作

当前求解 VRPPDT 的方法通常可以分为两类: 精确方法和启发式方法^[40-42]. 下面将详细介绍相关研究工作.

近年来, 许多研究人员提出求解 VRPPDT 的精确算法. 文献 [43] 首次尝试使用精确算法求解多达 20 个客户的 VRPPDT. 文献 [44] 研究如何将分支定价技术用于求解 VRPPDT, 并对比了两种

不同求解定价子问题的方法: 精确动态规划和状态空间松弛. 通过应用双向搜索, 文献 [44] 通过实验仿真验证分支定价技术在求解 VRPPDT 中的有效性. 文献 [45] 提出 VRPPDT 的两种混合整数线性模型公式, 即车流模型和商品流模型. 同时, 文献 [45] 提出缩小域的预处理技术和有效的切割平面来强化提出的模型, 并使用数学优化软件 CPLEX 求解源自实际问题的对称基准实例和新的不对称实例. 其中, CPLEX 通过将该实例转化为数学规划模型以快速获得候选解. 与启发式算法相比, 求解 VRPPDT (VRP with simultaneous pickup and delivery) 的精确方法很少. 因此, 文献 [46] 首先为 VRPPDT 开发一种剪支定价算法. 该算法在涉及多达 200 个客户的著名基准问题上进行了测试. 然而, 精确算法通常只适用于求解客户数量少于 100 的小问题实例, 在求解大规模问题上性能仍不理想^[47].

在过去十年中, 用于求解 VRPPDT^[43] 的启发式算法相比精确方法更受研究人员的欢迎. 这些启发式方法主要可以分为两类: 基于单一解的算法和基于群体的算法^[48]. 具体来说, 基于单一解的算法, 如模拟退火 (Simulated annealing, SA)^[49]、禁忌搜索 (Tabu search, TS)^[50]、大邻域搜索 (Large neighborhood search, LNS)^[51] 等, 称为轨迹方法. 轨迹方法只涉及单一解在搜索空间中移动以形成轨迹, 被视为局部搜索方法的智能版本. 为开发一种有效的启发式算法来解决 VRPPDT, 文献 [23] 提出一种并行模拟退火算法. 该算法包括基于剩余容量和径向超载的插入启发式算法. 文献 [52] 设计一种混合局部搜索算法, 用于求解同时取送货的异构车辆路径问题. 该算法将非单调阈值调整策略与禁忌搜索相结合, 且该阈值函数具有能够自我调整的自适应特性. 文献 [53] 提出一种基于分解的局部搜索方法用于优化多目标 VRPPDT, 可以在短时间内获得高质量的有效候选解. 为加快收敛速度, 该方法提出了一种使用 7 种邻域算子的新型局部搜索算法. 同时, 为保持多样性, 该方法采用分解的概念, 首先将问题分解为多个单目标问题, 然后利用局部搜索对这些子问题进行优化.

另一方面, 用于求解 VRPPDT 的基于种群的启发式算法包括进化算法 (即遗传算法^[54-55]、模因算法^[56-57]、差分进化算法^[58] 等) 和群体智能算法 (即粒子群优化算法^[59]、蚁群优化算法^[60-61] 等). 在求解 VRPPDT 时, 基于种群的启发式算法起着至关重要的作用, 并提供良好的 VRPPDT 候选解. 为求解大规模 VRPPDT, 文献 [62] 提出一种具有高效局部搜索和扩展邻域的新型模因算法 (Memetic algorithm with efficient local search and extended

neighborhood, MATE). 由于 MATE 具有新颖的初始化过程、交叉和大步长的局部搜索算子, 与现有算法相比, 它能够更高效地搜索决策空间. 此外, 由于其评估机制具有常数级时间复杂度, MATE 的局部搜索操作也更加高效. 为满足实际逆向物流中包含的所有复杂约束, 文献 [63] 构建一种新的数学模型, 用于处理具有时间窗和多个决策者的同时取送货问题. 同时, 文献 [63] 提出一种基于混合优先级的嵌套遗传算法, 结合模糊逻辑控制器和模糊随机模拟方法用于求解该问题. 文献 [64] 定义一个通用多目标 VRPPDT (Multiobjective VRPPDT, MO-VRPPDT), 并用于求解现实世界的一组 MO-VRPPDT. 同时, 文献 [64] 设计多目标局部搜索算法和多目标模因算法用于求解 MO-VRPPDT. 为尽可能降低路径规划的运营成本, 文献 [65] 提出一种改进的粒子群优化算法, 在满足客户取送货需求的同时最小化路径的总距离.

2 算法设计

本节详细介绍面向复杂物流配送场景的车辆路径规划多任务辅助进化算法 (MBEA), 用于解决同时取送货和时间窗的大规模车辆路径问题 VRPPDT. MBEA 的总体框架如图 2 所示. 首先, 第 2.1 节介绍 MBEA 的算法流程. 接着, 第 2.2 节介绍子任务生成. 然后, 第 2.3 节介绍基于多任务的知识迁移. 最后, 第 2.4 节介绍 MBEA 的环境选择操作.

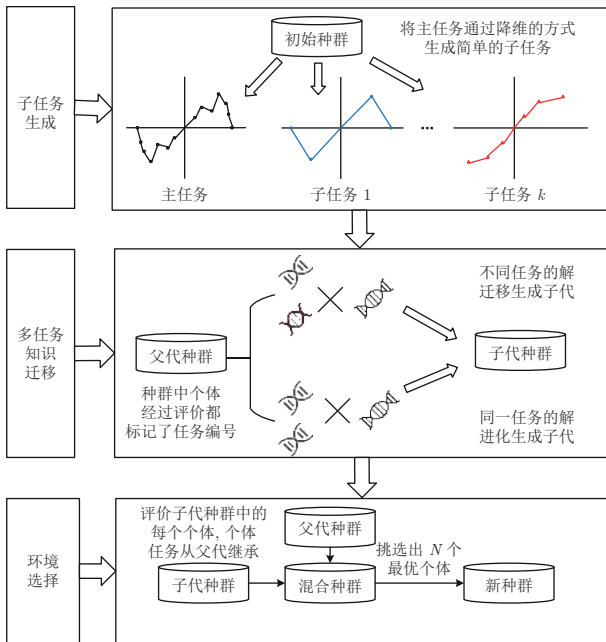


图 2 MBEA 总体框架图

Fig. 2 The overall framework diagram of MBEA

2.1 MBEA 总体框架

受最近提出的多任务辅助优化方法^[31]启发, MBEA 将一个原 VRPPDT 分解为 k 个较低维度的子任务, 并利用这些子任务使用进化多任务方法去优化原 VRPPDT. 在子任务生成中, MBEA 通过从原始任务中随机选择 $\lfloor ran \times |V| \rfloor$ ($ran \in (0, 1)$, $|V|$ 是原始任务的客户数量) 个客户订单来创建 k 个不同的子任务. 由于其客户数量比原任务少, 子任务相对容易求解. 而且, 子任务中的客户是原任务的子集, 所以原任务和子任务具有一定的相似性. 因此, 通过将子任务的最优解迁移到原任务上可以加快其求解的收敛速度. 在基于多任务的知识迁移中, MBEA 使用两种不同的交叉算子生成子代. 当两个父代个体属于同一个任务时, MBEA 将使用基于路径的交叉^[66]生成子代; 否则, MBEA 将使用顺序交叉^[67]生成子代. 在环境选择中, MBEA 先评估子代群体中的每个个体. 然后, MBEA 会从父代种群和子代种群的混合种群中选出 N 个 (N 是种群规模) 最好的个体存活到下一代. 经过不断地迭代进化, MBEA 可以得到 VRPPDT 的高质量候选解.

为详细说明 MBEA 的步骤, 算法 1 中给出了它的伪代码, 其中有两个输入: 一个要解决的 VRPPDT 实例 $F(V, E)$ 和子任务的数量 k . 在第 1 行, MBEA 首先使用基于序列的统一表示来初始化具有 N 个个体的种群 P , 每个个体都有 $|V|$ 个变量. 图 3 显示了一个具有 10 个客户的个体编码方法. 然后, MBEA 开始通过多阶段进化来优化种群 P . 在每个阶段, 子任务生成 (Task generation, TG) 操作首先创建 k 个不同的子任务 (第 3 行). 这些子任务的客户数量远少于原 VRPPDT. 然后, 当评估次数小于评价次数 TE 时, MBEA 执行基于多任务的知识迁移 (Multitask-based knowledge transfer, MBKT) 操作和环境选择 (Environmental selection, ES) 操作用于优化种群. 在第 5 行, MBEA 对父代种群 P 和 $(k+1)$ 个任务使用 MBKT 操作生成子代种群 Q . 然后, 在第 6 行, MBEA 使用 ES 操作从父代种群 P 和子代种群 Q 选择 N 个个体作为新种群 P 存活到下一代. 当评估次数超过 TE 时, MBEA 保留种群 P 中的 N_{be} 个最佳个体, 并重新初始化剩余的 $N - N_{be}$ 个个体以适应下一阶段的进化 (第 8 行).

