

文章编号:2095-7386(2016)02-0072-07

DOI:10.3969/j.issn.2095-7386.2016.02.013

基于多目标 VRP 的离散型萤火虫算法研究

董文波,周 康,刘 朔,高全胜

(武汉轻工大学 数学与计算机学院,湖北 武汉 430023)

摘 要: 本文以车辆路径问题为准,对萤火虫算法进行研究。建立了以最小化车辆数量和行驶路程为目标的多目标规划数学模型,提出一种结合变邻域搜索算法的离散型萤火虫算法。该离散型萤火虫算法的特色之处在于:重新定义了个体的生成方式和距离移动方式;采用变邻域搜索技术以增强算法的邻域搜索能力;在搜索过程中采用随机个体替代种群中的重复个体以维持种群的多样性;采取精英策略记录迭代过程中的最优解。通过不同规模的 Solomon 算例进行仿真实验,结果表明本文所提算法无论是在车辆数量还是行驶路程的求解质量都取得了很好的效果。

关键词: 离散型萤火虫算法;多目标;车辆路径问题;变邻域搜索;精英策略;

中图分类号: TP 391.9

文献标识码: A

Multi-objective vehicle routing problem based on DGSO algorithm

DONG Wen-bo, ZHOU Kang, LIU Shuo, GAO Quan-sheng

(School of Math and Computer, Wuhan Polytechnic University, Wuhan, 430023, China)

Abstract: In this paper, the multi-objective mathematical model is established in order to minimize the number of vehicles and driving distance in vehicle routing problem, and a discrete glowworm swarm optimization algorithm (DGSO) combined with variable neighborhood search is proposed. The characteristic of DGSO algorithm is that individual generation and movement mode are redefined; variable neighborhood search technique is adopted to balance the global search ability and local development ability of the algorithm; random individuals take place of repeated individuals in order to maintain the diversity of the population in the search process; the elite strategy records the optimal solution in the iterative process. Simulation results for different scale Solomon cases show that DGSO algorithm both in the number of vehicles and driving distance have achieved good results.

Key words: Discrete glowworm swarm optimization algorithm; Vehicle routing problem; Multi-objective optimization; Variable neighborhood search; Elite strategy

收稿日期:2015-4-8

作者简介:董文波(1992-),男,硕士生,E-mail:1623572531@qq.com

通信作者:周康(1965-),男,教授,E-mail:zhoukang_wh@163.com

基金项目:国家自然科学基金项目资助课题(项目编号 61179032);粮食公益性行业科研专项(项目编号 201513004-3);武汉轻工大学研究生教育教学改革研究与实践重点项目(项目编号 YZ2015002)。

1 引言

随着现代经济一体化和计算机与通信技术的不断进步,网上购物成为流行趋势,使得物流业得到了蓬勃发展。在现代社会经济中,物流承担着企业价值链的基础活动,其支出费用占生产成本的比重逐渐增加,缩减物流部分的支出费用,能够让企业在行业竞争中占得优势地位。所以现在许多生产企业已经把目光转向了物流配送供应链问题上。而车辆路径问题(Vehicle Routing Problem, VRP)是物流配送研究的重要内容^[1],对它进行合理的调度和优化,能够改善企业在物流经济方面的支出^[2]。VRP 的内容是对于一些地理位置分散的客户^[3],配送中心负责安排车队,组织合理的行驶路线向这些客户分送货物^[4],以满足这些客户不同单位的货物需求。在规定的约束限制下,取得运输成本最小。由于在实际生产和生活中,邮局投递安排、公交车路线安排、电力调度问题、快件的收发、航空和铁路时间表制定以及废品收集等很多实际生活问题都可以抽象映射成为 VRP。由此可见,VRP 在现实研究中具有非常重要的意义^[5]。所以车辆路径问题被提出后,很快便引起了包括数学、计算机学科等在内的众多学科和相关领域的专家极大重视,至今仍然是研究的热点问题之一^[6-8]。

