

## 需求点随机的分批配送VRP模型与算法研究

石建力<sup>†</sup>, 张 锦

(西南交通大学 交通运输与物流学院, 成都 610031)

**摘 要:** 针对城市配送中需求点不确定的现象, 在分批配送车辆路径问题中引入随机需求点进行建模. 建立带修正的随机规划模型, 采用先验优化策略, 根据分批配送的特点, 在自适应大邻域搜索算法中引入改进的分割插入算子进行求解. 在调整的Solomon算例上进行的测试表明, 允许分批配送在大部分算例中的费用低于不允许分批配送的情形. 通过分析计算过程中各个算子权重变化, 确定性最差删除算子和随机删除算子在求解此类问题时表现较好; 贪婪插入算子、后悔插入算子表现较好; 而分割插入算子虽然权重较低, 但能对解产生质的影响.

**关键词:** 需求点随机; 分批配送; 车辆路径问题; 自适应大邻域搜索; 先验优化

中图分类号: F270; F224.3; C931

文献标志码: A

### Model and algorithm for split delivery vehicle routing problem with stochastic customers

SHI Jian-li<sup>†</sup>, ZHANG Jin

(School of Transportation and Logistics, Southwest Jiaotong University, Chengdu 610031, China)

**Abstract:** According to the uncertainty of the customers in city logistics, a stochastic programming model with recourse is formulated to solve the split delivery vehicle routing problem (VRP) with stochastic customers. Based on the priori optimization strategy, an adaptive large neighborhood search heuristic with modified split insertion operator is developed. Computation tests on the modified Solomon's instances are carried out. The results show that, in most of the instances, the cost of split delivery is lower than unsplit-delivery. In the computation, the deterministic worst removal operator and the random removal operator are the best removal operators, and the greedy insertion and the regret insertion are the best insertion operator. Though the weight of the split insertion operator is lower, it makes the solution different from the unsplit-delivery solution.

**Keywords:** stochastic customers; split delivery; vehicle routing problem; adaptive large neighborhood search; priori optimization

## 0 引 言

随着我国物流业不断发展, 城市配送越来越受到人们的重视. 城市中连锁超市、便利店、大型卖场等商品配送以及城市垃圾运输等涉及到很多不确定的因素, 比如需求点位置、需求点需求量、配送时间、服务时间等<sup>[1]</sup>. 本文关注需求点是否出现需求, 即需求点随机问题. 对于连锁超市、便利店、大型卖场等配送问题, 物流公司在为其配送的过程中, 超市的位置是固定的(确定的), 但并非每个超市每天都需要进行配送, 也就是在一次配送中, 并非每个需求点都产生需求, 此时需对配送路径进行合理规划和优化, 以提高配送效率, 降低配送费用. 针对此类问题, 本文将需

求点随机引入分批配送车辆路径问题(SDVRP), 建立需求点随机的分批配送车辆路径问题(SDVRPSC)模型, 并根据问题特点设计改进的自适应大邻域搜索算法进行求解.

SDVRP是由Dror等<sup>[2]</sup>于1989年正式提出的, 是经典车辆路径问题(VRP)的一类变型, 在报纸配送、食品配送、零售物品配送、应急物资车辆路径问题等实际运作中得到了广泛应用<sup>[3]</sup>. SDVRP允许需求点需求量大于车容量, 允许多辆车对同一需求点进行配送. 此时, 车辆对被分割需求点的配送量成为一个变量, 而不再是常量(VRP中).

近10年来, 人们对SDVRP的研究较多, 主要集中在

收稿日期: 2016-01-14; 修回日期: 2016-04-27.

作者简介: 石建力(1985—), 男, 博士生, 从事车辆路径优化、城市配送的研究; 张锦(1963—), 男, 教授, 博士生导师, 从事物流系统规划与优化等研究.

<sup>†</sup>通讯作者. E-mail: shjl20043528@163.com

在SDVRP扩展问题及求解算法上. 在SDVRP扩展问题方面, 主要有带时间窗的、集货与配送混合的、多车型的、随机的等<sup>[3]</sup>. SDVRP求解算法大多借鉴VRP算法并加以改进得来, 主要包括精确求解算法和启发式算法. 目前求解效果较好的启发式算法包括禁忌搜索算法(TS)<sup>[4-6]</sup>、文化基因算法(MA)<sup>[7]</sup>、聚类算法<sup>[8]</sup>、蚁群算法(ACO)<sup>[9-10]</sup>、迭代局部搜索算法(ILS)<sup>[11]</sup>等. 此外, 文献[12]对求解带时间窗SDVRP的局部搜索算子进行了测试和对比研究.

以往的SDVRP所研究的问题基本都是确定性问题, 需求点位置、需求点需求、行驶时间等都是确定的, 而有关随机分批配送车辆路径问题的研究成果则较少. Bouzaïene-Ayari等<sup>[13]</sup>对需求随机的SDVRP进行了研究, 证实允许分批配送能产生一定的节约量.

Lei等<sup>[14]</sup>对带时间窗的需求随机的SDVRP进行了研究, 建立了带修正的随机规划模型, 并利用成对车辆修正策略, 使用自适应大邻域搜索算法进行求解, 通过测试证实了最大节约量是在需求点平均需求刚刚超过车容量的一半, 并且需求点需求量波动不大(方差低)的情况下得到的.

虽然迄今尚未见到对需求点随机的SDVRP进行的研究, 但人们对需求点随机的VRP已有一定的研究. Waters<sup>[15]</sup>对VRPSC进行了实证分析; Gendreau等<sup>[16]</sup>提出用精确算法求解需求点和需求随机的车辆路径问题(VRPSDSC); Tang等<sup>[17]</sup>使用精确求解算法求解需求点随机的旅行商问题(TSPSC); Cheung等<sup>[18]</sup>使用基于抽样的启发式算法对两阶段需求点随机的车辆路径问题(VRPSC)进行了求解; Lei等<sup>[19]</sup>对需求点随机的分区路径问题进行了研究, 并使用自适应大邻域搜索算法进行求解. 更多的是将需求点随机与需求随机一起研究的文献<sup>[20]</sup>.

本文对SDVRPSC进行研究, 建立带修正的随机规划模型, 并使用改进的自适应大邻域搜索算法(ALNS)进行求解.

## 1 问题描述及模型建立

SDVRPSC定义在无向图 $G = (V, E)$ 上. 其中:  $V = \{0, 1, \dots, N\}$ 代表节点集合, 0代表车场, 其余节点代表需求点. 需求点位置确定, 但部分需求点以一定的概率产生需求, 本文称这些需求点为潜在需求点. 各个潜在需求点是否产生需求是相互独立的, 产生需求的概率用 $p_i$ 表示. 一旦产生需求, 需求量就是确定的, 记为 $d_i, d_i > 0$ .  $E = \{(i, j), i, j \in V, i \neq j\}$ 代表边集合, 每条边 $(i, j)(\forall i, j \in V)$ 的行驶费用记为 $c_{ij}, c_{ij} > 0$ , 且满足三角不等式. 车辆数不限制且所

有车辆容量相同, 记为 $Q$ . 在配送过程中, 允许分批配送, 每个需求点可由多辆车进行配送.

