

改进的离散粒子群算法在物流配送路径问题中的应用

吴天红 王 玮 姜英姿

(江苏省徐州工程学院 江苏 徐州 221000)

【摘要】粒子群优化算法(PSO)是一种基于鸟群行为的进化元启发式算法,已成功地应用于求解多种问题。本文以容量约束车辆路径问题(CVRP)为研究对象,研究了运输和物流中的配送路径优化问题。提出了一种改进的离散粒子群优化算法(DPSO),它是PSO算法的一个衍生,结合迭代局部搜索(ILS)方法。在搜索过程中,改进的算法可以根据进化世代自适应调整粒子的飞行时间,ILS保证粒子从局部极小值逃逸。对部分基准问题的实验结果表明,该方法能获得较好的结果。

【关键词】旅行商问题;多目标优化;人工蜂群算法;迭代本地搜索;粒子群优化

一、介绍

旅行商问题^[1](TSP)是组合优化中一个众所周知的NP难题,几乎所有解决工程和优化问题的新方法都已在TSP上得到验证。但是,在现实世界中,TSP情况通常是不确定的,在我们必须做出决策时,尚不确定某些或全部TSP参数。随着概率论的极大改进,随机模型已在许多相关的TSP中广泛使用来表示不确定性,包括考虑存在客户时的概率、需求水平、旅行时间和客户现场的服务时间,通常假设已知分布决定了问题的某些参数。

众所周知,多目标优化是^[2]现实世界中最重要的优化问题之一,因此具有多个目标和相互冲突目标的UTSP更接近于实际情况。因此,基于不确定性理论和多目标优化技术,提出了一种在不确定环境下具有多目标的新型TSP,称为多目标不确定性TSP(UMTSP)。处理多目标规划问题时,最常用的方法之一是将一个函数中的所有目标聚合在一起,并将原始问题转换为单个目标优化问题。

车辆路径问题^[3](VRP)是一个具有重要经济意义的组合优化问题。一个典型的VRP可以描述为从一个仓库到一组地理上分散的点(城市、仓库、客户等)的最低成本路线设计问题。VRP因其在物流和供应链管理中的应用而受到广泛关注。

二、UMTSP的描述和数学公式

2.1 TSP说明

TSP是研究最广泛的NP-hard组合优化问题之一,其单目标版本是元启发式算法的典型基准。现代形式的TSP假设旅行推销员必须访问多个给定的城市,从同一城市开始和结束。然后,我们需要找到一条最小的路线,使一个人可以精确地到达每个城市一次,然后返回其故乡。为了建模TSP,用完整的边缘加权图G=(V, E)表示,其中V设置为n=|V|节点或代表城市和顶点E \subseteq V×V是组有向边或弧。每个弧(i, j)∈E被分配长度值d_{ij}。本文考虑了对称的TSP,其中d_{ij}=d_{ji}对于E中的所有弧均成立。因此,TSP中的目标是找到图的最小长度,它仅访问G中的n个节点中的每一个。因此,通向TSP的最佳路径是节点索引X的置换 $\pi(X)=1, 2, \dots, n$ 使得长度f($\pi(X)$)最小。

$$f(\pi(X)) = \sum_{i=1}^{n-1} d(K_i, K_{i+1}) + d(K_n, K_1) \quad (1)$$

2.2 制定UMTSP

基于上面介绍的TSP,UMTSP的公式可以表示为:

$$\min_{\pi(X), \xi} f(\pi(X), \xi) = (f_1(\pi(X), \xi^{(1)}), f_2(\pi(X), \xi^{(2)}), \dots, f_p(\pi(X), \xi^{(p)})), \quad (2)$$

在上面提出的UMTSP模型中,每个目标函数的形式取决于我们考虑的实际应用。例如,当第一个目标被认为是行车时间的最小化时,则f₁($\pi(X)$, $\xi(1)$)可以用总和形式表

$$\text{示如下: } \min_{\pi(X), \xi} f_1(\pi(X), \xi^{(1)}) = \sum_{i=1}^{n-1} \xi^{(1)}_{(K_i, K_{i+1})} = \xi^{(1)}_{(K_1, K_2)} \quad (3)$$