算法 1. MBEA 总体框架

输入. $F(V, E)$: VRPPDT 实例, k : 子任务个数

输出. VRPPDT 最好的解

1: 对 N 个个体使用基于排列的统一表示初始化种群 P , 每个个体有 $|V|$ 个变量;

```

2: for Restart := 1 to  $N_{re}$  do
3:   $(F(V_1, E_1), \dots, F(V_k, E_k), P) = TG(F(V, E), P, k)$ ;
4:  while 评价次数  $< TE$  do
5:     $Q = MBKT(P, F(V, E), F(V_1, E_1), \dots, F(V_k, E_k))$ ;
6:     $P = ES(P, Q, F(V, E), F(V_1, E_1), \dots, F(V_k, E_k))$ ;
7:  end while
8:  保留  $N_{be}$  个最好个体并初始化其他个体;
9: end for

```

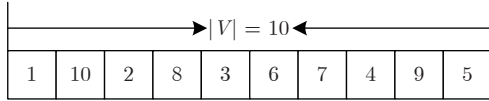


图 3 一个个体的编码方法

Fig.3 The coding method of an individual

2.2 子任务生成操作

本节介绍 TG 操作以及进化多任务中使用的一些概念. 为阐明子任务生成操作的步骤, 下面介绍算法 2 中的伪代码. TG 有 3 个输入: VRPPDT $F(V, E)$ 、种群 P 和子任务数 k . 在第 2 行, TG 首先生成一个随机数 $ran (ran \in (lower, 1))$, 其中 $lower$ 表示子任务的维度相对于原任务的最低比例. 然后, 在第 3 行, TG 通过公式 $D = \lfloor ran \times |V| \rfloor$ 计算第 i 个子任务中的客户数量来确定子任务的维度. 在第 4 行, TG 从原始任务 $F(V, E)$ 中选择索引为 1 到 D 的 D 个客户组成节点信息 V_i 和对应的边信息 E_i 作为子任务 $F(V_i, E_i)$. 图 4 上面的虚框中给出了从维度为 10 的候选解 $[1, 10, 2, 8, 3, 6, 7, 4, 9, 5]$ 中生成子任务候选解的示例. 若生成的随机数 $ran = 0.6$, 则表示子任务的维度是 6, 这时便从原任务中选择索引 1 到 6 的节点作为子任务的候选解 $[1, 2, 3, 6, 4, 5]$. 由于子任务是从原任务中随机选择 D 个客户生成的, 这确保了子任务的多样性, 并有助于原任务跳出局部最优解; 此外, 由于子任务是原任务的子集, 其拓扑结构与原任务相似, 且相对容易求解, 故子任务能够辅助加快原任务往全局最优

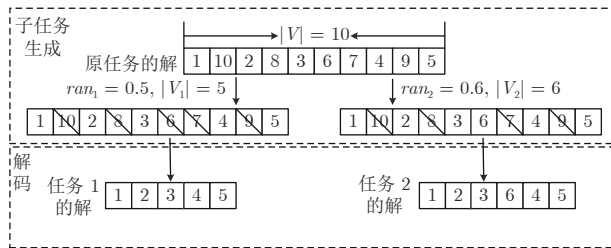


图 4 子任务生成及解码过程

Fig.4 The generation and decoding process of the subtask

解的收敛速度. 该子任务生成的有效性将在第 3.5 节实验中进行讨论与验证.

算法 2. 任务生成 (TG)

输入. $F(V, E)$: VRPPDT 实例, k : 子任务个数, P : 种群
输出. k 个子任务 $F(V_1, E_1), \dots, F(V_k, E_k)$, 种群 P

```

1: for  $i = 1$  to  $k$  do
2:   产生一个随机数  $ran (ran \in (lower, 1))$ ;
3:    $D = \lfloor ran \times |V| \rfloor$ ;
4:   选择  $F(V, E)$  中索引为 1 到  $D$  的  $D$  个客户形成节点信息  $V_i$  和对应边信息作为  $E_i$  作为子任务  $F(V_i, E_i)$ ;
5: end for
6: for  $p \in P$  do
7:   计算  $p$  在对应的  $k + 1$  个任务上的适应值向量作为因子代价  $f_p$ ;
8:   计算  $p$  的排位向量作为因子排位  $r_p$ ;
9:   计算  $p$  的标量适应度  $\varphi_p = [1 / \min_{j \in \{0, \dots, k\}} r_p^j]$ ;
10:  计算技能因子  $\tau_p = \arg \min r_p^j, j \in \{0, 1, \dots, k\}$ ;
11: end for

```

然后, 在第 6 ~ 11 行, TG 将计算进化多任务中的 4 个参数: 因子代价 (Factorial cost)、因子排位 (Factorial rank)、标量适应度 (Scalar fitness) 和技能因子 (Skill factor)^[31]. 个体 p 的因子代价 f_p 表示其在特定任务上的适应度或目标值. 对于 $k + 1$ 个任务, 会有一个长度为 $k + 1$ 的向量, 其中每个维度给出 p 在相应任务上的适应度. 因子排位 r_p 表示个体 p 在种群成员列表中的索引. 该列表是按种群个体在一个特定任务上的因子代价升序排序的. 个体 p 的标量适应度 φ_p 是根据其在所有任务中的最佳排名来定义的, 即 $\varphi_p = [1 / \min_{j \in \{0, \dots, k\}} r_p^j]$. 个体 p 的技能因子 τ_p 表示在所有任务中 p 表现最有效的任务, 即 $\tau_p = \arg \min r_p^j, j \in \{0, 1, \dots, k\}$. 最后, TG 输出 k 个子任务 $F(V_1, E_1), F(V_2, E_2), \dots, F(V_k, E_k)$ 和种群 P . 从算法 2 可以看出, 随机生成 k 个子任务的时间复杂度是 $O(k)$, 而计算种群 P (种群规模为 N) 的各个指标的时间复杂度是 $O(N)$, 故 MBEA 中的子任务生成操作需要的最坏时间复杂度是 $O(k + N)$.

2.3 基于多任务知识迁移

本节介绍基于多任务的知识迁移 MBKT 操作. 在生成子代过程中, 每次通过锦标赛选择法从父代种群 P 中选择两个父代个体. 当两个父代个体属于同一个任务时, MBEA 将使用基于路径的交叉生成子代; 否则, MBEA 将使用顺序交叉进行知识迁移生成子代. 为详细说明 MBKT 的步骤, 算法 3 介绍了它的伪代码. MBKT 有两个输入: 种群 P 和所

有任务 $F(V, E), F(V_1, E_1), \dots, F(V_k, E_k)$. 在第 1 行, MBKT 先将一个空种群 Q 初始化为子代种群. 然后, MBKT 会生成子代个体并将它们添加到 Q 中, 直到 Q 中的个体数量达到 N . 为生成子代个体, 在第 3 行, MBKT 首先通过锦标赛选择法从父代种群 P 中选择两个父代个体 p_1 和 p_2 . 然后, MBKT 会判断 p_1 的技能因子 τ_{p_1} 和 p_2 的技能因子 τ_{p_2} 是否相等. 如果 τ_{p_1} 等于 τ_{p_2} , 则 p_1 和 p_2 的解在该任务表现一样好. 然后, 在第 5 行, MBKT 将根据任务 $F(V_{\tau_{p_1}}, E_{\tau_{p_1}})$ 或 $F(V_{\tau_{p_2}}, E_{\tau_{p_2}})$ 将 p_1 和 p_2 解码为 p'_1 和 p'_2 . 图 4 显示了如何根据 $F(V_{\tau_{p_1}}, E_{\tau_{p_1}})$ 或 $F(V_{\tau_{p_2}}, E_{\tau_{p_2}})$ 将原始空间中的候选解解码为子空间. 从算法 3 可以得出, 这个基于多任务的知识迁移操作需要的最坏时间复杂度是 $O(N)$.

算法 3. 基于多任务的知识迁移 (MBKT)

输入. 种群 P , 所有任务 $F(V, E), F(V_1, E_1), \dots, F(V_k, E_k)$

输出. 子代种群 Q

- 1: 初始化子代种群 Q 为空;
- 2: **while** $|Q| \neq N$ **do**
- 3: 使用锦标赛选择法从 P 中选出两父代 p_1, p_2 ;
- 4: **if** $\tau_{p_1} == \tau_{p_2}$ **do**
- 5: 根据任务 $F(V_{\tau_{p_1}}, E_{\tau_{p_1}})$ 或 $F(V_{\tau_{p_2}}, E_{\tau_{p_2}})$ 将 p_1, p_2 解码成 p'_1, p'_2 ;
- 6: 将 p'_1, p'_2 切分成若干条路径;
- 7: p'_1, p'_2 通过交叉方法生成子代 c_1, c_2 ;
- 8: **else**
- 9: 根据原任务 $F(V, E)$ 将 p_1, p_2 解码成 p'_1, p'_2 ;
- 10: p'_1, p'_2 通过交叉-迁移生成子代 c_1, c_2 ;
- 11: **end if**
- 12: 将子代 c_1, c_2 加入种群 Q ;
- 13: **end while**