由于 VRP 已经被证明是 NP-Hard 问题,所以求解 VRP 的算法需要在合理的时间内得到尽可能优的解,这样就必须充分利用问题本身的限制来改进算法。复杂的 VRP 限制条件造成 VRP 分支众多,其中某一分支的研究成果可对其他分支的解决提供借鉴,能促进各个分支的共同发展。因此 VRP 在理论研究和学术研究中具有重要意义和价值。近些年来学者们对其进行实验研究,各种类型的求解算法层出不穷。目前这些算法大体上可分为包括分支定界法、整数规划和动态规划等在内的确定性型算法和包括构造法、两阶段法及改进型算法等在内的启发式算法这两大类^[9]。其中,确定型算法能够准确的求出问题的解,但是这些算法的缺点是只能解决小规模问题,并且消耗的时间比较长;启发式算法虽然可以求得问题的解,但是求解精度不高。另外,还有一类就是这二十年内新发展起来的群智能优化算法,例如:基于“优胜劣汰”规则的遗传算法^[10]、模仿蚂蚁行为的人工蚁群算法^[11]、基于物理现象的模拟退火算法^[12]、模仿鸟类飞行的粒子群优化算法^[13]以及禁忌搜索算法^[14]等,这些方法在某些文献分类

中亦被叫做亚启发式算法。

萤火虫群优化算法(Glowworm Swarm Optimization, GSO)是一种新型群智能仿生优化算法,由印度两位学者 Ghose 和 Krishnanand 提出^[15,16]。GSO 算法的仿生原理是:自然界中的萤火虫通过尾部的荧光素发出亮光,以此来吸引同伴向自己移动,以达到求偶或觅食的目的。萤火虫算法的特色之处在于每一只萤火虫本身都携带不同单位数量的荧光素^[17],在萤火虫种群中飞行的过程中能够进行实时的更新,并可利用自身的感知范围来决定搜索范围及路径。GSO 算法的特点是仿生计算模式简洁、算法的稳健性较强、可调参数较少。目前,GSO 算法已经成功应用于信号源定位、复杂多目标函数优化、感应器噪音处理、数值优化计算、有害气体泄漏检测、聚类分析和多模态函数优化等方面,并表现出了良好的性能。

2 VRP 和 GSO

2.1 VRP 数学模型

假设存在一个配送中心仓库对车辆进行调配任务,其中有 V 辆车可提供配送服务,容量分别为 $q_v (v = 1, 2, \dots, V)$ ^[18];现需对 L 个客户进行配送任务,客户分别以序号 $1, 2, \dots, L$ 来表示,若第 i 个客户的货物需求量为 $N_i (i = 1, 2, 3, \dots, L)$,且客户最大货物需求量小于当前配送车辆的载重量,求最少需要安排的车辆数和最短车辆行驶距离。

为便区别,以编号 0 代表配送中心,客户则分别以编号 $1, 2, 3, \dots, L$ 数字表示^[19],任务及配送中心均以点 $i (i = 0, 1, 2, \dots, L)$ 来表示。定义变量如下:

$$y_{ki} = \begin{cases} 1 & \text{发货点 } i \text{ 的任务由车 } k \text{ 完成} \\ 0 & \text{否则} \end{cases}$$

$$x_{ijk} = \begin{cases} 1 & \text{车 } k \text{ 从点 } i \text{ 行驶到点 } j \\ 0 & \text{否则} \end{cases}$$

用 c_{ij} 代表客户 i 到客户 j 的路程距离^[20]。最后可对车辆路径问题进行数学建模为:

$$\text{Min } Z = \sum_i \sum_j \sum_k c_{ij} x_{ijk}$$

$$\text{Min } V = \sum_{j=0}^n \sum_{k=1}^K x_{0jk}$$

$$s. t. \begin{cases} \sum_i N_i y_{ki} \leq q_k & \forall k \\ \sum_k y_{ki} = 1, i = 1, \dots, L \\ \sum_i x_{ijk} = y_{kj}, j = 0, 1, \dots, L, \forall k \\ \sum_j x_{ijk} = y_{ki}, i = 0, 1, \dots, L, \forall k \\ X = (x_{ijk}) \in S \\ x_{ijk} = 0 \text{ 或 } 1, j = 0, 1, \dots, L, \forall k \\ y_{ki} = 0 \text{ 或 } 1, i = 0, 1, \dots, L, \forall k \end{cases}$$

