

基于定向变异布谷鸟算法的配送路径问题

刘晓珍¹ 刘景森^{1,2}

(河南大学软件学院 河南 开封 475004)¹ (河南大学智能网络系统研究所 河南 开封 475004)²

摘 要 在货物配送路径规划问题中,为了保持基本布谷鸟算法中莱维飞行机制与偏好随机游动策略的特点,文中提出了基于定向变异的布谷鸟算法和求解配送路径问题的完整有效方法。首先采用快速排序法将实数编码个体的每一维元素映射成问题的城市编号,从而建立算法与问题模型之间的联系;然后运用邻域搜索法决定城市访问的次序,即通过各城市之间的距离寻找当前城市的邻近城市,以增强算法的收敛速度。同时,在算法局部搜索机制中,通过平均适应度函数将算法划分为双子群,然后针对不同的子群体采用相应的定向变异机制,从而使算法搜索具有目的性,以增强算法的局部搜索能力。对标准 TSP 数据库中测试算例的求解实验结果表明,所提算法在各个算例中的求解偏差率均有明显降低,无论在最优值还是平均值的偏差率上都小于其他几种对比算法,对于路径规划问题的求解效果较优。

关键词 布谷鸟算法,快速排序法,邻域搜索法,变异策略,路径规划问题

中图分类号 TP301.6 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2019.07.026

Distribution Routing Problem Based on Cuckoo Search Algorithm with Directional Mutation

LIU Xiao-zhen¹ LIU Jing-sen^{1,2}

(College of Software, Henan University, Kaifeng, Henan 475004, China)¹

(Institute of Intelligent Networks System, Henan University, Kaifeng, Henan 475004, China)²

Abstract In order to maintain the characteristics of the Lévy flight mechanism and the preference of the random walk strategy in the basic cuckoo search algorithm, this paper proposed a cuckoo search algorithm based on directional mutation to solve the distribution routing problem. Firstly, the quick sort method is used to map each dimensional element of a real-coded individual into the city number thus to establish the connection between the algorithm and the problem model. Then the neighboring-region search method is used to determine the city access order. In other words, it is to find the neighboring cities of the current city by the distance between cities so as to enhance the convergence speed of the algorithm. Meanwhile, in the algorithm local search mechanism, the average fitness function is used to divide the algorithm into two subgroups. Then the corresponding directional mutating mechanism is adopted according to different subgroups, so that the algorithm has purpose to search and the local search ability of the algorithm is enhanced. The experimental results of solution to the test case of the standard TSP database show that the deviation rate of the proposed algorithm has been significantly reduced in each case. The deviation rate of the algorithm is smaller than that of other comparison algorithms in both the optimal value and the average value, and its solution effect is better in the path planning problem.

Keywords Cuckoo search algorithm, Quick sort method, Local search method, Mutation strategy, Path planning problem

1 引言

随着电商行业的日益兴起,物流行业已成为市场经济发展的重要部分。在当今“网购潮”的大环境下,货物如何高效配送成为了物流行业发展与竞争的关键因素。货物配送简而言之则为“配”与“送”,“配”是将总仓库中的货物根据不同的

地点分配到各个相关城市的分仓库中;“送”则是根据车辆团队配送的目的地选择一条最优路线设计方案将货物送达消费者。因此,在保证货物配送成本、提高配送效率的基础上,设计一条优质的配送路线对于物流行业至关重要。

货物配送路线规划问题作为组合优化问题中的 NP-Hard 问题始终受到广大研究者的青睐。该问题是以寻找最

收稿日期:2018-08-05 返修日期:2018-10-13 本文受河南省重点研发与推广专项(182102310886),河南省科技攻关重点项目(162102110109)资助。

刘晓珍(1992-),女,硕士,主要研究方向为智能算法,E-mail:liuzhenu@163.com;刘景森(1968-),男,博士,教授,CCF 会员,主要研究方向为智能算法、网络与信息安全等,E-mail:ljs@henu.edu.cn(通信作者)。

短配送路径方案为目的,对物流行业的资源利用率及货物配送的效率等有重要影响,因此研究该问题具有理论意义和实际价值。起初,许多学者运用了各种精确算法来求解该问题,但是随着社会的发展,问题的规模越来越复杂化,传统的精确求解方法收敛速度慢、消耗时间长,实时性难以满足人们的需求。近年来,随着智能优化算法的迅速发展,研究者们不再使用传统的数值计算方法求解路线规划问题,而是应用启发式智能算法求解问题的近似最优值。目前,利用智能优化算法(如遗传算法^[1]、蚁群算法^[2]、粒子群算法^[3-4]、模拟退火算法^[5]、禁忌搜索算法^[6]等)来解决路线规划问题已成为众多学者研究的主要方向,并且这些算法在实际问题的应用中也获得了较好的求解效果。

2009年,剑桥大学学者 Yang 等通过模拟自然界布谷鸟的繁殖寄生行为提出了一种新型全局搜索算法——布谷鸟搜索算法(Cuckoo Search Algorithm,CS)^[7-8]。该算法运用莱维飞行和偏好随机游动两大机制有效地进行了全局探索与局部探索,且凭借其自身参数设置少、模型简单、寻优能力强等优势,在连续的函数极值优化问题中取得了较好的效果。同时,随着研究的不断深化,CS算法已被成功应用于多个领域,如函数优化^[9]、图像分割^[10-11]、电力系统优化^[12-13]、人脸识别^[14]、路径规划^[15-16]、结构工程优化^[17]、TSP^[18]等。