在图 4 中, 候选解在原空间的维数为 10. 如果子任务 $F(V_1, E_1)$ 中有 5 个客户, 那么 MBKT 会删除解 $[1, 10, 2, 8, 3, 6, 7, 4, 9, 5]$ 中大于 5 的编码, 从而得到子任务 $F(V_1, E_1)$ 的解 $[1, 2, 3, 4, 5]$. 然后, 在第 6 行, MBKT 将 p'_1 和 p'_2 的编码拆分为多条路径. 图 5 显示了将个体的编码拆分为多条路径的过程. 假设个体编码为 $[1, 2, 3, 4, 5]$, 车辆容量为 10, 车辆成本为 20, 客户服务时间为 0. 在满足时间窗和车辆容量约束的条件下, 解码序列可以分为 3 种不同的路径结果: $a = [0, 1, 0, 2, 0, 3, 0, 4, 5, 0]$, $b = [0, 1, 2, 0, 3, 4, 0, 5, 0]$ 和 $c = [0, 1, 0, 2, 3, 0, 4, 5, 0]$. MBKT 会计算不同路径的总成本, 最终选择成本最低的路径. 以路径 c 中的路线 $[0, 1, 0]$ 为例,

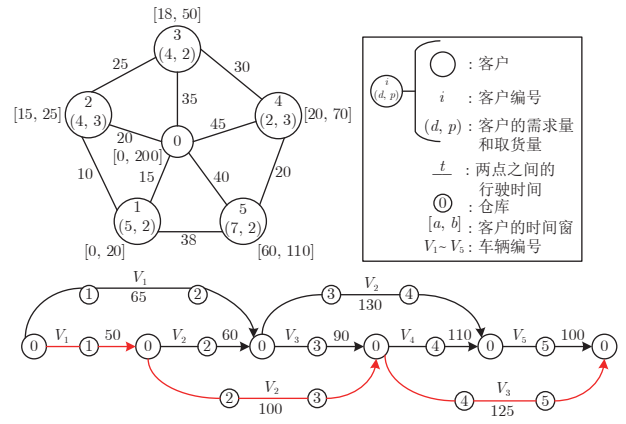


图 5 一个解的切分过程

Fig. 5 The splitting process of a solution

行程成本为 30, 车辆成本为 20, 因此总成本为 50. MBKT 计算每条可能路径的总成本. 最后, 我们可以得到成本最低的路径: $[0, 1, 0, 2, 3, 0, 4, 5, 0]$, 总成本为 275.

然后, 在第 7 行, MBKT 对两个父代个体 p'_1 和 p'_2 使用基于路径的交叉算子生成两个子代个体 c_1 和 c_2 . 图 6 显示了基于路径的交叉算子的操作过程. 对于两个父代个体 p'_1 和 p'_2 , 基于路径的交叉算子首先在 p'_1 中随机选择一条路径 $Route 2$, 同时在 p'_2 中随机选择一条路径 $Route 2$. 接着将 p'_2 中的路径 $Route 2$ 换为 p'_1 中的路径 $Route 2$ 以生成新的子代 c_2 . 在此替换过程中, c_2 中的一些客户会发生冲突 (如客户 A), 并且有些客户不会被访问 (如客户 B). 所以基于路径的交叉算子需要调整 c_2 中的路径. 先删除 c_2 的 $Route 1$ 中的客户 A , 然后将客户 B 插入 c_2 的 $Route 2$ 中, 生成调整后的个体 c_2 . 子代 c_1 也可以按同样的方法得到.

如果 τ_{p_1} 不等于 τ_{p_2} , 则 MBKT 将执行知识迁移操作. 首先, MBKT 根据原始任务 $F(V, E)$ 将 p_1 和 p_2 解码为 p'_1 和 p'_2 (第 9 行). 然后, MBKT 对两个父代个体 p'_1 和 p'_2 使用顺序交叉算子生成两个子代个体 c_1 和 c_2 (第 10 行). 顺序交叉算子的操作如图 7 所示. 顺序交叉算子先选择来自父代个体 p'_1 序列的一段 $[9, 2, 10, 4]$ 作为子代 c_1 对应段的位置. 然后, 顺序交叉算子将 p'_2 中不属于段 $[9, 2, 10, 4]$ 的编码按顺序插入到 c_1 中, 从而得到子代个体 c_1 . 子代个体 c_2 也可以用同样的方法生成. 当子代种群 Q 中的个体数量达到 N 时, 生成子代的操作结束. 最后, MBKT 返回子代种群 Q .

2.4 环境选择

本节介绍环境选择 ES 操作. 当得到父代种群 P 和子代种群 Q 时, MBEA 将选择更好的个体存

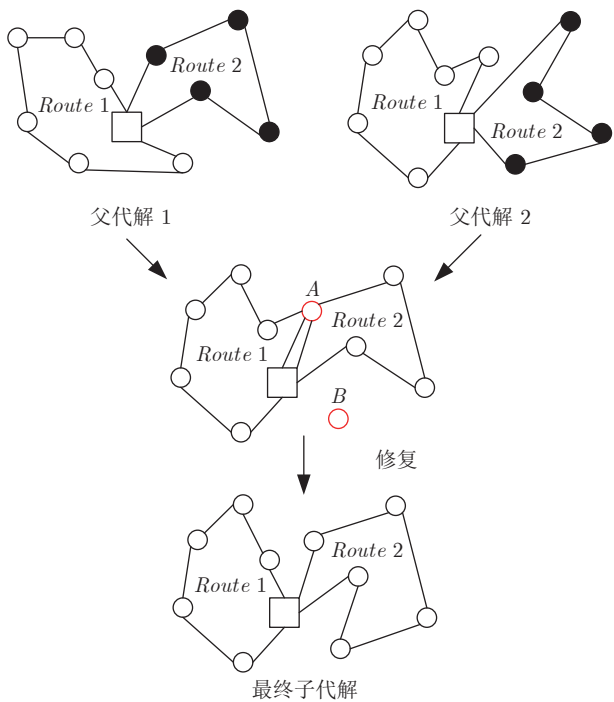


图 6 基于路径的交叉过程

Fig.6 The operation process of the route-based crossover

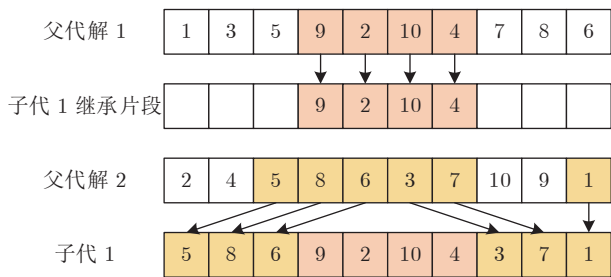


图 7 顺序交叉操作过程

Fig.7 The operation process of the order crossover

活到下一代. 为详细说明 ES 操作的步骤, 算法 4 给出了它的伪代码. ES 首先需要评估种群 Q 中的每个个体. 在第 2 行, ES 生成一个随机数 $ran \in (0, 1)$. 然后, 在第 3 ~ 7 行, 群体 Q 中的个体 c 以 50% 的概率继承父代个体 p_1 的技能因子 τ_{p_1} , 或以 50% 的概率继承父代个体 p_2 的技能因子 τ_{p_2} . 在第 8 行, ES 根据其技能因子 τ_c 对 c 执行局部搜索操作, 并更新 c 在任务 $F(V_{\tau_c}, E_{\tau_c})$ 的因子代价 f_c . 接下来, 在第 9 行, ES 将 c 在所有未评估任务的因子代价 f_c 设置为 ∞ . 在对种群 Q 中的所有个体进行评估后, 在第 11 行, ES 将父代种群 P 和子代种群 Q 合并为一个新种群 R . 然后, 在第 12 行, ES 更新种群 R 中所有个体的因子排位和标量适应度. 在第 13 行, 根据个体的标量适应度, ES 从种群 R 中

选出最好的 N 个个体作为新的种群 P 存活到下一代. 最后, 算法 4 将最终种群 P 作为结果并输出. 由于种群 Q 的大小为 N , 故由算法 4 易得, 环境选择的最坏时间复杂度是 $O(N)$.

算法 4. 环境选择 (ES)

输入. 种群 P , 子代种群 Q , 所有任务 $F(V, E), F(V_1, E_1), \dots, F(V_k, E_k)$

输出. 新种群 P

- 1: for $c \in Q$ do
- 2: 产生一个随机数 $ran \in (0, 1)$;
- 3: if $ran < 0.5$
- 4: c 继承父代节点 p_1 的技能因子 τ_{p_1} ;
- 5: else
- 6: c 继承父代节点 p_2 的技能因子 τ_{p_2} ;
- 7: end if
- 8: c 对任务 $F(V_{\tau_c}, E_{\tau_c})$ 使用局部搜索并更新其因子代价 f_c ;
- 9: 设置 c 对应其他未评估的任务的因子代价 f_c 为 ∞ ;
- 10: end for
- 11: 临时种群 $R = P \cup Q$;
- 12: 更新 R 中个体的因子排位和标量适应度;
- 13: 根据标量适应度选择最好的 N 个个体存活到下一代的种群 P .

