

应急物流中的累计时间式多车场车辆路径问题

曾正洋, 许维胜, 徐志宇, 刘竹馨

(同济大学 电子与信息工程学院, 上海 201804)

摘要: 为提升应急救援的快速性和公平性, 以最小化所有受灾点的累计等待时间为目标建立累计时间式多车场车辆路径问题(Cum-MDVRP)的模型. 由于该问题具有NP-hard性质, 设计了一种多起始点变邻域下降法对其进行快速求解. 每次迭代中, 多起始点方法通过改进的Split算法结合可行性修复程序生成随机的初始可行解, 然后由变邻域下降法对其进行进一步改进. 扩展的标准算例的测试结果验证了所提出模型和求解算法的有效性.

关键词: 应急物流; 多车场车辆路径问题; 累计等待时间; 多起始点方法; 变邻域下降法

中图分类号: TP273

文献标志码: A

Cumulative multi-depot vehicle routing problem in emergency logistics

ZENG Zheng-yang, XU Wei-sheng, XU Zhi-yu, LIU Zhu-xin

(School of Electronics and Information Engineering, Tongji University, Shanghai 201804, China. Correspondent: XU Wei-sheng, E-mail: xuweisheng@tongji.edu.cn)

Abstract: To promote the rapidity and equity(fairness) in emergency rescue, a cumulative multi-depot vehicle routing problem(Cum-MDVRP) model with the objective of minimizing the cumulative waiting time in all affected areas is built. Because of the NP-hard nature of this problem, a multi-start variable neighborhood descent algorithm is developed to solve it efficiently. In each iteration, multi-start methods produce a randomly initial feasible solution by an improved Split algorithm combined with a feasibility repairing procedure, and then this solution is further improved by a variable neighborhood descent algorithm. Results of computational tests on some extended benchmark instances show the effectiveness of the proposed model and the good performance of the developed algorithm.

Key words: emergency logistics; multi-depot vehicle routing problem; cumulative waiting time; multi-start methods; variable neighborhood descent

0 引言

经典的车辆路径问题(VRP)大多从物流企业本身考虑物资配送方案, 以配送成本(使用车辆数、行车距离等)最小化为目标函数. 而在各种突发事件发生后的应急物流中, 应从受灾点的角度考虑物资配送过程的快速性和公平性, 需要考虑救援物资到达每个受灾点的时间^[1]. 由于传统的成本最小化目标不能恰当反映受灾点的真实需求^[1], “累计等待时间^[2-4]”这一新目标随之出现. 累计等待时间等价于平均等待时间, 同属于公平性的一种度量指标^[5].

学术界对VRP及其变体已有了较为深入的研究, 但针对以累计等待时间为目标函数的VRP变体的研究却只有累计时间式带容量约束的车辆路径问题(CCVRP)^[2-4]. 由于实际的应急物流中通常存在多个

出救点(车场), 故将累计等待时间这一概念与多车场车辆路径问题(MDVRP)^[6-7]相结合, 得到的累计时间式多车场车辆路径问题(Cum-MDVRP)更具实际意义, 但目前尚未见到将二者结合起来研究的报道.

CCVRP已证明是NP难问题^[2], Cum-MDVRP是对CCVRP的一种多车场扩展, 比CCVRP更为复杂, 同样是NP难问题. 由于应急物流时效性^[1]的要求, 在较短的时间内给出较好的配送方案尤为重要, 为此本文选择用启发式方法求解Cum-MDVRP.

启发式方法在求解困难的组合优化问题时, 需要一定的多样性以搜索到全局最优解. 多起始点方法^[10-11]在遇到局部最优时, 生成新的初始解进行搜索改进, 以增加多样性, 避免在改进希望较小的解周围反复搜索^[10-11]. 多起始点方法的每次迭代均产生一个

收稿日期: 2013-08-05; 修回日期: 2014-01-09.

基金项目: 国家自然科学基金重大项目(71090404, 71090400).

作者简介: 曾正洋(1987-), 男, 博士生, 从事智能优化算法、应急物流的研究; 许维胜(1966-), 男, 教授, 博士生导师, 从事应急物流、智能控制等研究.

