基于混合蝙蝠算法的多车场车辆调度研究

曹庆奎[1,2], 高亚伟[1], 任向阳[1]
(1. 河北工程大学 管理工程与商学院,河北 邯郸 056038;2. 廊坊师范学院 经济与管理学院,河北 廊坊 065000)

[摘 要]多车场车辆调度问题是物流配送研究中的NP难题,同时也是现代物流的发展趋势。针对多车场车辆调度问题,考虑客户和物流运营商的利益,以客户不满意度最低、物流成本最低为目标,建立多目标车辆调度数学模型,并对目标函数进行规范化处理,将多目标问题化简为单目标问题。针对传统的蝙蝠算法在局部搜索能力上存在着不足的问题,将遗传算法中的自适应交叉操作引入到蝙蝠算法中,设计一种混合蝙蝠算法计算数学模型。通过MATLAB软件进行仿真,仿真结果与传统的蝙蝠算法进行对比。结果表明:此方法在解决多车场车辆调度问题上是可行的,并且优于传统的蝙蝠算法。

[关键词]多车场;混合蝙蝠算法;多目标;车辆调度
doi:10.3969/j.issn.1673-9477.2021.01.001
[中图分类号]C931 [文献标识码]A [文章编号]1673-9477(2021)01-001-06

随着客户需求量的增多,客户网点分布不规则,物流运营商若只考虑用一个车场对客户进行物流配送服务已远远不能满足现实生活的需要,配备多个车场必然是未来的趋势。物流运营商需要在多个车场间合理的安排车辆的行驶路线,在考虑物流配送成本的同时,还需考虑到时间窗、客户满意度、碳排放等因素,由此出现多车场车辆调度问题。该问题是基本车辆调度问题的扩展,是更为复杂的NP难题,近年来受到了国内外学者的广泛关注。Yoshinori等[1]以能耗和碳排放量最小为目标,构建了带时间窗限制的多车场车辆调度模型,并利用启发式算法进行求解。Nadjafi等[2]探讨了带时间窗约束和车辆限制的多车场车辆调度问题,以最小配送费用为目标进行规划,设计出一种启发式算法求解数学模型,并成功的运用到180个实验案例中。Zhou等[3]探讨了带有燃料限制的多车场车辆路径问题,以总配送成本最低为目标,提出了四种新的混合整数线性规划公式来计算模型的最优解。鲁建夏等[4]在多车场车辆调度问题中考虑了供给价格和运输成本的因素,以总成本最小为目标构建了多点配送车辆调度模型,并采用改进的变邻域搜索算法进行求解。

2010年,剑桥大学学者Yang[5]首次提出了蝙蝠算法,该算法是通过模拟蝙蝠回声定位系统而提出的一种随机寻优算法。蝙蝠算法自提出以来就受到国内外学者的广泛关注,现有的研究成果表明,蝙蝠算法拥有较好的收敛性和鲁棒性。近年来,蝙蝠算法已被学者们应用到了车辆调度领域并得到了相应的研究成果。Zhou等[6]将贪婪自适应搜索算法与路径重链接引入到蝙蝠算法中提出了一种混合蝙蝠算法,并将其运用到确定性VRP问题的求解。Osaba等[7]为了求解对称和非对称的旅行商问题,设计出一种离散型的蝙蝠算法,并与其他算法进行比较,证明了蝙蝠算法的优越性。殷亚等[8]在蝙蝠算法中引入了交叉算子和重组算子,并将其运用到多目标车辆调度问题的求解,通过算例证明了该方法可以很好地解决蝙蝠算法过早收敛的问题。孙奇等[9]在研究车辆调度问题上,将自适应启发式算法和病毒进化机制引入到蝙蝠算法中,并验证了该算法的有效性。

以上学者在多车场车辆调度问题以及蝙蝠算法上进行了深刻的研究。但是,现有的研究过分看重对物流运输成本的优化,往往忽视了客户的满意度,而且传统的蝙蝠算法在局部搜索能力上存在着缺陷。本文针对多车场车辆调度问题的特点,综合考虑了客户和物流运营商的利益,把遗传算法中的自适应交叉操作引入到蝙蝠算法中,设计了一种混合蝙蝠算法,并将其运用到本文的模型进行求解。