3 仿真实验

3.1 实验设置

为验证所提出的 MBEA 算法的有效性, 特别是在解决复杂物流配送场景的大规模 VRPPDT 方面, 本文使用一个大规模京东数据集^[62]作为测试问题. 在此数据集上, 本文将 MBEA 与最近提出的 5 种算法 (MATE^[62], EMA^[35], CCMO^[68], GVNS^[69] 和 VNSME^[70]) 进行比较. 在这 5 种对比算法中, MATE 是由种群进化和局部搜索相结合的模式算法, GVNS 是单个体的元启发式算法, VNSME 是由 4 种局部搜索策略和扰动算子构成的元启发式算法, EMA 是进化多任务迁移学习算法, CCMO 是子任务协同优化算法. 接着, 本文进行消融实验, 验证基于路径的交叉 (Route-based crossover, RBX) 算子和顺序交叉 (Order crossover, OX) 算子的有效性. 最后, 本文讨论了 MBEA 中参数 $lower$ 和参数 k 的设置对性能的影响, 参数 k 代表子任务的数量. 参数 $lower$ 表示子任务的客户数量的范围, 其范围在 $(N \times lower, N)$ 之间. 其中 $k = 0$ 时表示没有在 MBEA 中使用子任务, 可以用于验证子任务对 MBEA 性能提升的有效性. 对于参数 k 和

lower, 本文还将进行参数敏感性分析.

1) 数据集. 本文使用文献 [62] 提出的大规模数据集来评估 MBEA 的性能. 该数据集来自京东物流配送系统. 在该系统中, 除需派送客户购买的商品外, 还需要收取客户商品 (例如, 有缺陷的商品或需要维修的商品), 这两者都必须在预定的时间窗内提供客户满意的服务. 因此, 本质上京东物流配送问题可以建模为本文研究的 VRPPDT. 原始数据是从系统中一段时间内产生的多个客户服务请求中收集得到的. 然后, 原始数据用于生成具有 200、400、600、800 和 1000 个客户的 5 种规模测试问题. 对于每种规模问题, 会从原始数据中生成 4 个实例, 最终为本实验提供包含 20 个实例的问题测试集. 在这个数据集中, 对 NV (车辆数量) 或 TD (行驶距离) 的优化没有优先级. 式 (4) 中的参数 u_1 和 u_2 是根据物流配送系统中的估计值给出的, 其目标是 minimized TC (总成本), 即车辆调度成本和行驶路程成本之和. 大规模京东数据集的具体设置如表 1 所示. 表 1 中, $|V|$ 为客户数量, C 为车辆容量, J 为可使用车辆数量.

表 1 京东数据集的特性
Table 1 Properties of Jingdong dataset

问题	$ V $	C	J	u_1	u_2
F201 ~ F204	200	2.5	500	300	0.014
F401 ~ F404	400	2.5	500	300	0.014
F601 ~ F604	600	2.5	500	300	0.014
F801 ~ F804	800	2.5	500	300	0.014
F1001 ~ F1004	1 000	2.5	500	300	0.014

2) 参数设置. 本文选用最近提出的 5 种算法作为对比算法, 包括 MATE、EMA、CCMO、GVNS 和 VNSME. 对比算法的参数根据原论文中给出的推荐参数进行设置. 所有算法在大规模京东数据集的 20 个实例上独立运行 20 次. MBEA 算法的详细参数如表 2 所示. MBEA 有 7 个参数需要设置, 分别是 $Evaluation$ 、 TE 、 N 、 N_{re} 、 N_{be} 、 k 及 $lower$. $Evaluation$ 是总的评价次数, MBEA 和比较算法的评价次数都设置为 18 000. MBEA 是一种多阶段进化算法, 其中种群大小 N 设置为 36, 每个阶段的评价次数 TE 设置为 3 600, 即每一个阶段种群进化 100 代. MBEA 的阶段数 N_{re} 设置为 5. 当 MBEA 完成一个阶段并进入下一个阶段时, 种群将保留前一阶段的 N_{be} 个最好的个体, 而种群中的其他个体重新初始化.

3.2 大规模车辆路径问题的实验结果和分析

表 3 给出了 MBEA 与 5 种比较算法 (MATE、

表 2 MBEA 算法参数设置
Table 2 Parameter settings in MBEA

参数	含义	值
$Evaluation$	算法总评价次数	18 000
TE	每阶段的评价次数	3 600
N	种群大小	36
N_{re}	阶段数	5
N_{be}	保留个体的数量	18
k	子任务个数	1
$lower$	子任务维度最低占比	0.7

EMA、CCMO、GVNS 和 VNSME) 在大规模京东数据集上的对比结果, 记录了车辆数量 NV 、行驶距离 TD 、总成本 TC ($TC = 300 \times NV + TD$) 和 CPU 时间. 在表 3 的 20 个测试问题中, MBEA 在 18 个测试问题中取得最小的 TC 值, 因此 MBEA 是当中优化性能最好的算法. 在客户规模为 200 的测试问题上, MBEA 和 MATE 取得的优化效果差不多. MBEA 在 F202 和 F204 问题上表现最好, MATE 在 F201 和 F203 问题上取得最好结果, 而 EMA、CCMO、GVNS 和 VNSME 在以上 4 个问题上都表现较差, 均未取得最好结果. 当测试问题的客户规模增加到 400、600、800 和 1000 时, MBEA 的优化效果明显好于其他 5 种算法. 例如, 在客户规模为 400 的测试问题上, MBEA 的平均 TC 值是 124 669, EMA 的平均 TC 值是 149 109, MATE 的平均 TC 值是 136 450, CCMO 的平均 TC 值是 155 794, GVNS 的平均 TC 值是 180 370, 而 VNSME 的平均 TC 值是 141 789. 因此, MBEA 取得了最好的平均 TC 值. 与 MATE, EMA, CCMO, GVNS 和 VNSME 相比, MBEA 的优化目标值 (TC) 分别提高了 8.63%, 16.39%, 19.98%, 30.88% 和 12.07%.

在其他客户规模数据集 (600、800、1 000) 上, MBEA 的平均 TC 值分别为 188 614, 220 804 和 327 730; MATE 的平均 TC 值分别为 214 861, 243 393 和 359 545; EMA 的平均 TC 值分别为 250 678, 296 887, 443 675; CCMO 的平均 TC 值分别为 241 545, 293 280 和 444 823; GVNS 的平均 TC 值分别为 266 184, 308 754 和 446 854; 而 VNSME 的平均 TC 值分别为 215 069, 242 376 和 346 777. 因此, 在这 3 个客户规模数据集上, 与 MATE 相比, MBEA 的优化目标值 (TC) 分别提高了 12.22%, 9.28% 和 8.85%; 与 EMA 相比, MBEA 的优化目标值 (TC) 分别提高了 24.75%, 25.62% 和 26.13%; 与 CCMO 相比, MBEA 的优化目标值 (TC) 分别提高了 21.91%, 24.71% 和 26.32%; 与 GVNS 相比, MBEA 的优化目标值 (TC) 分别提高了 29.14%,

表 3 MBEA 和 5 种对比算法在京东数据集对比实验结果
Table 3 Comparative experimental results of MBEA and five compared algorithms on Jingdong dataset

问题	MBEA				EMA				MATE				CCMO				GVNS				VNSME			
	NV	TD	TC	运行时间(s)	NV	TD	TC	运行时间(s)	NV	TD	TC	运行时间(s)	NV	TD	TC	运行时间(s)	NV	TD	TC	运行时间(s)	NV	TD	TC	运行时间(s)
F201	43	53 851	66 751	3 291	45	54 918	68 418	4 252	42	53 997	66 597	8 712	51	66 099	81 399	2 976	52	84 808	100 408	81	50	60 494	75 494	3
F202	44	53 155	66 355	3 270	47	56 288	70 388	4 340	43	53 649	66 609	14 194	52	63 782	79 382	2 839	53	67 756	83 656	252	49	59 728	74 428	2
F203	43	54 899	67 679	3 356	46	59 009	72 809	5 416	42	54 544	67 024	13 635	49	67 608	82 308	2 881	51	83 079	98 379	103	51	65 951	81 251	2
F204	43	53 311	66 211	2 983	46	56 456	70 256	3 986	43	54 398	67 238	9 929	48	62 329	76 729	2 970	52	74 571	90 171	300	51	60 415	75 715	2
F401	81	99 380	123 620	11 538	93	120 041	147 941	2 567	84	109 863	135 123	16 852	94	124 412	152 612	8 580	98	144 757	174 157	1 229	96	112 942	141 742	15
F402	84	103 091	128 351	13 338	101	122 636	152 936	2 535	87	113 871	139 971	10 742	100	130 655	160 655	7 923	101	160 822	191 122	268	98	117 970	147 370	12
F403	80	98 175	122 055	12 119	95	122 289	150 789	2 731	84	109 212	134 412	10 154	97	123 599	152 699	8 364	98	160 018	189 418	420	93	111 171	139 071	15
F404	83	99 809	124 649	12 656	95	116 269	144 769	2 694	86	110 555	136 295	13 709	100	127 209	157 209	8 154	101	136 483	166 783	651	94	110 775	138 975	17
F601	118	149 868	185 148	15 779	153	202 915	248 816	2 663	126	174 424	212 344	17 795	148	192 176	236 576	15 623	144	240 941	284 141	702	138	171 997	213 397	41
F602	121	153 129	189 429	19 571	164	204 772	253 972	2 656	129	177 851	216 551	18 839	146	199 278	243 078	15 505	141	227 723	270 023	1 624	143	175 068	217 968	49
F603	120	153 681	189 741	16 090	151	202 985	248 285	2 922	128	176 806	215 146	17 636	151	198 996	244 296	15 032	143	219 879	262 779	395	142	171 057	213 657	37
F604	122	153 477	190 137	18 569	157	204 541	251 641	2 886	128	176 943	215 403	18 789	154	196 028	242 228	15 201	145	204 293	247 793	757	141	172 956	215 256	33
F801	159	175 009	222 709	11 565	200	244 506	304 506	3 679	164	196 076	245 156	20 421	200	234 549	294 549	25 467	189	278 179	334 879	1 654	178	189 502	242 902	82
F802	157	173 598	220 577	13 077	210	226 736	289 736	3 657	164	194 325	243 465	20 835	199	236 794	296 494	25 879	184	271 798	326 998	1 153	179	192 243	245 943	107
F803	159	173 474	221 173	14 682	206	240 358	302 158	3 355	165	195 539	244 919	24 212	201	236 025	296 325	25 387	186	231 297	287 097	1 130	180	188 245	242 245	71
F804	156	171 956	218 756	12 743	213	227 247	291 147	3 324	161	191 853	240 033	21 884	198	226 353	285 753	25 707	181	231 743	286 043	1 490	174	186 214	238 414	81
F1001	212	265 385	329 044	9 698	275	363 035	445 535	3 874	222	293 298	359 838	25 957	279	364 136	447 836	34 957	239	391 293	462 993	1 236	232	278 192	347 792	154
F1002	211	264 034	327 213	8 655	279	356 200	439 900	3 858	225	291 180	358 740	27 482	284	354 899	440 099	34 582	240	352 092	424 092	2 847	234	278 465	348 665	126
F1003	212	265 409	329 008	8 910	275	358 768	441 268	3 917	227	295 806	363 786	26 217	283	359 276	444 176	33 748	243	408 770	481 670	554	231	274 553	343 853	126
F1004	212	262 117	325 656	10 331	285	362 496	447 996	3 914	223	289 035	355 815	26 180	289	360 481	447 181	33 515	234	348 460	418 660	890	233	276 896	346 796	123
最佳/全部			18/20				0/20				2/20				0/20				0/20					0/20