局部最优值, 迭代过程中遇到的最好的局部最优值作为最终输出. 多起始点方法的局部开发能力需要通过局部搜索来强化.

变邻域下降法 (VND)^[8-9] 是一种简化的变邻域搜索 (VNS)^[8-9] 算法. VND 的核心思想是为系统切换一组邻域: 一方面通过下降的过程寻找到局部最优; 另一方面通过扰动该局部最优以获得新的搜索起始点^[9]. VND 不包含扰动过程, 虽优化效率较高, 但易于陷入局部最优, 通常用于强化局部搜索能力.

多起始点方法与变邻域下降法的组合平衡了启发式算法搜索的多样性和深度, 已成功用于卡车带拖车路径问题 (TTRP)^[12]、旅行维修员问题 (TRP)^[13]、校车路径问题 (SBRP)^[14] 等. 受这些成果启发, 本文选择合适的局部搜索算子并结合多起始点方法, 设计了一种多起始点变邻域下降法 (MS-VND) 来求解所提出的 Cum-MDVRP 模型, 以平衡计算结果质量和计算时间.

1 Cum-MDVRP 的数学模型

1.1 问题描述

假设某地存在多个应急救援出救点 (车场), 每个出救点有一定数目的出救车辆和其他应急物资. 某个突发事件发生后, 出现多个受灾点等待出救车辆运送应急物资. 受灾点的需求量已知, 且均需要及时通过出救车辆一次满足 (不可拆分); 出救车辆相同, 且存在相同的运载容量约束; 各节点 (出救点、受灾点) 之间的行车时间已知且具有对称性.

需要解决的问题是: 在已有的约束条件下, 如何合理地进行各出救点出救车辆和应急资源的调配, 以及安排各出救车辆的配送路径, 满足所有受灾点的物资需求量, 并使所有受灾点的等待时间总和最小. 图 1 为 Cum-MDVRP 的一个简单示例.

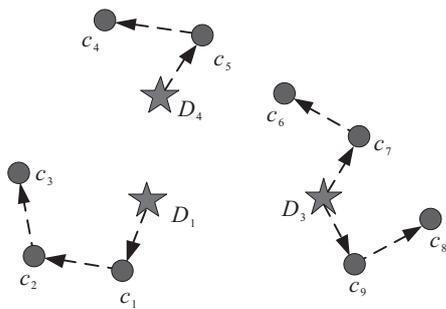


图 1 Cum-MDVRP 示例

图 1 中每条路径均未闭合是因为 Cum-MDVRP 的每辆车返回出救点的时间对受灾点累计等待时间没有影响. 由于目标函数的不同, Cum-MDVRP 不同于 MDVRP, 也不同于多车场开放式车辆路径问题 (MDOVRP).

1.2 符号说明

标号 C 为受灾点集合, n_c 为受灾点总数; D 为出救点集合, n_d 为出救点总数; K 为出救车辆集合, N_V 为可用车辆总数. 参数 c_{ij} 为节点 i 到节点 j 的行车时间; q_i 为受灾点 i 所需要的应急物资量; N_j 为第 j 个出救点的可用车辆数; W 为所使用车辆的最大运载容量. 决策变量 x_{ij}^k 为车辆 k 从节点 i 直接到节点 j 时为 1, 否则为 0; t_i^k 为出救车辆 k 到达受灾点 i 所用的时间, 若车辆 k 不出救受灾点 i , 或 i 为出救点时, 则定义 $t_i^k = 0$.

1.3 数学模型

根据以上假设和定义, 并综合 MDVRP^[6-7] 和 CCVRP^[2-4] 的数学模型, 建立如下累计时间式多车场车辆路径问题 (Cum-MDVRP) 的混合整数规划模型. 目标函数

$$\min Z = \sum_{k \in K} \sum_{i \in C} t_i^k. \quad (1)$$

约束条件

$$\sum_{i \in C \cup D} x_{ij}^k = \sum_{i \in C \cup D} x_{ji}^k, \quad \forall j \in C \cup D, k \in K; \quad (2)$$

$$\sum_{k \in K} \sum_{i \in C \cup D} x_{ij}^k = 1, \quad \forall j \in C; \quad (3)$$