其中, $\xi^{(1)}_{(i,j)}$ 表示城市i和j之间的旅行时间。

由于目标通常在UMTSP中相互冲突,因此没有最佳途径同时最小化所有目标功能。在这种情况下,应该在UMTSP中引入帕累托有效路径^[4]的概念,这意味着在不牺牲一个或多个其他目标的情况下,不可能改善任何一个目标。因此,应提出一种新的解决方案,即不确定方法,以获取UMTSP中的帕累托有效路径。

三、离散粒子群优化

在粒子群优化算法中,被称为粒子群种群中的每一个个体,都朝着它自己先前的最佳位置,以及它的拓扑邻域中的任何一个成员先前获得的最佳位置调整其轨迹。在粒子群优化算法的全局变量中,将整个粒子群视为邻域。因此,在寻找有希望的景观区域过程中,全球共享信息,粒子从所有其他同伴的发现和先前的经验中获益。

粒子根据以下方程式进行操作(上标表示迭代次数):

$$v_{id}^{i+1} = w v_{id}^i + c_1 \times rand_1() \times (p_{id}^i - x_{id}^i) + c_2 \times rand_2() \times (p_{jd}^i - x_{jd}^i) \quad (4)$$

$$x_{id} = x_{id} + v_{id} \quad (5)$$

其中 $1 \leq i \leq m$, $1 \leq d \leq D$, w是惯性权重, c_1 和 c_2 是两个正单位,称为认知参数。

虽然PSO的开发工作主要集中在连续搜索空间中的优化问题上,但也对该算法在离散问题中的应用进行了研究。通过将每个矢量中的速度改变为处于一种或另一种状态的概率来解决粒子在空间中移动的问题。

四、不确定的方法

为了直接在不确定目标上定义UMTSP中的帕累托有效路径,提出了不确定变量之间关系的定义。

(1) ξ 和 η 为两个不确定变量,我们说 $\xi < (\text{或} \leq)$ η 当且仅当其中 $\rho(\xi) \leq \rho(\eta)$ 表示的估值 ξ 是低于或等于 η 的, ρ 表示用于定义不确定变量评估的原理。

(2) 值得指出的是,对于不确定变量之间的比较,关系是在不确定变量的评估下定义的。现实生活中存在的不同问题,需要不同的估值含义来满足其在实际应用中的需要,因此,相应的原理 ρ 应该根据问题的实际情况来定义此评估,其中 ρ 是此评估含义的概括。例如,在UMTSP中,由于我们希望从长远角度最小化行进时间 $T(\pi(X), \xi)$,因此将其期望值 $E[T(\pi(X), \xi)]$ 用于定义关系。

由于不确定变量的期望值已被广泛应用于现实生活中,因此解决了UMTSP问题。 ρ 原则获得本文的帕累托有效路线。

用于有效解决 UMTSP 问题。

五、CVRP 的离散粒子群算法

有容量约束的车辆路径问题^[5] (CVRP) 是 VRP 的基本形式。这一名称源于拥有容量有限车辆的限制。CVRP 是 NP-hard 问题, 因为它包含一个或多个 TSP 作为子问题。显然, CVRP 比 TSP 更难求解。

经典的 CVRP 可以描述为: N 个地理上分散在一个平面区域的客户必须从一个独特的仓库获得服务。每个客户都要求货物数量 q_i ($i=1, 2, \dots, n$)。从节点 i 到节点 j 的传输成本为 c_{ij} 。 m 固定容量的车辆 Q 可用于运送储存在仓库中的货物。每一位顾客只能被一辆车访问一次。该问题的目标是在不违反每辆车能力的情况下, 最小化所有路线的总成本。

在这个亚组中, 我们提出了一个改进的 DPSOILS 算法, 本地搜索算法^[6] 被应用到 PSO 中。

要应用 ILS 算法, 基本上必须指定三个过程。给定当前 s_0 , 这是一个过程扰动, 它扰动当前解, 导致相同的中间解 s , 过程局部搜索将 s'' 带到局部最优 s''' , 以及决定应用下一个扰动步骤解的可接受准则。

