

基于客户需求的现金押运路线问题研究*

徐国勋¹, 李妍峰^{1,2}, 李 军¹

(1. 西南交通大学 经济管理学院, 四川 成都 610031)

(2. 西南交通大学 服务科学与创新四川省重点实验室, 四川 成都 610031)

摘 要 本文以现金物流为研究背景, 不同于以往对客户需求的研究, 提出了一类考虑客户券别要求的现金押运路线问题, 并以现金券别均衡和运输成本减少为目标, 建立了相应的混合整数规划模型。根据模型的性质, 设计了一种基于局部搜索和多样性管理机制的遗传算法进行求解。数值实验对模型特性和算法性能进行了分析, 结果表明券别因素增加了运输成本, 影响了押运路线, 改进的遗传算法能求解更大规模的问题, 得到质量较好的解。

关键词 现金物流, 客户需求, 券别均衡, 车辆路径问题, 遗传算法

中图分类号 F253.4

1 引言

随着社会经济发展, 流通于各个环节的现金量逐年增加。2012 年, 全世界的现金交易额超过了 11.6 万亿美元^[1]。2013 年, 欧洲中央银行和美国联邦储备局均报告称, 过去三年流通中的现金以每年 9% 的速度增加, 在可预见的将来这种趋势并不会下降^[2]。在美国, 超过 40% 的交易借助于现金完成^[3]。据中国人民银行发布的 2016 年 5 月金融统计数据报告, 流通中的现金达到 6.28 万亿元。现金押运作为现金流转过程中核心的一个环节, 对现金发行渠道和物流成本影响非常大。现实中, 押运部门通常基于经验规划路线, 这些路线或多或少相同并无太大变化, 使现金物流成本较高, 影响现金的流通效率。因此, 研究运钞车的路线优化问题具有重要的现实意义。

现金押运问题的主要理论来源于车辆路径问题 (vehicle routing problem, VRP), VRP 一直是组合优化领域的前沿热点问题。Partyka 和 Hall^[4]、Emir^[5]、Eksioglu 等^[6]、Hosny^[7]认为 ATM 和银行间的现金押运是车辆路径问题的应用, Iori 和 Riera-Ledesma^[8]提出银行之间的现金押运是多旅行商问题的应用。目前对现金这类特殊货品的配送研究中, 考虑客户需求的文章很少, 并集中于准点到达和现金总量被满足这两点。由于现金不同于其他商品, 不允许缺货, 因此 Lambert 等把集取送货一体的货币运输问题归结为带硬时间窗的车辆路径问题, 并采用整数规划和节约法解决了确定和随机旅行时间两种情况下的比利时银行网络运钞问题^[9]。Boonsam 等研究了曼谷一家银行现金押运问题, 将总金额需求预先设定, 并把问题描述为指派问题和带时间窗的车辆路径问题, 并且以最小化行驶时间为目标函数^[10]。Dai 和 Liu 同样预设客户的总金额需求, 优化运钞车行驶时间, 并采用改进的蚁群算法进行求解^[11]。Talarico 等^[12]、

* 基金项目: 国家自然科学基金项目 (71571150, 71361006), 教育部人文社会科学研究项目 (14YJA630026), 四川省哲学社会科学重点研究基地项目 (QGXH15-05), 中央高校基本科研业务费项目 (26815WCX03), 成都市科技局项目 (2015-RK00-00038-ZF), 四川省科技厅项目 (2017JY0225)。

通信作者: 李妍峰, 西南交通大学经济管理学院, 副教授, E-mail: yanwaa@126.com。

Talarico 等^[13]在满足客户现金需求的情况下，将车上运载的现金总量作为量化风险的参数，研究了一种带风险约束的现金押运问题。

与以往研究不同，本文有如下贡献。

(1) 考虑了客户对券别均衡的要求。这是由于在现实中，银行、ATM、购物中心和大型零售商等客户为提高企业竞争力，不希望现金券别结构失衡，如出现零钱荒问题^[14]，因为券别结构失衡会导致企业服务质量下降，降低客户满意度，影响企业声誉，增加运营成本。

(2) 研究了券别因素对运输路线的影响。

(3) 设计了一种遗传算法对问题进行求解，该算法可有效地求解大规模算例。

2 问题描述及模型建立

2.1 问题描述

银行委托押运公司为分布在城市各个路段的客户提供现金押运服务，押运过程中产生的运输费用由银行承担。客户根据前一天的营业情况对现金的总金额提出期望需求，这一需求必须被满足。同时客户为维持一定的服务水平，对各面值的券别数目提出了期望需求并设定了下限，在满足下限的前提下，可以不满足期望需求，但若不满足期望需求，会给银行带来一定的惩罚成本。此外，考虑服务的便利性，客户要求现金一次性送达。但不同的券别组合会产生不同的现金重量，进而产生不同的路线。银行要考虑的是如何设计路线，使运输成本和惩罚成本最小。图 1 用一个简单的例子来说明整个押运过程。