$$\sum_{i \in D} \sum_{j \in C} x_{ij}^k \leq 1, \quad \forall k \in K; \quad (4)$$

$$\sum_{i \in D} \sum_{j \in C} x_{ij}^k = \sum_{i \in C} \sum_{j \in D} x_{ij}^k, \quad \forall k \in K; \quad (5)$$

$$\sum_{i \in C} \sum_{j \in C \cup D} q_i x_{ij}^k \leq W, \quad \forall k \in K; \quad (6)$$

$$\sum_{k \in K} \sum_{j \in C} x_{ij}^k \leq N_i, \quad \forall i \in D; \quad (7)$$

$$\sum_{k \in K} \sum_{i \in D} \sum_{j \in C} x_{ij}^k \leq N_V; \quad (8)$$

$$N_V \leq \sum_{i \in D} N_i; \quad (9)$$

$$t_i^k + c_{ij} - (1 - x_{ij}^k)M \leq t_j^k, \quad \forall i \in C \cup D, j \in C, k \in K; \quad (10)$$

$$t_i^k \geq 0, \quad \forall i \in C \cup D, k \in K; \quad (11)$$

$$x_{ij}^k \in \{0, 1\}, \quad \forall i \in C \cup D, j \in C \cup D, k \in K. \quad (12)$$

式 (1) 为目标函数, 表示最小化所有受灾点的累计等待时间; 式 (2) 为流量守恒约束, 出救车辆进、出某个节点的次数相等; 式 (3) 表示每个受灾点恰好被服务一次; 式 (4) 表示每辆出救车辆最多出动一次; 式 (5) 为每辆出救车辆对出救点集合的进出平衡条

件; 式 (6) 表示每辆车的配送量不得超过其容量限制; 式 (7) 为各出救点可用车辆数限制; 式 (8) 为可用车辆总数限制; 式 (9) 表示可用车辆总数不得超过所有出救点可用车辆数之和; 式 (10) 计算每个点的车辆到达时间, 同时消除由受灾点组成的子回路, 其中 M 为一个很大的正的常数; 式 (11) 表示决策变量 t_i^k 的非负约束; 式 (12) 表示决策变量 x_{ij}^k 为 0-1 变量.

2 Cum-MDVRP 的求解算法

2.1 求解思路

采用多起始点方法和变邻域下降法 (VND) 组合求解 Cum-MDVRP. 首先不考虑各出救点 (车场) 可用车辆数的限制, 循环生成由所有受灾点组成的随机排列, 通过改进的最优分割算法进行分割, 直至获得总车辆数可行的分配方案; 然后根据各车场可用车辆数限制修复该方案, 获得初始可行解; 最后由几种局部搜索算子组成的 VND 改进初始可行解. 当 VND 无法改进当前解时, 由多起始点方法按上述过程重新生成随机的初始可行解, 再使用 VND 改进. 到达最大迭代次数后, 终止搜索过程, 输出找到的最好结果.

2.2 多起始点方法中初始可行解的构建

为增加搜索的多样性, 多起始点方法中的初始可行解通过合理分割由所有受灾点组成的随机排列产生. Cum-MDVRP 存在可用车辆总数约束和各车场 (出救点) 可用车辆数约束. 首先考虑可用车辆总数约束, 对所有受灾点进行随机排列, 通过对 Nguyen 等^[15] 的 Split 算法进行扩展与改进, 引入受灾点等待时间的计算方法, 同时以优先考虑车辆数最少, 其次考虑累计等待时间最短为原则, 按随机排列中的顺序将受灾点合理分配至各个车场. Split 算法最早由 Beasley^[16] 提出, 由 Prins^[17] 给出详细的实现过程. 基本的 Split 算法^[16-17] 只适用于单车场 VRP, 并不适用于 MDVRP 和 Cum-MDVRP; Nguyen 等的 Split 算法^[15] 虽考虑了多个车场, 但目标函数仍为配送成本最小化, 并未考虑累计等待时间, 也未考虑最小化所使用的车辆数. 针对当前 Split 算法的缺陷, 并结合 Cum-MDVRP 的目标函数, 对 Split 算法进行改进, 改进后的伪代码如下:

```

Time0 := 0; V0 := 0;
for i := 1 to nc do Timei := +∞, Vi := i; endfor
for i := 1 to nc do
  load := 0; COST := 0; j := i;
  repeat
    load := load + Q(Rj);
    cost1 := min{T(d, Ri) | d ∈ D} × (j - i + 1);
    cost2 := 0;