在前一阶段 s_0 中建立的每个解都被带到局部最优。然后, 从通过扰动当前搜索点获得的起始解迭代地应用局部搜索过程。在本文中, DPSOILS 扩展背后的主要思想是使用 ILS 来增加多样性。标准粒子和 DPSOILS 粒子之间的唯一区别是: 对速度向量 v_i 的每个元素附加随机扰动, 如等式 (7) 所述。根据粒子在不同位置, 加入随机扰动, 可以平衡局部和全局搜索范围。

$$x_{i,t} = x_{i,t} + rand(0, 1) \times (1 - k \times iter / I_{max}) \times v_{i,t} \quad (6)$$

其中 $rand()$ 是在 $[0, 1]$ 范围内均匀分布的随机数。 I_{max} 是迭代的总数。 $iter$ 是当前迭代大小。常数 k 平衡 I_{max} 和 $iter$ 。通常有 $I_{max} > iter$ 。

六、TSP 的 ABC 算法变体

6.1 人口初始化

在我们提出的算法中, TSP 的路径表示为整数置换 $\pi(X)$ 。以 14 个城市的 TSP 为例, 整数置换 (1、4、14、2、9、3、12、8、11、13、6、5、10、7) 是 TSP 的可行路线。基于路径表示, 我们可以使用 SN 整数排列随机初始化种群, 其中 SN 是食物来源的数量。

6.2 反向运算符

通常使用的蜜蜂使用相同的搜索运算符来执行漏洞利用搜索。为了表示利用搜索的行为, 引入了反向运算符, 其过程描述如下:

(1) 从 1 到 n (位置总数) 随机生成两个整数 r_1 和 r_2 。
(2) 将 max_r 设置为 r_1 和 r_2 之间的最大值, 并将 min_r 设置为 r_1 和 r_2 之间的最小值。
(3) 将所选解决方案中的排列顺序从位置 min_r 反转到位置 max_r 。

6.3 交叉算子

为了进一步提高开发能力, 引入了交叉算子, 以选择的食物来源和当前的最佳食物来源为父母, 以创造更好的后代, 其过程描述如下。

(1) 从 1 到 n (位置总数) 随机生成两个整数 r_1 和 r_2 。
(2) 将 max_r 设置为 r_1 和 r_2 之间的最大值, 并将 min_r 设置为 r_1 和 r_2 之间的最小值。
(3) 选择 max_r 和 min_r 作为两个交叉点, 然后在两个父级的 max_r 和 min_r 之间交换数据。
(4) 删除后代中的重复数字, 并通过映射关系从 (3) 中

交换的数据中替换它们。

6.4 变异算子

为了提高勘探能力, 当最佳食物来源在一定周期后没有变化时, 应用变异算子对当前的最佳食物来源进行变异以发现新的食物来源, 其过程描述如下。

(1) 从 1 到 n (位置总数) 随机生成两个整数 r_1 和 r_2 。

(2) 交换所选解决方案中的 r_1 位置和 r_2 位置。

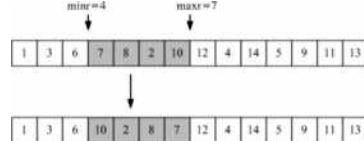


图 1. 逆向运算符的利用搜索插图、

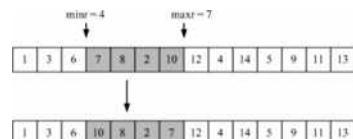


图 2. 用于开发搜索的变异算子插图

交叉和变异算子的例子在图 1、图 2 中示出, 其中 r_1 和 r_2 分别是 4 和 7。

6.5 受雇佣蜜蜂阶段

在本文中, 受雇佣蜜蜂通过反向和交叉算子对给定的食物来源执行开发搜索, 并在当前食物来源及其新的相邻食物来源之间进行贪婪选择, 以确保为更好的发展保留更好的食物来源。贪婪选择基于食物来源的适合度, 其计算公式如下:

$$fit_i = \begin{cases} \frac{1}{1-f_i}, & \text{if } f_i \geq 0 \\ 1 - abs(f_i), & \text{if } f_i < 0 \end{cases} \quad (8)$$

