

一种求解带时间窗车辆路径问题的混合差分进化算法

宋晓宇 朱加园 孙焕良

(沈阳建筑大学信息与控制工程学院 沈阳 110168)

摘要 对带时间窗的车辆路径问题进行研究,建立以最小化车辆数量和行驶路程为目标的多目标数学模型,提出一种结合改进差分进化算法和变邻域下降搜索的基于 Pareto 支配的混合差分进化算法。首先重新定义了个体的生成方式。其次,结合双种群策略和变邻域下降搜索技术来平衡算法的全局探索能力和局部开发能力,并在搜索过程中用随机个体替代种群中的重复个体,维持种群的多样性。然后引入 Pareto 支配的概念来评价个体的优劣性,并采用擂台法则构造非支配解集。最后对 18 个不同规模的 Solomon 算例的求解结果表明,算法在行驶路程和车辆数量上的求解质量比人工蜂群算法分别平均提高了 2.04% 和 14.95%,且与已知最优解相比,在车辆数量的求解质量上平均提高了 14.53%,验证了所提算法的有效性。

关键词 带时间窗车辆路径问题,多目标,差分进化算法,双种群,变邻域下降搜索

中图分类号 TP301 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2014.12.048

Hybrid Differential Evolution Algorithm for Vehicle Routing Problem with Time Windows

SONG Xiao-yu ZHU Jia-yuan SUN Huan-liang

(School of Information and Control Engineering, Shenyang Jianzhu University, Shenyang 110168, China)

Abstract Aiming at the vehicle routing problem with time windows, a multi-objective mathematical model was built with goals of minimizing the vehicle number and the travel distance, and it was solved by a hybrid differential evolution algorithm which is based on Pareto-dominance, combined with improved differential evolution algorithm and variable neighborhood descent. First, redefined method to produce new individual was employed. Then, the global exploration and the local exploitation were balanced by combining the dual populations strategy and variable neighborhood descent technology, and the diversity of the population was maintained through the method which uses the random individuals to replace the repetition individuals. Third, Pareto-dominance was applied to compare different solutions, and Arena's principle was adopted to construct non-dominated solution set. Finally computational results on the 18 Solomon problems with the different sizes show that the solution quality of the proposed algorithm averagely increases 2.04% and 14.95% than artificial bee colony algorithm in the travel distance and the vehicle number, and averagely increases 14.53% than the best known solutions in the vehicle number, verifying the effectiveness of the proposed algorithm.

Keywords Vehicle routing problem with time windows, Multi-objective, Differential evolution algorithm, Dual populations, Variable neighborhood descent

1 引言

车辆路径问题(VRP)^[1]一直是计算机科学、运筹学与组合优化领域的热点前沿问题,现实生活中很多问题都可以抽象为车辆路径问题,如物流配送问题、电力调度问题、邮政投递问题、校车及公交车路径安排问题等。近年来随着 JIT、敏捷制造等生产理念的影响的不断扩大,以及顾客对物流配送的时间准时性的要求也越来越高,时间因素在物流配送尤其是车辆路径问题中显得越发重要,因此对带时间窗的车辆路径问题(VRPTW)的研究具有很重要的实用价值。VRPTW 是在 VRP 的基础上增加了顾客要求服务的时间窗口,即给定每个顾客一个时间范围,车辆对顾客的服务时间必须在这个

范围内开始。VRPTW 是 VRP 的拓展,也是 NP 难问题,它的求解方法大体可分为精确算法和启发式算法两大类。随着问题规模增大,精确算法的计算量会呈指数增长,从而使该类算法只能求解小规模 VRP。现在广大学者主要把精力放在启发式算法上,如遗传算法^[2]、蚁群算法^[3]、粒子群算法^[4]等都被用来求解 VRPTW。

差分进化(Differential Evolution, DE)算法^[5]是一种基于群体的启发式算法,最早用以求解连续域函数优化问题而被提出。DE 采用实数编码,基于差分的变异操作和一对一竞争选择策略。DE 属于基于群体的算法,其优势是善于发现搜索空间中可能存在的最优解的区域。由于 DE 具有容易实现、结构简单以及鲁棒性强等特点,近年来已经有很多学者用

到稿日期:2013-06-25 返修日期:2013-08-16 本文受国家自然科学基金(61070024),国家科技支撑计划(2006BAJ06B08-03)资助。

宋晓宇(1963-),男,博士,教授,CCF 会员,主要研究方向为计算智能、模式识别和数据挖掘等, E-mail: sxy@sju.edu.cn; 朱加园(1990-),男,硕士生,主要研究方向为物流工程和智能优化算法; 孙焕良(1969-),男,博士,教授,CCF 高级会员,主要研究方向为数据库和数据挖掘等。