```

```

if j > i then
  for k := 1 to (j - i) do
    cost2 := cost2 + T(Ri+k-1, Ri+k) ×
      (j - i + 1 - k);
  endfor
  COST := cost1 + cost2;
  if load ≤ W then
    if Vj > Vi-1 + 1,
      {Vj := (Vi-1 + 1),
      Timej := Timei-1 + COST, Pj := i - 1;}
    if (Vj = Vi-1 + 1) and
      (Timei-1 + COST < Timej),
      Timej := Timei-1 + COST, Pj := i - 1;
      j = j + 1;
    until (j > nc) or load > W;
  endfor

```

改进的代码中, R 为由所有受灾点组成的随机排列, V_i 为 R 中前 i 个节点所需的出救车辆数, Time_i 为对应的累计等待时间, $T(i, j)$ 为节点 i 到节点 j 的行车时间, $Q(R_i)$ 为受灾点 R_i 的需求量. 计算完毕后, 得到配送方案信息: V_{n_c} 为配送方案所需要的最少车辆数; Time_{n_c} 为全部各受灾点的累计等待时间; P_i 记录了以需求点 R_i 为终点的最优可行弧的起点.

初始分配方案由节点信息从后向前复现^[15]: 根据 P_{n_c} 获得最后一条配送路径, 同时获得前一条路径的结尾, 继续向前运行直至所有受灾点分配完毕.

接着考虑各车场的可用车辆数约束, 对获得的配送方案进行可行性修复: 随机选择车辆数超出约束的车场所分配的路径, 并随机安排给其他车辆有剩余的车场. 可行性修复完毕后, 获得初始可行解.

2.3 变邻域下降法中局部搜索算子的选择

变邻域下降法通过系统性切换一组邻域来下降搜索改进当前解, 邻域的选择需要与所求解的问题相适应. 针对本文的 Cum-MDVRP, 采用 VRP 常用的“路径内部”和“路径之间”两类局部搜索算子^[18] 的组合, 并扩展了部分算子的搜索范围.

1) 路径内部改进. 采用 2-Opt、swap 和 move 算子^[18-20]. 2-Opt 算子通过左右翻转路径内部的一段连续的节点来搜索改进; swap 算子通过将单个节点与单个节点交换位置, 或单个节点与连续的两个节点交换位置来搜索改进; move 算子通过将单个节点移动位置或连续的两个节点移动位置来搜索改进, 该算子的作用范围包括一条路径所属的车场.

2) 路径之间改进. 采用 2-Opt*、swap 和 move 算

子^[18-20], 并针对 Cum-MDVRP 扩大了 swap 和 move 算子的搜索范围. 2-Opt* 算子将两条路径各自去除一条边, 再重新组合成两条新的路径. swap 算子交换两条路径中的连续的节点, 限定每条路径可交换的连续节点数最多为 3 个; 此外, 扩展了 swap 算子搜索范围: 当两条路径属于不同车场时, 交换它们所属的车场来搜索改进. move 算子将一条路径中的连续的节点移动到另外一条路径中, 限定可移动连续节点数最多为 3 个; 此外, 针对单条路径扩展了 move 算子的搜索范围: 将一条路径所属的车场更换为其他的车场来搜索改进. 为增强搜索的效能, 当路径之间局部搜索算子每次产生的新路径均满足车容量约束和车场车辆数约束时, 采用路径内部搜索算子进一步优化^[19-20].

2.4 求解 Cum-MDVRP 的 MS-VND 算法流程

本文求解 Cum-MDVRP 的 MS-VND 算法框架可用图 2 所示的流程图简要表述.