布谷鸟优化算法作为一种新型启发式算法,在路径规划领域也得到了较广泛的应用,其中最为典型的就是对 TSP 问题的研究,如:文献[18]中的经典离散布谷鸟算法(Discrete Cuckoo Search Algorithm,DCS)通过重建鸟巢种群将 CS 算法离散化,然后依据莱维飞行移动的距离来决定个体不同的操作(2-opt 移动与双桥移动),从而使算法有效求解离散型组合优化问题。文献[19]基于随机键的布谷鸟算法(Random-Key Cuckoo Search,RKCS),运用随机产生的小数对个体所代表的城市进行编码,然后对其进行排序以形成一条可行解,将离散型问题巧妙地转化为连续型求解模式,仿真实验表明了该算法的有效性与优越性。文献[20]提出了一种混合模拟退火的布谷鸟算法,该算法采用退火时机的判断准则对 CS 算法进行了改进,增强了算法的局部搜索能力,且在 TSP 问题的求解中表现较好。文献[21]提出的基因-表现型的布谷鸟算法在求解旅行商问题时,首先对每个城市赋予一个基因(城市编号与分配给该城市的随机小数)形成算法的编码方式,然后依据莱维飞行移动的大小进行重新定位操作或替换操作,以增强算法的收敛速度与局部搜索能力,此改进算法在求解大规模 TSP 问题上表现较佳。虽然这些改进算法在求解路径规划问题时都取得了不错的结果,但在使用莱维飞行机制、寻优性能和收敛速度等方面,仍有继续提升和改进的空间。而货物配送路径规划问题的实质也是寻找最短路径,因此在此 CS 算法求解 TSP 问题的启发下,本文将布谷鸟算法应用到货物配送路径规划问题中,提出基于定向变异策略的布谷鸟算法以求解配送路径问题。首先,本文对基本布谷鸟算法进行了改进,在算法的偏好随机游动策略中,利用平均适应度函数将种群划分为优质子群体与劣质子群体,并对不同的子群体采取相应的定向变异策略,以增强算法的局部搜索能力;然后,在运用改进算法求解货物配送路径规划问题时,本

文通过一种映射的关系对算法的编码方式进行了解码,即对实数编码的个体通过快速排序的方法映射出货物配送路径问题的城市编号,同时运用邻域搜索模式决定城市访问的次序,从而形成问题的可行性路径;最后,运用标准 TSP 数据库中的测试算例对改进后的算法进行测试,以全面验证改进算法的可行性与有效性。仿真实验结果表明:改进算法在求解货物配送路径规划问题时具有较好的寻优效果。

2 货物配送路径规划问题模型及布谷鸟算法的思想原理

2.1 问题模型设计

货物配送路线规划问题是配货车辆团队为了高效到达各配送城市或地点而寻求最短配送路径的问题,此类路径规划问题也可抽象为 TSP 问题。即该问题可描述为:配送车辆团队随机从某个城市的仓库出发,将货物送达每个目的地城市,并且该车辆团队必须要经过每个目的地且仅经过一次,最终回到出发地点,从而形成一条可行的封闭回路,且尽可能地使目标函数达到最小值的行程路线。

基于上述分析,该问题的模型设计也可应用 TSP 问题模型^[22-23],则问题的图论描述可以表示为:设无向加权图 $G=(V,E)$,其中 V 代表城市编号的集合,即 $V=\{v_1,v_2,\dots,v_n\}$, E 是两点间边的集合,即 $E=\{e_1,e_2,\dots,e_m\}$,各城市之间的欧氏距离可设为 D_{ij} ,则其计算公式可以表示为:

$$D_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \quad (1)$$

其中, (x_i,y_i) 代表城市 i 的坐标位置, (x_j,y_j) 代表城市 j 的坐标位置。由此可知, $D_{ij} > 0 \parallel D_{ij} = 0,(i,j \in V)$,则旅行者是否从城市 i 经过城市 j 的关系可用式(2)表示:

$$x_{ij} = \begin{cases} 1 \\ 0 \end{cases} \quad (2)$$

其中, $i,j=1,\dots,n$ 且 $i \neq j,x_{ij} = 1$ 表示旅行者从城市 i 经过了城市 $j,x_{ij} = 0$ 则表示未从城市 i 经过城市 j 。

综上所述,TSP 问题的数学模型可表示为:

$$\min f(x) = \sum_{i=1}^n \sum_{j=1}^n D_{ij} x_{ij} \quad (3)$$

$$\text{s. t. } \begin{cases} \sum_{j=1}^n x_{ij} = 1, & i \in V \\ \sum_{i=1}^n x_{ij} = 1, & j \in V \\ \sum_{i \in S} \sum_{j \in S} x_{ij} \leq |S| - 1, & \forall S \subset V, 2 \leq |S| \leq n-1 \\ x_{ij} \in \{0,1\}, & i,j \in V, i \neq j \end{cases} \quad (4)$$

式(3)是该问题的目标函数,即走遍所有城市的最短路径长度。约束条件式(4)中的前两个约束条件代表对于每个城市,旅行者都必须经过且只能经过一次,第3个约束则保证了行程路径中没有任何回路产生。