其来解决离散域问题,例如旅行商问题^[6]、车间调度问题^[7]等。但是在求解带时间窗的车辆路径问题方面研究较少,曹二保等人^[8]采用整数序规范的辅助算子实现了个体连续位置到顾客排序离散值的转化,使 DE 适用于求解 VRPTW 问题,在该文献中采用一种顺序优化的方法,对两个目标设定一个求解优先级;王君^[9]设计了一种具有离散操作的 DE,并将其与局部禁忌搜索策略相结合来求解 VRPTW 问题,在该文献中采用线性加权法将 VRPTW 的两个目标转化成一个目标进行求解。但是这两种方法一次求解都只能得到一个最优解,很难较好地反映 VRPTW 的多目标性。文献^[10]将 DE 用于多目标优化,但是由于 DE 本身存在解质量较低的缺点,因此直接将其应用于多目标优化不能得到质量较好的全局最优解,所以如何使 DE 在求解多目标问题时具有良好的全局探索能力和局部开发能力,以及在 Pareto 解集分布的均匀性和多样性方面,还有待提高。变邻域下降搜索 (Variable Neighborhood Descent, VND) 是变邻域搜索^[11]的一种变形,主要是通过一种确定的方式来改变邻域结构的使用。VND 属于轨迹法,其优势是善于在一个解的附近进行深度的开发。陈萍等人^[12]提出一种结合蚁群系统和变邻域下降搜索的混合启发式算法来求解卸装一体化的车辆路径问题,并验证了该方法的可行性与高效性。

VRPTW 需要同时考虑使用的车辆数量和行驶路程两个目标,因此它也是一个多目标优化问题。本文根据 DE 和 VND 各自的优势,将一种改进的差分进化算法 (IDE) 与 VND 相结合,提出一种基于 Pareto 支配的混合差分进化算法 IDE_VND 用于求解多目标 VRPTW 问题。对不同规模的 Solomon 算例进行实验,结果表明本文提出的混合差分进化算法是一种求解 VRPTW 问题的有效方法。

2 问题描述及数学模型建立

VRPTW 问题可定义在图 $G=(V,E)$ 上,其中 $V=\{0,1,2,\dots,n\}$ 表示节点集,0 节点表示配送中心,且该配送中心拥有相同载重量的车辆 K 辆; $1,\dots,n$ 节点表示待服务的 n 个顾客点。车辆从配送中心出发,完成对所有顾客的服务之后再返回配送中心,每个顾客点都有一个固定的需求且只能被一辆车服务一次。车辆必须在给定的时间窗 $[e_i, l_i]$ 对顾客进行服务,如果车辆在时间 e_i 之前就到达顾客点 i ,则必须等待直到 e_i 才能进行服务。每条路线中的所有顾客需求总和不能超过车辆的载重量。在满足顾客需求和时间窗的前提下,需要规划车辆行驶路线,使得使用车辆数量最少且行驶路程最短。

符号、参数变量及决策变量说明:

d_{ij} 表示从顾客 i 到顾客 j 的运输成本(时间、路程或者费用等); Q 表示车辆的最大载重量; q_i 表示顾客点 i 的需求量; t_i 表示车辆到达顾客点 i 的时间; t_{ij} 表示从顾客 i 到顾客 j 的行驶时间; st_i 表示车辆在顾客 i 处的服务时间; I 表示车辆集合, $I=\{k\}, k=1,2,\dots,K$ 。

$$x_{ijk} = \begin{cases} 1, & \text{车辆 } k \text{ 服务完顾客 } i \text{ 后立即服务顾客 } j \\ 0, & \text{否则} \end{cases}$$

$$y_{ik} = \begin{cases} 1, & \text{车辆 } k \text{ 服务顾客 } i \\ 0, & \text{否则} \end{cases}$$

则 VRPTW 的数学模型可表示为:

目标函数:

$$\min f_1 = \sum_{k=1}^K \sum_{i=0}^n \sum_{j=0}^n d_{ij} x_{ijk} \quad (1)$$

$$\min f_2 = \sum_{j=0}^n \sum_{k=1}^K x_{0jk} \quad (2)$$

约束条件:

$$\sum_{i=1}^n q_i y_{ik} \leq Q; \forall k \in I \quad (3)$$

$$\sum_{i=0}^n x_{ijk} = y_{jk}; j=1,2,\dots,n; \forall k \in I \quad (4)$$

$$\sum_{j=0}^n x_{ijk} = y_{ik}; i=1,2,\dots,n; \forall k \in I \quad (5)$$

$$\sum_{k=1}^K y_{ik} = \begin{cases} 1, & i=1,2,\dots,n; \forall k \in I \\ K, & i=0 \end{cases} \quad (6)$$

$$\sum_{i \in S} \sum_{j \in S} x_{ijk} \leq |S| - 1; \forall \substack{S \\ 2 \leq |S| \leq n-1} \subseteq V \setminus \{0\} \quad (7)$$

$$t_i + t_{ij} + st_i - M(1 - x_{ijk}) \leq t_j; \quad (8)$$

$$i, j=0,1,\dots,n; \forall k \in I \quad (9)$$

$$e_i \leq t_i \leq l_i; i=1,2,\dots,n \quad (10)$$

$$x_{ijk} = 0 \text{ 或 } 1; i, j=0,1,\dots,n; \forall k \in I \quad (10)$$

$$y_{ik} = 0 \text{ 或 } 1; i=0,1,\dots,n; \forall k \in I \quad (11)$$

模型中,式(1)、式(2)表示的是目标函数,其中式(1)表示最小化总的行驶路程,式(2)表示最小化使用车辆数量;式(3)表示车辆 k 服务的所有顾客的需求量不能超过车辆的最大载重量;式(4)~式(6)表示每个顾客只能被一辆车服务一次,每辆车均从配送中心出发并且在服务完顾客后都回到配送中心;式(7)表示消除子回路;式(8)、式(9)表示时间窗约束关系,其中 M 是一个较大的数;式(10)、式(11)表示变量关系。