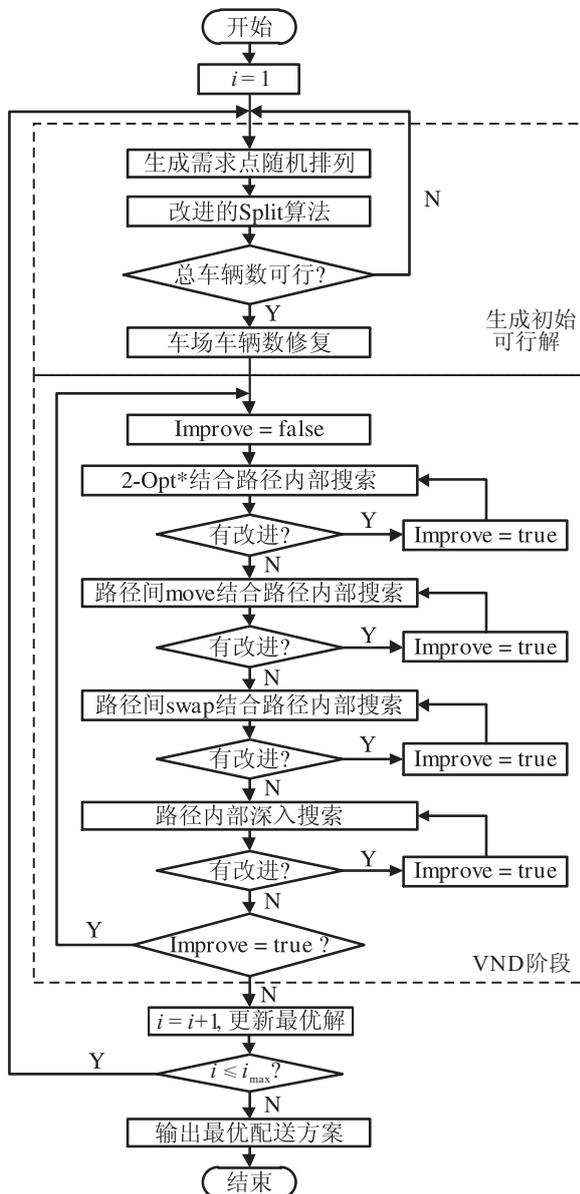


图 2 MS-VND 算法流程

图 2 中: 参数 i 为当前的迭代次数, i_{\max} 为算法设定的最大迭代次数; Improve 用于指示迭代过程中是否出现改进; “车场车辆数修复”是指随机分配车辆数超出限制的车场下属的路径使其满足可用车辆数约束; “路径内部深入搜索”是指利用路径内部的 2-Opt、swap 和 move 算子循环改进每条路径, 直至这 3 种算子均无法改进.

VND 阶段的局部搜索算子均采用 First-Accept 策略^[18]以减少计算量: 当搜索到更优解时, 立即替换当前解, 并继续局部搜索过程. 每次生成一个初始可行解后, 局部搜索算子按顺序循环执行, 直至所有算子均无法改进当前解. 由于路径内部 3 种局部搜索算子执行次数较多, 限定其与路径之间算子结合使用时, 仅按顺序执行; 在“路径内部深入搜索”中用这 3 种算子循环改进每条路径^[19-20]. 当所有算子均无法改进当前解时, 终止 VND 阶段.

3 仿真与分析

3.1 算例描述与实验设置

本文算例取自 MDVRP 标准算例, 算例及当前最优解可从 <http://neo.lcc.uma.es/radi-aeb/WebVRP/> 获得. 算例中包含不同的需求点数、车场数、车容量约束和各车场可用车辆数限制, 详见表 1. 通过使用 Cum-MDVRP 的累计等待时间这一目标函数重新进行实验测试. 为减少受灾点的等待时间, 需要出动尽可能多的出救车辆^[2], 实验设定 Cum-MDVRP 可用车辆总数 N_V 等于各车场可用车辆数之和, 或者等于各车场可用车辆数之和减 1. 由于改进的 Split 算法生成的初始解的车辆数一般小于 N_V , 这可通过对 Split 算法中使用的车容量 W 进行适当的减小来产生刚好满足总车辆数要求的初始解.

在使用主频为 2.8 GHz 的 Intel 奔腾双核 E5500 处理器 (只使用单核心)、内存为 2 GB 的硬件平台上通过 Microsoft Visual C++ 6.0 实现本文求解 Cum-MDVRP 的 MS-VND 算法. 设定最大迭代次数为 200, 每个算例均进行 10 轮独立的计算.