一、问题描述及模型建立

以车场和客户点间的配送网络为研究对象,假设物流服务商有M个车场,每个车场配备容量为Q的货车$K_m(m=1,2,\cdots,M)$辆,货车给N个客户实施

[投稿日期]2020-10-05
[基金项目]河北省高等学校人文社会科学研究项目(SD181012);河北省社会科学基金项目(HB17GL022)
[作者简介]曹庆奎(1963-),男,河北唐山人,博士,教授,研究方向:供应链管理。

物流配送服务,客户点 i 的需求量为 $q_i(i=1,2,\cdots,N)$,车辆从各自的初始车场发车送货,给客户配送完货物后返回初始车场。所有客户都能被任何一个车场的车辆服务,但每个客户仅仅可以被一辆车服务一次,且需求一次得到满足,已知客户的期望时间窗,客户收到货物的时间不可早于客户能接受的最早服务时间,也不可晚于客户能接受的最晚服务时间。设客户的编码为 $1,2,\cdots,N$,车场编码为 $N+1,N+2,\cdots,N+M$。

(一)模型假设

为了简化模型的构建与求解,让模型在现实中有更高的实用价值,对构造的模型给出如下假设:

(1)所有车场的坐标位置、配备货车的数量以及货车的最大运输量已知,且每辆货车的型号相同;

(2)所有的运输车辆在车辆的初始车场发车,且完成送货任务后回到原车场;

(3)车场可利用的车辆数不可多于该车场配备的车辆总数;

(4)每辆车运输的重量不可多于该车辆的最大承重量;

(5)客户收到货物的时间不可早于客户能接受的最早服务时间,也不可晚于客户能接受的最晚服务时间;

(6)每辆运输车辆匀速行驶,且单位行驶成本固定;

(7)所有客户都能被任意一个车场的车辆服务,但每个客户仅仅可以被一辆车服务一次,且需求一次得到满足。

(二)变量说明

下面对多车场车辆调度问题数学模型中的变量进行说明:

d_{ij}:表示客户节点 i 到 j 的距离;

Q:表示车辆的最大载重量;

C_0:表示车辆的单位运输成本;

V_0:表示车辆的行驶速度;

t_i^{mk}:表示车场 m 的车辆 k 到达客户点 i 的时刻;

ST_i:表示车辆在客户点 i 的服务时间;

$[E_i,L_i]$:表示客户 i 的期望时间窗;

δ:表示超出客户期望时间窗的最大忍耐时间;

$$x_{ij}^{mk}=\begin{cases}1,\text{车场 } m \text{ 的车辆 } k \text{ 从点 } i \text{ 行驶到点 } j\\0,\text{否则}\end{cases};$$

(三)运输成本度量

物流的运输成本为运输车辆从配送中心或客户点开往下一个客户点所产生的费用。本文假设物流运营商的运输成本只与运输车辆行驶的距离相关,运输成本等于所有车辆的行驶距离乘以车辆的单位运输成本。故本文的运输成本计算公式如下:

$$D_F=\sum_{k=1}^{k_m}\sum_{m=1}^{M}\sum_{i=1}^{N+M}\sum_{j=1}^{N+M}C_0d_{ij}x_{ij}^{mk} \tag{1}$$

(四)客户不满意度度量

本文假设客户的不满意度只与客户收到货物的时间相关。当客户的收货时间在期望时间窗内时,客户的不满意度最低,当客户的收货时间提前或延迟于期望时间窗,会增加客户的不满意度,但客户不会选择拒绝收货。客户时间窗惩罚图如图1所示:

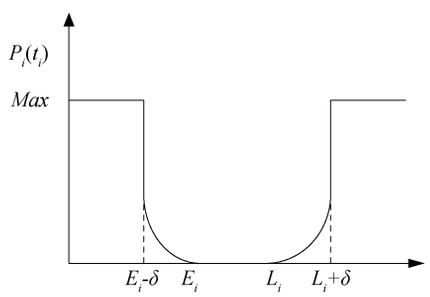

图1 客户时间窗惩罚图

图中当 $E_i\leqslant t_i^{mk}\leqslant L_i$ 时,代表 m 车场的车辆 k 在客户期望时间内抵达,惩罚为0。当 $E_i-\delta<t_i^{mk}<E_i$ 或者 $L_i\leqslant t_i^{mk}\leqslant L_i+\delta$ 时表示货物没有在客户的期望时间内送到,此时需要进行相应的惩罚。因此,这里引用相应的惩罚函数 $P_i(t_i)$[10]:

$$P_i(t_i)=\begin{cases}Max & t_i^{mk}<E_i-\delta\\ a_i(E_i-t_i^{mk})^{m_i} & E_i-\delta\leqslant t_i^{mk}\leqslant E_i\\ 0 & E_i\leqslant t_i^{mk}\leqslant L_i\\ b_i(t_i^{mk}-L_i)^{n_i} & L_i\leqslant t_i^{mk}\leqslant L_i+\delta\\ Max & t_i^{mk}>L_i+\delta\end{cases} \tag{2}$$