2.2 基本布谷鸟算法的思想原理

布谷鸟具有一种特殊的孵育行为,它们从不构建自己的鸟窝去孵化幼卵,而是寻找与其生活习性相似的鸟类,让它们代替自己孵化幼卵,但是这些鸟蛋一旦被宿主发现,宿主便会抛弃这些鸟蛋或重新筑鸟巢。为了模拟布谷鸟的寄生行为,首先假设以下3个理想状态^[7]:

(1)布谷鸟一次只能产一只卵,并随机选择鸟窝来孵化它。

(2)在随机选择的一组鸟窝中,最好的鸟窝会被保留到下一代。

(3)可利用的鸟窝数量 N 是固定的,一个宿主发现一个外来鸟蛋的概率为 P_a 。

在以上3个理想状态下,布谷鸟寻窝的路径和位置更新公式如下:

$$x_i^{t+1} = x_i^t + \delta \oplus Levy(\lambda) \quad (5)$$

其中, x_i^t 表示第 i ($i=1,2,\dots,n$) 个鸟巢在第 t 代的位置; δ 为步长因子,用来控制步长大小,在多数情况下取 $\delta=1$; \oplus 表示点对点乘法; $Levy(\lambda)$ 表示莱维随机搜索路径,其服从莱维分布:

$$Levy(\lambda) \sim \mu = t^{-\beta}, 1 < \beta \leq 3 \quad (6)$$

为了方便计算,文献[8]使用式(7)产生莱维随机数。

$$Levy(\lambda) = \frac{\phi * \mu}{|v|^{\frac{1}{\beta}}} \quad (7)$$

其中, μ 和 v 均服从标准正态分布, $\beta=1.5$ 。

$$\phi = \left(\frac{\Gamma(1+\beta) * \sin(\pi * \frac{\beta}{2})}{\Gamma(\frac{1+\beta}{2}) * \beta * 2^{\frac{\beta-1}{2}}} \right)^{\frac{1}{\beta}} \quad (8)$$

综合上述公式,布谷鸟的位置更新公式如下:

$$x_i^{t+1} = x_i^t + \delta \frac{\phi * \mu}{|v|^{\frac{1}{\beta}}} (x_i^t - x_{best}^t) \quad (9)$$

按发现概率 P_a 丢弃部分解后,采用偏好随机游走重新生成相同数量的新解,公式如下:

$$x_i^{t+1} = x_i^t + \gamma(x_{g'}^t - x_i^t) \quad (10)$$

其中, γ 是服从均匀分布的缩放因子, $x_{g'}^t, x_k^t$ 表示第 t 代的两个随机数。

3 改进布谷鸟算法在货物配送路径规划问题中的应用

在基本布谷鸟算法中,莱维飞行机制与偏好随机游荡策略是平衡算法全局搜索与局部搜索的关键因素,影响着算法的寻优性能,且在解决连续型函数极值优化问题中可以很好地利用两者的关系进行高效求解。但在实际应用中包括货物配送路径规划在内的许多问题都具有离散性,为了更好地将算法与问题模型相结合,研究者们会依据问题模型改变算法的求解策略以将算法离散化。在路径规划问题中,一些学者也常常采用路径表示法、二进制表示法、次序表示法、近邻表示法等对算法进行编码,应用离散化的思想来解决问题,但是上述方法在布谷鸟算法中将很难利用莱维飞行在全局搜索能力上的独特优势进行位置更新移动。针对上述不足,本文在维持算法原有机制的基础上,对算法的编码进行了处理,使其更好地求解货物配送路径规划问题。

3.1 编码与解码

在货物配送路径问题的求解过程中,如何进行编码对问题的寻优性能具有一定影响。许多学者在使用群智能优化算法求解路径规划问题时,常采用路径表示法初始化种群个体,

从而将算法离散化^[18-19,21]。而本文为了有效地利用布谷鸟算法中莱维飞行机制的特点,借鉴了随机键^[19,21,24]的编码思想,将离散化问题转化为连续型问题进行求解。

在货物配送路线规划问题中,假设1个车辆团队须向 N 个城市仓库进行配货。运用改进布谷鸟算法求解此问题时,采用实数编码方式初始化的个体位置可表示为: $X = \{x_1, x_2, \dots, x_d\}$, 其中个体的空间维度 d 为车辆要访问的城市个数。运用位置更新式(9)产生的新鸟巢位置可表示为: $X' = \{x_1', x_2', \dots, x_d'\}$ 。然后运用快速排序法将更新后个体中的各维元素进行排序,从而将个体元素的索引位置映射成每个目的城市编号。例如:现有一个车辆团队需往6个城市进行送货,而寄生鸟巢的位置为: $nest = (4.08, 1.46, 2.53, 1.78, 3.69, 2.96)$, 通过排序后各元素所映射的城市编号如表1所列。

表1 个体元素映射的城市编号

序号	1	2	3	4	5	6
<i>nest</i>	4.08	1.46	2.53	1.78	3.69	2.96
排序后 <i>nest</i>	1.46	1.78	2.53	2.96	3.69	4.08
城市编号	2	4	3	6	5	1