本文假定至少有一个点不产生需求, 且每个需求点需求均为整数单位. 所有需求必须全部满足; 允许分批配送, 每个需求点可接受多于一辆车的配送服务, 但每辆车最多对同一个需求点配送一次, 配送量不超过车容量.

符号说明:

$a$ 为车辆固定费用, 使用每辆车的固定费用;

$b$ 为车辆可变费用, 单位路径行驶费用;

$K = \{1, 2, \dots, m\}$ 表示车辆集, 车场内有 $m$ 辆车;

$\Omega$ 表示所有需求点出现与否的组合情况的集合;

$\xi$ 为包含所有需求点的伯努利随机变量的向量, 对 $\forall \omega \in \Omega, \xi(\omega)^T = [\xi_1(\omega), \dots, \xi_n(\omega)]$ , 若需求点 $i$ 产生需求, 则 $\xi_i(\omega) = 1$ , 否则 $\xi_i(\omega) = 0$ ;

解 $R = \{r_1, \dots, r_m\}$ 的路径期望费用为

$$E_{\xi}[F(R, \xi(\omega))] = \sum_{k=1}^m E_{\xi}[F(r_k, \xi(\omega))],$$

其中 $E_{\xi}[F(R, \xi(\omega))]$ 表示路径 $r_k$ 上的期望费用.

决策变量有两组, 分别为 $x_{ij}^k$ 和 $y_{ik}$ . 令 $x_{ij}^k$ 为0-1变量, 当车辆 $k$ 从需求点 $i$ 直接行驶到需求点 $j$ 时,  $x_{ij}^k = 1$ , 否则 $x_{ij}^k = 0$ ;  $y_{ik}$ 为整型变量, 表示车辆 $k$ 对需求点 $i$ 的配送量.

SDVRPSC模型如下:

$$\min_R a|R| + bE_{\xi}[F(R, \xi(\omega))]. \quad (1)$$

s.t.

$$\sum_{i=0}^n x_{0i} + \sum_{i=1}^n x_{i0} = 2m; \quad (2)$$

$$\sum_{i=0}^n x_{ip}^k - \sum_{j=0}^n x_{pj}^k = 0, \forall p \in V, \forall k \in K; \quad (3)$$

$$\sum_{i \in S} \sum_{j \in S} x_{ij}^k \leq |S| - 1, \forall S \subseteq V \setminus \{0\}, \forall k \in K; \quad (4)$$

$$y_{ik} \cdot \xi_i(\omega) \leq d_i \cdot \sum_{j=0}^n x_{ij}^k \cdot \xi_i(\omega),$$

$$\forall i \in V \setminus \{0\}, \forall k \in K, \forall \omega \in \Omega; \quad (5)$$

$$\sum_{k=1}^m y_{ik} \cdot \xi_i(\omega) = d_i, \forall i \in V \setminus \{0\}, \forall \omega \in \Omega; \quad (6)$$

$$\sum_{i=1}^n y_{ik} \cdot \xi_i(\omega) \leq Q, \forall k \in K, \forall \omega \in \Omega; \quad (7)$$

$$\sum_{i=0}^n \sum_{k=1}^m x_{ij}^k \cdot \xi_i(\omega) \geq \xi_j(\omega), \forall j \in V \setminus \{0\}, \forall \omega \in \Omega; \quad (8)$$

$$x_{ij}^k \in \{0, 1\}, \forall i \in V, \forall j \in V, \forall k \in K; \quad (9)$$

$$y_{ik} \in N, \forall i \in V, \forall k \in K. \quad (10)$$

式(1)为目标函数,期望费用由两部分构成,车辆使用费用和与行驶路径长度有关的期望费用; $a$ 表示使用每辆车的固定费用; $|R|$ 表示解中路径数,即使用的车辆数; $b$ 表示单位路径行驶费用; $E\xi[F(R, \xi(\omega))]$ 表示解 $R$ 的期望形式距离.

式(2)表示使用的总车辆数为 $m$ ,每辆车都从车场出发,并最终回到车场.其中: $\sum_{i=0}^n x_{0i}$ 表示从车场出发的车辆数, $\sum_{i=1}^n x_{i0}$ 表示回到车场的车辆数.

式(3)为流平衡约束,表示对某一需求点而言,进入的车辆数与从此需求点离开的车辆数相等,即每个需求点的出度与入度相等.其中: $\sum_{i=0}^n x_{ip}^k$ 表示车辆 $k$ 到需求点 $p$ 的次数, $\sum_{j=0}^n x_{pj}^k$ 表示车辆 $k$ 从需求点 $p$ 出发到下一个需求点的次数,这两个都只可能是1或0,且两个一定是相等的,这是由于每辆车最多对同一个需求点配送1次.

式(4)为子路径消减约束,排除所有不可行路径,包括出发点和最终到达点不是车场的所有路径, $\sum_{i \in S} \sum_{j \in S} x_{ij}^k$ 表示车辆 $k$ 在子集 $S$ 中经过的边的数量,所有车辆所经过的边之和应该小于需求点子集合 $S$ 的秩(集合中元素数量)减1.

式(5)表示只对出现需求的点进行配送,且经过需求点 $i$ 的车辆才能对其进行配送,配送量不超过需求点需求量.其中: $y_{ik} \cdot \xi_i(\omega)$ 表示需求点 $i$ 出现需求时( $\xi_i(\omega) = 1$ ),才出现某一车辆 $k$ 对它的配送量 $y_{ik}$ ; $d_i \cdot \sum_{j=0}^n x_{ij}^k \cdot \xi_i(\omega)$ 表示需求点 $i$ 出现需求时,某一车辆 $k$ 才经过需求点 $i$ ( $\sum_{j=0}^n x_{ij}^k \cdot \xi_i(\omega) = 1$ ),且对其配送量最多为其需求量 $d_i$ .

式(6)表示出现需求的需求点需求被完全满足,其中 $\sum_{k=1}^m y_{ik} \cdot \xi_i(\omega)$ 表示在每种需求点组合 $\omega$ (一部分出现,一部分不出现)情况下,需求点 $i$ 的需求量恰被完全满足.

式(7)表示每辆车的配送量不超过其容量,其中 $\sum_{i=1}^n y_{ik} \cdot \xi_i(\omega)$ 表示在每种需求点组合 $\omega$ (一部分出现,一部分不出现)情况下,车辆 $k$ 每次的载重量不大于车容量.