28.48% 和 26.65%; 与 VNSME 相比, MBEA 的优化目标值 (TC) 分别提高了 12.30%, 8.90% 和 5.49%. 由于 MBEA 将原始问题分解为 k 个辅助问题, 所以辅助问题的客户规模相对较小, 因此辅助问题比原问题更容易求解并能更快地获得较好的收敛解. 基于辅助问题与原问题的相似性, 求解辅助问题得到的收敛解有助于传递优化知识帮助求解原问题. 在表 3 中, MBEA 得到的结果充分验证了子任务优化知识迁移的有效性. 另一方面, 当问题客户规模达到 800 和 1000 时, MBEA 的平均运行时间分别是 13 017 s 和 9 399 s, 然而 MATE 的平均运行时间分别为 21 838 s 和 26 459 s, 所以 MBEA 的运行效率明显高于 MATE, 这也验证了 MBEA 在解决大规模问题上的优势.

MBEA 在 5 种客户规模的数据集上取得上述实验结果的原因如下. 当测试问题的客户数量较少 (即 200) 时, 对该类问题的求解相对容易, 因此与其他对比算法相比, MBEA 使用迁移优化的优势并不明显. 但是, 当测试问题的客户数量逐步增加到 400、

600、800 和 1000 时, 求解问题的难度急剧增加, 对比算法便很难找到最优解. 在这种情况下, MBEA 利用迁移优化具有显著优势. 因为 MBEA 可以通过从更简单和相似的子任务中迁移有用的路径信息到原任务, 所以能有效加快原任务的求解速度. 因此, MBEA 的优化性能明显好于其他 5 种对比算法.

3.3 收敛速度分析

图 8 给出了 MBEA 和其他 3 种比较算法 (EMA、CCMO、MATE) 的进化曲线. 由于 GVNS 和 VNSME 的终止条件是当前解经过连续迭代邻域搜索与多次扰动后依然没有得到改进, 所以按照 GVNS 和 VNSME 的原始设置, 当前最优解所使用的迭代次数并不是固定的. 因此, 图 8 没有给出 GVNS 和 VNSME 的收敛轨迹. 从图 8 可以看出, 与其他 3 种算法相比, MBEA 具有更快的收敛速度, 并且在大多数问题上都取得更好结果.

3.4 知识迁移中的交叉算子有效性分析

本文的多任务知识迁移操作使用两种不同的交

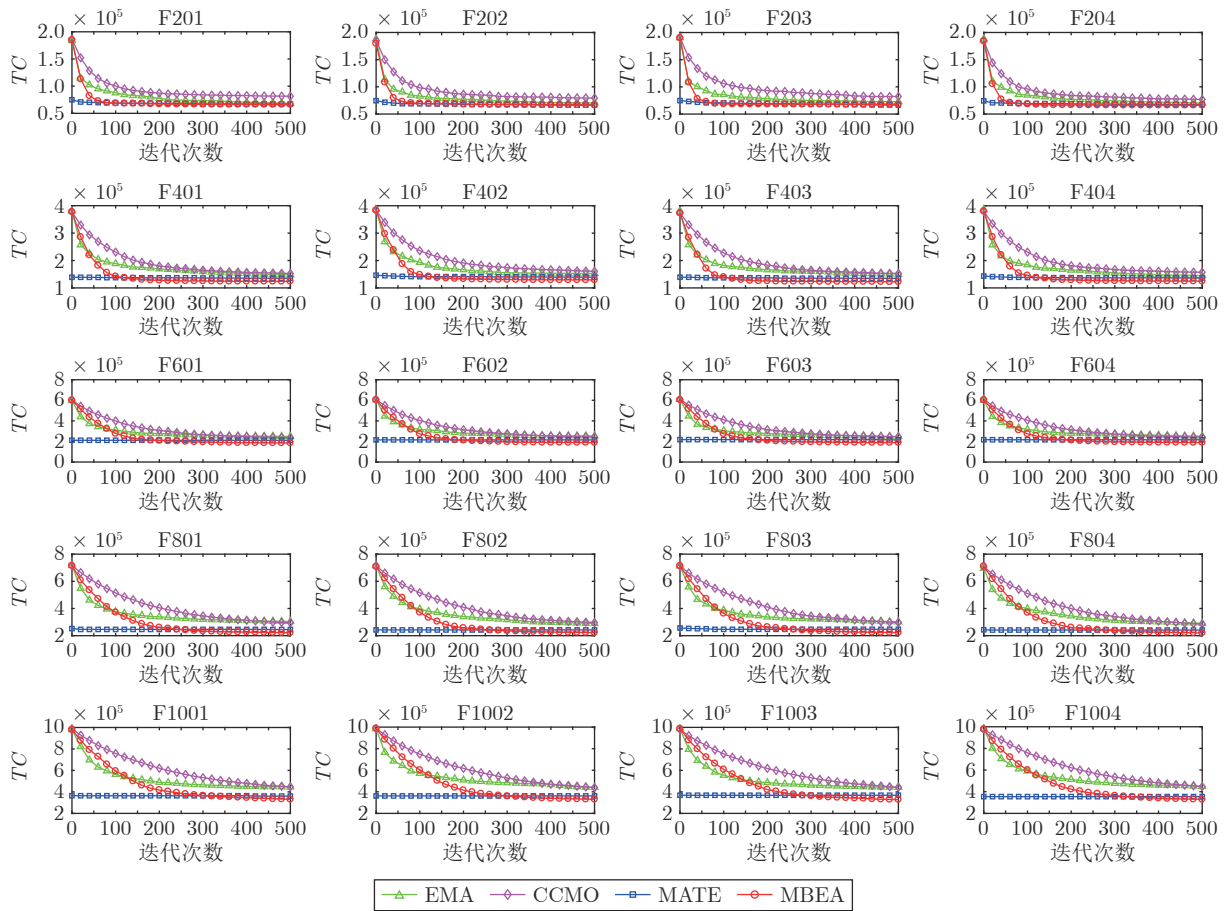


图 8 本文提出的方法和对比算法的平均搜索收敛轨迹

Fig. 8 Averaged search convergence traces of the proposed method and the compared algorithms

叉算子来生成子代种群. 当两个个体来自同一个任务时, MBEA 使用 RBX 算子优化该任务; 当两个个体来自不同的任务时, MBEA 使用 OX 算子交换两个个体的有用信息进行迁移优化. 表 4 给出了 MBEA 分别使用 OX 算子、RBX 算子和 RBX + OX 算子的实验结果. 从表 4 可以看出, OX 对大规模问题的优化效果较差. 特别是当问题客户规模超过 200 时, OX 算子得到的 TC 值明显大于 RBX 和 RBX + OX 算子取得的 TC 值. 另一方面, RBX + OX 算子的效果在一定程度上优于 RBX 算子. 当两个个体来自不同的任务时, RBX 算子不能对来自不同任务的两个个体进行操作, 而 OX 算子可以交换任意两个个体的信息. 因此, 与单独使用 RBX 算子相比, 使用 RBX + OX 算子在知识迁移中更有效.