根据表1数据可知,鸟巢位置 *nest* 所映射的路径为(2,4,3,6,5,1)。但为了进一步增强算法的收敛能力,本文对鸟巢位置映射的路径做了相应的处理,首先将城市2作为车辆团队的出发点,然后依据各城市之间的欧氏距离运用邻域搜索的方法决定下一次需要访问的城市,从而对上述路径的访问次序进行调整,以加速算法的收敛速度。现假设以上6个城市之间的距离如表2所列。

表2 各城市间的距离

Table 2 Distance among cities

序号	1	2	3	4	5	6
1	0	5.88	5.42	3.34	10.96	8.25
2	5.88	0	1.29	4.48	16.75	14.10
3	5.42	1.29	0	4.83	16.38	13.46
4	3.34	4.48	4.83	0	12.88	11.00
5	10.96	16.75	16.38	12.88	0	4.40
6	8.25	14.10	13.46	11.00	4.40	0

结合表2数据对鸟巢 *nest* 映射的城市序列进行相应调整。首先,以城市2作为出发点将其存放到城市访问次序列表(见表3)中,而后依据各城市之间的距离锁定与城市2距离最小的城市3作为下一次访问的城市,同时查看城市序列表中是否已访问过该城市,若已访问,则从原始序列中找到第一个未被访问的城市放入此表中,反之则将城市3纳入表3中,然后按照此方式进行遍历直到所有城市已被访问。此时,城市访问次序表中的数据如表3所列。

表3 城市访问次序

Table 3 City access order

序号	1	2	3	4	5	6
访问次序	2	3	4	1	6	5

基于上述解码方式的操作,寄生鸟巢 *nest* 所映射出的最终城市访问路径为:2-3-4-1-6-5。

此种解码方式有效地将离散优化问题转化为连续优化问题,保持了基本布谷鸟算法中的求解模式,发挥了算法本质上的优势,简化了问题的操作方式,能够在一定程度上增强算法的运行效率。

3.2 定向变异策略的布谷鸟算法

基本布谷鸟算法在寻优过程中有时也存在收敛速度慢、求解精度低等不足,为了优化算法的寻优性能,本文设计了定向变异策略的布谷鸟算法(Directional Variant Cuckoo Search Algorithm, DVCS)。一方面,该算法对位置更新公式的步长控制因子 δ 进行了改进,引入了动态惯性权重策略来动态调整莱维飞行移动的大小,从而使算法具有自适应性,提高算法的收敛速度。改进后的鸟巢位置更新公式为:

$$x_i^{t+1} = x_i^t + \omega \frac{\phi * \mu}{|v|^{\frac{1}{\beta}}} (x_i^t - x_{\text{best}}) \quad (11)$$

$$\omega = (\exp(1 - \frac{g}{G}) - 1) * (a - \omega_{\min}) + \omega_{\min} \quad (12)$$

其中, g 为算法当前的迭代次数; G 为最大迭代次数; a 为常数,用于控制惯性权重 ω 的最大取值范围; ω_{\min} 为惯性权重 ω 的最小值。

另一方面,该算法对偏好随机游动机制产生新解的公式(见式(10))也进行了改进。首先依据当前种群的平均适应度值将个体进行分类,当个体适应度值小于平均适应度时,当前个体所处的位置较佳,此时个体将被列入优质子群体;反之,个体将被列入劣质子群体。其中,种群的平均适应度值函数可用式(13)表示:

$$\begin{cases} f_{\text{avg}} = \frac{1}{n} \sum_{i=1}^n f(x_i) \\ \Delta = f_i - f_{\text{avg}} \end{cases} \quad (13)$$

其中, f_{avg} 代表当代种群的平均适应度值; f_i 代表当前个体的适应度值; Δ 代表个体适应度与平均适应度的差值,即衡量个体优劣的标准。

基于上述分析,本文运用式(14)对优质子群体进行变异,从而使个体朝自身方向进行深度搜索,以增强算法搜索的目的性,避免算法因盲目搜索而低效求解。对于劣质子群体,本文引入了差分进化算法的 DE/best/1/bin 变异策略,该变异策略不仅可以增强算法的局部搜索能力,还可以加快算法的收敛速度,但对于复杂的优化问题,此策略也会加大算法陷入局部解的可能。因此,本文通过引入扰动因子对此机制进行改进,此改进策略如式(15)所示。该变异策略不仅可以使当前较差的个体朝着全局最优位置进行搜索,还能够加快算法的收敛速度、增强算法的局部搜索能力。此时,偏好随机游动策略的变异机制如下:

$$x_i^{t+1} = x_i^t + \gamma(x_i^t - x_k^t) \quad (14)$$

$$\begin{cases} x_i^{t+1} = x_{\text{best}} * \eta + \gamma(x_g^t - x_k^t) \\ \eta = 1 + \text{trnd}() \end{cases} \quad (15)$$

其中, x_i^t 是第 t 代种群的第 i 个个体; x_k^t, x_g^t 是第 t 代种群中任意挑选的两个鸟巢的位置; x_{best} 是当前种群的最优位置; η 是具有 t 分布特性的扰动因子。

3.3 算法的实现流程

Step 1 设置算法参数(种群大小、循环次数、维度、发现概率、界值大小),并随机初始化 N 个鸟窝的位置 $X = (x_1, x_2, \dots, x_n)^T$ 。