在该数学模型中,数学表达式表示的约束条件的含义为:被配送的客户都能够得到车辆的配送服务,但是每个客户只能在配送路径中出现一次^[21]。同时每辆车的最大载重量要能够满足该车服务的所有客户的总需求量。规划目标是找出尽可能少的车辆数参加配送任务,使得运输代价最小,即尽可能使所有车辆路径之和 Z 和车辆数 V 达到最小。

2.2 基于变邻域搜索的 GSO 及其数学模型

标准萤火虫算法中,在初始阶段,每只萤火虫随机分布在解空间中的某个位置上^[26],其尾部均含有相同浓度的 l_0 ,并且拥有相同的邻域半径 r_0 ^[22]。在萤火虫移动的过程中,每只萤火虫个体携带浓度为 l_i 的荧光素,通过发光来吸引四周其他的萤火虫,并拥有各自的邻域半径 r_d^i ($0 < r_d^i \leq r_s$)。处在萤火虫邻域半径内,并且亮度比当前萤火虫更亮的萤火虫组成萤火虫的邻域集合 N_i ^[23]。在集合中,亮度越亮的萤火虫就越能吸引其他萤火虫向它处的位置方向飞行。萤火虫邻域半径的大小受邻居数量的影响,在迭代过程中可动态调整。当萤火虫邻居密度小时,邻域半径就会变大;相反,当萤火虫邻居密度大时,邻域半径则会相应缩小。最终大部分萤火虫都会飞聚在解空间的多个最佳位置上^[24-26]。

标准 GSO 包括以下 5 个步骤:

Step1 将函数目标值 $J(x_i(t))$ 用公式(1)转化为相应的荧光素值 $l_i(t)$;

Step2 构建每只萤火虫的邻域集合。种群中的每只萤火虫将处于各自邻域半径 $r_d^i(t)$ 内并且荧光素值 $l_i(t)$ 比自身大的个体加入到自己的邻域集合 $N_i(t)$ 内;

Step3 计算路径选择概率,确定萤火虫的飞行路径。用公式(2)计算萤火虫个体 i 对于自身邻域集合内其他萤火虫的路径选择概率 $p_{ij}(t)$;

Step4 向目标萤火虫飞行。对于萤火虫 i ,在其邻域集中选择移动对象 j ,向之方向飞行,根据公式

(3)刷新飞行过后萤火虫 i 所处的位置;

Step5 根据萤火虫邻居密度的影响,由公式(4)调整萤火虫 i 的邻域半径的值。

$$l_i(t) = (1 - \rho)l_i(t-1) + \gamma J(x_i(t)) \quad (1)$$

$$p_{ij}(t) = [l_j(t) - l_i(t)] / \sum_{k \in N_i(t)} l_k(t) - l_i(t) \quad (2)$$

$$x_i(t+1) = x_i(t) + s * \left[\frac{x_j(t) - x_i(t)}{\|x_j(t) - x_i(t)\|} \right] \quad (3)$$

$$r_d^i(t+1) = \min\{r_s, \max\{0, r_d^i(t) + \beta(n_i - |N_i(t)|)\}\} \quad (4)$$

其中

$$\begin{cases} \rho \in [0, 1] \text{ 荧光素挥发因子} \\ t \text{ 当前迭代次数} \\ x_i(t) \text{ 第 } i \text{ 只萤火虫在时刻 } t \text{ 所处的位置} \\ \gamma \in [0, 1] \text{ 荧光素更新率} \\ r_s \text{ 萤火虫个体的最大邻域半径} \\ \beta \text{ 邻域半径更新率} \\ n_i \text{ 萤火虫个体邻域集内包含的萤火虫数目的阈值} \end{cases}$$

s 为萤火虫每次飞行的前进步长^[17-18]。

3 离散型萤火虫算法的设计

标准 GSO 只能解决连续论域中的函数优化问题,为解决 VRP,要离散化 GSO,所以下面提出离散型萤火虫算法(DGSO)。在 DGSO 算法中,萤火虫个体 i 被编码后表示一条可行路径的编码,则萤火虫在向目标萤火虫飞行的过程中,每一步的飞行,其飞行动作对应于相应编码的一种变换操作^[10]。