表 4 RBX 和 OX 的消融实验结果

Table 4 Ablation experiment results of RBX and OX

问题	RBX	OX	RBX + OX
F201	66 517	67 966	66 751
F202	66 365	67 744	66 355
F203	68 948	71 718	67 679
F204	66 970	69 372	66 211
F401	124 851	148 685	123 620
F402	128 798	146 954	128 351
F403	123 550	149 781	122 055
F404	125 247	155 403	124 649
F601	187 048	246 299	185 148
F602	192 623	253 252	189 429
F603	193 915	247 466	189 741
F604	193 410	245 400	190 137
F801	224 758	298 889	222 709
F802	228 345	296 430	220 577
F803	226 138	302 783	221 173
F804	220 988	294 788	218 756
F1001	342 544	442 939	329 044
F1002	339 143	440 007	327 213
F1003	341 946	446 173	329 008
F1004	336 077	445 518	325 656
最佳/全部	1/20	0/20	19/20

3.5 参数敏感性实验和分析

本节对 MBEA 中的参数 $lower$ 和参数 k 进行参数敏感性分析实验, 其中 k 表示子任务的数量, $lower$ 表示子任务的客户数量的范围, 其范围在 $(N \times lower, N)$ 之间. 表 5 和表 6 分别展示了 MBEA

表 5 MBEA 中参数 $lower$ 的敏感性分析Table 5 Sensitivity analysis of $lower$ in MBEA

问题	$TC(0.7)$	$TC(0.5)$	$TC(0.6)$	$TC(0.8)$	$TC(0.9)$
F201	66 751	66 664	66 926	66 895	66 971
F202	66 355	66 587	66 597	66 677	66 751
F203	67 679	68 206	67 919	68 027	67 783
F204	66 211	66 215	65 920	66 209	66 150
F401	123 620	123 826	123 103	122 861	122 672
F402	128 351	127 154	127 696	127 771	127 986
F403	122 055	122 428	122 703	122 343	122 270
F404	124 649	124 771	125 365	125 138	124 936
F601	185 148	185 831	186 163	185 579	186 371
F602	189 429	190 317	190 661	190 363	190 731
F603	189 741	189 160	189 740	189 986	189 683
F604	190 137	189 300	189 128	189 743	188 712
F801	222 709	221 057	221 905	221 221	220 910
F802	220 577	221 957	220 655	219 998	220 951
F803	221 173	222 172	222 227	221 495	221 377
F804	218 756	218 002	216 628	217 680	218 297
F1001	329 044	330 379	330 059	330 929	329 632
F1002	327 213	327 346	327 565	326 923	327 199
F1003	329 008	329 596	326 035	328 369	327 482
F1004	325 656	325 790	325 812	326 352	327 106
最佳/全部	9/20	3/20	3/20	2/20	3/20

在大规模京东数据集上的参数分析结果. 表 5 分析了子任务规模变化 (即客户数量变化) 对 MBEA 性能的影响. 根据文献 [36], 在两个车辆路径问题中, 如果 50% 的客户相同, 则可以认为这两个任务是低度相似的; 如果 90% 的客户相同, 则可以认为这两个任务是高度相似的. 因此本文将参数 $lower$ 设置为 5 组, $lower$ 的值分别为 0.5、0.6、0.7、0.8 和 0.9. 然后, 将 MBEA 的子任务数设置为 1, 在不同的 $lower$ 设置下求解 20 个大规模京东数据集的优化性能. 从表 5 可以看出, 在 $lower = 0.7$ 的设置下, MBEA 在 20 个测试问题上的整体表现是最好的, 可以在 9 个 (总共 20 个) 测试问题上取得最好结果. 在其他参数设置下, MBEA 分别在 3 ($lower = 0.5$)、3 ($lower = 0.6$)、2 ($lower = 0.8$) 和 3 ($lower = 0.9$) 个测试问题上取得最好结果. 因此, 当参数 $lower$ 设置为 0.7 时, MBEA 可以达到整体最佳性能. 具体原因分析如下: 当 $lower$ 的值设置远小于 0.7 时, 子任务与原 VRPPDT 之间的相似性相对较低, 导致有用的路径特征很少能被 MBEA 从子任务迁移到原 VRPPDT. 相反, 当 $lower$ 的值设置远大于 0.7 时, 子任务与原 VRPPDT 的相似度过高, 导致

表 6 MBEA 中参数 k 的敏感性分析
Table 6 Sensitivity analysis of parameter k in MBEA

问题	$TC(1)$	$TC(0)$	$TC(2)$	$TC(3)$
F201	66 751	67 985	66 298	66 731
F202	66 355	67 920	67 002	67 111
F203	67 679	68 180	67 917	67 680
F204	66 211	66 621	66 524	65 730
F401	123 620	124 871	124 825	123 839
F402	128 351	127 377	128 292	127 603
F403	122 055	123 494	122 305	121 883
F404	124 649	126 338	124 959	125 486
F601	185 148	187 055	185 004	187 381
F602	189 429	192 343	192 178	189 516
F603	189 741	190 448	190 047	190 610
F604	190 137	191 208	189 940	189 096
F801	222 709	223 158	223 670	224 248
F802	220 577	222 758	223 576	225 523
F803	221 173	222 456	223 684	223 939
F804	218 756	218 393	217 485	220 373
F1001	329 044	330 374	331 387	332 643
F1002	327 213	329 829	329 525	326 603
F1003	329 008	331 877	330 099	329 200
F1004	325 656	330 725	326 082	331 858
最佳/全部	12/20	1/20	3/20	4/20

生成许多高维的子任务. 在这种情况下, 子任务也很难在有限的时间内求得较优解. 因此, 当 $lower$ 的值设置得远小于或大于 0.7 时, MBEA 都无法将有用的路径特征从子任务迁移到原 VRPPDT. 综上, 当 $lower = 0.7$ 时, MBEA 可以在子任务的求解难度和子任务与原任务的相似度之间取得较好的均衡.

另一方面, 表 6 分析了子任务数量 (即参数 k) 对 MBEA 性能的影响. 在表 6 中, 参数 $lower$ 设置为 0.7, 子任务的数量 k 分别设置为 0、1、2、3, 其中 $k = 0$ 时直接使用进化算法进行优化, 没有生成子任务进行知识迁移过程. 当参数 k 设置为 1 (即只有一个子任务) 时, MBEA 在 12 个测试问题上取得最好的结果. 同时, 在参数 $k = 0$ 的情况下, MBEA 仅在 1 个测试问题上取得最好的结果. 这表明设置子任务来解决大规模 VRPPDT 是有效的. 但子任务的数量不宜设置太多. 因为每个子任务都需要种群中的一些个体进行优化, 所以子任务过多会导致种群中优化原始问题的个体较少, 最终无法在有限代数内达到收敛状态.

4 结束语

本文提出面向复杂物流配送场景的车辆路径规划多任务辅助进化算法 MBEA, 通过对多个简单且相关的子任务使用知识迁移操作来辅助优化原大规模 VRPPDT, 加快算法收敛速度. 与现有算法相比, MBEA 的创新性表现为两个方面: 子任务生成和基于多任务的知识迁移. 子任务生成操作将原始 VRPPDT 划分为多个简单且相关的子任务. 然后 MBEA 在进化多任务迁移中使用两种不同的交叉算子产生下一代种群. 当两个父代个体来自同一个任务时, 使用基于路径的交叉算子 (优化) 产生子代个体; 否则, 使用顺序交叉算子 (迁移) 产生子代个体. 在本文的实验分析中, 与最近提出的 5 种算法相比, MBEA 在京东大规模 VRPPDT 数据集上的路径规划结果更优.

在未来的工作中, 我们将在 MBEA 算法的基础上进一步探索解决 VRPPDT 的有效策略. 首先, 我们将进一步结合问题特征设计出更高效的子任务生成方法. 其次, 我们将在 MBEA 中探索使用更多高效的局部搜索方法. 最后, 我们将逐步把 MBEA 构建成一个通用的路径规划算法框架, 从而支持求解更多的 VRP 变体, 例如动态 VRP、多站点 VRP 和电动车队 VRP.

References

- Hu M, Liu W D, Peng K, Ma X Q, Cheng W Q, Liu J C, et al. Joint routing and scheduling for vehicle-assisted multidrone surveillance. *IEEE Internet of Things Journal*, 2019, 6(2): 1781-1790
- Yang Pei-Ying, Tang Jia-Fu, Yu Yang, Pei Jin-Xiang. Minimizing carbon emissions for vehicle routing and scheduling in picking up and delivering customers to airport service. *Acta Automatica Sinica*, 2013, 39(4): 424-432
(杨培颖, 唐加福, 于洋, 裴金翔. 面向最小碳排放量的接送机场服务的车辆路径与调度. *自动化学报*, 2013, 39(4): 424-432)
- Han Yue-Qi, Zhang Kai, Bin Yang, Qin Chuang, Xu Yun-Xiao, Li Xiao-Chuan, et al. Convex approximation based avoidance theory and path planning MPC for driver-less vehicles. *Acta Automatica Sinica*, 2020, 46(1): 153-167
(韩月起, 张凯, 宾洋, 秦闯, 徐云霄, 李小川, 等. 基于凸近似的避障原理及无人驾驶车辆路径规划模型预测算法. *自动化学报*, 2020, 46(1): 153-167)
- Xu Yang, Lu Li-Ping, Chu Duan-Feng, Huang Zi-Chao. Unified modeling of trajectory planning and tracking for unmanned vehicle. *Acta Automatica Sinica*, 2019, 45(4): 799-807
(徐杨, 陆丽萍, 褚端峰, 黄子超. 无人车辆轨迹规划与跟踪控制的统一建模方法. *自动化学报*, 2019, 45(4): 799-807)
- Wu Wei, Liu Yang, Liu Wei, Wu Guo-Hong, Ma Wan-Jing. A novel autonomous vehicle trajectory planning and control model for connected-and-autonomous intersections. *Acta Automatica Sinica*, 2020, 46(9): 1971-1985
(吴伟, 刘洋, 刘威, 吴国弘, 马万经. 自动驾驶环境下交叉口车辆路径规划与最优控制模型. *自动化学报*, 2020, 46(9): 1971-1985)
- Abbatecola L, Fanti M P, Pedroncelli G, Ukovich W. A distrib-