Step 2 运用 3.1 节的解码方式将初始化的 N 个鸟巢位置映射成问题的 N 条可行性路径。

Step 3 利用式(3)求出每条路径的长度,然后选出当前最好的行走路线并记录。

Step 4 采用式(11)对初始化个体位置进行更新,将位置更新后的新解应用映射函数进行解码;然后求出更新后每条路径的长度,并对新、旧路径的目标函数值进行比较,若更新后的路径较好则代替旧解。

Step 5 用式(13)将当前种群分为优、劣两个子种群,比较随机数 r 与鸟巢被淘汰概率 P_e ,若 $r > P_e$,则对优质子种群中的个体采用式(14)进行变异,而对劣质子种群中的个体采用式(15)进行变异,从而生成相同数量的新个体。

Step 6 将新解解码,求出全部个体的适应度值,挑选并记录全局最小解。

Step 7 判断是否满足算法设置的循环次数;若满足,终止迭代过程,输出最短路径;否则,重复 Step4—Step7 进行迭代优化。

4 仿真实验与分析

为了验证本文算法求解货物配送路径问题的有效性与可行性,在保证客观、公正的前提下,选用了标准 TSP¹⁾ 数据库中的算例对算法进行测试。同时,将改进算法 DVCS 与基本布谷鸟算法(Cuckoo Search Algorithm, CS)、文献[21]中的 GPCS 算法以及经典的遗传算法(GA)进行实验对比分析,以全面检验 DVCS 算法的寻优性能。

4.1 实验环境及参数设置

为了保证对比实验的公平性,CS,GA,GPCS,DVCS 算法均采用了 Windows 8.1 专业版 64 位的操作系统、i5-3230M CPU @ 2.6 GHz 的处理器、4 GB 的运行内存、MATLAB R2014a 版的开发环境及编程语言进行仿真实验。

实验过程中,4 种算法的种群大小、寻优进化代数、空间维度均保持一致,其中种群大小 $N=50$;最大进化代数 $G=3000$ 。其他参数的设置如表 4 所列。

表 4 4 种算法的参数设置

Table 4 Parameter settings for four algorithms

算法	变量名	取值大小
CS	控制因子 δ	0.01
	发现概率 P_a	0.25
GA	交叉概率 p_c	0.8
	变异概率 p_m	0.1
GPCS	控制因子 δ	1
	发现概率 P_a	0.25
DVCS	惯性权重 ω	[0.2, 1]
	发现概率 P_a	0.25

1) <https://www.proxy.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/>

对于仿真实验而言,算法参数的设置也是影响算法求解性能的重要因素,但目前参数设置并没有统一的标准,本文在查阅大量文献与反复实验的基础上对各个算法中的参数进行了统一、合理的设置。

在上述参数的设置中,CS算法的控制因子 δ 影响着个体移动的距离,当 δ 取值较大时,个体因获得较大步长而快速收敛,增强了算法的收敛速度,但在一定程度上降低了算法的求解精度;当 δ 取值较小时,个体因获得较小步长而精细搜索,提高了算法的寻优精度,但易增加算法陷入局部极值的可能。因此,参数 δ 对算法的收敛速度和求解精度都具有一定的影响。为保证实验数据的客观性,本文对此参数的设置与基本布谷鸟算法保持一致。发现概率 P_a 是维持算法全局搜索与局部搜索平衡的关键参数,如果 P_a 取值较小,则被更新鸟巢的数量较多,算法种群的灵活性较强,从而提高了算法的全局搜索能力,但也降低了算法的收敛速度;若 P_a 取值较大,则被更新鸟巢数量变少,算法种群的多样性降低,收敛速度加快,但易陷入局部极值。为了更好地平衡算法全局与局部搜索之间的关系,基本布谷鸟算法中一般取 P_a 为0.25。交叉概率 p_c 和变异概率 p_m 是遗传算法中的两个重要参数, p_c 的取值大小关系着算法通过交叉操作产生新个体的比例,对全局和局部搜索能力都具有较大影响,一般取值为0.25~1.0之间,在基本遗传算法中此参数常取值为0.8; p_m 主要维持算法种群的多样性,避免算法陷入局部极值,增强算法的全局搜索能力,在基本遗传算法中此参数常取值为0.1。以上这些基本算法参数的常态取值都是经过长期、反复实验和应用检验后的结果,具有合理性。惯性权重 ω 是为提高本文算法DVCS的寻优性能而提出的一种具有非线性递减思想的步长调整函数,DVCS算法用动态取值的 ω 代替了固定取值的步长控制因子 δ ,使个体在寻优过程中随算法迭代次数的增加而自适应地调整步长大小,从而增强了算法的收敛速度与寻优精度。经过大量实验测试,惯性权重 ω 的取值范围为0.2~1.0时,算法的稳定性较好。

4.2 评估函数

算法寻优结果的优劣是评价算法性能的关键因素,本文为了全面展示算法的寻优性能,运用最优解、平均解、最优偏差率以及平均偏差率来评价各算法的实验结果。其中,最优偏差率和平均偏差率分别是算法寻找的最优解、平均值与理论最小值之间的偏差情况,能够直观地体现算法寻优结果的优劣。基于实验结果与分析的客观性考虑,本文采用了最常用的标准偏差率评估函数^[18,25]来验证改进算法的寻优特性,其中最优偏差率(Optimal Deviation rate, OD)与平均偏差率(Average Deviation rate, AD)的表达式如式(16)和式(17)所示:

$$OD = \frac{\text{所求最优值} - \text{理论最小值}}{\text{理论最小值}} \times 100\% \quad (16)$$

$$AD = \frac{\text{所求平均值} - \text{理论最小值}}{\text{理论最小值}} \times 100\% \quad (17)$$