式中:a_i、b_i、m_i 和 n_i 均为惩罚系数。客户平均不满意度的计算方法如下:

$$D_P=\frac{\sum_{i=1}^{N}\sum_{m=1}^{M}\sum_{k=1}^{K_m}P_i(t_i)}{N} \tag{3}$$

(五)多车场车辆调度模型

根据上述条件,本文以客户不满意度最低、物流运输成本最低为目标构造如下的多车场车辆调度数学模型:

$$f_1 = minD_P \tag{4}$$

$$f_2 = minD_F \tag{5}$$

$$\sum_{j=1}^{N}\sum_{k=1}^{K_m} x_{ij}^{mk} \leqslant K_m, i = m \in (N+1,\cdots,N+M) \tag{6}$$

$$\sum_{j=1}^{N} x_{ij}^{mk} = \sum_{j=1}^{N} x_{ji}^{mk} \leqslant 1, i = m \in (N+1,\cdots,N+M), k \in K_m \tag{7}$$

$$\sum_{j=1}^{N+M}\sum_{m=1}^{M}\sum_{k=1}^{K_m} x_{ij}^{mk} = 1, i \in N \tag{8}$$

$$\sum_{i=1}^{N+M}\sum_{m=1}^{M}\sum_{k=1}^{K_m} x_{ij}^{mk} = 1, j \in N \tag{9}$$

$$\sum_{i}^{N} q_i \sum_{j=1}^{N+M} x_{ij}^{mk} \leqslant Q, k \in K_m, m \in (N+1,\cdots,N+M) \tag{10}$$

$$\sum_{j=N+1}^{N+M} x_{ji}^{mk} = \sum_{j=N+1}^{N+M} x_{ij}^{mk} = 0, i = m \in (N+1,\cdots,N+M), k \in K_m \tag{11}$$

$$E_i - \delta \leqslant t_i^{mk} \leqslant L_i + \delta, i \in N, k = K_m, m \in (N+1,\cdots,N+M) \tag{12}$$

模型中式(4)、(5)为目标函数,代表客户不满意度和物流运输成本最低;式(6)表示车场可利用的车辆数不可超过该车场配备的车辆总数;式(7)表示所有的运输车辆在车辆的初始的车场发车,且完成送货后回到原车场;式(8)和(9)表示一个客户仅仅可以被一辆车服务一次;式(10)表示每辆车的装载量不可多于该车的最大运输量;式(11)表示车辆不可直接从车场到车场;式(12)表示客户时间窗约束。

二、多目标处理

针对多目标优化问题的处理,本文首先按照各目标之间的重要性给予对应的权重,然后对各目标函数进行规范化处理,将规范化处理后的目标函数乘以相应的权重,最后相加当作本文的目标函数,进而把多目标问题化简为单目标问题。

(一)确定权重

为了保证权重赋值的客观性,本文按照各目标的重要性程度对权重进行随机化处理,在提高算法搜索能力的同时可以得到多个最优解。目标函数的权重 l_e 为:

$$l_e = \frac{ran\ d_1}{ran\ d_1 + ran\ d_e} \tag{13}$$

式中,$0 \leqslant l_e \leqslant 1, e \in [1,2], l_1 > l_2$,且 $\sum_{e=1}^{2} l_e = 1$,$rand_e$ 为非负随机实数。

(二)规范化处理

本文要求解的多个目标之间的量纲不是统一的,不能直接采用线性加权法进行求解。本文采用如下权重模型[16]对两个目标函数进行规范化处理。

$$F = 1 - \sum_{e=1}^{2} l_e\left(\frac{f_e - f_e^*}{f_e^*}\right) \tag{14}$$

式中,f_e^* 为第 e 个目标在单目标限制条件下的最佳配送路线的值,F 越高说明配送路线越理想。本文将 F 作为最终的目标函数。

三、混合蝙蝠算法

蝙蝠是一种神奇的动物,拥有特殊的回声定位功能,可以通过向周围发出声音脉冲,接受周围物体反射回来的回声,基于接收回声的时间差及响度变化去构建周围环境的三维场景,从而锁定捕食目标[12]。