其中 f_i 是模型中描述的不确定单一目标模型的期望值。

6.6 围观蜂阶段

本文在围观蜂阶段使用反向算子和交叉算子在所选食物来源周围生成新的有希望的解决方案, 该解决方案是根据与所选解决方案适用性成正比的某个概率进行的。所选解决方案的相应概率值可以计算如下:

$$p_i = fit_i / \sum_{i=1}^{SN} fit_i \quad (9)$$

就像所采用的蜜蜂阶段一样, 采用贪婪选择来保证解决方案的质量。

6.7 侦察蜂阶段

在本文中, 如果所雇用的蜜蜂和围观蜂无法在有限的时间内找到更好的邻近食物来源, 则可以使用变异算子围绕当前的首选解决方案进行搜索。

所提出的 ABC 算法不仅应用反向算子和交叉算子来生成新的相邻食物源, 而且还应用了变异算子来保护搜索免受局部最小值的影响。它不仅强调了全球勘探与本地开发之间的平衡, 而且强调了搜索过程中种群的多样性。

七、案例分析

我们现在在 12 个 CVRP 基准问题的实例上评估 DPSOIL 的性能。其中包括每个问题最著名的解决方案。这些问题的解决范围从 32 个客户到 80 个客户, 从 5 辆车到 14 辆车。对于数据集的每个实例, 客户数由实例名称上的第一个数字给出。这些问题之间的主要区别在于它们的紧密性(需求和容量的比率)

和客户的位置。

实验是在奔腾 IV, 1GB 内存, 2.0GHz 处理器上进行的。然后对每种问题类型的解进行平均, 结果在表 1 中报告。我们使用 $n=20$ 个人工蚂蚁, 设置 $\alpha=1$, $q_0=0.9$, $\beta=2$ 和 $\rho=0.1$ 。对于所有问题, 最大迭代次数为 $2 \times N = 40$ 。

在表 1 中, 我们展示了 DPSOILS 取得的结果。下表显示了所提出算法找到的最佳解以及 50 次运行中每次运行找到的最佳解平均值。当我们的研究开始时, 色谱柱的最佳值表明了已知的最佳解决方案。结果表明, DPSOILS 能够找到所有情况下的最优解。用 DPSOILS 得到的平均值接近最佳解的值。实际上, 距离可以小于 3%。另一个有趣的点是 DPSOILS 能够找到迄今为止最好的解决方案 (实例 A32k5、E76k7、E76k8 和 E76k14)。

表 1. DPSOILS 的结果

实例	最优值	DPSOILSB 最优	DPSOILS 均值
A32k5	784	784	792.1
A54k7	1167	1170	1180.5
A60k9	1358	1364	1375.6
A69k9	1167	1169	1181.7
A80k10	1764	1775	1813.9
B57k7	1153	1163	1176.2
B63k10	1496	1532	1534.2
B78k10	1266	1272	1275.6
E76k7	682	682	686.6
E76k8	735	735	738.3
E76k10	832	846	861.3
E76k14	1032	1032	1040.5

八、结论

为了提高离散粒子群优化算法的性能, 我们提出了一种改进的迭代局部搜索离散粒子群优化算法。计算结果表明, DPSOILS 方法能够为 CVRP 问题产生高质量的解, 从而证明 DPSOILS 是一种新颖的随机优化 VRP 方法。

作为未来的工作, 我们打算对 DPSOILS 的重要性进行详细的研究, 进行实验来评估 DPSOILS 在现实问题和更广泛的组合问题上的有效性。

参考文献:

- [1] 高尚, 韩斌, 吴小俊, et al. 求解旅行商问题的混合粒子群优化算法 [J]. 控制与决策, 2004(11): 86-89.
- [2] 公茂果, 焦李成, 杨咚咚, et al. 进化多目标优化算法研究 [J]. 软件学报, 2009, 20(2): 271-289.
- [3] 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.
- [4] 白翠玲, 苗泽华, 杨建朝. 基于帕累托最优的京津冀旅游企业合作路径选择. 商业研究, 2008(10): 45-48.
- [5] 晁晓菲, 杨晓龙. 带容量约束的车辆路径问题算法综述 [J]. 价值工程, 2012(05): 26-27.
- [6] 蒋宗礼, 张婷. 基于用户行为分析的本地搜索排序算法优化 [J]. 计算机技术与发展, 2014(02): 21-24+30.

基金项目:

大学生创新创业训练计划项目 (xcx2019033)

通讯作者:

姜英姿 (1970-), 男, 副教授 / 高级实验师, 徐州工程学院数学与统计学院副院长