3.1 路径编码、解码和构造初始萤火虫群的设计说明

设定位置编码采用自然数编码机制($c_1, c_2, \dots, c_n$),其中 $c_1, c_2, \dots, c_n$ 是自然数 1 到 n 的自然排列,坐标 c_i 代表一个顾客点,坐标位的顺序表示车辆访问顾客点的次序。解码时,按照位置编码顺序依次将顾客点插入到路径中;若插入的顾客点违反车辆最大载重量约束,则重新使用一辆车来服务该顾客点;直到所有的顾客被服务。

初始萤火虫群采用 N 个顾客结点随机排列的方式来产生。若萤火虫群规模过大则计算量增加,影响算法收敛的时间;若萤火虫群数量过少,则智能搜索算法的搜索性能下降,所以将问题规模和萤火虫群规模相关联,动态确定萤火虫群规模。

3.2 萤火虫个体间距离的创新

两个编码的差异性是编码对应的顾客点之间的

差异性累积而成的,因此,如何刻画两个编码对应的顾客点之间的差异性就是设计萤火虫个体间距离的核心问题。由于两个顾客点之间的实际差异就是这两个顾客点之间的实际路程长度,因此,为了更好地刻画并反映萤火虫个体之间真实的位置距离,对两个顾客点之间的实际差异选择了两个顾客点的弧距离,由此设计了新的萤火虫距离。

设萤火虫个体 i 和 j 在 t 的编码分别为: $x_i(t) = (x_{i1}, x_{i2}, \dots, x_{in})$ 和 $x_j(t) = (x_{j1}, x_{j2}, \dots, x_{jn})$, 则萤火虫个体 i, j 在 t 时刻的编码差异度可以按如下形式进行定义:

$$\delta_{ij}(t) = \frac{\sum_{k=1}^n d_{ij}(t, k)}{M} \quad (5)$$

其中 $d_{ij}(t)$ 为个体 i, j 在 t 迭代每一维上的弧距离; M 为权矩阵中每一行的最大值之和。所以 $\delta_{ij}(t) \in [0, 1]$; 因为权矩阵在数学形式上是对称的, 所以 $\delta_{ij}(t)$ 也是对称的, 即 $\delta_{ij}(t) = \delta_{ji}(t)$ 。

基于上面定义的针对两只萤火虫编码的差异度, 可以设计萤火虫个体 i 与萤火虫个体 j 在 t 时刻的距离计算公式 $Dist_{ij}(t)$ 为:

$$Dist_{ij}(t) = c \times \delta_{ij}(t). \quad (6)$$

其中 c 为一个常数, $\delta_{ij}(t)$ 为萤火虫个体 i 和 j 编码的差异度。

该距离既反映了萤火虫编码实际的差异性, 也为下面的萤火虫位置更新规则的设计奠定了基础。

3.3 萤火虫位置更新规则的创新

为了在离散化萤火虫算法过程中继承连续萤火虫算法的位置更新思想, 在此基于萤火虫个体间距离的概念设计了移动步长 s , 给出了向移动对象移动 s 步的更新规则。

在移动阶段, 首先产生一个 s 维的随机数 $r(s) = (r_1, r_2, \dots, r_s)$, 其中 $r_i \neq r_j (i, j \in [1, n])$ 。对于顾客数为 n 的 VRP, 位置编码是一个 n 维序列, 对萤火虫位置编码 $x = (x_1, x_2, \dots, x_n)$, 移动目标 $y = (y_1, y_2, \dots, y_n)$ 和 s 维的随机数 r_k , 位置编码更新如下:

$$\begin{aligned} temp &= x[r_k]; \\ x[r_k] &= y[r_k]; \\ x[y[r_k]] &= temp, \end{aligned}$$

其中 $k \in [1, s]$ 。该编码更新方式继承了标准 GSO 的位置更新公式。根据 3.2 小节求出萤火虫 x 和萤火虫 y 之间的距离, 萤火虫 x 在每次迭代过程中向萤火虫 y 飞行 s 步长的距离, x 和 y 编码的差异