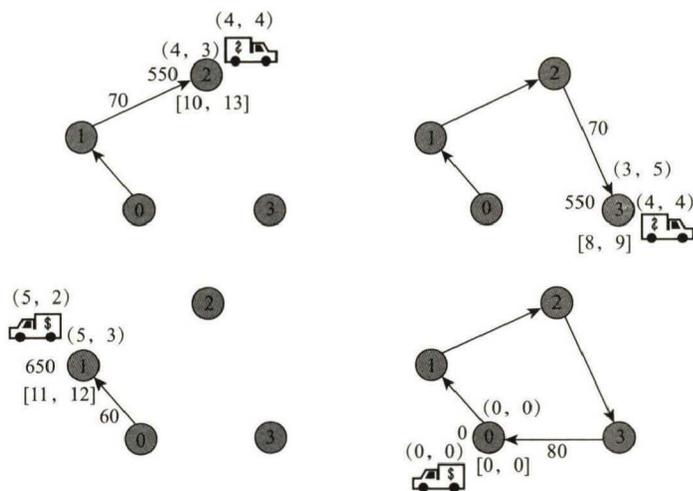


图 1 问题论证举例

为了方便论述，只考虑 100 元和 50 元两种券别。在图 1 中，车辆行驶路线为 $0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 0$ ，表示车辆从车场（0 点）出发，依次向客户 1、客户 2 和客户 3 派送现金，最后回到车场。弧上数字表示两点间的运输成本，图 1 左下图，从车场到客户 1 的运输成本为 60。客户点上方的数字表示客户对每种券别数目的需求，图 1 左下图，客户点 1 上方数字 (5, 3) 表示客户需要 100 元 5 个单位，50 元 3 个单位。点下方数字为惩罚值，表示未满足客户的期望券别数目而带来的单位惩罚，图 1 左下图，客户点 1 下方数字 [11, 12] 分别对应 100 元和 50 元的惩罚值。客户点左方的数字代表客户的金额需求，图 1 左下图，客户点左方数字 650 表示客户需要 650 单位的现金。车上方的数字代表实际送达券别数

目, 图1左下图, 车上方数字为(5, 2), 表示实际押送给客户100元5个单位, 50元2个单位。故现金送至客户1后, 带来100元的券别惩罚 $|5-5|\times 11=0$, 50元的券别惩罚 $|3-2|\times 12=12$, 总金额惩罚 $650-(5\times 100+2\times 50)=50$, 因此客户1的总惩罚为 $0+12+50=62$ 。依次类推, 可求得客户2和客户3的惩罚成本。

2.2 符号说明

本文问题定义在完备图 $G=(V, A)$ 上, V 表示点集合, A 表示弧集合。令 n 表示客户数目, 则 $V=\{0, 1, \dots, n\}$, 其中0表示车场, 车场的任何类型的需求均为零, $A=\{(i, j): i, j \in V, i \neq j\}$ 。在建模过程中需要用到以下符号。

(1) 集合。

M 券别集合;

N 客户集合。

(2) 参数。

q_i : 客户 i 的总金额期望;

a_i^m : 客户 i 第 m 种券别数日期望;

H_i^m : 客户 i 第 m 种券别数目的需求上限;

p_i^m : 未满足客户 i 第 m 种券别数日期望而产生的单位惩罚;

u_i : 未满足客户 i 总金额期望而产生的单位惩罚;

a_m : 第 m 种券别的面值, 如100元, 50元等;

w_m : 第 m 种券别的单位重量;

c_{ij}^k : 车辆 k 从客户 i 到客户 j 的运输成本;

C_k : 车辆 k 的装载能力;

M_0 : 表示一个很大的数。

(3) 决策变量。

y_{im}^k : 车辆 k 向客户 i 派送的第 m 种券别的数目;

Q_i^k : 车辆 k 离开客户 i 时的载重;

B_i^k : 车辆 k 到达客户 i 的时间;

$x_{ij}^k = \begin{cases} 1, & \text{若车辆 } k \text{ 从客户 } i \text{ 驶向客户 } j \\ 0, & \text{否则} \end{cases}$ 。

2.3 数学模型

$$\begin{aligned} \min & \sum_{(i,j) \in A} \sum_{k \in K} c_{ij}^k x_{ij}^k \\ & + \sum_{i \in N} u_i \left| q_i - \sum_{k \in K} \sum_{m \in M} a_m y_{im}^k \right| \\ & + \sum_{i \in N} \sum_{m \in M} P_i^m \left| d_i^m - \sum_{k \in K} y_{im}^k \right| \end{aligned} \quad (1)$$

s.t.