3 混合差分进化算法 IDE_VND 设计

3.1 算法思想

在混合差分进化算法中,使用 Pareto 支配的思想评价种群中个体的优劣性,并采用擂台法则^[13]构造非支配集。重新设计差分进化操作,使 DE 能直接在离散域内进行个体更新。

以差分进化算法作为求解主体,将求解 VRPTW 问题的启发式规则 PFH^[14]结合随机方式生成规模为 N 的初始种群;使用擂台法则和拥挤距离机制^[15]将种群个体分层排序,为每个个体赋与非支配层等级和拥挤距离值,由文献^[15]可知,处在不同非支配层的个体,非支配层等级序号越小,其个体适应度值越高;处在同一非支配层的个体,拥挤距离越大,其适应度值越高。通过个体的非支配层等级和拥挤距离将种群划分为两个子群:将非支配层等级序号小的和拥挤距离大的个体放入子群 P1 中,采用 DE/best/1 变异策略,用于局部开发,此种变异策略由最优个体做引导,局部开发能力较强,搜索精度较高;非支配层等级序号大的和拥挤距离小的个体放在子群 P2 中,采用 DE/rand/1 变异策略,用于全局探索,这种变异策略由 3 个互不相同的随机个体来完成,有利于种群多样性的保持,具有较强的全局探索能力。

在优化的前期为使算法具有较强的全局探索能力,设定 P2 子群的个体数多;而在后期则需要较强的局部开发能力,设定 P1 子群的个体数量逐渐增多,以达到较快的收敛速度。因此在进化过程中动态地调整两个子群的规模。P1、P2 的个体数由式(12)、式(13)确定

$$P1 = \left[\left(\frac{1}{5} + \frac{2}{3} \times \frac{gen}{gen_{max}} \right) \times N \right] \quad (12)$$

$$P2 = N - P1 \quad (13)$$

其中, N 是种群规模, gen 为当前进化代数, gen_{max} 为最大进化代数。

在进化过程中, 采用双种群机制, 使算法既能从局部极值的邻域跳转到全局最优解的邻域, 又能在全局最优解的邻域内进行精细搜索, 在每代进化完后通过子种群重组实现信息交流和融合, 平衡算法的全局探索能力和局部开发能力。

随着进化过程的进行, 种群中的个体会趋于一致, 因此在每次执行完变异、交叉、选择操作后, 采用随机个体替换掉种群中的重复个体, 维持种群的多样性, 以增强种群的全局探索能力, 然后从种群中随机选取若干个个体进行变邻域下降搜索, 进一步提高算法的局部开发能力, 降低算法陷入局部最优的风险。

3.2 算法步骤

基于以上的算法思想描述, 混合差分进化算法的具体步骤如下:

步骤 1 设置算法的相关参数, 生成算法的初始种群, 设进化代数 $gen=0$;

步骤 2 根据 Pareto 支配思想对种群中的个体适应值进行评价, 利用擂台法则和拥挤距离机制将种群个体分层排序, 得到每个个体的非支配层等级和拥挤距离值;

步骤 3 按照个体的非支配层等级和拥挤距离, 并根据式(12)、式(13)将种群划分为两个不同大小的子群 P1 和 P2;

步骤 4 P1 子群执行 DE/best/1 变异策略, P2 子群执行 DE/rand/1 变异策略, 并根据 3.3.4 节执行交叉操作;

步骤 5 将初始种群与子群 P1、P2 重组为一个混合种群;

步骤 6 对混合种群多次使用擂台法则构造多层非支配集;

步骤 7 对多层非支配集, 使用基于非支配层等级和拥挤距离的选择操作生成下一代目标种群;

步骤 8 用随机个体替代新一代目标种群中的重复个体, 以保证种群的多样性;

步骤 9 从目标种群中随机选取若干个个体执行变邻域下降搜索, 根据 Pareto 支配思想评价个体并更新种群;

步骤 10 判断是否满足终止条件 $gen \leq gen_{max}$, 如果是, 则输出 Pareto 非支配解集, 算法结束; 如果不是, 则转至步骤 3, $gen=gen+1$ 。

3.3 改进差分进化算法

3.3.1 染色体编码设计及路径解码

本文算法中的染色体编码采用自然数编码机制, 用 $(l_1, l_2, l_3, \dots, l_i, \dots, l_n)$ 表示一条染色体, 其中染色体中的每个基因位 l_i 代表一个顾客点, l_i 是 $[1, n]$ 之间的一个不重复的自然数, 基因位间自然数的顺序体现了车辆访问顾客点的次序, 这样的编码方式满足了每个顾客点只能被车辆访问一次的约束。该种编码方式的染色体中没有作为子路径分隔符的基因位, 这样就不需要预先估计所使用的车辆数量。