性会相应减少 s 个单位, 经过一定的迭代过程, x 和 y 的编码会趋于一致性, 即萤火虫 x 飞到萤火虫 y 所在的位置上。

3.4 引入变邻域搜索技术

引入变邻域搜索技术可以增强 DGSO 算法的邻域寻优能力。变邻域搜索的具体操作步骤是: 将更新后的萤火虫编码个体作为初始解, 随机选择以下两种邻域操作中的一种操作进行邻域搜索, 如果当前解得到优化, 但在优化后期一直保持不变, 则说明达到了局部最优解, 结束邻域搜索过程, 进行精英比较; 如果当前解在算法所规定的迭代次数内仍没有得到优化时, 则结束邻域搜索过程。由于算法不断迭代, 萤火虫群中的萤火虫个体会飞于相同的位置, 即表现为萤火虫个体的编码会逐渐相同, 因此在算法每次迭代过程中, 随机产生一组萤火虫个体编码, 代替重复编码个体, 维持萤火虫种群的多样化, 使得萤火虫有更广阔的空间以飞行。

邻域操作:

(1) 交换: 对随机产生的两个不同的数, 交换这两个数代表的编码。

(2) 反转: 对随机产生的两个不同的数, 将这两个随机数之间的编码反转。

在标准萤火虫算法中, 迭代过程中产生的最好的状态的解随着迭代的进行, 在迭代后期可能不会再出现, 造成该次迭代运算的计算精度下降。为了克服以上缺陷, 需要引入精英策略来记录解的最优状态。在算法每次迭代中, 将当代最优位置的萤火虫和精英萤火虫对比, 如果当代最优位置的萤火虫比精英萤火虫所处位置更好, 则更新精英萤火虫, 否则保持精英萤火虫位置不变。

3.5 基于变邻域搜索的 DGSO 算法

基于变邻域搜索的 DGSO 算法(以下简称“本算法”)的具体步骤如下:

Step1 初始化萤火虫种群, 并设定以下参数: 萤火虫群大小, 最大迭代次数, $l_i(0), r_d^i(0), \gamma, r_s, \rho, \beta, s, n_i$;

Step2 对于萤火虫个体 i , 计算萤火虫个体 i 路径编码代表路径的目标函数值, 根据式(1)将目标函数值转化为萤火虫荧光素值;

Step3 根据式(5)和式(6)计算萤火虫个体 i 与其他萤火虫个体之间的距离; 在邻域半径 $r_d^i(t)$ 内, 选择亮度比自己亮的萤火虫加入到邻域集 $N_i(t)$ 内;

Step4 计算路径选择概率, 确定萤火虫的飞行路径。根据公式(2)计算萤火虫个体 i 对于自身邻域

集合内其它萤火虫的路径选择概率 $p_{ij}(t)^{[19]}$ 。根据概率机制(轮盘赌方法)选定目标萤火虫;若邻域集为空,则说明该萤火虫是一个局部最亮萤火虫,编码表示是问题的一个局部最优值,则转到 Step6 进行邻域优化;

Step5 产生一个 s 维序列 $r = (r_1, r_2, \dots, r_s)$,其中 $r_i \neq r_j (i, j \in [1, n])$ 。根据 3.3 节更新萤火虫个体 i 编码 $x_i(t)$ 上的每一维数据;

Step6 对新得到的萤火虫个体编码用变邻域搜索算法进行局部优化;

Step7 根据萤火虫邻居密度的大小和公式(4)动态调整萤火虫个体 i 的邻域半径的大小;

Step8 判断当前迭代次数是否达到了最大迭代次数或者违反算法结束条件,如果未达到结束条件,返回到 Step2,赋值 $i = i + 1$,按步骤顺序继续迭代;否则,程序终止,输出最优萤火虫代表的全局最优解和全局最优路径。

4 仿真实验与分析

为了测试本算法的正确性和有效性,且使测试结果更具可比性和说服力,使用公开的标准测试算例(benchmark instances)对所提出的算法进行测试。数据来源于国际上公认的 VRP 数据库(<http://www.branchandcut.org/>)。采用 VC++6.0 进行编程,实验环境为:

处理器:Pentium(R) Dual - Core;
CPU:E53002.60GHz;
内存:2.00GB;
操作系统:Windows732 位操作系统。

4.1 DGSO 算法参数的选取

本算法的参数取值可分为固定参数和可调参数^[27-28]。其中迭代次数对算法收敛性能的影响比较复杂,其取值不能过小,否则难以得到满意的结果,但也并不是值越大越好,迭代次数过大,运算时间就会加长。其实际取值需要根据问题的规模 n 而设定。如果想要获得或接近最优解就需要将迭代次数取值大些。根据不同的问题规模,设置不同的迭代次数做多次运行。参数调优的参考文献[27]和文献[28],并根据 VRP 问题做了不同的调整。我们分别取了两组参数,并对这两组参数进行了算法对比实验,对于每组实验参数,程序分别执行 20 次。具体结果见表 4-1。

参数组 1: $\rho=0.5, \gamma=0.5, \beta=0.09, n_1=6, s=3, l_0=5$, 种群 150;

参数组 2: $\rho=0.3, \gamma=0.7, \beta=0.08, n_1=10, s=3, l_0=5$, 种群 200。

表 4-1 参数组对比实验结果

算例	出现的最优解		出现最优解次数		平均值	
	参数组 1	参数组 2	参数组 1	参数组 2	参数组 1	参数组 2
An55k9	1074.96	1074.46	24	30	1075.76	1074.46
Bn45k5	753.96	753.96	6	15	756.62	755.67
En51k5	524.61	524.61	8	30	534.94	524.61
Fn72k4	244.91	242.44	10	21	250.34	245.36
Pn50k7	560.03	559.86	15	25	561.52	560.87

从表 4-1 可以看出,从出现的最优解来看,在算例 Bn45k5 求解结果相同,而在其余的算例上,参数组 2 出现的最优解要优于参数组 1 出现的最优解;从出现最优解次数来看,参数组 2 出现最优解的次数要多于参数组 1,这说明参数组 2 在解空间中的搜索能力要强于 1。所以采取参数组 2 作为算法在接下来的仿真实验中的参数。

4.2 实验对比与分析

4.2.1 与现有算法的对比实验

首先,根据文献[14]中的实例 1,对其做了仿真实验,并与文献[14]中提出的禁忌搜索算法做了对比。具体对比结果见表 4-2。(注:“/”表示“DGSO/禁忌搜索算法”;“C”表示配送总距离;“V”表示使用的车辆数;“T”表示程序运行的时间)

表 4-2 实例 1 的对比实验结果

计算次序	1	2	3	4	5	6	7	8	9	10	平均
C	109/106	109/107	109/103	109/104	109/112	109/108	109/107	109/110	109/104	109/109	109/107
V	3/4	3/4	3/4	3/4	3/4	3/4	3/4	3/3	3/4	3/3	3/3.8
T	2	2	2	2	2	2	2	2	2	2	2

从表 4-2 可以看出,本算法对实例 1 求解 10 次,每次均得到较高质量的解,其求得的车辆行驶距

离平均值为 109.7Km,平均使用车辆数为 3。车辆对应的配送路径具体为:

路径 1: (0 - 18 - 17 - 3 - 4 - 14 - 0); 路径 2: (0 - 20 - 11 - 6 - 19 - 7 - 1 - 5 - 0);

路径 3: (0 - 10 - 2 - 12 - 9 - 15 - 16 - 13 - 8 - 0)。

与文献[14]中提出的禁忌搜索算法相比,本文算法求出的车辆数要少于禁忌搜索算法;在车辆数相同时,本算法求出的车辆行驶路径更短,即总体优

于禁忌搜索算法。从平均值可以得出本算法的求解结果更稳定,即算法鲁棒性较强。从计算时间来看,本算法求解此规模问题仅需 2 秒,计算效率较高。

为了进一步说明本算法的优势,下面分别与爬山算法、遗传算法和模拟退火算法三种算法做对比实验。具体对比结果见表 4-3。

表 4-3 各种算法对实例 1 计算结果的对比

算法类型	爬山算法	模拟退火算法	遗传算法	本算法
平均配送总距离	128.0	109.5	140.1	109.7
平均使用车辆数	3.9	3.9	4	3
首次搜索到最终解的平均搜索次数	893.5	12012	11812	500