$$\sum_{k \in K} \sum_{j: (i,j) \in A} x_{ij}^k = 1 \quad \forall i \in N \quad (2)$$

$$\sum_{j:(0,j) \in A} x_{0j}^k = 1 \quad \forall k \in K \quad (3)$$

$$\sum_{i:(i,0) \in A} x_{i0}^k = 1 \quad \forall k \in K \quad (4)$$

$$\sum_{i:(i,j) \in A} x_{ij}^k - \sum_{i:(j,i) \in A} x_{ji}^k = 0 \quad \forall j \in N, k \in K \quad (5)$$

$$L_i^m \leq \sum_{k \in K} y_{im}^k \leq H_i^m \quad \forall i \in N, m \in M \quad (6)$$

$$y_{im}^k \leq M_0 x_{ij}^k \quad \forall (i, j) \in A, k \in K, m \in M \quad (7)$$

$$Q_0^k = \sum_{i \in N} \sum_{m \in M} w_m y_{im}^k \quad \forall k \in K \quad (8)$$

$$Q_j^k \geq Q_i^k - \sum_{m \in M} w_m y_{jm}^k + M_0(1 - x_{ij}^k) \quad \forall (i, j) \in A, k \in K \quad (9)$$

$$\max \left\{ 0, \sum_{m \in M} w_m y_{im}^k \right\} \leq Q_i^k \leq \min \left\{ C_k, C_k + \sum_{m \in M} w_m y_{jm}^k \right\} \quad \forall i \in V, k \in K \quad (10)$$

$$B_j^k \geq B_i^k + 1 - M_0(1 - x_{ij}^k) \quad \forall (i, j) \in A, k \in K \quad (11)$$

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

目标函数(1)总成本最小化,第一部分指运输成本,第二部分指未满足总金额而带来的惩罚成本,第三部分指未满足期望的券别数目而带来的惩罚成本;约束条件(2)保证每个客户仅由一辆车服务一次;约束条件(3)和(4)保证每辆车从车场出发,并最终回到车场;约束条件(5)是流守恒条件,车辆服务过客户后必须从此客户处离开,以保证路线的连续性;约束条件(6)保证派送给客户的各券别数目在需求区间之内;约束条件(7)保证车辆只服务行驶路线上的客户;约束条件(8)保证车辆出发时的载重等于派送至所有客户的券别重量之和,即保证空车返回车场;约束条件(9)和(10)是载重平衡条件,确保车辆在整条行驶路线中不会超载;约束条件(11)是子环消除条件,保证一条路线上各个点的先后访问顺序,如一条路线 $0 \rightarrow 1 \rightarrow 2 \rightarrow 0$,若不加子环消除条件,可能会生成 $0 \rightarrow 1 \rightarrow 0$ 和 $0 \rightarrow 2 \rightarrow 0$ 两条子环;约束条件(12)是0~1变量条件。

3 求解算法

遗传算法被成功地应用于车辆路径问题的求解,针对车辆路径问题的特性,Vidal等改进了遗传算法,设计的算法具有高效局部搜索能力和多样性管理机制^[15-17],Cherkesly等^[18]分别对算法进行了改进。由于本问题是车辆路径问题的应用,故参考了Vidal等的做法,设计了遗传算法对问题进行求解^[15]。与传统遗传算法不同,本文对遗传算法做了关键性的改进,包括评估机制,通过局部搜索对子代个体进行教育,以及种群多样性管理。

算法流程如下所示。

Step 1: 随机生成初始种群;

Step 2: 如果达到最大迭代次数 It_{max} 则转 Step 9, 否则继续以下步骤;

Step 3: 由选择算子生成两个父代个体 P_1 和 P_2 ;

Step 4: 由交叉算子对父代个体 P_1 和 P_2 进行操作生成子代 C ;

Step 5: 若子代 C 可行,则放入可行子种群,否则放入非可行子种群;

Step 6: 以概率 P_{repair} 对非可行子种群每个个体进行修复;

Step 7: 若达到子种群规模上限, 则筛选出幸存个体;

Step 8: 若迭代 It_{div} 次无法改善最优解, 则进行多样性操作, 并加大不可行个体的惩罚系数;

Step 9: 返回最好的可行个体。

3.1 解的表示

由于车辆路径问题是基于次序的组合优化问题, 故本文对染色体采用自然数编码, 将每辆车的路线连接成一条序列来表示染色体, 一个个体表示一个解。图 2 所示的序列代表了 3 条子路线, 为路线 $r_1=(0,3,2,0)$ 、 $r_2=(0,4,1,0)$ 和 $r_3=(0,6,5,7,0)$ 。其中 0 点代表车场, 数字 1 至 n 代表被服务的客户, 路线中点的前后顺序代表服务客户的先后顺序, 如路线 r_1 , 车辆服务过客户 3 后才能服务客户 2。若出现路线 (0, 0) 表示不使用这辆车。