式(8)表示出现需求的需求点至少接受一次配

送,其中 $\sum_{i=0}^n \sum_{k=1}^m x_{ij}^k \cdot \xi_i(\omega)$ 表示在每种需求点组合 $\omega$ (一部分出现,一部分不出现)情况下,需求点 $i$ 接收到的配送次数,当 $\xi_i(\omega) = 1$ 时,配送次数至少为1,当 $\xi_i(\omega) = 0$ 时, $\sum_{i=0}^n \sum_{k=1}^m x_{ij}^k \cdot \xi_i(\omega) = 0$ .

式(9)和(10)为决策变量.

期望费用 $E_\xi[F(r_k, \xi(\omega))]$ 可根据PTSP中计算路径期望费用的方法进行计算.令路径 $r_k$ 经过的需求点数量为 $N_{r_k}$ ,路径 $r_k$ 有 $2^{N_{r_k}}$ 种不同的组合情况(需求点是否出现需求),则

$$E_\xi[F(r_k, \xi(\omega))] = \sum_{i=1}^{2^{N_{r_k}}} p(r_{ki}) \cdot L[r_{ki}(r_k)]. \quad (11)$$

其中: $p(r_{ki})$ 为路径 $r_k$ 以 $r_{ki}$ 方式实现的概率, $L[r_{ki}(r_k)]$ 表示路径 $r_k$ 第 $r_{ki}$ 种组合情况的路径长度.按式(11)计算期望费用效率较低, Branke等<sup>[21]</sup>在2004年提出的近似估计期望费用的公式如下:

$$E_{\xi, \lambda}^{AP}[F(r_k, \xi(\omega))] = \sum_{i=0}^{N_{r_k}} \min(N_{r_k} + 1, i + \lambda) \sum_{j=1}^{N_{r_k} - i} \{c_{r_k(i)r_k(j)} p_{r_k(i)} p_{r_k(j)}\} \times \prod_{q=i+1}^{j-1} (1 - p_{r_k(q)}). \quad (12)$$

其中: $c_{r_k(i)r_k(j)}$ 为点 $r_k(i)$ 到点 $r_k(j)$ 的费用; $p_{r_k(i)}$ 表示 $r_k(i)$ 出现需求的概率.为了避免估计值偏离过大,本文采用精确计算方法与近似估计算法相结合的方法进行计算.

## 2 问题求解及算法设计

SDVRPSC是一个随机优化问题,最常用的求解方法是由Bertsimas等<sup>[22]</sup>提出的先验优化,此方法已被广泛用于解决随机VRP问题<sup>[14, 16, 23-24]</sup>.先验优化中,第1阶段,不考虑随机因素,计算出一个先验路径;第2阶段,考虑随机因素的影响,对先验路径进行修正,本文中对于不产生需求的潜在需求点采用的策略是直接跳过.

图1为需求点分布和路径结构示意图,图中共19个需求点,其中12个需求点为确定的需求点,7个为随机的需求点,最终有5个随机需求点产生需求,2个需求点不产生需求,不产生需求的点被直接跳过(如图1(c)所示).

Dror等<sup>[25]</sup>得到如下确定性SDVRP的性质:如果SDVRPSC存在可行解,且距离满足三角不等式,则在该问题优化的任意2条路线中最多只能存在1个共同点.仿照确定性SDVRP性质,本文在求解过程中

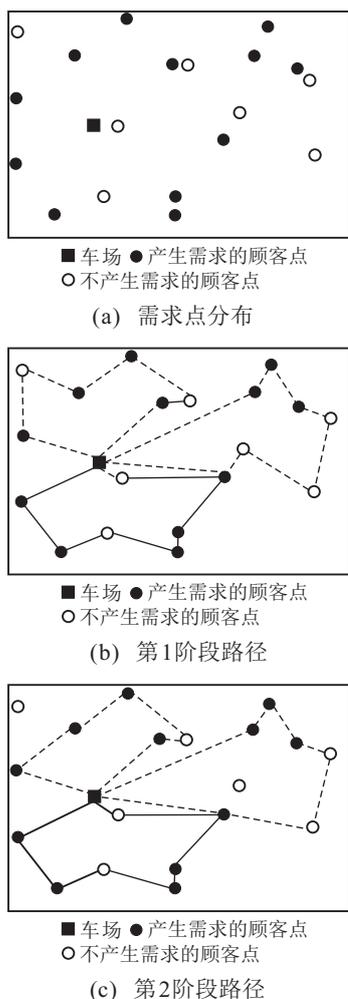


图1 需求点分布及第1阶段路径、第2阶段路径示意图

假定任意两条路径中最多有一个相同的需求点(此时必定是被分割的需求点),以降低解空间规模,提高搜索效率。

ALNS是由Shaw<sup>[26]</sup>提出的LNS的一种扩展,由Ropke等<sup>[27]</sup>将其应用于求解车辆路径问题。其基本思想与LNS相同,通过破坏现有解的一部分,以不同的方式重新构建新的解,以得到更好的解。与LNS中使用一个删除算子和插入算子不同,ALNS使用多个删除算子和插入算子,在搜索过程中根据各个算子的权重动态选取删除算子和插入算子。每次迭代使用选中的删除算子删除 $q$ 个点,然后使用插入算子将删除的点重新插入到路径中。经试验, $q$ 取区间 $[[0.1n], [0.2n]]$ 中的随机数时较好。

### 2.1 初始解构造

针对分批配送车辆路径中需求点可被分割的特点,设计改进的插入算法构造初始解,任一需求点都可能被分割多次,根据车辆剩余容量判断需求点是否被分割以及被分割的次数。构造算法可描述如下。

**算法1** 初始解构造算法。

1) 将所有需求点按期望需求量从小到大排列。

2) 按顺序将所有需求点依次插入到费用增加最小的路径中,插入时只考虑确定的行驶距离;若现有路径没有可行的插入,则寻找仍有剩余容量的车辆,将需求点按增加路径长度最小的原则分割插入到多条路径的合适位置,否则构建一条新的路径,直到所有需求点都被插入到路径中为止。

### 2.2 删除算子

文中使用4个删除算子:随机删除、相似性删除、确定性最差删除和期望最差删除,每个算子都根据一定的规则删除 $q$ 个需求点。特别地,若被删除需求点 $i$ 是被分割的需求点,则将所有路径中的需求点 $i$ 全部删除。

1) 随机删除(RR)。此算子随机选择 $q$ 个需求点,从相应的路径中删除,以增加搜索的广度。

2) 相似性删除(SR)。基本思想是删除“相似”的两个需求点。两个节点的相似度可定义为

$$S(i, j) = \frac{1}{c_{ij}/c_i^{\max} + \tau_{ij}}$$

其中: $c_i^{\max}$ 表示所有被删除的点与节点 $i$ 之间的最大行驶距离;若 $i, j$ 在不同路径中则 $\tau_{ij}$ 等于1,否则等于0。此算子首先随机选择一个需求点加入到被删除点集合 $R$ 中,然后从未被删除的需求点中选取与已删除集合中所有需求点相似度最高的点进行删除,直到删除集合中有 $q$ 个需求点时结束。