由于此种编码方式不能从染色体中直接得到问题的解, 即问题的调度方案, 因此设计一种解码算子来将染色体转换成可行的行驶路径。解码时按照染色体中基因位上的自然数排列顺序依次将每个自然数表示的顾客点插入到路径中, 当插入某个顾客点违反了时间窗约束或者车辆最大载重量约束时, 就重新使用一辆车来服务该顾客点, 开辟一条新的路径,

按照这种方法, 直到所有的顾客都被服务。这种解码方法可以使解码后的路径和解码前染色体中所对应的路径方案一致, 并且使用车辆的数量可以在解码过程中灵活动态地获得, 从而实现对车辆数量的自动寻优。例如染色体串 3 6 1 8 5 7 2 9 4, 经过路径解码为: 路线 1: 0→3→6→0; 路线 2: 0→1→8→5→7→0; 路线 3: 0→2→9→4→0。

3.3.2 初始种群生成

产生初始种群时, 为了保证种群的多样性, 其中 90% 的个体采用 N 个顾客节点随机排列的方式来产生, 应用前向插入启发式算法(PFIH)^[14]来生成剩下 10% 的个体。

3.3.3 变异操作

鉴于标准差分进化算法采用实数编码, 不能直接应用于 VRPTW 问题, 由于采用了自然数编码, 因此重新设计了变异操作方式来产生变异个体。由标准 DE 算法可知, 变异个体是由目标种群中随机选择的 3 个目标个体相互作用的结果, 记 $X_i^G = [x_{i,1}^G, x_{i,2}^G, \dots, x_{i,n}^G]$, $V_i^G = [v_{i,1}^G, v_{i,2}^G, \dots, v_{i,n}^G]$, $U_i^G = [u_{i,1}^G, u_{i,2}^G, \dots, u_{i,n}^G]$ 分别为第 G 代目标种群、变异种群和试验种群的第 i 个个体。

(1) P1 子种群采用“DE/best/1”变异策略, 重新定义:

$$V_i^G = g(F \otimes g(X_{r_1}^G, X_{r_2}^G), X_{best}^G) \quad (14)$$

式中, i, r_1, r_2 是区间 $[1, n]$ 里互不相等的整数; X_{best}^G 是当前目标种群中最好的个体, 在本文中从非支配层等级序号最小的非支配层中随机选取; F 为缩放因子, 且 $F \in [0, 1]$ 。

式(14)由两部分组成, 第一部分为:

$$\Delta_i^G = F \otimes g(X_{r_1}^G, X_{r_2}^G) = \begin{cases} g(X_{r_1}^G, X_{r_2}^G), & rand() < F \\ X_{r_1}^G, & rand() \geq F \end{cases} \quad (15)$$

通过式(15)产生一个扰动个体, 其中 $g(X_{r_1}^G, X_{r_2}^G)$ 的实现过程如下: ①从 $X_{r_1}^G$ 中随机选择一段基因, 将其放在 $X_{r_2}^G$ 的前面或者后面; ②删除 $X_{r_2}^G$ 与该段基因相同的顾客。

第二部分由 $V_i^G = g(\Delta_i^G, X_{best}^G)$ 构成, 其实现方法与 $g(X_{r_1}^G, X_{r_2}^G)$ 相同。

(2) P2 子种群采用“DE/rand/1”变异策略, 重新定义:

$$V_i^G = g(F \otimes g(X_{r_2}^G, X_{r_3}^G), X_i^G) \quad (16)$$

式(16)的具体操作过程与 P1 子群的操作过程类似。

3.3.4 交叉操作

交叉操作是通过变异个体和目标个体之间的交叉作用来产生试验个体:

$$U_i^G = CR \otimes g(V_i^G, X_i^G) = \begin{cases} g(V_i^G, X_i^G), & rand() < CR \\ V_i^G, & rand() \geq CR \end{cases} \quad (17)$$

式中, CR 为交叉概率且 $CR \in [0, 1]$, $g(V_i^G, X_i^G)$ 的操作方式与 $g(X_{r_1}^G, X_{r_2}^G)$ 相同。

3.3.5 选择操作

在单目标优化中, 标准 DE 采用一对一竞争的选择操作方式来生成下一代目标个体, 但是不适合直接用在多目标优化问题中。在本文中, 当目标个体生成试验个体时, 不比较它们之间的优劣, 而是在两个目标子种群中的个体都产生试验个体后, 将初始种群与两个试验子种群合并成混合种群; 然后使用基于非支配层等级和拥挤距离^[15]的选择操作从混合种群中选择下一代目标种群。使用这种选择操作方式可以实现精英保留策略, 从而加快算法的收敛速度。