0	3	2	0	4	1	0	6	5	7
---	---	---	---	---	---	---	---	---	---

图 2 解的表示

3.2 解的评估

该算法加入了非可行个体的操作, 非可行个体通过松弛载重约束 (即路线上所有客户的需求总重量可以大于车辆的载重上限) 来获得, 加入非可行个体的目的是扩大搜索范围。本文将路径 R 第 k 条子路径 R_k 的超载量定义为 $W(R_k) = \max\{0, \sum_{i=0}^n D_i - C_k\}$, 其中, D_i 表示从客户 i 运走的现金重量, C_k 表示车辆 k 的载重能力。如果子路径 R_k 满足载重约束, 令 $W(R_k) = 0$, 否则 $W(R_k) > 0$ 。于是路径 R 便有如公式 (13) 的惩罚成本:

$$C(R) = f(R) + \alpha W(R) \quad (13)$$

其中, $f(R)$ 通过目标函数 (1) 得到, α 为超过载重上限而带来的惩罚值。由于每个个体通过车的路线连接成, 因此路径 R 已生成, 决策变量 x_{ij}^k 和 B_i^k 的数值被确定, 路线策略被确定。于是路径约束条件 (2)~(5), 子环消除约束条件 (11) 便可以删除, 原模型从复杂的混合整数规划问题变成了简单的线性规划问题。本文使用大规模数学规划优化器 Gurobi optimizer 6.5 求解该线性规划问题, 从而求解决策变量 y_{im}^k 和 Q_i^k , 即券别策略被确定。当决策变量 x_{ij}^k 和 y_{im}^k 求得后, 目标函数 (1) 便求得, 从而得出 $f(R)$ 。

对于个体 s , 偏差值有如下定义:

$$BF(s) = \text{fit}(s) + (1 - \frac{N_{\text{elite}}}{N_{\text{indiv}}}) \text{dc}(s) \quad (14)$$

其中, $\text{fit}(s)$ 代表个体 s 在 $C(R)$ 集合中的排序值。 $\text{dc}(s)$ 代表个体 s 与其他个体相似度的排序值, Cherklesly 等把两个个体相同弧的数目定义为相似度, 如 $s_1 = (0, 1, 2, 3, 0, 5, 7, 0)$, $s_2 = (0, 1, 2, 3, 0, 6, 8, 0)$, 则弧 (0, 1)、(1, 2)、(2, 3) 和 (3, 0) 为 s_1 和 s_2 的相似弧, 两个个体的相似度为 4^[18]。 N_{elite} 代表存活到下一代的精英个体数目, N_{indiv} 代表子种群的个体数目。Vidal^[15]等证明偏差值的设定有益于种群多样性, 本文采用 Cherklesly 等^[18]对相似度的定义, 并用偏差值函数表示评价函数。

3.3 选择与交叉

文献[15~18]使用了二元联赛选择算子 (binary tournament selection), 认为这种方法选择出的父代

个体，经交叉操作后容易得到质量较高的子代个体。本文亦采用了这种方法，从种群中随机选出两个个体，选出偏差值最小的个体，根据其可行性放入相应的可行或不可行子种群。

Vidal 等表明，对基于次序的组合优化问题，OX (ordered crossover) 算子效果较好，考虑到本问题亦是基于次序的组合优化问题，故选用 OX 算子作为交叉算子^[17]。

3.4 教育与修复

本文设计了 6 种邻域操作算子作为教育算子，以概率 $P_{educate}$ 对子代个体进行教育，用来改进子代个体的质量。

(1) 单点随机插入。该算子从解中随机选出一个点，并将该点随机插入原解中。如图 3 (a) 所示例子，算子选中点 6，并将该点插入点 2 和点 3 之间。

(2) 子序列随机插入。该算子从解中随机选取一条序列，并将该序列随机插入原解中。如图 3 (b) 所示例子，算子选中子序列 (0, 6, 5)，并将该子序列插入点 2 和点 3 之间。

(3) 两点交换。该算子从解中随机选出两个点，交换两者位置。如图 3 (c) 所示例子，算子选中点 2 和点 6，然后交换两者位置。

(4) 子序列随机交换。该算子从解中随机选出两条序列，交换两者位置。如图 3 (d) 所示例子，算子选中子序列 (2, 0) 和 (0, 6, 5)，然后交换两者位置。

(5) 子序列逆序。该算子从解中随机选出一条序列，然后将其逆序。如图 3 (e) 所示例子，算子选中子序列 (0, 6, 5)，然后逆序变为 (5, 6, 0)。

(6) 子序列交换后逆序。该算子从解中随机选出两条序列，然后将两者逆序并交换位置。如图 3 (f) 所示例子，选中子序列 (2, 0) 和 (0, 6, 5)，然后逆序为 (2, 0) 和 (5, 6, 0)，并交换两者位置。



图 3 邻域算子

具体的教育操作步骤如下。

- Step 1: 将 6 种邻域操作算子放入选择集合 S 中;
- Step 2: 从集合 S 中随机选取一种邻域算子;
- Step 3: 若被选中算子能改进解，反复执行该算子 It_{edu} 次后终止教育操作，否则转 Step 4;
- Step 4: 将被选中的算子从集合 S 中移除;