3) 确定性最差删除(DWR)。此算子删除现有解中插入到“错误”位置的点,只考虑确定费用。计算每条路径中每个需求点的费用差,然后选择每条路径中最大费用差对应的需求点。分两种情况考虑: i) 路径数大于 $q$ ,将所有路径的最大费用差按降序排列,前 $q$ 条路径最大费用差对应的需求点被删除; ii) 路径数小于 $q$ ,循环删除各条路径中具有最大费用的需求点,直到删除 $q$ 个需求点为止。每个循环都要重新计算每条路径最大费用差。

4) 期望最差删除(EWR)。此删除算子关注插入到“错误”位置的关键点。与确定性最差删除不同的是,此算子根据路径上每个点的期望费用进行删除。具体步骤与确定性最差删除相同,但此算子跳过只有一个节点的路径。

### 2.3 插入算子

文中使用3个插入算子,即贪婪插入、后悔插入和分割插入,每个算子都根据相应的规则将删除 $q$ 个需求点重新插入到路径中。特别地,只有分割插入能将需求点分割,并插入到多条路径中。

1) 贪婪插入(GI)。此算子将 $q$ 个被删除点依次插

入到费用增加最少的路径中. 为了节省计算量, 此算子只考虑确定的费用.

2) 后悔插入(RI). 后悔插入算子结合先前信息选择需要插入的点. 算子首先计算每个需求点在每条路径中的最小增加费用  $\Delta\phi(r_k)$ , 然后根据如下方式确定被插入的点:

$$v := \arg \max_{v \in M} \left\{ \frac{1}{z} \sum_{j=1}^z (\Delta_v \phi(r_j) - \Delta\phi_v(r_*)) \right\}.$$

其中:  $M$  为删除点集合;  $\Delta\phi_v(r_*)$  表示将需求点  $v$  插入到所有路径中的最低增加费用, 而且是在  $v$  插入到路径  $r_*$  时得到的;  $\Delta_v \phi(r_j)$  表示将需求点  $v$  插入到除  $r_*$  之外的所有路径中相应的最低增加费用;  $z$  表示需求点  $v$  可插入的路径数. 选定插入的需求点  $v$  后, 将其插入到路径  $r_*$  中的最优位置. 若被选中的需求点  $v$  不能插入到现有路径中, 则构造新路径.

3) 分割插入(SI). 此插入算子是由 Lei 等<sup>[14]</sup> 在研究需求随机的 VRPSD 时提出的成对路径分割插入算子改进而来. 文献[14]中的分割比例由两条路径的期望需求决定, 会导致产生不必要的新路径. 本文不对单个需求点分割数进行限定, 算子首先确定不存在分割点且剩余容量不为 0 的路径(即可插入路径集), 然后将插入点插入到增加期望费用最少的路径中, 配送量尽量大; 进而依次插入到期望费用最小的可插入路径中. 若不存在可插入路径, 则构建新的路径.

## 2.4 自适应选择规则

ALNS 在每次迭代中分别选择一个合适的删除算子和插入算子. 一般地, 每个算子  $o$  都有与之相应的权重  $w_o$ ,  $o = 1, 2, \dots, h$ , 每个算子的初始权重都是 1. 在每次迭代中,  $h$  个算子中算子  $o$  被选中的概率为  $w_o / \sum_{i=1}^h w_i$ . 各个算子的权重每  $\delta$  次迭代(称为一个阶段)更新一次, 在第  $N + 1$  个阶段时, 算子  $o$  的权重  $w_{o,N+1}$  计算方法如下:

$$w_{o,N+1} = w_{o,N}(1 - \chi) + \chi\pi_{oN}/\varepsilon_{oN}.$$

其中:  $\varepsilon_{oN}$ ,  $\pi_{oN}$  分别代表算子  $o$  在第  $N$  阶段被选中的次数和所得的分数;  $\chi \in [0, 1]$  为常数; 文中选取  $\chi = 0.1$ .

算子  $o$  在第  $N$  阶段的得分按下式计算:

$$\pi_{o,N}^{\gamma+1} = \pi_{o,N} + \begin{cases} \sigma_1, & \text{如果得到新的最优解;} \\ \sigma_2, & \text{如果得到比现有解更优的解;} \\ \sigma_3, & \text{如果得到比现有解差但可接受的解.} \end{cases}$$

其中:  $\gamma = 1, 2, \dots, \delta - 1$ , 表示第  $N$  阶段中的迭代

次数, 每个阶段第一次迭代时所有算子得分重置为 0, 即  $\pi_{oN}^1 = 0, \forall o, N$ ;  $\sigma_1, \sigma_2, \sigma_3$  为常数, 表示算子在相应情况下的得分,  $\sigma_1 > \sigma_2 > \sigma_3$ , 这里取  $\sigma_1 = 30, \sigma_2 = 10, \sigma_3 = 6$ .

## 2.5 迭代结束条件

解的接受条件基于 Dueck<sup>[28]</sup> 的记录旅行算法(RRT). 令  $f^*$  表示最优的现有解的目标函数值, 称为一条记录;  $\beta > 0$  是一个参数, 称为离差, 本文中令  $\beta = 0.01f^*$ . 若  $R$  表示一个解,  $R'$  表示产生于  $R$  邻域的一个新的解, 则为节约运算时间, 新的解产生后首先计算新解目标函数的估算值(式(12)), 计算出  $f_{R'}^b$ . 若  $f_{R'}^b < f^*$ , 则计算新解目标函数值的精确值(式(11)) $f_{R'}$ ; 如果  $f_{R'} < f^* + \beta$ , 则解  $R'$  被接受; 若  $f_{R'} < f^*$ , 解  $R'$  优于解  $R$ , 则更新记录. 算法在迭代一定次数或者一定次数未产生更优的解时结束. 文中, 这两个参数分别取为 50 000 和 800.

## 3 算例分析

本程序使用 Matlab(2012b) 编写, 运行平台内存 4 GB, 显卡 2.5 GHz.

### 3.1 算例设计及生成

Solomon<sup>[29]</sup> 提出的 VRP 算例是最经典的算例集, 很多 VRP 文章都将 Solomon<sup>[29]</sup> 中的算例作为基本算例. 本文算例由 Solomon<sup>[29]</sup> 提出的经典算例调整而来, 通过对需求点分布、潜在需求点占总需求点数量的比例等进行调整, 生成 48 个算例进行测试.

1) 需求量及位置. 节点(包括停车场和需求点)的坐标和需求点需求量与 Solomon 的例子中都是相同的, 每个需求点以一定的概率产生需求.

2) 需求点分布及数量. 考虑 Solomon 的 6 类算例: 随机分布(R), 第 2 类随机分布(R2), 第 1 类集中分布(C1), 第 2 类集中分布(C2), 第 1 类随机集中分布(RC1) 和第 2 类随机集中分布(RC2). 其中 R1 和 R2、RC1 和 RC2 坐标分别相同, 因此本文中考虑 4 类不同算例: R, C1, C2, RC, 并分别考虑各类算例的规模分别为 25、50、100 的情况.