上述公式中偏差率的大小可直观地体现算法寻优结果的好坏。偏差率越小代表算法的寻优结果越接近理论最优值,求解效果较好;反之,求解效果较差。

4.3 仿真实验结果的对比分析

为了充分检验改进算法在求解货物配送路径规划问题中的性能,本文运用了标准TSPLIB库中的5个算例:Burma14, Ulysses16, Ulysses22, Eil51, China31(中国31个城市)进行仿真实验。同时,为了保证实验的客观性,实验过程中CS, GA, GPCS, DVCS算法针对上述每个算例在同等环境下分别独立运行20次,从而获得每种算法在不同算例的最优值、平均值、最优偏差率和平均偏差率。上述5个算例的仿真实验结果如表5所列。

表5 5个算例的寻优结果
Table 5 Optimization results of five examples

算例名称	理论最优值	算法	最优值	偏差率/%	平均值	偏差率/%
Burma14	30.8	CS	30.8	0	31.215	1.34
		GA	30.8	0	31.354	1.79
		GPCS	30.8	0	31.145	1.12
		DVCS	30.8	0	31.107	0.99
Ulysses16	72	CS	74.001	2.77	74.313	3.21
		GA	74.154	2.99	74.533	3.51
		GPCS	74.096	2.91	74.384	3.31
		DVCS	74.001	2.77	74.251	3.12
Ulysses22	74	CS	75.876	2.53	77.139	4.24
		GA	76.089	2.82	77.367	4.55
		GPCS	76.383	3.22	77.354	4.53
		DVCS	75.309	1.76	75.974	2.66
Eil51	426	CS	719.221	68.83	759.468	78.27
		GA	492.499	15.61	548.472	28.74
		GPCS	601.74	41.25	655.662	53.91
		DVCS	449.203	5.44	460.847	8.18
China31	—	CS	15936	—	18009	—
		GA	15879	—	17011	—
		GPCS	17448	—	18363	—
		DVCS	15381	—	15808	—

由表5可知:在相同的实验环境下,本文DVCS算法的寻优结果均优于其他3种算法。在Burma14算例中,4种算法均能求到全局最小值,其最优偏差率为0,但是从平均值来看,本文算法的平均偏差率只有0.99%,明显小于CS, GA, GPCS算法;对于算例Ulysses16,4种算法的最优偏差率和平均偏差率均保持在2%~3%的数量级上,寻优结果并无太大的差别,但本文算法的寻优结果仍然优于其他3种算法;对于算例Ulysses22,4种算法的寻优结果中本文算法的最优值与平均值均小于其他算法,且最优偏差率比其他3种算法降低了1%~1.5%,平均偏差率则比其他3种算法降低了2%,寻优结果最接近理论最小值;对于算例Eil51,本文算法的寻优明显优于其他3种算法,从算法的最优偏差率和平均偏差率来看,本文算法的偏差率可达5%~8%,而其他3种算法的偏差率均在15%以上。对于China31算例,许多学者应用了不同的方法(蚁群算法、禁忌搜索算法、最短路径法等)来寻求最小值,且求得的结果在不断更新。因此,目前尚没有该算例的理论最优值,但是从寻优结果可知,DVCS算法的最小值和平均值均小于CS, GA, GPCS算法。综上所述,仿真实验的数据结果表明了本文算法的可行性与有效性。