Step 5: 如果集合 S 不为空, 则转 Step 2, 否则终止教育操作。

经过教育操作后的个体可能会违背载重约束导致不可行, 故需要修复操作。将可行个体和非可行个体分别放入对应的可行集合和非可行集合, 以概率 P_{repair} 对不可行个体进行修复, 每个不可行个体最多被修复两次, 第一次将惩罚值 α 变大 10 倍并进行教育操作, 若解仍旧不可行则进行第二次修复, 将惩罚值 α 变大 100 倍并再次进行教育操作。被解如果修复成功 (即满足载重约束), 则将其放入可行个体集合, 同时将原非可行个体从非可行集合删除。在修复算子中加入单位惩罚值 α 的目的是使搜索朝可行解的方向前进。

3.5 种群管理

可行和不可行两个子种群的规模被控制在区间 $[\mu, \mu + \lambda]$, 其中 μ 表示子种群规模的下限, λ 表示每次迭代生成的子代数目。每个个体经过交叉、教育和修复操作后, 根据可行性分别放入对应的子种群。如果子种群规模达到上限 $\mu + \lambda$, 则进行幸存者挑选操作, 即将子种群中的个体按照偏差值从大到小的顺序排序, 然后删除 λ 个个体, 使子种群规模减至 μ 。

在算法的最开始, 需生成一定规模的初始种群, 本文采用文献[15~17]的做法, 随机生成 4μ 个个体。初始种群中的每个个体都要进行教育操作, 如果被教育后的个体不可行, 则以 0.5 的概率对其进行修复操作。然后根据个体可行性, 将被操作后的 4μ 个个体分别放入可行和不可行子种群。

不可行个体的惩罚参数 α , 随着算法的进行而动态调整, 目的是平衡可行个体和不可行个体的比例。本算法每迭代 20 次, 调整一次 α 的数值。取参数 ξ^{REF} 为可行个体的期望占比, 若可行个体的实际比例比 ξ^{REF} 高 5% 以上, 则 α 调整为原来的 0.8 倍, 若可行个体的实际比例比 ξ^{REF} 低 5% 以上, 则 α 调整为原来的 1.2 倍。

如果迭代 It_{div} 次 (It_{div} 取 $0.4It_{\text{max}}$ ^[15]), 最优解仍无改进, 则对种群进行多样性操作: 将每个子种群最好的 $\mu/3$ 个个体保留, 然后生成 4μ 个新个体, 从新个体中挑选幸存者, 分别放入对应的子种群。

4 数值实验

由于没有针对本问题的算例, 故在 VRP 算例 (见网站 <http://neo.lcc.uma.es/vrp/vrp-instances/capacitated-vrp-instances/>) 的基础上生成了 10 组不同规模的数据。其中车辆数目、客户数目、车辆载重和客户坐标来自标准算例, 券别面值参数 a_m 和重量参数 w_m 分别为人民币纸币各券别实际的面值和重量, 券别集合 $M=6$, 总金额期望 q_i 均匀分布于 $[0, 1000]$, 券别数目期望 d_i^m 、需求下限 l_i^m 和需求上限 H_i^m 均匀分布于 $[0, 20]$, 风险系数 p_i^m 均匀分布于 $[0, 100]$, 风险系数 u_i 均匀分布于 $[0, 10]$ 。所有程序均用 Python 编写, 并在 i7-2600CPU 主频 3.40GHz 内存 16GB 的 PC 上进行实验。

4.1 节用一个小规模算子 (数据规模 2×9) 来阐述模型特性, 分析了现金因素对路线选择的影响。4.2 节对遗传算法性能进行了分析。

4.1 现金券别因素对路线的影响

本节分析现金因素对车辆运输路线的影响, 比较车辆押运路线在策略 T1 (保持原模型不变) 和策略 T2 (不考虑现金因素, 令模型中总金额惩罚系数和券别惩罚系数 $u_i, p_i^m=0$) 的差异。表 1 记录了两种策略下车辆的运输成本和运行路线。

表 1 两种现金策略下的路线

策略	总成本	运输成本	运输成本占比/%	押运路线
T1	1 274.6	381.4	29.9	0 → 5 → 8 → 4 → 2 → 3 → 0 0 → 1 → 7 → 6 → 0
T2	339.2	339.2	100	0 → 6 → 32 → 4 → 8 → 5 → 0 0 → 7 → 1 → 0