3) 潜在需求点占需求点总数比例. 仿照孔媛<sup>[24]</sup> 对不同程度满意度进行试验, 本文考虑潜在需求点(出现需求点概率小于 1 的需求点)的数量分别占总节点数量的 25%、50%、75% 和 100% 的情形, 随机选取相应数量的潜在需求点, 并随机初始化潜在需求点的出现概率.

4) 车型. 本文考虑单一车型, 所有车容量均取 95. 车辆使用费用(固定费用)系数取为 100, 行驶距离费用参数取 1.

### 3.2 允许分批配送结果分析

首先对允许分批配送的结果进行分析,并对允许分批与不允许分批的结果进行对比分析.所有结果中  $\text{propotion}$  表示潜在需求点数量占总需求点数量的比例;  $\text{cost}$  表示费用,也就是最优解目标函数值;  $n$  表示问题规模,即需求点数量;  $\text{number of customers}$  表示最优解中配送点的数量,这里的配送点是每条路径上的需求点数量的总和,被分割的需求点重复计算;  $\text{weight}$  表示算子的权重,  $\text{weight of RO}$  表示删除算子的权重,  $\text{weight of IO}$  表示插入算子的权重;  $\text{cars}$  表示最优解中使用的车辆数.

1) 潜在需求点比例越大,期望费用越大.

C1、C2、R、RC 四类算例在不同规模下,潜在需求点比例与期望费用关系如表1所示.

表1 不同类型算例潜在需求点与最优值

算例类型	潜在需求点比例/%	算例规模		
		25	50	100
C1	25	750.8	1 557.2	3 510.7
	50	765.9	1 564.9	3 495.7
	75	799.8	1 582.2	3 572.7
	100	804.8	1 612.9	3 601.8
C2	25	782	1 641.6	3 520.7
	50	808.5	1 629.5	3 530.2
	75	828.5	1 683.1	3 613.3
	100	856.6	1 692.9	3 641.7
R	25	733.4	1 420.3	2 728.5
	50	744.7	1 434.5	2 749.1
	75	742.1	1 454.1	2 783.3
	100	766.1	1 483.2	2 809.3
RC	25	1 031.1	2 010.7	3 514.6
	50	1 051.1	2 014.2	3 553.3
	75	1 075.2	2 072	3 583.3
	100	1 108	2 120	3 625.6

从表1中可以看出,潜在需求点比例越大,期望费用越大,这是由于潜在需求点比例变大,导致路径不确定性增加,期望费用增大.规模为25和50的算例中,期望费用增加的幅度最大的达3%以上;规模为100的算例中,期望费用增加幅度最大的超过2%.虽然在一次配送中增加的幅度不大,但是通过一定时间的积累,物流企业配送成本增加极大,因此,需求点稳定性对降低物流企业配送成本而言极为重要.

2) 配送点数量与潜在需求点所占比例无关,与需求点分布有关.

需要说明的是,本文中各个算例的平均需求水平基本相同,除了R类算例,平均需求占车容量的比例均在17%~22%之间,如表2所示.在分析配送点与潜在需求点所占比例的关系、配送点与需求点分布的关系时,可不考虑平均需求对两者关系的影响.

表2 不同类型、不同规模算例平均需求 %

算例类型	算例规模		
	25	50	100
C1	18.4	17.2	18.1
C2	18.4	17.2	18.1
R	13.28	14.42	14.58
RC	21.6	19.4	17.24

将C1、C2、R、RC 四类算例中规模相同、潜在需求点所占比例相同的最优解中配送点的数量相加,然后求平均,分析配送点数量与潜在需求点所占比例的关系,如表3所示.

表3 不同潜在需求点比例和规模的算例配送点数量

潜在需求点比例/%	算例规模		
	25	50	100
25	25.75	50	100.5
50	25.5	50	100.5
75	25.5	50	100.5
100	25.5	50	100.5

从表3可以看出,除了规模为25的算例,最优解中配送点数量在潜在需求点所占比例为25%时与其他情况下相差0.25,其他规模中潜在需求点比例变化时,最优解中需求点总数保持不变,需求点分割数与潜在需求点规模无关.

将C1、C2、R、RC 四类算例中相同规模算例的最优解中配送点数量相加,然后求平均,分析配送点数量与需求点分布之间的关系,如表4所示.

表4 不同类型及规模的算例配送点数量

算例类型	算例规模		
	25	50	100
C1	26	50	101
C2	26	50	101
R	25	50	100
RC	25	50	100

从表4可以看出:规模为25和100的集中分布(C1、C2类)的算例最优解中平均需求点数量大于需求点规模,而随机分布(R类)和随机集中分布(RC类)的算例最优解中平均需求点数量等于需求点规模;需求点集中分布时更容易被分割.

3) 算子权重变化.

将每个算例(共12个算例)10次计算的算子权重进行平均,从不同角度分析各个算子的权重变化及其在计算过程中的重要性.

各个删除算子在不同规模和不同潜在需求点所占比例算例中的权重,以及各个算子在所有算例中权重的平均值如图2所示,其中  $\text{average}$  表示所有算例中删除算子权重的平均值.

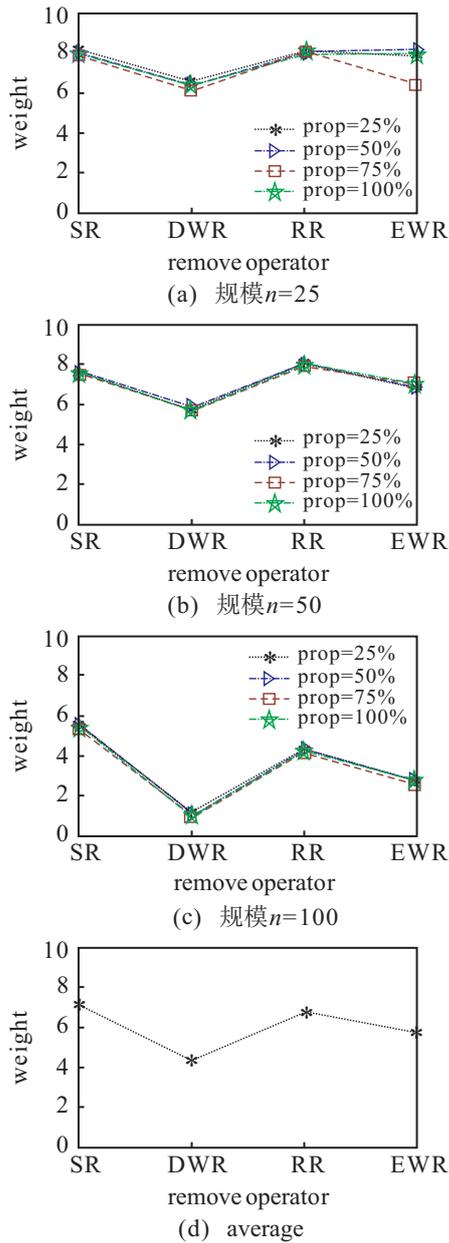


图2 不同规模算例删除算子权重及平均权重

从图2可以看出,规模为25和50的算例中,各个删除算子平均权重相差不大,所有算子平均权重都在6以上.从图2的4幅图中可以看出,相似性删除和随机删除平均权重较大,在计算过程中起主要作用.