(一)蝙蝠的速度与位置

假设在 t 时刻,n 只蝙蝠分散在 D 维搜索空间内,X_i^t、V_i^t 表示蝙蝠 i 的位置和速度,则在 $t+1$ 时刻,蝙蝠的位置和速度 X_i^{t+1}、V_i^{t+1} 更新规则如下:

$$f_i = f_{\min} + (f_{\max} - f_{\min})\beta \tag{15}$$

$$V_i^{t+1} = V_i^t + (X_i^t - X^*)f_i \tag{16}$$

$$X_i^{t+1} = X_i^t + V_i^{t+1} \tag{17}$$

式中:f_i 为当前时刻蝙蝠 i 的脉冲频率,$f_{\min}$ 和 $f_{\max}$ 表示蝙蝠脉冲频率的最小值和最大值,β 表示[0,1]之间的一个随机数,X^* 为当前的全局最优解。

蝙蝠算法在寻优过程中,同其他启发式算法一样,可以进行局部搜索产生新的位置,更新公式如下:

$$X_{new}(i) = X_{old}(i) + \varepsilon A^t \tag{18}$$

式中:ε 表示[-1,1]之间的一个随机数,A^t 为当前时刻全部蝙蝠响度的均值。

(二)蝙蝠的脉冲与频率

当蝙蝠接近目标时,脉冲响度会减少,同时增加发射频率。设 r_i^0 为蝙蝠 i 的初始发射频率,A_i^T 代表蝙蝠 i 在 T 代的脉冲响度,则蝙蝠在 $T+1$ 代的脉冲响度 A_i^{T+1} 和发射频率 r_i^{T+1} 为:

$$A_i^{t+1} = \alpha A_i^t \tag{19}$$

$$r_i^{t+1} = r_i^0[1 - \exp(-\gamma t)] \tag{20}$$

式中:$\alpha \in [0,1]$ 为响度衰减系数,$\gamma > 0$ 为频率增加系数。对于任意的 α 和 γ 有:$\lim_{t\to\infty} A_i^t = 0$,$\lim_{t\to\infty} \gamma_i^t = r_i^0$。

(三)编码与解码

传统的蝙蝠算法难以直接用来计算组合优化问

题,需要对蝙蝠的编码方式进行重新定义。本文采用如下的编码方式,设有 N 个客户,M 个车场,车场配备的车辆数为 K_m,则总车辆数 $W=\sum_{m=N+1}^{N+M} k_m$,令维度 $w=N+W$。用"A、B、C"代表车场,在蝙蝠的位置中随机插入车场编号。例如,A_1[123]B_3[456]C_2[789]代表车辆的行驶路径,车场A编号为1的车辆路径为:A-1-2-3-A;车场B编号为3的车辆路径为:B-4-5-6-B;车场C编号为2的车辆路径为:C-7-8-9-C。经过不断的循环迭代,找到能够满足约束条件并且结果最好的解码方案,以此来制定配送方案。

(四)交叉操作

蝙蝠算法在前期通过速度和位置的更新具有快速收敛、鲁棒性强等优点,但同时也有"早熟"的现象[13]。本文把遗传算法中的自适应交叉操作引入到蝙蝠算法中,该方法能够提高蝙蝠算法的寻优性能。

本文采用顺序交叉操作。随机选取两个蝙蝠个体,将蝙蝠1中与蝙蝠2中后三个相同的客户编码删除,得到删除编码后的蝙蝠1,以同样的方法得到删除编码后的蝙蝠2。然后将蝙蝠2中的后三位客户编码顺序插入蝙蝠1中的空缺编码位,得到新的蝙蝠个体1,以同样的方法获得新的蝙蝠个体2。如图2所示:

选择两个蝙蝠个体	A_1 1 2 3 B_3 4 5 6 C_2 7 8 9	A_1 4 5 2 B_3 1 8 7 C_2 6 9 3
删除编码后的蝙蝠个体	A_1 1 2 3 B_3 4 5 6 C_2 7 8 9	A_1 4 5 2 B_3 1 8 7 C_2 6 9 3
生成两个新蝙蝠个体	A_1 1 2 6 B_3 4 5 9 C_2 7 8 3	A_1 4 5 2 B_3 1 7 8 C_2 6 9 3