由表 1 可知,当采用策略 T2 时,模型的目标函数的优化仅仅是路线的优化,运输成本即是总成本,因此运输成本占比 100%。而考虑策略 T1 时,运输成本和总成本都在增加,这是由于考虑 T1 加入了现金券别因素时,惩罚系数 p_i^m 的存在会产生不同的券别组合,进而产生不同的现金金额和车辆载重,使目标函数的改善就不仅仅是路线的改善,还要在运输成本和券别组合之间实现最优平衡以实现最小化所有成本,因此车辆无法选择策略 T2 最优运输成本路线,在路线上做出了妥协,选择一条运输成本相对较高的路线。这表明现金券别因素在模型中得到了成功表达,影响了车辆运输路线。

4.2 算法性能分析

Vidal 等为得到遗传算法最优的参数值,使用车辆路径问题的算例做了大规模的测试^[15]。文献[16]、文献[17]采用了和文献[15]相同的参数设置。考虑到本问题也是车辆路径问题,因此采用和文献[15]相同的参数设置,详细内容见表 2。

表 2 遗传算法参数设置

参数	描述	数值
N_{indiv}	子种群个体数目	25
N_{elite}	精英个体数目	10
$P_{educate}$	个体被教育的概率	1.0
It_{edu}	邻域算子最大连续执行次数	20
P_{repair}	个体被修复的概率	0.5
α	对非可行个体的惩罚值	2
μ	最小种群规模	25
λ	一次迭代生成的子代数	40
ζ_{REF}	可行个体的期望比例	0.2
It_{max}	最大迭代次数	200

考虑到本文所设计算法增加了非可行解的操作,其目的是扩大邻域的搜索范围,提高邻域搜索的效率。为比较算法的优劣,本文选取两种基于邻域搜索求解车辆路径问题的算法,分别是 Chen 等^[19]改进的变邻域搜索算法(variable neighborhood search, VNS),Pisinger 和 Ropke^[20]改进的大规模邻域搜索算法(large neighbor search, LNS)。表 3 中算法的各项结果包括:程序运行 20 次得到的目标函数的平均值(avg)、标准差(std)和平均运行时间(cpu/s)。

表3 算法性能比较

问题规模	VNS			LNS			GA		
	avg	std	cpu/s	avg	std	cpu/s	avg	std	cpu/s
2×9	1 289	15.5	0.4	1 277	4.6	0.6	1 275	2.7	0.6
3×15	1 198	24.9	0.5	1 186	18.4	1.9	1 175	9.2	0.9
4×20	1 504	38.9	0.6	1 488	37.7	1.1	1 464	6.2	1.0
5×32	2 430	55.2	0.9	2 382	46.3	1.1	2 285	18.6	1.1
5×37	2 661	37.4	1.1	2 665	36.4	1.2	2 598	9.9	1.2
6×45	4 884	41.0	1.8	4 885	46.4	2.6	4 554	9.8	2.8
7×48	3 066	43.8	1.7	2 982	40.7	2.8	2 850	14.5	2.9
8×62	4 280	51.9	1.8	4 055	42.7	3.5	3 995	11.1	3.3
9×69	4 830	43.2	1.8	4 556	37.1	3.9	4 256	25.5	4.5
10×80	4 981	25.9	1.9	4 784	27.4	4.8	4 759	15.5	5.2
均值	3 112	37.8	1.2	2 951	33.8	2.3	2 896	12.3	2.4

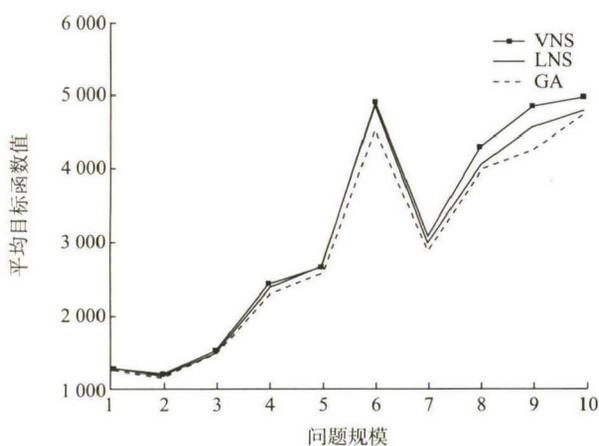
注: VNS 表示变邻域搜索算法, LNS 表示大规模邻域搜索算法, GA 表示遗传算法

根据表3的实验数据,得到以下结论。

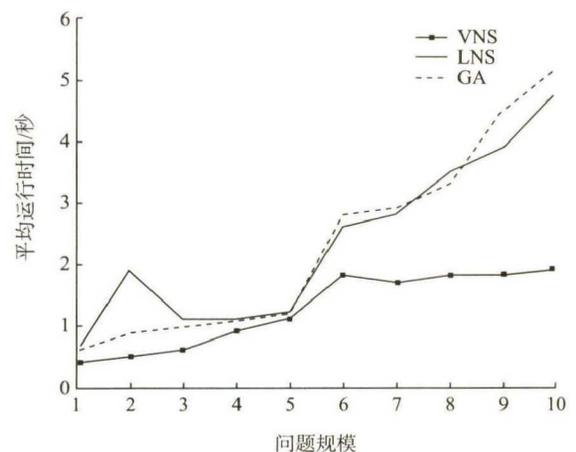
(1) 与 VNS 和 LNS 算法相比,在全部 10 组数据中,本文所设计的遗传算法目标函数值最小[图 4(a)],虽然时间开销相对较大[图 4(b)],但时间均值也仅仅比 VNS 多了 1.2 秒,比 LNS 多了 0.1 秒,但其均值比 VNS 改进了 7.4%,比 LNS 改进了 1.9%,显示具备相对均衡的求解效率。