3.4 变邻域下降搜索

变邻域下降搜索的具体步骤:将经过变异、交叉、选择操作之后得到的个体作为初始解,选择一种邻域结构进行局部搜索,直到找到局部最优解;然后再以当前局部最优解为初始解,使用另外一种邻域结构继续进行局部搜索;当任意一种邻域结构都不能继续优化当前解时,则结束 VND 过程。

3.4.1 邻域结构

在使用变邻域下降搜索前,需要定义一组邻域结构。本文采用基于顾客排序的自然数编码方式,因此采用以下 2 种邻域结构:

(1)插入(*Insert*)邻域:在路径序列中随机选择两个不同的位置 a, b ,把位置 a 对应的顾客插入到位置 b 上,得到新的路径序列。

(2)互换(*Swap*)邻域:在路径序列中随机选择两个不同的位置 a, b ,交换这两个位置对应的顾客,得到新的路径序列。

插入邻域结构对解的结构破坏程度较小,有利于算法的小范围迁移;互换邻域结构对解的结构破坏程度稍大,可以保证解的广域搜索。但插入邻域和互换邻域两种邻域结构的直径都相对较小,而在车辆路径问题领域中,局部最优解与全局最优解在解空间上的位置彼此比较靠近,因此采用这两种邻域结构可以将搜索操作控制在一个紧凑的解空间里,从而达到有效的搜索。

3.4.2 VND 算法伪码

算法 1 VND

1. 输入初始解 x ,选择一组邻域结构 $N_k, k=1, 2, \dots, k_{max}$;令 $t=0$;
2. WHILE $t < k_{max}$ DO
3. $k=1$;
4. WHILE $k \leq k_{max}$ DO
5. 以 x 为初始解,在 N_k 定义的邻域中进行局部搜索,直到找到局部最优解 x^* 为止;
6. IF x^* 支配 x THEN
7. $x=x^*$;
8. $t=0$;
9. ELSE
10. $t=t+1$;
11. END IF
12. $k=k+1$;
13. END WHILE
14. END WHILE
15. 输出 x ,算法结束

在使用一种邻域结构局部搜索时,由于邻域解较多,为了减少搜索成本,因此设置最大无改进迭代次数来避免对整个邻域空间的可行解进行穷举搜索,当满足最大无改进迭代次数时,则变换另一种邻域结构继续搜索。当达到邻域搜索的最大迭代次数时,停止 VND 搜索。实验采用随机的邻域结构顺序来进行变邻域下降搜索。

4 实验分析

为了测试算法的有效性,使用著名的 Solomon 算例^[16]作为本文的测试实例。该算例有 3 种不同规模顾客数(25, 50 和 100 个顾客点)的数据集,每种数据集包含 56 个测试问题,并且根据顾客的地理位置分布分为 6 类: C1、C2、R1、R2、RC1、RC2。其中 C 类问题中包含多个组,每组中的顾客点坐

标都是服从聚类分布的;R 类问题中的所有顾客点坐标则全部服从均匀分布;RC 类问题中的顾客点则兼有聚类分布和均匀分布的特征。C1、R1、RC1 这 3 类问题的车辆最大载重量较小且配送中心的时间窗也较窄,因此每辆车能服务的顾客较少;而 C2、R2、RC2 这 3 类问题的车辆载重量较大且配送中心的时间窗较宽,因此每辆车能服务较多的顾客。

本文的实验结果都是通过使用 VC++6.0 编程,在 Intel Pentium CPU E5400 2.72GHz, 2.00GB 内存、Windows XP SP3 的主机上运行所得的。本文算法的实验参数设置:对于 25 个顾客集,种群规模为 40,进化代数 gen_{max} 为 400;对于 50 个顾客集,种群规模为 100,进化代数 gen_{max} 为 1000;对于 100 个顾客集,种群规模为 100,进化代数 gen_{max} 为 1000。VND 搜索的最大迭代次数为 100,最大无改进迭代次数为 10。P1、P2 子种群的缩放因子 F 分别取 0.4 和 0.8,交叉概率 CR 取 0.08。

4.1 混合算法的性能测试

为了验证本文所提的混合差分进化算法的性能,分别进行两组实验:第一组实验不加入 VND,只使用改进的差分进化算法;第二组实验使用本文提出的混合差分进化算法 IDE_VND。不失一般性,分别从 Solomon 算例的 25 个顾客集、50 个顾客集和 100 个顾客集中选取 18 个算例作为实验测试数据,实验结果如表 1 所列。

表 1 混合启发式算法 IDE_VND 的性能测试结果

算例	顾客数	IDE		IDE_VND	
		车辆数量	行驶路程	车辆数量	行驶路程
C104	25	3	211.0	3	187.5
C201	25	2	215.5	2	215.5
R103	25	5	500.9	5	455.7
R201	25	4	489.1	4	464.4
		2	696.1	2	526.8
RC101	25	4	490.3	4	462.2
RC208	25	2	327.1	2	269.6
C103	50	1	400.6	1	309.6
		5	388.2	5	362.2
C201	50	3	361.8	3	361.8
R101	50	2	458.4	2	445.0
		13	1063.8	12	1046.7
R201	50	12	1072.2		
		7	854.0	6	794.3
RC101	50	3	1327.7	3	904.0
		9	993.7	8	947.0
RC204	50	3	566.1	3	449.2
C101	100	2	602.0	2	483.0
		13	989.3	10	828.9
C201	100	12	997.1		
		3	603.9	3	591.6
R103	100	16	1496.6	15	1235.5
R201	100	9	1557.4	9	1174.8
		6	1761.8	6	1268.7
RC101	100	19	1921.5	17	1675.1
		18	1957.5		
RC205	100	8	1669.6	8	1177.4
		5	2115.5	5	1461.0