3.2 实验结果与对比分析

实验结果见表 1. “MDVRP 结果”是指用 Cum-MDVRP 的目标函数评价 MDVRP 各算例的当前最优解获得的结果, 考虑路径整体左右翻转导致累计等待时间的差异, 选择其中较小值作为该路径的累计等待时间. MS-VND 为本文的多起始点变邻域下降法获得的结果, 每个算例均给出了 10 轮计算结果的最优值 (Best)、平均值 (Average) 和每轮计算的最优值的平均出现时间 (Time).

表 1 算例特性与测试结果

序号	算例名称	n_c	n_d	W	N_i	N_V	MDVRP 结果	MS-VND		
								Best	Average	Time/s
1-1	p01	50	4	80	4	16	1 256.04	846.62	846.62	0.62
1-2	p01	50	4	80	4	15	1 256.04	873.37	874.01	1.97
2-1	p02	50	4	160	2	8	2 232.96	1 296.81	1 296.81	1.21
2-2	p02	50	4	160	2	7	2 232.96	1 469.08	1 470.43	36.41
3-1	p03	75	5	140	3	15	2 128.52	1 389.70	1 391.86	28.70
3-2	p03	75	5	140	3	14	2 128.52	1 457.05	1 458.30	39.36
4-1	p04	100	2	100	8	16	3 113.54	2 426.84	2 434.12	46.43
4-2	p04	100	2	100	8	15	3 113.54	2 569.01	2 588.06	81.73
5-1	p05	100	2	200	5	10	4 794.78	3 124.58	3 132.39	429.89
5-2	p05	100	2	200	5	9	4 794.78	3 391.34	3 396.85	470.42
6-1	p06	100	3	100	6	18	2 870.60	1 975.89	1 981.27	82.26
6-2	p06	100	3	100	6	17	2 870.60	2 047.21	2 057.21	52.68
7-1	p07	100	4	100	4	16	2 769.28	2 120.07	2 143.32	46.17
7-2	p07	100	4	100	4	15	2 769.28	2 291.20	2 308.58	81.40
8-1	p12	80	2	60	5	10	6 245.52	4 510.11	4 510.11	42.62
8-2	p12	80	2	60	5	9	6 245.52	4 914.65	4 914.65	87.43
9-1	p15	160	4	60	5	20	13 511.47	8 720.73	8 731.82	642.21
9-2	p15	160	4	60	5	19	13 511.47	9 104.99	9 131.68	811.04

由表 1 可知, 当以累计等待时间为目标来考察 MDVRP 的当前最优解时, 结果明显不如 MS-VND 算法迭代 200 次所获得的结果, 说明 Cum-MDVRP 与 MDVRP 差异较大, 以配送成本最小化为目标求解 MDVRP 的算法不适合直接求解 Cum-MDVRP. 同样的算例下, 当 N_V 减少时, 对应的累计等待时间有明显的增加, 说明 Cum-MDVRP 的可用车辆总数对累计等待时间影响较大.

从计算结果的质量看, MS-VND 的 10 轮计算的平均值较为接近最优值, 且 1-1、2-1、8-1 及 8-2 下 10 轮计算均获得相同的结果, 算法的鲁棒性较好. 从计算时间看, 除 p05 与 p15 算例的时间稍长外, 其他算例最优解的平均出现时间均在 100s 以内; p05 与 p15 算例由于有较多需求量较小的受灾点, 每条路径可容纳的受灾点数远远多于其他算例, 因而大大增加了局部搜索的计算量, 明显增加了计算时间. 此外, p15 算例由于受灾点个数及可用车辆总数明显增多, 问题规模扩大, 计算时间随之增加.

综合上述对算法的计算结果质量和计算时间的讨论, 本文设计的 MS-VND 算法可迅速有效地求解所提出的 Cum-MDVRP 模型.

4 结 论

本文针对应急物流中新出现的最小化所有受灾点的累计等待时间这一目标, 结合常见的多出救点救灾情形和 MDVRP, 提出了累计时间式多车场车辆路径问题 (Cum-MDVRP) 的数学模型, 并设计一种多起

始点变邻域下降法 (MS-VND) 进行快速求解. 对扩展标准算例的测试结果表明, Cum-MDVRP 与 MDVRP 存在较大差异, 本文的 MS-VND 算法可迅速有效求解本文提出的 Cum-MDVRP 模型. 此外, 本文所提出的模型和算法可更好地服务于其他以需求点为中心的需要快速服务的物流实践.