(2) 与 VNS 和 LNS 算法相比,在全部 10 组数据中,遗传算法标准差最小[图 4(c)],并且其均值比 VNS 改进了 200%,比 LNS 改进了 175%。这表明,本文所设计的遗传算法的稳定性远远超过了 VNS 和 LNS 算法。

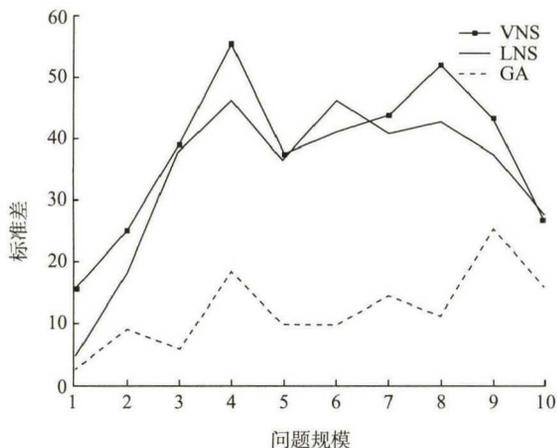
(3) 以 5×37 为例[图 4(d)],分析了三种算法的收敛性能,遗传算法在 50 代以内便开始收敛,而 VNS 在 80 代左右开始收敛,LNS 则在 100 代以后才开始收敛。这表明,本文所设计的遗传算法收敛能力比 VNS 和 LNS 算法更加优异。



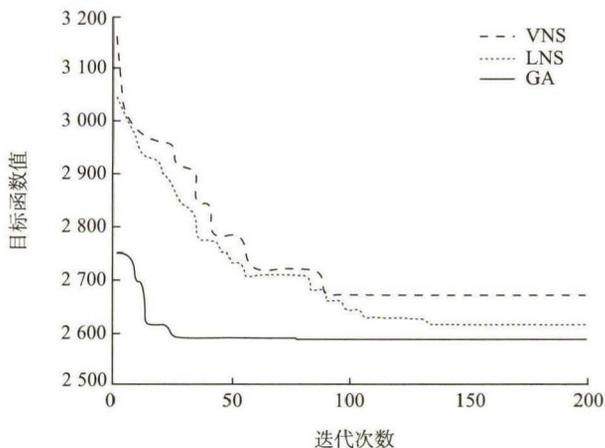
(a) 平均目标函数值对比



(b) 平均运行时间对比



(c) 标准差对比



(d) 收敛性对比

图4 遗传算法收敛性分析

5 结束语

本文利用车辆路径问题理论研究了考虑客户券别均衡要求的现金押运问题，以运输成本减少，券别组合最优为目标建立了混合整数规划模型，并针对问题特性设计了一种基于局部搜索和多样性管理机制的遗传算法进行求解，所设计的遗传算法嵌入了精确求解方法，其中嵌入算法用来确定券别决策，遗传算法用来确定路线决策并进行全局寻优。数值实验表明现金因素在模型中得到了成功表达，影响了押运路线。本文所设计的遗传算法求解效率超过了 VNS 算法和 LNS 算法，表现出了更好的求解性能和收敛性能。

高效的物流效率，可以降低人民币的运输储存成本，保障人民币的安全，满足社会合理的现金供应。而我国目前的货币调运管理，仍然采用固定路线或者老员工按照经验制定路线的方式，效率较低并且效果较差，不能适应新时期的经济发展需求。希望本文的研究结论能给相关部门制定押运路线时提供科学的理论依据。

本文仍有如下不足：①仅研究了单车场情况下现金的押运路线问题，所设计的遗传算法仅适用于单车场的情况，并不能适用于多车场情况；②由于无法获得真实数据，本文采用了模拟数据进行仿真，因此相关结论可能无法做到和实际情况完全一致。

参考文献

- [1] Bagnall J, Bounie D, Huynh K P, et al. Consumer cash usage: a cross-country comparison with payment diary survey data[J]. Dnb Working Papers, 2015, 12 (4): 1-61.
- [2] Kim M, Raghuvanshi K, Bucciarelli R, et al. Optimizing the retail bank supply chain: how retail banks can learn from the consumer business industry to lower cost, reduce inventory, and boost productivity[R]. Technical Report, Deloitte, 2013.
- [3] Bennett B, Conover D, O'Brien S, et al. Cash continues to play a key role in consumer spending: evidence from the diary of consumer payment choice[J]. Federal Reserve Bank of San Francisco Fednotes, 2014.
- [4] Partyka J G, Hall R W. On the road to service[J]. Or/ms Today, 2000, 27 (4): 26-30.
- [5] Emir A. Delivery pricing for different demand price elasticity functions[D]. University of Florida, 2002.