通过表 1,从求解具有最短路程的解来看,在 2 个算例(25 个顾客规模算例 C201 和 50 个顾客规模算例 C201)上 IDE_VND 与 IDE 求解结果相同,而在其余的算例上,本文算法 IDE_VND 的求解结果比不混合 VND 的 IDE 算法更优;相比于 IDE,IDE_VND 在此类解的行驶路程目标上最大提高 31.53%,而在此类解的车辆数量目标上最大提高 23.08%。

从求解具有最少车辆数量的解来看,在5个算例(25个顾客规模算例 R103,50个顾客规模算例 RC101 以及100个顾客规模算例 C101、R103、RC101)上,IDE_VND 可以求得比 IDE 具有更少车辆数量的解;而且在解具有相同车辆数量的情况下,IDE_VND 求得的路程更短。由此可以得出,将 IDE 和 VND 混合是有效的。

为了更好地验证 IDE_VND 算法的稳定性,本文对每个算例分别求解 20 次。表 2 给出 20 次实验的统计结果,主要列举了各算例在行驶路程目标上 20 次求解得到的最优解的平均值、平均值与已知最优解之间的相对偏差以及标准差。

表 2 对每个算例求解 20 次得到的平均结果

算例	顾客数	已知最优解	IDE_VND 运行 20 次的结果		
			平均值	平均相对偏差	标准差
C104	25	186.9	188.2	0.70%	1.22
C201	25	214.7	215.5	0.37%	0
R103	25	454.6	461.6	1.54%	6.43
R201	25	463.3	464.4	0.24%	0
RC101	25	461.1	468.0	1.50%	7.88
RC208	25	269.1	270.7	0.59%	1.67
C103	50	361.4	362.2	0.22%	0
C201	50	360.2	361.8	0.44%	0
R101	50	1044	1049.6	0.54%	2.84
R201	50	791.9	802.9	1.39%	5.01
RC101	50	944	958.5	1.53%	5.62
RC204	50	444.2	459.5	3.44%	7.47
C101	100	827.3	828.9	0.19%	0
RC201	100	589.1	591.6	0.42%	0
R103	100	1208.7	1252.9	3.66%	11.18
R201	100	1143.2	1188.2	3.94%	12.60
RC101	100	1619.8	1690.2	4.35%	8.91
RC205	100	1154	1201.3	4.10%	16.80

从表 2 可以看出,IDE_VND 算法求解每个算例的平均

值与目前最优解的平均相对偏差较小,均控制在 4.5% 以内;所有算例的平均值与目前已知最优解的平均值的相对偏差为 2.21%,每个算例的标准差相对平均值较小。由此可以看出,算法 IDE_VND 在求解 VRPTW 时具有较强的稳定性。

以算例 R101(50 个顾客)为例,给出了 IDE 和 IDE_VND 的运行收敛情况,如图 1 所示。从图中可以看出,不论是在行驶路程上还是在车辆数量上,IDE_VND 的收敛速度均快于 IDE,并且在收敛结果的质量上也好于 IDE。

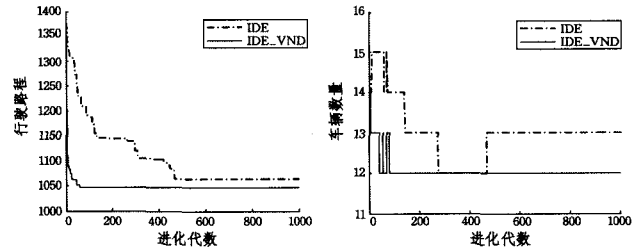


图 1 算例 R101(50 个顾客)的收敛情况

4.2 与现有算法的对比实验

为进一步验证本文算法的性能,将算法 IDE_VND 求得的最优解及 20 次平均值分别与文献[17]中的 ABC(人工蜂群算法)算法和已知最优解的结果进行比较,结果如表 3 所列。表中平均值为 20 次运行中所得到的所有非支配解分别在车辆数量和行驶路程上的平均值;最后一列的相对偏差表示的是本文算法的最优解与已知最优解在车辆数量和行驶路程两个目标上的差距,可采用式(18)计算:

$$\text{相对偏差} = (\text{最优解} - \text{已知最优解}) * 100\% / \text{已知最优解} \quad (18)$$

表 3 算法 IDE_VND 和现有算法在不同规模算例上的实验结果比较

算例	顾客数	ABC				IDE_VND				已知最优解		相对偏差		
		最优解		平均值		最优解		平均值		车辆数量	行驶路程			
		车辆数	行驶路程	车辆数	行驶路程	车辆数	行驶路程	车辆数	行驶路程					
C104	25	3	187.5	3	189.0	3	187.5	3	188.2	3	186.9	0.00%	0.32%	
C201	25	2	215.5	2	216.7	2	215.5	2	215.5	2	214.7	0.00%	0.37%	
R103	25	5	455.7	5.4	468.1	5	455.7	4.7	461.6	5	454.6	0.00%	0.24%	
R201	25	4	479.2	5.4	468.1	4	473.4	4.7	461.6			-20.00%	4.14%	
		4	464.4	3.6	482.7	4	464.4	3	501.8	4	463.3	0.00%	0.24%	
RC101	25	3	481.5	3.6	482.7	2	526.8	3	501.8			-50.00%	13.71%	
		4	463.6	4.9	478.7	4	462.2	4.3	468.0	4	461.1	0.00%	0.24%	
RC208	25	2	269.6	2	270.8	2	269.6	1.5	293.4	2	269.1	0.00%	0.19%	
C103	50	1	309.6	1.5	293.4								-50.00%	15.05%
		5	392.2	6.1	439.9	5	362.2	5	362.2	5	361.4	0.00%	0.22%	
C201	50	3	373.8	3.8	411.5	3	361.8	2.5	387.2	3	360.2	0.00%	0.44%	
		2	445.0	2.5	387.2								-33.33%	23.54%
R101	50	13	1049.5	13.1	1081.8	12	1046.7	12.2	1049.6	12	1044	0.00%	0.26%	
		12	1061.2	13.1	1081.8									
R201	50	7	829.8	7.4	872.7	6	794.3	4.5	902.5	6	791.9	0.00%	0.30%	
		5	872.8	7.4	872.7	3	904.0	4.5	902.5			-50.00%	14.16%	
RC101	50	10	977.1	10	986.6	8	947.0	9	958.5	8	944	0.00%	0.32%	
		9	982.6	10	986.6									
RC204	50	3	459.4	3	508.1	3	449.2	2.5	474.4	3	444.2	0.00%	1.13%	
		2	524	3	508.1	2	483.0	2.5	474.4			-33.33%	8.73%	
C101	100	10	828.9	10.2	846.2	10	828.9	10	828.9	10	827.3	0.00%	0.19%	
C201	100	4	618.6	4	631.8	3	591.6	3	591.6	3	589.1	0.00%	0.42%	
R103	100	15	1264.5	15.3	1273.9	15	1235.5	15.3	1252.9	14	1208.7	15.38%	2.22%	
R201	100	11	1227.2	10.8	1245.9	9	1174.8	7.6	1245.1	8	1143.2	12.50%	2.76%	
		9	1261.7	10.8	1245.9	6	1268.7	7.6	1245.1			-25.00%	10.98%	
RC101	100	18	1698	18.2	1734.1	17	1675.1	17.2	1690.2	15	1619.8	13.33%	3.41%	
RC205	100	9	1221.6	8.9	1249.3	8	1177.4	7.6	1312.3	7	1154	14.29%	2.03%	
		8	1249.7	8.9	1249.3	5	1461.0	7.6	1312.3			-28.57%	26.60%	

在表 3 中,从最小化行驶路程的目标来看,IDE_VND 在 6 个算例(25 个顾客规模算例 C104、C201、R103、R201、

RC208 和 100 个顾客规模算例 C101) 上与 ABC 求解结果相同, 在其余算例上 IDE_VND 的求解质量优于 ABC; 相比于 ABC, IDE_VND 的求解质量最大提高 7.65%; 并且 IDE_VND 与已知最优解的相对偏差均控制在 3.5% 以内。从最小化车辆数量的目标来看, IDE_VND 在 9 个算例(25 个顾客规模算例 R201、RC208, 50 个顾客规模算例 C201、R201、RC101 以及 100 个顾客规模算例 C201、R201、RC101、RC205) 上可以找到比 ABC 具有更少车辆数量的解; 与已知最优解相比, IDE_VND 可以在 8 个算例(25 个顾客规模算例 R103、R201、RC208, 50 个顾客规模算例 C201、R201、RC204 以及 100 个顾客规模算例 R201、RC205) 上找到具有更少车辆数量的解。从 20 次运行所得的非支配解的平均值来看, 在具有唯一解的算例中, IDE_VND 的优势相比 ABC 更加明显; 在一些具有多个非支配解的算例中, ABC 的行驶路程的平均值优于 IDE_VND 的行驶路程的平均值, 而在车辆数量的平均值上要劣于 IDE_VND, 原因是 IDE_VND 可以求得更少的车辆数量, 但是相对增加了行驶路程; 在车辆数量平均值和行驶路程平均值上, IDE_VND 相比于 ABC 分别最大提高 39.19% 和 17.66%。

由此表明, IDE_VND 算法充分利用了差分进化算法较强的全局搜索能力和双种群的协同搜索能力; 同时变邻域下降搜索有效地增强了算法的局部开发能力, 避免了算法过早陷入局部最优, 提高了算法的求解质量。