- uted cluster-based approach for pick-up services. *IEEE Transactions on Automation Science and Engineering*, 2019, **16**(2): 960–971
- 7 Wang Su-Xin, Gao Li, Cui Xiao-Guang, Cao Hong-Mei. Study on multi-requirement points vehicle scheduling model and its swarm mix algorithm. *Acta Automatica Sinica*, 2008, **34**(1): 102–104
(王素欣, 高利, 崔小光, 曹宏美. 多需求点车辆调度模型及其群体智能混合求解. *自动化学报*, 2008, **34**(1): 102–104)
- 8 Zeng Zheng-Yang, Xu Wei-Sheng, Xu Zhi-Yu. Modeling and multi-start variable neighborhood descent solution of two-echelon open vehicle routing problem. *Computer Science*, 2014, **41**(10): 232–237
(曾正洋, 许维胜, 徐志宇. 开放式两级车辆路径问题建模与多起始点变邻域下降法求解. *计算机科学*, 2014, **41**(10): 232–237)
- 9 Kheirkhah A, Navidi H, Bidgoli M M. An improved benders decomposition algorithm for an arc interdiction vehicle routing problem. *IEEE Transactions on Engineering Management*, 2016, **63**(2): 259–273
- 10 Hu Rong, Li Yang, Qian Bin, Jin Huai-Ping, Xiang Feng-Hong. An enhanced ant colony optimization combined with clustering decomposition for solving complex green vehicle routing problem. *Acta Automatica Sinica*, 2022, **48**(12): 3006–3023
(胡蓉, 李洋, 钱斌, 金怀平, 向凤红. 结合聚类分解的增强蚁群算法求解复杂绿色车辆路径问题. *自动化学报*, 2022, **48**(12): 3006–3023)
- 11 Xiao J H, Zhang T, Du J G, Zhang X Y. An evolutionary multiobjective route grouping-based heuristic algorithm for large-scale capacitated vehicle routing problems. *IEEE Transactions on Cybernetics*, 2021, **51**(8): 4173–4186
- 12 Lin C H, Choy K L, Ho G T S, Chung S H, Lam H Y. Survey of green vehicle routing problem: Past and future trends. *Expert Systems With Applications*, 2014, **41**(4): 1118–1138
- 13 Kassem S, Chen M Y. Solving reverse logistics vehicle routing problems with time windows. *The International Journal of Advanced Manufacturing Technology*, 2013, **68**(1–4): 57–68
- 14 Min H. The multiple vehicle routing problem with simultaneous delivery and pick-up points. *Transportation Research Part A: General*, 1989, **23**(5): 377–386
- 15 Dethloff J. Vehicle routing and reverse logistics: The vehicle routing problem with simultaneous delivery and pick-up. *OR-Spektrum*, 2001, **23**(1): 79–96
- 16 Berbeglia G, Cordeau J F, Laporte G. Dynamic pickup and delivery problems. *European Journal of Operational Research*, 2010, **202**(1): 8–15
- 17 Wang H F, Chen Y Y. A genetic algorithm for the simultaneous delivery and pickup problems with time window. *Computers and Industrial Engineering*, 2012, **62**(1): 84–95
- 18 Eksioglu B, Vural A V, Reisman A. The vehicle routing problem: A taxonomic review. *Computers and Industrial Engineering*, 2009, **57**(4): 1472–1483
- 19 Fan Hou-Ming, Liu Peng-Cheng, Liu Hao, Hou Deng-Kai. The multi-depot vehicle routing problem with simultaneous deterministic delivery and stochastic pickup based on joint distribution. *Acta Automatica Sinica*, 2021, **47**(7): 1646–1660
(范厚明, 刘鹏程, 刘浩, 侯登凯. 多中心联合配送模式下集货需求随机的 VRPSDP 问题. *自动化学报*, 2021, **47**(7): 1646–1660)
- 20 Chen Ping, Huang Hou-Kuan, Dong Xing-Ye. A hybrid heuristic algorithm for the vehicle routing problem with simultaneous delivery and pickup. *Chinese Journal of Computers*, 2008, **31**(4): 565–573
(陈萍, 黄厚宽, 董兴业. 求解卸装一体化的车辆路径问题的混合启发式算法. *计算机学报*, 2008, **31**(4): 565–573)
- 21 Gupta A, Heng C K, Ong Y S, Tan P S, Zhang A N. A generic framework for multi-criteria decision support in eco-friendly urban logistics systems. *Expert Systems With Applications*, 2017, **71**: 288–300
- 22 Wang J H, Weng T Y, Zhang Q F. A two-stage multiobjective evolutionary algorithm for multiobjective multidepot vehicle routing problem with time windows. *IEEE Transactions on Cybernetics*, 2019, **49**(7): 2467–2478
- 23 Wang C, Mu D, Zhao F, Sutherland J W. A parallel simulated annealing method for the vehicle routing problem with simultaneous pickup-delivery and time windows. *Computers and Industrial Engineering*, 2015, **83**: 111–122
- 24 Mingyong L, Erbao C. An improved differential evolution algorithm for vehicle routing problem with simultaneous pickups and deliveries and time windows. *Engineering Applications of Artificial Intelligence*, 2010, **23**(2): 188–195
- 25 Huang Wu-Lan, Zhang Tao. Vehicle routing problem with simultaneous pick-up and delivery and time-windows based on improved global artificial fish swarm algorithm. *Computer Engineering and Applications*, 2016, **52**(21): 21–29
(黄务兰, 张涛. 基于改进全局人工鱼群算法的 VRPSDPTW 研究. *计算机工程与应用*, 2016, **52**(21): 21–29)
- 26 Shi Y, Zhou Y J, Boudouh T, Grunder O. A lexicographic-based two-stage algorithm for vehicle routing problem with simultaneous pickup-delivery and time window. *Engineering Applications of Artificial Intelligence*, 2020, **95**: Article No. 103901
- 27 Hof J, Schneider M. An adaptive large neighborhood search with path relinking for a class of vehicle-routing problems with simultaneous pickup and delivery. *Networks*, 2019, **74**(3): 207–250
- 28 Iqbal M, Xue B, Al-Sahaf H, Zhang M J. Cross-domain reuse of extracted knowledge in genetic programming for image classification. *IEEE Transactions on Evolutionary Computation*, 2017, **21**(4): 569–587
- 29 Di S, Zhang H G, Li C G, Mei X, Prokhorov D, Ling H B. Cross-domain traffic scene understanding: A dense correspondence-based transfer learning approach. *IEEE Transactions on Intelligent Transportation Systems*, 2018, **19**(3): 745–757
- 30 Chalmers E, Contreras E B, Robertson B, Luczak A, Gruber A. Learning to predict consequences as a method of knowledge transfer in reinforcement learning. *IEEE Transactions on Neural Networks and Learning Systems*, 2018, **29**(6): 2259–2270
- 31 Gupta A, Ong Y S, Feng L. Multifactorial evolution: Toward evolutionary multitasking. *IEEE Transactions on Evolutionary Computation*, 2016, **20**(3): 343–357
- 32 Gupta A, Ong Y S, Feng L, Tan K C. Multiobjective multifactorial optimization in evolutionary multitasking. *IEEE Transactions on Cybernetics*, 2017, **47**(7): 1652–1665
- 33 Bali K K, Ong Y S, Gupta A, Tan P S. Multifactorial evolutionary algorithm with online transfer parameter estimation: MFEA-II. *IEEE Transactions on Evolutionary Computation*, 2020, **24**(1): 69–83
- 34 Hao X X, Qu R, Liu J. A unified framework of graph-based evolutionary multitasking hyper-heuristic. *IEEE Transactions on Evolutionary Computation*, 2021, **25**(1): 35–47
- 35 Feng L, Zhou L, Gupta A, Zhong J H, Zhu Z X, Tan K C, et al. Solving generalized vehicle routing problem with occasional drivers via evolutionary multitasking. *IEEE Transactions on Cybernetics*, 2021, **51**(6): 3171–3184
- 36 Feng L, Huang Y X, Zhou L, Zhong J H, Gupta A, Tang K, et al. Explicit evolutionary multitasking for combinatorial optimization: A case study on capacitated vehicle routing problem. *IEEE Transactions on Cybernetics*, 2021, **51**(6): 3143–3156