参考文献(References)

- [1] Campbell A M, Vandenbussche D, Hermann W. Routing for relief efforts[J]. *Transportation Science*, 2008, 42(2): 127-145.
- [2] Ngueveu S U, Prins C, Calvo R W. An effective memetic algorithm for the cumulative capacitated vehicle routing problem[J]. *Computers & Operations Research*, 2010, 37(11): 1877-1885.
- [3] Ribeiro G M, Laporte G. An adaptive large neighborhood search heuristic for the cumulative capacitated vehicle routing problem[J]. *Computers & Operations Research*, 2012, 39(3): 728-735.
- [4] Ke Liangjun, Feng Zuren. A two-phase metaheuristic for the cumulative capacitated vehicle routing problem[J]. *Computers & Operations Research*, 2013, 40(2): 633-638.
- [5] Ogryczak W. Inequality measures and equitable locations[J]. *Annals of Operations Research*, 2009, 167(1): 61-86.
- [6] Mirabi M, Fatemi Ghomi S M T, Jolai F. Efficient stochastic hybrid heuristics for the multi-depot vehicle

- routing problem[J]. *Robotics and Computer-Integrated Manufacturing*, 2010, 26(6): 564-569.
- [7] Lim A, Wang F. Multi-depot vehicle routing problem: A one-stage approach[J]. *IEEE Trans on Automation Science and Engineering*, 2005, 2(4): 397-402.
- [8] Mladenović N, Hansen P. Variable neighborhood search[J]. *Computers & Operations Research*, 1997, 24(11): 1097-1100.
- [9] Hansen P, Mladenović N, Brimberg J, et al. Variable neighborhood search[M]. New York: Spring Science + Business Media, 2010: 61-86.
- [10] Martí R, Moreno-Vega J M, Duarte A. Advanced multi-start methods[M]. New York: Spring Science+ Business Media, 2010: 265-281.
- [11] Martí R, Resende M G C, Ribeiro C C. Multi-start methods for combinatorial optimization[J]. *European J of Operational Research*, 2013, 226(1): 1-8.
- [12] Villegas J G, Prins C, Prodhon C, et al. GRASP/VND and multi-start evolutionary local search for the single truck and trailer routing problem with satellite depots[J]. *Engineering Applications of Artificial Intelligence*, 2010, 23(5): 780-794.
- [13] Salehipour A, Sörensen K, Goos P, et al. Efficient GRASP+VND and GRASP+VNS metaheuristics for the traveling repairman problem[J]. *4OR*, 2011, 9(2): 189-209.
- [14] Schittekat P, Kinable J, Sörensen K, et al. A metaheuristic for the school bus routing problem with bus stop selection[J]. *European J of Operational Research*, 2013, 229(2): 518-528.
- [15] Nguyen V P, Prins C, Prodhon C. Solving the two-echelon location routing problem by a GRASP reinforced by a learning process and path relinking[J]. *European J of Operational Research*, 2012, 216(1): 113-126.
- [16] Beasley J E. Route first-Cluster second methods for vehicle routing[J]. *Omega*, 1983, 11(4): 403-408.
- [17] Prins C. A simple and effective evolutionary algorithm for the vehicle routing problem[J]. *Computers & Operations Research*, 2004, 31(12): 1985-2002.
- [18] Bräysy O, Gendreau M. Vehicle routing problem with time windows, Part I: Route construction and local search algorithms[J]. *Transportation Science*, 2005, 39(1): 104-118.
- [19] Subramanian A, Drummond L M A, Bentes C, et al. A parallel heuristic for the vehicle routing problem with simultaneous pickup and delivery[J]. *Computers & Operations Research*, 2011, 37(11): 1899-1911.
- [20] Subramanian A, Uchoa E, Ochi L S. A hybrid algorithm for a class of vehicle routing problems[J]. *Computers & Operations Research*, 2013, 40(10): 2519-2531.

(责任编辑: 孙艺红)