图 2—图 4 给出了 3 种算法以 25 个顾客规模算例 R201、50 个顾客规模算例 R201 以及 100 个顾客规模算例 RC205 为例的 Pareto 前沿对比情况。比较 3 种算法可以看出, 在 Pareto 解的数量和质量上, IDE_VND 具有更加明显的优势, 所求得的 Pareto 前沿更加逼近实际 Pareto 前沿。

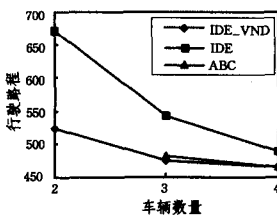


图 2 算例 R201(25 个顾客)中的 Pareto 前沿比较

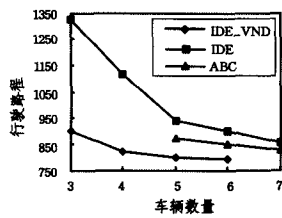


图 3 算例 R201(50 个顾客)中的 Pareto 前沿比较

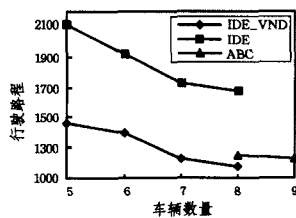


图 4 算例 RC205(100 个顾客)中的 Pareto 前沿比较

结束语 本文对带时间窗的车辆路径问题进行了研究, 从多目标的角度出发, 同等地考虑行驶路程和车辆数量这两个目标, 提出一种结合改进差分进化算法和变邻域下降搜索的基于 Pareto 支配的混合差分进化算法 IDE_VND。利用改进差分进化算法良好的全局探索能力和变邻域下降搜索较强

的局部寻优能力, 提高了算法的求解质量, 加速了算法的收敛。最后对 18 个不同规模的 Solomon 算例进行实验测试, 结果表明本文算法是求解 VRPTW 的一种有效方法, 为 VRPTW 提供了一种新的求解思路, 同时对研究其他类型的车辆路径问题具有一定的借鉴意义。

参考文献

- [1] Dantzig G, Ramser J. The truck dispatching problem [J]. *Management Science*, 1959(6): 80-91
- [2] Jung S, Moon B. R. A Hybrid genetic algorithm for the vehicle routing problem with time windows [C]// *Proceedings of Genetic and Evolutionary Computation*. San Francisco, CA, USA, 2002: 1309-1316
- [3] Balseiro S R. An ant colony algorithm hybridized with insertion heuristics for the time dependent vehicle routing problem with time windows [J]. *Computers and Operations Research*, 2011, 38(6): 954-966
- [4] 魏明, 靳文舟. 求解车辆路径问题的离散粒子群算法 [J]. *计算机科学*, 2010, 37(4): 187-191
- [5] Storn R. Differential evolution design of an IIR-filter [C]// *Proceedings of IEEE Conference Evolutionary Computation (S07803-29023)*. Nagoya, Japan, 1996: 268-273
- [6] Mei Mi, Xue Hui-feng, Zhong Ming, et al. An improved differential evolution algorithm for TSP problem [C]// *Proceedings of Intelligent Computation Technology and Automation*. Washington DC, USA, 2010: 544-547
- [7] Onwubolu G, Davendra D. Scheduling flow shops using differential evolution algorithm [J]. *European Journal of Operational Research*, 2006, 171(2): 674-69
- [8] 曹二保, 赖明勇, 聂凯. 带时间窗的车辆路径问题的改进差分进化算法研究 [J]. *系统仿真学报*, 2009, 21(8): 2420-2423
- [9] 王君. 带时间窗车辆路径问题的差分进化混合算法 [J]. *计算机工程与应用*, 2013, 49(2): 24-28, 66
- [10] Babu B V, Jehan M M L. Differential evolution for multi-objective optimization [C]// *Proceedings of the IEEE Congress on Evolutionary Computation (CEC 2003)*. Canberra, Australia, 2003: 2696-2703
- [11] Hansen P, Mladenovic N. Variable neighborhood search: principles and applications [J]. *European Journal of Operational Research*, 2001, 130(3): 449-467
- [12] 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 [J]. *计算机学报*, 2008, 31(4): 565-573
- [13] 郑金华, 蒋浩, 邝达, 等. 用擂台法则构造多目标 Pareto 最优解集的方法 [J]. *软件学报*, 2007, 18(6): 1287-1297
- [14] 刘敏. 多目标遗传算法在车辆路径优化中的应用研究 [D]. 湘潭: 湘潭大学, 2006
- [15] Deb K, Pratap A, Agarwal S, et al. A fast and elitist multiobjective genetic algorithm: NSGA-II [J]. *IEEE Transactions on Evolutionary Computation*, 2002, 6(2): 182-197
- [16] <http://web.cba.neu.edu/~msolomon/problems.htm>
- [17] Nahum O E, Hadas Y, Spiegel U. Multi-objective vehicle routing problem with time windows: a vector evaluated artificial bee colony approach [J]. *International Journal of Computer and Information Technology*, 2014, 3(1): 41-47