图2 交叉操作

式中,交叉概率并不是一个不变的值,而是根据每代种群中最佳蝙蝠个体的适应值和其它蝙蝠个体的之间差额关系决定的。交叉概率公式[14]见下式:

$$pc=\frac{1}{1+\exp(k(\min-s))} \tag{21}$$

式中:min是个体的最小适应值,f是种群中所有适应度值低于平均适应度值个体的均值。$k>0$,exp()为以e为底数的指数函数。因为 $min-f<0$,所以 $0<\exp(k(\min-f))<1$。当min越靠近f,表示种群内的适应度值差异越小,则 pc 越靠近于0.5,此时优秀个体的结构能够保持稳定,减少了求解的时间。当 min 越偏离f,表示种群内适应度差异越大,则 pc 越接近于1,此时有助于依靠交叉形成新的个体,扩大了搜寻空间。

(五)混合蝙蝠算法实现过程

步骤1:初始化各参数,设置种群数量 n,迭代次数 iter_max,蝙蝠初始响度 A_i^0,脉冲发射频率 r_i^0,脉冲发射频率的最大值、最小值 $f_{\max}$、$f_{\min}$,响度衰减系数 α,频率增加系数 γ。

步骤2:通过约束条件确定每只蝙蝠路径上的运输车辆及车辆的运输路线,将适应度值最优的蝙蝠作为当前最优蝙蝠 X^*。

步骤3:通过公式(15)至(17)调整频率,更新当前蝙蝠的速度和位置,产生新的解。

步骤4:随机产生一个数 rand1,若 rand1$>r_i$,则对当前最优解进行局部搜索,生成一个新解 X_{new}。

步骤5:随机产生一个数 rand2,若 rand2$<A_i$,且 X_{new} 的适应度值好于 X^*,则将 X_{new} 视为新解。

步骤6:按照公式(19)、(20)更新 A_i 和 r_i,排列所有蝙蝠找出最优解。

步骤7:把当前最优解的客户编码进行自适应交叉操作,若适应度值优于当前最优解则更新当前最优解。

步骤8:令 iter=iter+1,如果 iter<iter_max,则返回步骤3。

四、模型仿真及分析

(一)问题描述

某物流公司拥有的车场数量 M 为3,每个车场配备2辆相同型号的车辆,客户点的数量 N 为16,每辆运输车辆的车速 V_0 为40 km/h,车辆的单位运输成本 C_0 为每公里5元。其中,3个车场的具体坐标分别为(20,20),(75,45),(50,80),每辆车的最大运输量为9 t。具体的数据如表1所示:

表1 车场信息

车场	X坐标	Y坐标	车辆数/辆	车辆载重/t
1	20	20	2	9
2	75	45	2	9
3	50	80	2	9

16个客户地址的具体坐标、货物需求量、客户期望的时间窗以及服务每个客户所需要的时间,如表2所示:

表 2　客户信息

N	X_i	Y_i	q_i	ET_i	LT_i	ST_i	N	X_i	Y_i	q_i	ET_i	LT_i	ST_i
1	78	30	3	2	3.5	3.5	9	55	35	3.5	2	3	2
2	39	31	3	4	5	3	10	88	55	2.5	4	5	1.5
3	20	76	2	2	3	0.5	11	9	12	3	0.5	2	1.5
4	65	98	2	1.5	3	2	12	35	93	2.5	2	3.5	2
5	95	23	1.5	5	6.5	3	13	73	85	4	3	5	3
6	90	9	4.5	1	2.5	1	14	62	53	2	3	4.5	1
7	33	65	3.5	3	4	2.5	15	33	13	3	1	2	1.5
8	79	75	2.5	5	6.5	1.5	16	11	20	2.5	3	4.5	0.5