同样地,各个插入算子在不同规模和不同潜在需求点所占比例算例中的权重,以及各个算子在所有算例中权重的平均值如图3所示,其中average表示所有算例中插入算子权重的平均值.

从图3可以看出:规模25的算例中,贪婪插入和分割插入平均权重较大,在计算中起主要作用;规模为50的算例中,贪婪插入和后悔插入平均权重较大,在计算中起主要作用,分割插入算子平均权重较小,与计算结果中无分割点相对应;规模为100的算例中,后悔插入算子权重最大,起主要作用,从被分割的需

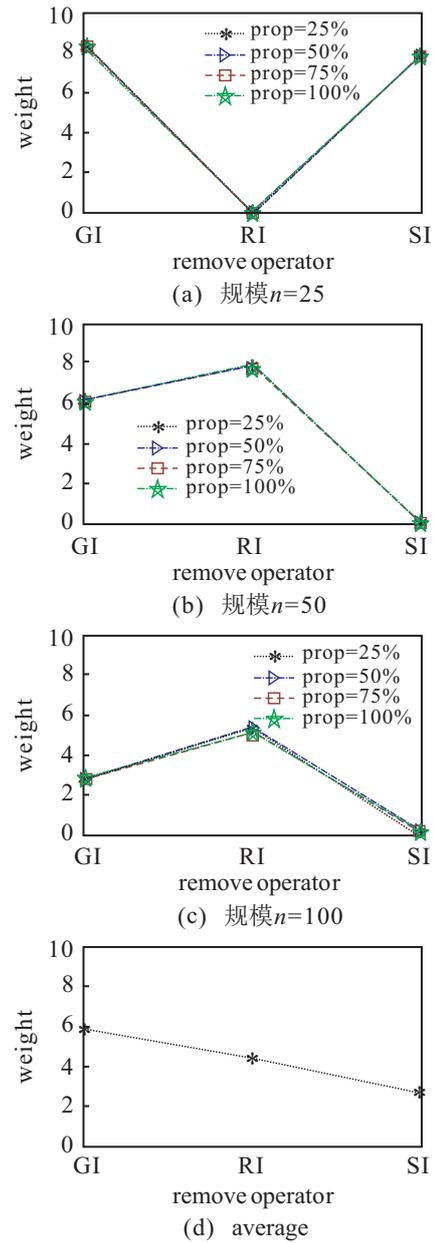


图3 算例规模对插入算子权重及平均权重影响

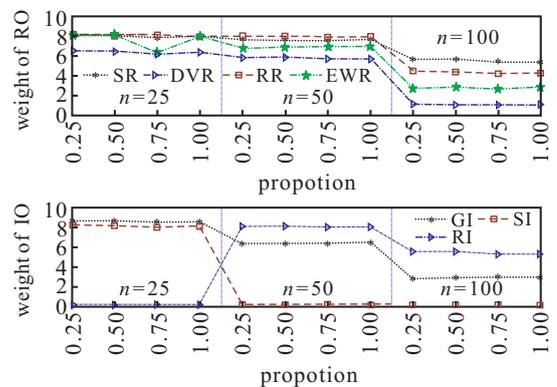


图4 潜在需求点所占比例对算子权重影响

求点数量上看,虽然分割插入平均权重较小,但仍起较大作用.从插入算子平均权重上看,贪婪插入、后悔插入、分割插入3个算子的平均权重呈现递减趋势,贪婪插入起主要作用.

将各个删除算子、插入算子在不同规模、不同潜在需求点所占比例下权重变化绘制成如图4所示的曲线,分析其在不同规模、不同潜在需求点所占比例下的变化。

从图4可以看出:规模为25和50的算例中,所有删除算子都比其在规模为100的算例中权重大;规模为25和50的算例中随机删除权重与相似性删除权重相差不大,且所有删除算子权重相差不大;在规模为100的算例中相似性删除权重最大,且4个删除算子权重分层明显.相似性删除在所有算例中表现最好,特别是在较大规模( $n = 100$ )算例中,明显优于其他算子;而每个删除算子在相同规模下,潜在需求点所占比例不同并不影响算子权重大小。

从图4还可以看出:规模为25的算例中,贪婪插入和分割插入权重较大,贪婪插入权重最大;在规模为50和100的算例中,后悔插入和贪婪插入算子权重较大,后悔插入算子权重最大,分割插入权重几乎为0.综合起来看,贪婪插入算子表现最好,优于后悔插入算子;而分割插入算子虽然权重较小,但由于其分割需求点的特性,能起到改变解的结构和特征的作用,使配送总费用更低。

### 3.3 允许分批配送与不允许分批配送结果对比分析

调整第2节中算法,去掉分批插入算子,形成不允许分批配送的自适应大邻域搜索算法,并对第3.1节中的算例进行测算,每个算例同样测算10次,比较允许分批配送与不允许分批配送在费用、求解时间和车辆数的差异。