为了更全面地展示改进算法的适应性,本文还给出了4种算法求解每个算例的适应度对比曲线图,从而有效分析算法的收敛性能。上述5个算例的收敛曲线如图1—图5所示。

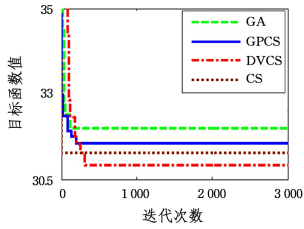


图1 Burma14的适应度对比曲线
Fig. 1 Fitness curve of Burma14

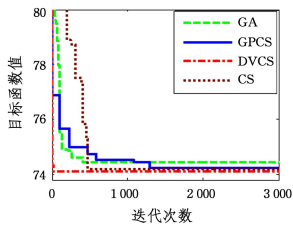


图2 Ulysses16的适应度对比曲线
Fig. 2 Fitness curve of Ulysses16

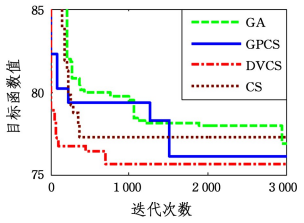


图3 Ulysses22的适应度对比曲线
Fig. 3 Fitness curve of Ulysses22

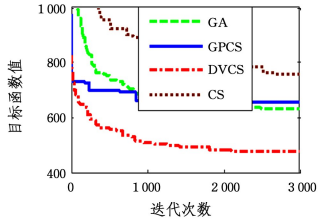


图4 Eil51的适应度对比曲线
Fig. 4 Fitness curve of Eil51

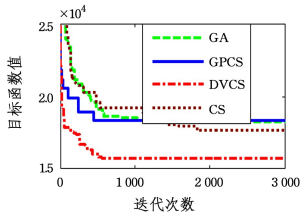


图5 China31的适应度对比曲线
Fig. 5 Fitness curve of China31

上述实验结果显示:本文算法在5个算例上的收敛速度明显快于其他3种算法,且在寻优过程中出现拐点的次数较少,尤其是算例 Ulysses16 上的收敛曲线在算法搜索前期收敛速度非常快,收敛曲线几乎垂直向下并且曲线较为光滑。对于算例 Burma14,虽然 DVCS 算法前期的收敛速度与其他3种算法相差无几,但是该算法仍在进行深度挖掘。对于算例 Ulysses22,4种算法均出现了多处拐点,但是本文算法的收敛速度仍快于其他3种算法,且跳离局部极值的速度也较快。对于算例 Eil51、China31(中国31个城市),本文算法呈弧状形态并不断进行深度搜索,而其他3种算法在搜索后期几乎呈水平直线状态,搜索能力减弱。基于上述分析,本文算法寻优曲线的下降速度均快于其他算法,收敛速度得到了提升。这是因为本文算法在映射过程中,通过邻域搜索方式中邻近表的决策使每次访问的城市都以自身最近城市为出发点进行访问,从而缩短了路径的距离,增强了算法的收敛速度。

以上仿真实验结果表明,DVCS算法在求解货物配送路径规划问题时具有较好的求解效果,从而有效验证了改进算法的可行性与适应性。与此同时,本文也通过图6—图10对DVCS算法在5个算例上求解的最优路径进行了展示。

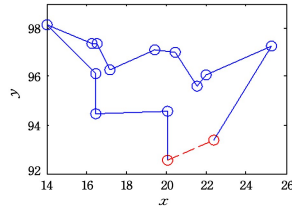


图6 Burma14上的最优路径
Fig. 6 Optimal path map on Burma14

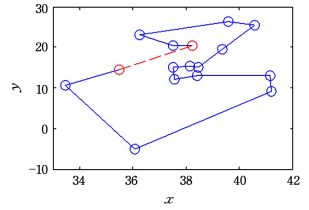


图7 Ulysses16上的最优路径
Fig. 7 Optimal path map on Ulysses16

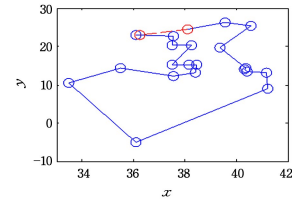


图8 Ulysses22上的最优路径
Fig. 8 Optimal path map on Ulysses22

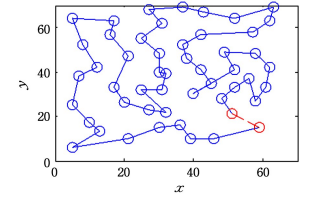


图9 Eil51上的最优路径
Fig. 9 Optimal path map on Eil51

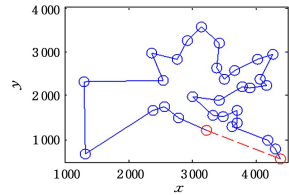


图10 China31上的最优路径
Fig. 10 Optimal path map on China31

由图6—图10可知,本文算法所求最优路径的走势清晰,并未出现任何路径交叉现象,这说明 DVCS 算法在求解货物配送路径规划问题时具有一定的优势,且此算法在实际问题的应用中也具有较好的适应性与可行性。

结束语 本文针对货物配送路径规划问题,设计了一种映射关系,从而将离散型的问题转化为连续型问题进行求解,并提出了一种 DVCS 求解算法。首先运用快速排序法将实数编码的种群个体映射成问题中车辆团队需要访问的城市编号,从而建立算法与实际问题的关系。然后运用各城市之间的距离进行邻域搜索,决定各城市的访问次序,此种方式通过对邻近城市的搜索可以加快算法的收敛速度,提高算法搜索的效率。同时,采用定向变异的方式优化布谷鸟算法的局部搜索机制,通过平均适应度函数将种群个体分为优质子群体与劣质子群体两类,针对不同子群体,采用不同的变异机制,从而保持算法种群的灵活性与多样性,增强算法的局部搜索能力。对上述算例的仿真实验结果表明,基于 DVCS 算法的货物配送路径规划方法的求解性能均优于其他对比算法,且具有较好的寻优效果。

参考文献

[1] GOLDBERG D E. Genetic algorithm in search, optimization and machine learning [M]. Boston: Addison-Wesley Longman Publishing Co. Inc., 1989.