(二)仿真结果

为了检验本文多车场车辆调度数学模型与混合蝙蝠算法的可行性,本文在 MATLAB 2016a 版本上对传统蝙蝠算法、混合蝙蝠算法进行编程实现。2 种算法均在 Intel Core i3-3217U 1.80 GHz (8.00 GB RAM),操作系统为 64 位 Win8.1 的环境下进行。2 种算法的设置参数如下:$\alpha = 0.95$,$\gamma = 0.95$,$A_0 \in [0,1]$,$r_0 \in [0,1]$,蝙蝠种群数 $n = 100$,迭代次数 100。时间窗惩罚参数设置如下:$a_i = 0.05$,$b_i = 0.5$,$m_i = 0.02$,$n_i = 0.03$。将 2 种方法计算的结果进行比较,其中车场 1,车场 2,车场 3 分别用数字 17,18,19 表示,具体结果如表 3 所示:

表 3　传统蝙蝠算法与本文方法对比

方案	车辆路线	客户不满意度	物流成本
传统蝙蝠算法	车场 1:[17-15-14-13-17] [17-11-16-7-17] 车场 2:[18-4-10-5-18] [18-6-1-18] 车场 3:[19-3-12-8-19] [19-9-2-19]	9.11%	3931.1
混合蝙蝠算法	车场 1:[17-11-16-2-17] [17-15-7-17] 车场 2:[18-6-1-5-18] [18-4-14-8-18] 车场 3:[19-3-12-13-19] [19-9-10-19]	4.54%	3505.95

两种算法求解得出的车辆运输路线对比如图 3、图 4 所示:

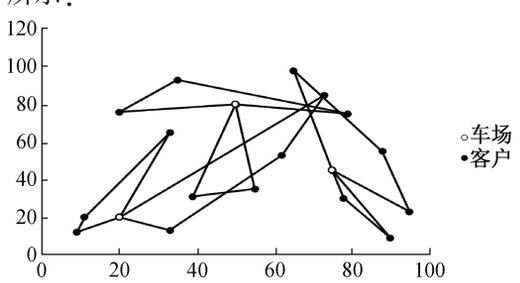

图 3　传统蝙蝠算法路线图

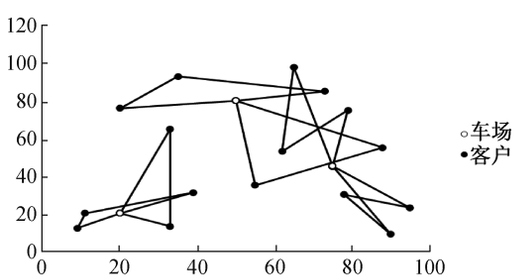

图 4　混合蝙蝠算法路线图

两种算法求解过程中的成本收敛情况对比如图 5 所示:

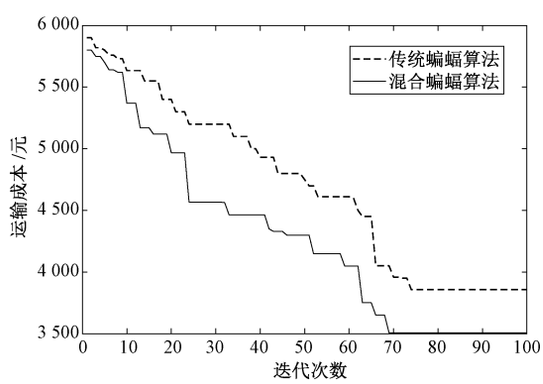

图 5　成本收敛对比图

从两种算法的成本收敛过程来看,本文采用的混合蝙蝠算法与传统的蝙蝠算法相比拥有更强的寻优能力,这主要得益于将自适应交叉操作引入到蝙蝠算法中,该方法可以对结果较差的蝙蝠进行交叉操作,从而能够扩大搜索范围。

(三)对比分析

通过对比传统的蝙蝠算法和本文的求解结果方法可以得出结论:

(1)从客户的利益来看,本文的方法在客户不满

意度方面比传统的蝙蝠算法降低了4.57%,这表明本文的方法在减少客户不满意度方面的效果是显著的。

(2)从运营商的利益来看,本文的方法在配送成本优化方面不仅提高了求解的速度,而且运输成本比传统的蝙蝠算法减少了425.15元,这表明本文的方法在减少配送成本方面优于传统的蝙蝠算法。

综上,从物流运营商的长远利益来看,本文的方法在减少配送成本的同时降低了客户的不满意度,这有助于物流运营商与客户维持长期的合作关系,使物流运营商能够持久盈利。

五、结束语

本文综合考虑了客户和物流运营商的利益,设计了适合本文研究的多目标车辆调度模型。并对蝙蝠算法进行了改进,使算法更适用于求解本文的数学模型,并经过仿真检验了本文算法的可行性。本文是蝙蝠算法在多车场车辆调度领域的成功运用,为求解多车场领域车辆调度难题进行了有益的探索。

须指出的是,在现实生活中物流运输的过程必然伴随着多种扰动因素,因此,将扰动因素与车辆调度相结合是下一步研究的方向。

参考文献

[1]Suzuki Y. A dual-objective meta heuristic approach to solve practical pollution routing problem[J]. International Journal of Production Economics, 2016, 176(30): 143-153.