#### 1) 费用对比.

将允许分批配送时与不允许分批配送时的最优解的总费用和多次求解的平均费用绘制成如图5和

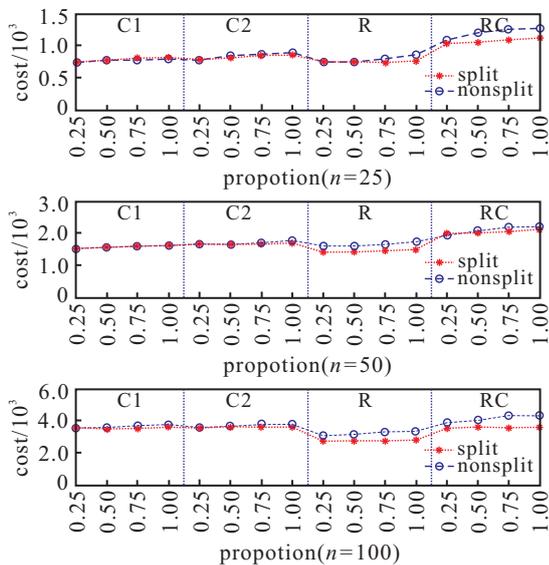


图5 总费用对比

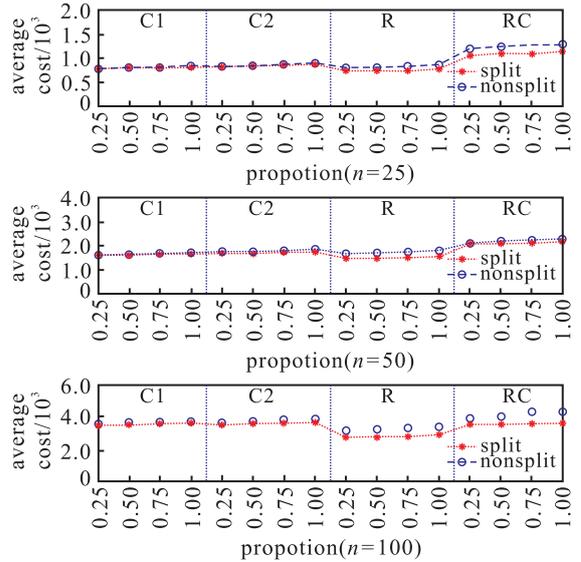


图6 平均费用对比

图6所示的曲线.从图5和图6可以看出:大部分算例允许分批配送的费用和平均费用均不高于不允许分批配送的总费用,特别是R类算例和RC类算例,允许分批配送时的费用和平均费用节约较明显,平均在8%以上;在随机分布时,允许分批配送表现较好.同时,规模为100的所有算例允许分批配送时总费用和平均费用节约量比规模为25和50的算例节约量更明显,平均在10%以上;分批配送在较大规模算例中表现更好。

#### 2) 运行时间对比.

将允许分批配送时与不允许分批配送时,得到最优解的求解时间和多次求解的平均求解时间绘制成如图7和图8所示的曲线.从图7和图8可以看出:规模为25的所有算例,允许分批配送时的求解时间和平均求解时间都超过不允许分批配送时的时间.规模为50的算例,RC类算例允许分批配送时的求解时

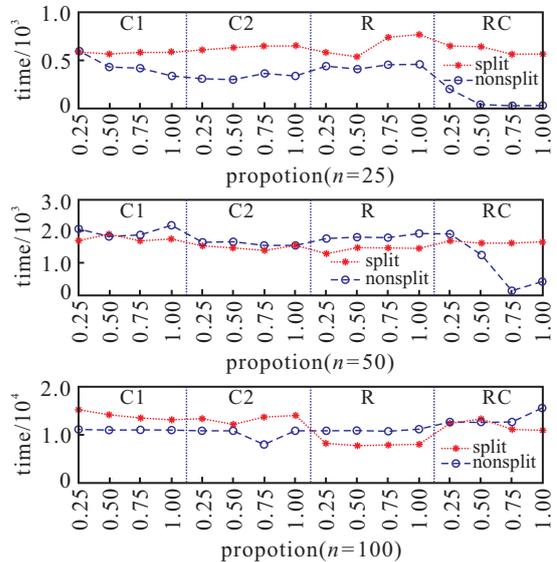


图7 最优解求解时间对比

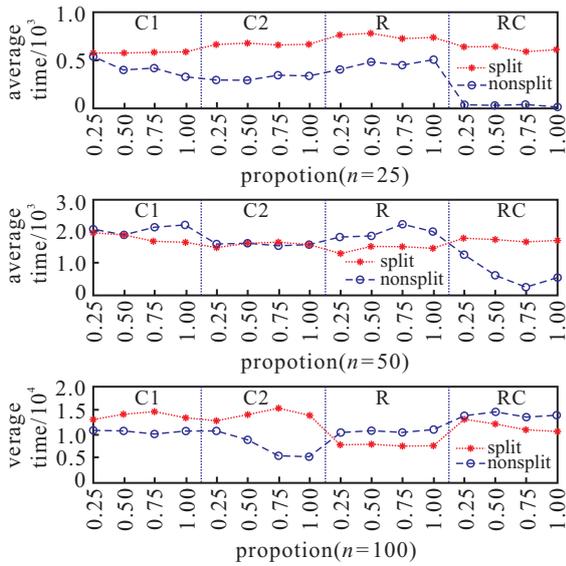


图8 平均求解时间对比

间和平均求解时间都高于不允许分批配送时的时间; C1、C2和R类算例允许分批配送时的求解时间和平均求解时间比不允许分批配送时的少. 规模为100的算例, C1和C2类算例允许分批配送时的求解时间和平均求解时间高于不允许分批配送时的时间; R类算例和RC类算例则相反.

3) 车辆数对比.

将允许分批配送时与不允许分批配送时所使用的车辆数绘制成如图9所示的曲线. 从图9可以看出: 规模为25的所有算例, 允许分批配送时和不允许分批配送时所使用的车辆数相同. 规模为50的算例, C1、C2和R类算例, 两种情形使用的车辆数相同; 而RC类算例中, 允许分批配送时所使用的车辆数高于不允许分批配送时所使用的车辆数. 规模为100的算例中, 除了R类算例所使用的车辆数是允许分批配送时低于不允许分批配送时所使用的车辆数外, C1、C2

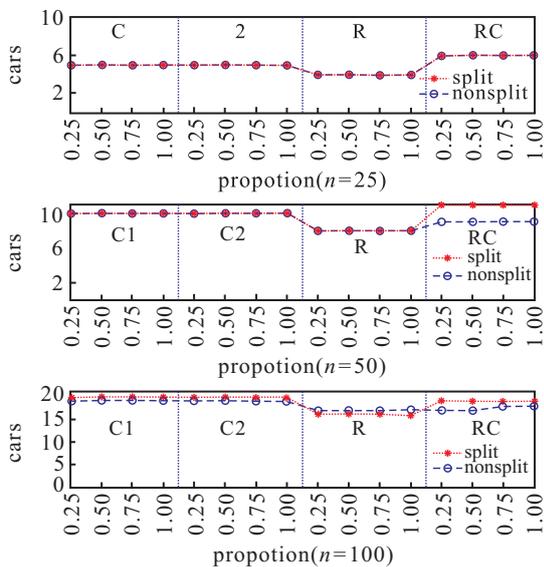


图9 车辆数对比

和RC类算例, 允许分批配送时所使用的车辆数均多于不允许分批配送时所使用的车辆数.

4) 算子权重.

不允许分批配送时算例求解过程中算子权重如表5所示. 从表5可以看出: 在不允许分批配送时, 删除算子中确定性删除算子权重较大, 在计算过程中表现较好; 贪婪插入和后悔插入权重大小相差较小, 而贪婪插入算子权重略大. 与允许分批配送时不同, 随机删除算子权重和相似性删除权重均小于确定性删除算子权重, 表现不如确定性删除算子.