- 37 Feng L, Zhou L, Zhong J H, Gupta A, Ong Y S, Tan K C, et al. Evolutionary multitasking via explicit autoencoding. *IEEE Transactions on Cybernetics*, 2019, **49**(9): 3457–3470
- 38 Feng L, Huang Y X, Tsang I W, Gupta A, Tang K, Tan K C, et al. Towards faster vehicle routing by transferring knowledge from customer representation. *IEEE Transactions on Intelligent Transportation Systems*, 2022, **23**(2): 952–965
- 39 Shang Q X, Huang Y X, Wang Y, Li M, Feng L. Solving vehicle routing problem by memetic search with evolutionary multitasking. *Memetic Computing*, 2022, **14**(1): 31–44
- 40 Braekers K, Ramaekers K, Van Nieuwenhuysse I. The vehicle routing problem: State of the art classification and review. *Computers and Industrial Engineering*, 2016, **99**: 300–313
- 41 Huang M F, Hu X P. Large scale vehicle routing problem: An overview of algorithms and an intelligent procedure. *International Journal of Innovative Computing, Information and Control*, 2012, **8**(8): 5809–5819
- 42 Shin K, Han S Y. A centroid-based heuristic algorithm for the capacitated vehicle routing problem. *Computing and Informatics*, 2011, **30**(4): 721–732
- 43 Angelelli E, Mansini R. The vehicle routing problem with time windows and simultaneous pick-up and delivery. *Quantitative Approaches to Distribution Logistics and Supply Chain Management*. Berlin: Springer, 2002. 249–267
- 44 Dell'Amico M, Righini G, Salani M. A branch-and-price approach to the vehicle routing problem with simultaneous distribution and collection. *Transportation Science*, 2006, **40**(2): 235–247
- 45 Rieck J, Zimmermann J. Exact solutions to the symmetric and asymmetric vehicle routing problem with simultaneous delivery and pickup. *Business Research*, 2013, **6**(1): 77–92
- 46 Subramanian A, Uchoa E, Pessoa A A, Ochi L S. Branch-cut-and-price for the vehicle routing problem with simultaneous pickup and delivery. *Optimization Letters*, 2013, **7**(7): 1569–1581
- 47 Cao W J, Yang W S. A survey of vehicle routing problem. *MATEC Web of Conferences*, 2017, **100**: Article No. 01006
- 48 Elshaer R, Awad H. A taxonomic review of metaheuristic algorithms for solving the vehicle routing problem and its variants. *Computers and Industrial Engineering*, 2020, **140**: Article No. 106242
- 49 Yu V F, Redi A A N P, Hidayat Y A, Wibowo O J. A simulated annealing heuristic for the hybrid vehicle routing problem. *Applied Soft Computing*, 2017, **53**: 119–132
- 50 Schermer D, Moeini M, Wendt O. A hybrid VNS/Tabu search algorithm for solving the vehicle routing problem with drones and en route operations. *Computers and Operations Research*, 2019, **109**: 134–158
- 51 Ribeiro G M, Laporte G. An adaptive large neighborhood search heuristic for the cumulative capacitated vehicle routing problem. *Computers and Operations Research*, 2012, **39**(3): 728–735
- 52 Avci M, Topaloglu S. A hybrid metaheuristic algorithm for heterogeneous vehicle routing problem with simultaneous pickup and delivery. *Expert Systems With Applications*, 2016, **53**: 160–171
- 53 Zhou Y, Kong L J, Cai Y Q, Wu Z Y, Liu S P, Hong J M, et al. A decomposition-based local search for large-scale many-objective vehicle routing problems with simultaneous delivery and pickup and time windows. *IEEE Systems Journal*, 2020, **14**(4): 5253–5264
- 54 Baker B M, Ayechew M A. A genetic algorithm for the vehicle routing problem. *Computers and Operations Research*, 2003, **30**(5): 787–800
- 55 Mohammed M A, Abd Ghani M K, Hamed R I, Mostafa S A, Ahmad M S, Ibrahim D A. Solving vehicle routing problem by using improved genetic algorithm for optimal solution. *Journal of Computational Science*, 2017, **21**: 255–262
- 56 Sabar N R, Bhaskar A, Chung E, Turkey A, Song A. An adaptive memetic approach for heterogeneous vehicle routing problems with two-dimensional loading constraints. *Swarm and Evolutionary Computation*, 2020, **58**: Article No. 100730
- 57 Ouaddi K, Mhada F, Benadada Y. Memetic algorithm for multi-tours dynamic vehicle routing problem with overtime (MD-VRPOT). *International Journal of Industrial Engineering Computations*, 2020, **11**(4): 643–662
- 58 Zhang X Y, Duan H B. An improved constrained differential evolution algorithm for unmanned aerial vehicle global route planning. *Applied Soft Computing*, 2015, **26**: 270–284
- 59 Venkatesan S R, Logendran D, Chandramohan D. Optimization of capacitated vehicle routing problem using PSO. *International Journal of Engineering Science and Technology*, 2011, **3**(10): 7469–7477
- 60 Yu B, Yang Z Z, Yao B Z. An improved ant colony optimization for vehicle routing problem. *European Journal of Operational Research*, 2009, **196**(1): 171–176
- 61 Lee C Y, Lee Z J, Lin S W, Ying K C. An enhanced ant colony optimization (EACO) applied to capacitated vehicle routing problem. *Applied Intelligence*, 2010, **32**(1): 88–95
- 62 Liu S C, Tang K, Yao X. Memetic search for vehicle routing with simultaneous pickup-delivery and time windows. *Swarm and Evolutionary Computation*, 2021, **66**: Article No. 100927
- 63 Ma Y F, Li Z M, Yan F, Feng C Y. A hybrid priority-based genetic algorithm for simultaneous pickup and delivery problems in reverse logistics with time windows and multiple decision-makers. *Soft Computing*, 2019, **23**(15): 6697–6714
- 64 Wang J H, Zhou Y, Wang Y, Zhang J, Chen C L P, Zheng Z B. Multiobjective vehicle routing problems with simultaneous delivery and pickup and time windows: Formulation, instances, and algorithms. *IEEE Transactions on Cybernetics*, 2016, **46**(3): 582–594
- 65 Lagos C, Guerrero G, Cabrera E, Moltedo A, Johnson F, Paredes F. An improved particle swarm optimization algorithm for the VRP with simultaneous pickup and delivery and time windows. *IEEE Latin America Transactions*, 2018, **16**(6): 1732–1740
- 66 Potvin J Y, Bengio S. The vehicle routing problem with time windows Part II: Genetic search. *INFORMS Journal on Computing*, 1996, **8**(2): 165–172
- 67 Oliver I M, Smith D J, Holland J R C. A study of permutation crossover operators on the traveling salesman problem. In: Proceedings of the Second International Conference on Genetic Algorithms on Genetic Algorithms and Their Application. Cambridge, USA: L. Erlbaum Associates Inc., 1987.
- 68 Tian Y, Zhang T, Xiao J H, Zhang X Y, Jin Y C. A coevolutionary framework for constrained multiobjective optimization problems. *IEEE Transactions on Evolutionary Computation*, 2021, **25**(1): 102–116
- 69 Fleszar K, Osman I H, Hindi K S. A variable neighbourhood search algorithm for the open vehicle routing problem. *European Journal of Operational Research*, 2009, **195**(3): 803–809
- 70 Cai J C, Zhu Q L, Lin Q Z. Variable neighborhood search for a new practical dynamic pickup and delivery problem. *Swarm and Evolutionary Computation*, 2022, **75**: Article No. 101182



李坚强 深圳大学计算机与软件学院教授. 2008 年获华南理工大学博士学位. 主要研究方向为嵌入式系统和物联网. E-mail: lijq@szu.edu.cn

(LI Jian-Qiang Professor at the College of Computer Science and Software Engineering, Shenzhen University. He received his Ph.D. degree from South China University of Technology in 2008. His research interest covers embedded systems and internet of things.)



蔡俊创 深圳大学计算机与软件学院博士研究生. 主要研究方向为进化计算及其在物流领域中的应用.

E-mail: caijunchuang2020@email.szu.edu.cn

(CAI Jun-Chuang Ph.D. candidate at the College of Computer Science and Software Engineering, Shenzhen University. His research interest covers evolutionary computation and its applications in the field of logistics.)



孙涛 中兴通讯股份有限公司工程师. 2022 年获深圳大学硕士学位. 主要研究方向为进化计算和路径规划.

E-mail: 1910272020@email.szu.edu.cn

(SUN Tao Engineer at ZTE Corporation. He received his master degree from Shenzhen University in

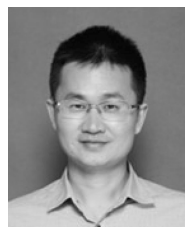
2022. His research interest covers evolutionary computation and path planning.)



朱庆灵 深圳大学计算机与软件学院博士后. 2021 年获香港城市大学博士学位. 主要研究方向为进化多目标优化和机器学习.

E-mail: zhuqingling@szu.edu.cn

(ZHU Qing-Ling Postdoctor at the College of Computer Science and Software Engineering, Shenzhen University. He received his Ph.D. degree from the City University of Hong Kong in 2021. His research interest covers evolutionary multiobjective optimization and machine learning.)



林秋镇 深圳大学计算机与软件学院副教授. 2014 年获香港城市大学博士学位. 主要研究方向为人工免疫系统, 多目标优化和动态系统. 本文通信作者. E-mail: qiuzhlin@szu.edu.cn

(LIN Qiu-Zhen Associate professor at the College of Computer Science and Software Engineering, Shenzhen University. He received his Ph.D. degree from the City University of Hong Kong in 2014. His research interest covers artificial immune system, multiobjective optimization, and dynamic system. Corresponding author of this paper.)