- [6] Eksioglu B, Vural A V, Reisman A. The vehicle routing problem: a taxonomic review[J]. *Computers & Industrial Engineering*, 2009, 57 (4): 1472-1483.
- [7] Hosny M I. Investigating heuristic and meta-heuristic algorithms for solving pickup and delivery problems[D]. Cardiff University, 2010.
- [8] Iori M, Riera-Ledesma J. Exact algorithms for the double vehicle routing problem with multiple stacks[J]. *Computers & Operations Research*, 2015, 63: 83-101.
- [9] Lambert V, Laporte G, Louveaux F. Designing collection routes through bank branches[J]. *Computers & Operations Research*, 1993, 20 (7): 783-791.
- [10] Boonsam P, Suthikarnnarunai N, Chitphaiboon W. Assignment problem and vehicle routing problem for an improvement of cash distribution[C]. *Proceedings of the World Congress on Engineering and Computer Science*, 2011, 2: 1160-1164.
- [11] Dai M, Liu X C. An improved ant colony algorithm for single vehicle route optimization[J]. *Journal of Computational Information Systems*, 2012, 10 (8): 3963-3969.
- [12] Talarico L, Sörensen K, Springael J. Metaheuristics for the risk-constrained cash-in-transit vehicle routing problem[J]. *European Journal of Operational Research*, 2015, 244 (2): 457-470.
- [13] Talarico L, Springael J, Sörensen K, et al. A large neighbourhood metaheuristic for the risk-constrained cash-in-transit vehicle routing problem[J]. *Computers & Operations Research*, 2017, 78 (C): 547-556.
- [14] 梁永明, 赖厚玮, 石晓熊. 零钱荒现象分析及其对策研究[J]. *科技致富向导*, 2011, (18): 57.
- [15] Vidal T, Crainic T G, Gendreau M, et al. A hybrid genetic algorithm for multidepot and periodic vehicle routing problems[J]. *Operations Research*, 2012, 60 (3): 611-624.
- [16] Vidal T, Crainic T G, Gendreau M, et al. A hybrid genetic algorithm with adaptive diversity management for a large class of vehicle routing problems with time-windows[J]. *Computers & Operations Research*, 2013, 40 (4): 475-489.
- [17] Vidal T, Crainic T G, Gendreau M, et al. A unified solution framework for multi-attribute vehicle routing problems[J]. *European Journal of Operational Research*, 2014, 234 (3): 658-673.
- [18] Cherkesly M, Desaulniers G, Laporte G. A population-based metaheuristic for the pickup and delivery problem with time windows and LIFO loading[J]. *Computers & Operations Research*, 2015, 62 (C): 23-35.
- [19] Chen P, Huang H K, Dong X Y. Iterated variable neighborhood descent algorithm for the capacitated vehicle routing problem[J]. *Expert Systems with Applications*, 2010, 37 (2): 1620-1627.
- [20] Pisinger D, Ropke S. A general heuristic for vehicle routing problems[J]. *Computers & Operations Research*, 2005, 34 (8): 2403-2435.

The Cash-in-transit Vehicle Routing Problem with Customer Denomination Demand

XU Guoxun¹, LI Yanfeng^{1, 2}, LI Jun¹

(1. School of Economics and Management, Southwest Jiaotong University, Chengdu 610031, China

2. Key Laboratory of Service Science and Innovation, Southwest Jiaotong University, Chengdu 610031, China)

Abstract Based on cash logistics problem, different from the previous researches focusing on arriving on schedule and meeting demand, this paper proposes a kind of vehicle routing problem by considering cash denominations, and presents an integer programming formulation to balance cash denominations and reduce transportation costs. According to the characteristics of the model, this paper develops a genetic algorithm with efficient local search based improvement heuristics and diversity management mechanism. The numerical studies are adopted to analyze problem characteristics and the method performance. The results reveal that cash factors increase the transportation costs, change the transportation routing. The improvement genetic algorithm can solve much larger problem and produce high quality solutions.

Key words cash logistic, customer demand, denomination balance, vehicle routing problem, genetic algorithm

作者简介

徐国勋（1987—），男，博士生，河南商丘人，研究方向为物流优化、交通优化。E-mail: xgxbing@126.com。

李妍峰（1980—），女，副教授、博士生导师，研究方向为物流优化、交通优化。E-mail: yanwaa@126.com。

李军（1967—），女，教授、博士生导师，研究方向为物流优化、合作博弈。E-mail: lijun@home.swjtu.edu.cn。