表5 不允许分批配送时算例求解过程中算子权重

算子	相似性删除	确定性删除	随机删除	期望最差删除	贪婪插入	后悔插入
权重	0.933	1.320	0.934	0.951	1.090	0.992

4 结 论

本文通过对城市配送实际问题的分析, 建立了需求点随机的分批配送车辆路径问题模型, 并使用改进的自适应大邻域搜索算法进行求解. 在调整的Solomon算例上进行了测试, 结果表明: 潜在需求点的比例对车辆使用数、被分割的需求点数等几乎不产生影响; 允许分批配送时比不允许分批配送时得到的最优解更优; 相似性删除算子、随机删除算子、贪婪插入算子、后悔插入算子在计算过程中表现较好.

在未来的研究工作中, 一方面可以对不同的删除和插入算子作更深入的研究, 特别是对分割插入算子的改进, 以提高算子和算法的效率, 或研究使用更有效的启发式算法, 甚至是精确求解算法来求解随机SDVRP; 另一方面, 可以对更加复杂或其他因素随机的分批配送车辆路径问题进行研究, 比如对需求随机、服务时间随机、行驶时间随机等情形进行研究.

参考文献(References)

- [1] Cattaruzza D, Absi N, Feillet D, et al. Vehicle routing problems for city logistics[J]. EURO J on Trans and Logistics, 2015.
- [2] Dror M, Trudeau P. Savings by split delivery routing[J]. Transportation Science, 1989, 23(2): 141-145.
- [3] Archetti C, Speranza M G. Vehicle routing problems with split deliveries[J]. Int Trans in Operational Research, 2012, 19(1/2): 3-22.
- [4] Ho S C, Haugland D. A tabu search heuristic for the vehicle routing problem with time windows and split deliveries[J]. Computers & Operations Research, 2004, 31(12): 1947-1964.
- [5] Berbotto L, García S, Nogales F J. A randomized granular tabu search heuristic for the split delivery

- vehicle routing problem[J]. *Annals of Operations Research*, 2014, 222(1): 153-173.
- [6] Bolduc M, Laporte G, Renaud J, et al. A tabu search heuristic for the split delivery vehicle routing problem with production and demand calendars[J]. *European J of Operational Research*, 2010, 202(1): 122-130.
- [7] Boudia M, Prins C, Reghioui M. An effective memetic algorithm with population management for the split delivery vehicle routing problem[J]. *Hybrid Metaheuristics*, 2007, 4771(1): 16-30.
- [8] 刘旺盛, 杨帆, 李茂青, 等. 需求可拆分车辆路径问题的聚类求解算法[J]. *控制与决策*, 2012, 27(4): 535-541.  
(Liu W S, Yang F, Li M Q, et al. Clustering algorithm for split delivery vehicle routing problem[J]. *Control and Decision*, 2012, 27(4): 535-541.)
- [9] 侯立文, 谭家美, 赵元. 求解带时间窗的客户需求可分条件下的车辆路径问题[J]. *中国管理科学*, 2007, 15(6): 46-51.  
(Hou L W, Tan J M, Zhao Y. Solving the split delivery vehicle routing problem with time windows[J]. *Chinese J of Management Science*, 2007, 15(6): 46-51.)
- [10] Tang J F, Ma Y, Guan J, et al. A max-min ant system for the split delivery weighted vehicle routing problem[J]. *Expert Systems with Applications*, 2013, 40(18): 7468-7477.
- [11] Silva M M, Subramanian A, Ochi L S. An iterated local search heuristic for the split delivery vehicle routing problem[J]. *Computers & Operations Research*, 2015, 53(1): 234-249.
- [12] McNabb M E, Weir J D, Hill R R, et al. Testing local search move operators on the vehicle routing problem with split deliveries and time windows[J]. *Computers & Operations Research*, 2015, 56(1): 93-109.
- [13] Bouzaiene-Ayari B, Dror M, Laporte G. Vehicle routing with stochastic demand and split deliveries[J]. *Foundations of Computing and Decision Sciences*, 1993, 18(2): 63-69.
- [14] Lei H T, Laporte G, Guo B. The vehicle routing problem with stochastic demands and split deliveries[J]. *Information Systems and Operational Research*, 2012, 2(50): 59-71.
- [15] Waters C D J. Vehicle-scheduling problems with uncertainty and omitted customers[J]. *J of the Operational Research Society*, 1989, 40(12): 1099-1108.
- [16] Gendreau M, Laporte G, Se Guin R. An exact algorithm for the vehicle routing problem with stochastic demands and customers[J]. *Transportation Science*, 1995, 29(2): 143-155.
- [17] Tang H, Miller-Hooks E. Solving a generalized traveling salesperson problem with stochastic customers[J]. *Computers & Operations Research*, 2007, 34(7): 1963-1987.
- [18] Cheung R K, Xu D, Guan Y. A solution method for a two-dispatch delivery problem with stochastic customers[J]. *J of Mathematical Modelling and Algorithms*, 2007, 6(1): 87-107.
- [19] Lei H, Laporte G, Guo B. Districting for routing with stochastic customers[J]. *EURO J on Transportation and Logistics*, 2012, 1(12): 67-85.
- [20] 曾华. 随机顾客和需求的配送优化模型与算法[D]. 济南: 山东大学控制科学与工程学院, 2012.  
(Zeng H. Distribution optimization with stochastic customers and demands-modeling and algorithms[D]. Jinan: School of Control Science and Engineering, Shandong University, 2012.)
- [21] Branke J, Guntsch M. Solving the probabilistic TSP with ant colony optimization[Z]. 2004, 3: 403-425.
- [22] Bertsimas D J, Jaillet P. A priori optimization[J]. *Operations Research*, 1990, 38(6): 1019-1033.
- [23] Gendreau M, Laporte G, SeGuin R. A tabu search heuristic for the vehicle routing problem with stochastic demands and customers[J]. *Operations Research*, 1996, 44(3): 469-477.
- [24] 孔媛. 航空票务公司免费机场接送服务中基于最小化成本的车次分配与调度方法研究[D]. 沈阳: 东北大学信息科学与工程学院, 2010.  
(Kong Y. Research on minimizing cost methods of vehicle allocation and schedule problems for free pickup & delivery service to airport of flight ticket sales company[D]. Shenyang: College of Information Science and Engineering, Northeastern University, 2000.)
- [25] Dror M, Trudeau P. Split delivery routing[J]. *Naval Research Logistics*, 1990, 37(3): 383-402.
- [26] Shaw P. Using constraint programming and local search methods to solve vehicle routing problems[J]. *Principles and Practice of Constraint Programming-CP98, Lecture Notes in Computer Science*, 1998, 1520(1): 417-431.
- [27] Ropke S, Pisinger D. An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows[J]. *Transportation Science*, 2006, 40(4): 455-472.
- [28] Dueck G. New optimization heuristics: The great deluge algorithm and the record-to-record travel[J]. *J of Computational Physics*, 1993, 104(1): 86-92.
- [29] Solomon M M. Algorithms for the vehicle routing and scheduling problems with time window constraint[J]. *Operations Research*, 1987, 35(2): 254-265.

(责任编辑: 曹洪武)