在表 4-3 中,从求解结果看,本算法的寻优能力明显优于爬山算法和遗传算法,虽车辆行驶距离值略高于模拟退火算法,但是本算法求出的平均车辆数低于模拟退火算法;从首次搜索到最终解的平均搜索次数来看,本算法的计算效率较高,在较短的迭代次数内就可以搜索到满意解;从算法的稳健性来看,本算法无论从车辆数还是距离值,求出的结果都更稳定,优于这三种算法。

根据以上实验对比与分析,可以归纳本算法具有以下特点:(1)对于 VRP,在问题规模一定的情况下,求得问题解的质量较高;(2)本算法寻优效果强,可以在很短时间内收敛到问题的解,计算效率较高;(3)从平均值可以看出,本算法所求得的问题结果很稳定,说明本算法求解性能比较稳定。

4.2.2 Solomon 算例对比实验

本算法采用标准数据库 ([http://www. bran-chandcut. org](http://www.bran-chandcut.org)) 中的 VRP 算例。并和人工蜂群算法 (ABC) 做了对比。其计算结果见表 4-4。

表 4-4 Solomon 算例的计算对比结果

算例	ABC	本算法	最优解	平均相对偏差
An33k5	675	662	661	0.15%
An33k6	750	742	742	0
An34k5	790	782	778	0.51%
An37k6	980	949	949	0
Bn31k5	680	676	672	0.60%
Bn34k5	799	788	788	0
Bn38k6	814	805	805	0
Bn39k5	565	553	549	0.73%
Bn41k6	857	837	829	0.97%
En22k4	375	375	375	0
En33k4	844	835	835	0
Pn23k8	531	529	529	0

在表 4-4 中,ABC 与本算法求解 VRP 的结果进行了对比,我们选取了每种类型的部分算例。实验共计算了 14 个算例,通过 ABC 与本算法的最优值对比发现除了在算例 En22k4 和算例 Pn23k8 中取得相同的最优值,其他 12 个算例中,本算法比 ABC 的最优值更好。对比中还发现,随着求解问题的规模增大,本算法的寻优能力有了明显提升。本算法的相对误差率仅在 1% 以下。针对多目标 VRP 的数学模型,应该得出的是一个最优解集。但是,由于本文算例的特殊性,使得实验过程中车辆数量和车辆路径同时达到了最优,没有表现出解的多样性,所以表 4 中只是给出了对应问题的最优解。

5 结束语

本文对 VRP 和萤火虫算法进行研究,建立了以最小化车辆数量和车辆行驶距离为目标的多目标数学模型,提出一种结合变邻域搜索算法的离散型萤火虫算法。通过对不同规模的 Solomon 算例进行仿真实验,结果表明,算法无论是在车辆数量还是行驶路程的求解质量都取得了很好的效果。从实验分析和实验对比的结果来看,本算法具有下面几项优点:(1)求解问题相同,所求得的问题解的质量较高。与其他算法相比,本文算法可以求得更少的配送车辆数;在求得车辆数相同的情况下,本算法求得的车辆行驶距离更短;(2)本文算法在解空间中的寻优能力比较强,收敛速度也较快,可以在较短的时间内收敛到问题的最优解或者满意解,计算效率较高;(3)本算法求得的结果很稳定,稳健性较强。

今后的研究工作主要是一方面进行更多的数值实验来优化和验证本算法的性能,并对多目标车辆路径问题做进一步研究;另一方面研究本算法在其他领域的可行性,以期扩大本算法的应用范围。

参考文献:

- [1] 李雪竹. 基于免疫萤火虫算法的 RFID 仓储车辆动态调度[J]. 计算机工程与应用, 2014, (6): 235-239.
- [2] 刘烨. 物流系统中智能优化技术的应用研究[D]. 合肥工业大学, 2007.
- [3] 王柏根, 汪勋, 张子臻等. 遗传算法在电力维护人员调度问题中的应用[J]. 现代计算机(专业版), 2015, (8): 3-8.
- [4] 顾坤坤. 不确定环境下物流配送有关问题的研究[D]. 中南大学, 2009.
- [5] 王刚. 遗传算法在 VRP 中的应用与研究[D]. 重庆交通大学, 2011.
- [6] 贾楠, 吕永波, 付蓬勃等. 物流配送问题中 VRP 的数学模型及其求解算法[J]. 物流技术, 2007, 26(4): 54-56.
- [7] 刘云忠, 宣慧玉. 车辆路径问题的模型及算法研究综述[J]. 管理工程学报, 2005, 19(1): 124-130.
- [8] 符卓, 陈斯卫. 车辆路径问题的研究现状与发展趋势[C]. //中国运筹学会第七届学术交流会议论文集下卷. 2004: 997-1002.
- [9] 崔雪丽, 马良, 范炳全等. 车辆路径问题(VRP)的蚂蚁搜索算法[J]. 系统工程学报, 2004, 19(4): 418-422.
- [10] 张丽萍, 柴跃廷. 车辆路径问题的改进遗传算法[J]. 系统工程理论与实践, 2002, 22(8): 79-84.
- [11] 崔雪丽, 马良, 范炳全等. 车辆路径问题(VRP)的蚂蚁搜索算法[J]. 系统工程学报, 2004, 19(4): 418-422.
- [12] 车辆路径问题的模拟退火算法[J]. 中国公路学报, 2006, 19(4): 123-126.
- [13] 李宁, 邹彤, 孙德宝等. 车辆路径问题的粒子群算法研究[J]. 系统工程学报, 2004, 19(6): 596-600.
- [14] 郎茂祥, 胡思继. 车辆路径问题的禁忌搜索算法研究[J]. 管理工程学报, 2004, 18(1): 81-84.
- [15] K N Krishnand, D Ghose. Detection of multiple source locations using a glowworm metaphor with applications to collective robotics [A]. Proceedings of IEEE Swarm Intelligence Symposium. Pisatway [C]. Pasadena California: IEEE Press, 2005. 84-91.
- [16] K N Krishnand, D Ghose. Glowworm swarm based optimization algorithm for multimodal functions with collective robotics applications [J]. Multiagent and Grid Systems, 2006, 2(3): 209-222.
- [17] 周永权, 黄正新, 刘洪霞等. 求解 TSP 问题的离散型萤火虫群优化算法[J]. 电子学报, 2012, 40(6): 1164-1170.
- [18] 何文玲, 倪郁东, 汪婷婷等. 基于混合行为蚁群算法的车辆路径问题[J]. 合肥工业大学学报(自然科学版), 2014, (7): 883-887.
- [19] 陈则辉, 刘诚, 吕品等. 不确定环境下应急物资配送问题研究[J]. 铁道科学与工程学报, 2014, (5): 82-89.
- [20] 刘诚, 陈治亚. 含装卸工调配的物流车辆配送路径问题的研究[J]. 铁道科学与工程学报, 2006, 3(4): 79-83.
- [21] 陈昌敏, 谢维成, 范颂颂等. 自适应和最大最小蚁群算法的物流车辆路径优化比较[J]. 西华大学学报(自然科学版), 2011, 30(3): 5-8.
- [22] 欧阳喆, 周永权. 自适应步长萤火虫优化算法[J]. 计算机应用, 2011, 31(7): 1804-1807.
- [23] 骆东松, 李雄伟, 赵小强等. 基于人工萤火虫的模糊聚类算法研究[J]. 工业仪表与自动化装置, 2013, (2): 3-6.
- [24] 黄正新. 人工萤火虫群优化算法分析改进及应用研究[D]. 广西民族大学, 2011.
- [25] 张军丽. 一种用 Powell 方法局部优化的人工萤火虫算法[J]. 模式识别与人工智能, 2011, 24(5): 680-684.
- [26] 黄正新, 周永权. 自适应步长萤火虫群多模态函数优化算法[J]. 计算机科学, 2011, 38(7): 220-224.
- [27] K N Krishnand, D Ghose. Glowworm swam optimisation: a new method for optimizing multi-modal functions[J]. Int J Computational Intelligence Studies, 2009, 1(1): 93-119.
- [28] Wenbo Dong, Kang Zhou. Adaptive neighborhood search's DGSO applied to travelling salesman problem. 10th International Conference, BIC-TA 2015 Hefei, China, September 25-28, 2015, Proceedings.