[2]Afshar-Nadjafi B, Afshar-Nadjafi A. A constructive heuristic for time-dependent multi-depot vehicle routing problem with time-windows and heterogeneous fleet[J]. Journal of King Saud University Engineering Sciences, 2017, 29(1): 29-34.

[3]Zhou L, Baldacci R, Vigo D, et al. A multi-depot two-echelon vehicle routing problem with delivery options arising in the last mile distribution[J]. European Journal of Operational Research, 2017, 265(2): 765-778.

[4]鲁建厦, 洪欢蕾, 陈青丰. 考虑供给商品价格的多车场车辆路径问题[J]. 浙江工业大学学报, 2016, 44(5): 553-558.

[5]Yang X S. A new metaheuristic bat-inspired algorithm[J]. Computer Knowledge and Technology, 2010, 284: 65-74.

[6]Zhou Y, Xie J, Zheng H. Praveen Agarwal. A hybrid bat algorithm with path relinking for capacitated vehicle routing problem[J]. Mathematical Problems in Engineering, 2013, 9: 831-842.

[7]Osaba E, Yang X S, Diaz F, et al. An improved discrete bat algorithm for symmetric and asymmetric traveling salesman problems[J]. Engineering Applications of Artificial Intelligence, 2016, 48: 59-71.

[8]殷亚, 张惠珍. 求解带硬时间窗的多目标车辆路径问题的多种混合蝙蝠算法[J]. 计算机应用研究, 2017, 34(12): 3632-3636.

[9]孙奇, 马良. 基于客户满意度的车辆路径问题的混合蝙蝠算法[J]. 上海理工大学学报, 2019, 41(2): 160-166.

[10]丁秋雷, 胡祥培, 姜洋. 物流配送受扰延迟问题的干扰管理两阶段决策方法[J]. 运筹与管理, 2012, 21(6): 84-93.

[11]郭森, 秦贵和, 张晋东, 等. 多目标车辆路径问题的粒子群优化算法研究[J]. 西安交通大学学报, 2016, 50(9): 97-104.

[12]朱颢. 基于改进蝙蝠算法的带模糊需求的车辆路径问题[J]. 计算机测量与控制, 2017, 25(7): 276-281.

[13]Fister I, Rauter S, Yang X S, et al. Planning the sports training sessions with the bat algorithm[J]. Neurocomputing, 2015, 149: 993-1002.

[14]符俊波, 马慧民, 张爽, 等. 有垃圾量变动的生活垃圾收运车辆调度干扰管理研究[J]. 上海理工大学学报, 2017, 39(4): 368-375.

[责任编辑 王云江]

Research on multi-depot vehicle scheduling based on hybrid bat algorithm

CAO Qing-kui[1,2], GAO Ya-wei[1], REN Xiang-yang[1]

(1. School of Management Engineering and Business, Hebei University of Engineering, Handan 056038, China;
2. Faculty of Economics and Management, Langfang Normal University, Langfang 065000, China)

Abstract: Multi-depot vehicle scheduling problem is a NP hard problem in logistics distribution, and it is also the development trend of modern logistics. Aiming at the problem of multi-depot vehicle scheduling, this paper took the interests of customers and logistics operators into consideration, built a multi-objective vehicle scheduling mathematical model with the objective of minimum customer dissatisfaction and minimum transportation cost, and normalized the objective function to transform the multi-objective problem into a single objective problem. The traditional bat algorithm has some shortcomings in local search ability, the crossover operation of genetic algorithm was introduced into bat algorithm, and a hybrid bat algorithm solution model was proposed. Through the simulation of MATLAB software, the simulation results are compared with the traditional bat algorithm. The results show that this method is feasible and superior to the traditional bat algorithm.

Key Words: multi-depot; hybrid bat algorithm; multi-objective; vehicle scheduling