- [2] DORIGO M, MANIEZZO V, COLORNI A. Ant System: optimization by a colony of cooperating agents [J]. *IEEE Transactions on Systems Man & Cybernetics Part B Cybernetics a Publication of the IEEE Systems Man & Cybernetics Society*, 1996, 26(1): 29-41.
- [3] EBERHART R, KENNEDY J. A new optimizer using particle swarm theory[C]//*Proceedings of the Sixth International Symposium on Micro Machine and Human Science*. 1995: 39-43.
- [4] KENNEDY J, EBERHART R. Particle swarm optimization [C] // *Proceedings of IEEE International Conference on Neural Networks*. 1995: 1942-1948.
- [5] KIRKPATRICK S, GELATT C, VECCHI M. Optimization by simulated annealing [J]. *Science*, 1983, 220: 671-680.
- [6] GLOVER F. Future Paths for integer programming and links to artificial intelligence [J]. *Computers and Operations Research*, 1986, 13(5): 535-549.
- [7] YANG X S, DEB S. Cuckoo search via levy flight [C]//*Proceedings of World Congress on Nature & Biologically Inspired Computing*. India, Washington: IEEE Publications, 2009: 210-214.
- [8] YANG X S, DEB S. Engineering optimization by cuckoo search [J]. *Int'l Journal of Mathematical Modeling and Numerical Optimization*, 2010, 1(4): 330-343.
- [9] LI Y, MA L. A new metaheuristic cuckoo search algorithm [J]. *Systems Engineering*, 2012, 30(8): 64-69. (in Chinese)
李煜, 马良. 新型元启发式布谷鸟搜索算法[J]. *系统工程*, 2012, 30(8): 64-69.
- [10] SURESH S, LAL S. An efficient cuckoo search algorithm based multilevel thresholding for segmentation of satellite images using different objective functions [J]. *Expert Systems with Applications*, 2016, 58(C): 184-209.
- [11] CHAKRABORTY S, CHATTERJEE S, DEY N, et al. Modified cuckoo search algorithm in microscopic image segmentation of hippocampus [J]. *Microscopy Research & Technique*, 2017 (10): 1051-1072.
- [12] ABD-ELAZIM S M, ALI E S. Optimal location of STATCOM in multimachine power system for increasing load ability by cuckoo search algorithm [J]. *International Journal of Electrical Power & Energy Systems*, 2016, 80: 240-251.
- [13] CHITARA D, NIAZI K R, SWARNKAR A, et al. Cuckoo search optimization algorithm for designing of multimachine power system stabilizer [C]//*IEEE International Conference on Power Electronics, Intelligent Control and Energy Systems*. IEEE, 2017: 1-6.
- [14] NAIK M K, PANDA R. A novel adaptive cuckoo search algorithm for intrinsic discriminant analysis based face recognition [J]. *Applied Soft Computing*, 2016, 38(C): 661-675.
- [15] MOHANTY P K, PARHI D R. Optimal path planning for a mobile robot using cuckoo search algorithm [J]. *Journal of Experimental & Theoretical Artificial Intelligence*, 2016, 28(1/2): 35-52.
- [16] SANTILLAN J H, TAPUCAR S, MANLIGUEZ C, et al. Cuckoo search via levy flights for the capacitated vehicle routing problem [J]. *Journal of Industrial Engineering International*, 2017(1): 1-12.
- [17] PAULINE O, SIN H C, SHENG D D C V, et al. Design optimization of structural engineering problems using adaptive cuckoo search algorithm [C]//*International Conference on Control, Automation and Robotics*. IEEE, 2017: 745-748.
- [18] OUAARAB A, AHIOD B, YANG X S. Discrete cuckoo search algorithm for the travelling salesman problem [J]. *Neural Computing & Applications*, 2014, 24(7/8): 1659-1669.
- [19] OUAARAB A, AHIOD B, YANG X S. Random-key cuckoo search for the travelling salesman problem [J]. *Soft Computing*, 2015, 19(4): 1099-1106.
- [20] MA C, LIU J, YU F P. Research on cuckoo algorithm with simulated annealing [J]. *Journal of Chinese Computer Systems*, 2016, 37(9): 2029-2034. (in Chinese)
马灿, 刘坚, 余方平. 混合模拟退火的布谷鸟算法研究[J]. *小型微型计算机系统*, 2016, 37(9): 2029-2034.
- [21] LIN M, ZHONG Y W, LIU B X, et al. Genotype-phenotype cuckoo search algorithm for traveling salesman problem [J]. *Computer Engineering and Applications*, 2017, 53(24): 172-181. (in Chinese)
林敏, 钟一文, 刘必雄, 等. 基因-表现型的布谷鸟算法求解旅行商问题[J]. *计算机工程与应用*, 2017, 53(24): 172-181.
- [22] JAFARZADEH H, MORADINASAB N, ELYASI M. An enhanced genetic algorithm for the generalized traveling salesman problem [J]. *Engineering, Technology & Applied Science Research*, 2017, 7(6): 2260-2265.
- [23] LIU J. Applied research of hybrid genetic algorithm and simulated annealing algorithm in traveling salesman problem [D]. Guangzhou: South China University of Technology, 2014. (in Chinese)
刘锦. 混合遗传算法和模拟退火算法在 TSP 中的应用研究[D]. 广州: 华南理工大学, 2014.
- [24] BEAN J C. Genetic algorithms and random keys for sequencing and optimization [J]. *Inform Journal on Computing*, 1994, 6: 154-160.
- [25] WU H S, ZHANG F M, LI H, et al. Discrete wolf pack algorithm for traveling salesman problem [J]. *Control and Decision*, 2015(10): 1861-1867. (in Chinese)
吴虎胜, 张凤鸣, 李浩, 等. 求解 TSP 问题的离散狼群算法[J]. *控制与决策*, 2015(10): 1861-1867.