

求解带容量约束的车辆路径问题的改进伊藤算法

易云飞^{1,2} 蔡永乐² 董文永¹ 林郭隆²

(武汉大学计算机学院 武汉 430079)¹ (河池学院计算机与信息科学系 宜州 546300)²

摘要 针对车辆路径问题中路径选择未能确定的缺陷,引入蚁群算法对客户点选取规则进行决策。此外,采用冷却进度表作为控制温度变化的参数,将漂移和波动过程同步进行来改进根据伊藤随机过程而设计的伊藤算法,并将改进后的算法应用于 CVRP 的求解。实验结果表明,改进后的算法能有效求解带容量约束的车辆路径问题,取得了理想的结果。

关键词 车辆路径问题,伊藤算法,漂移算子,波动算子

中图分类号 TP301 文献标识码 A

Improved ITO Algorithm for Solving the CVRP

YI Yun-fei^{1,2} CAI Yong-le² DONG Wen-yong¹ LIN Guo-long²

(School of Computer, Wuhan University, Wuhan 430079, China)¹

(Department of Computer and Information Science, Hechi University, Yizhou 546300, China)²

Abstract In order to overcome the shortcoming of the path selection for the vehicle routing problem, the method of the Ant Colony Algorithm was used to select the customer point. In addition, cooling schedule was used to control the parameters of temperature changes, and the process of drifting operator and fluctuation operator simultaneously to improve the Ito algorithm which is based on the hypothesis testing and Ito stochastic process. When it comes to solve the capacitated vehicle routing problems, the improved ITO algorithm is effective, and the numerical results show that the improved algorithm is feasible.

Keywords Vehicle routing problem, ITO algorithm, Drifting operator, Fluctuation operator

1 引言

车辆路径问题^[1] (Vehicle Routing Problem, VRP)由 G. Dantzig 和 J. Ramser 于 1959 年首先提出,一直是组合优化领域的热点和前沿问题。经典 VRP 被提出后,根据现实生活条件的变化,问题本身有多个扩展类别,文献[2]中的带时间窗约束的分车运输同时收发车辆路径问题(简称 SVRPSP-DTW)就是 VRP 的扩展。求解 VRP 问题的近似算法包括蚁群算法^[3]、蛙跳算法^[4]、文化基因算法^[5]、粒子群算法^[6-8]等。在求解大规模的 VRP 时,各种近似算法通常可以在有限时间里找到满意的次优解或可行解。

伊藤算法^[9-16] (Ito Algorithm, ITO)是董文永教授等人基于随机过程中的伊藤过程,模仿粒子系统中粒子相互碰撞与作用的动力学规律进行算法设计和问题求解,是从微观的角度来分析粒子的运动规律,然后通过抽象及模拟的方法提出的新算法。伊藤算法一方面体现了仿生演化算法群体搜索的特征,解的表示被看成是一个粒子,大量的粒子组成粒子系统;另一方面又运用了伊藤过程的理论来分析算法,利用伊藤

随机积分的方法建立算法的动力学方程,并根据爱因斯坦、朗之万等人提出的大分子运动规律来设计伊藤算法的两个关键算子:漂移率和波动率,结合粒子热运动规律,使之具有模拟退火的一些特征。目前伊藤算法已经在求解组合优化、函数优化、系统辨识、时间序列建模等问题中取得了很好的效果。

VRP 问题是一个经典的组合优化问题,本文设计了一种求解带容量约束的车辆路径问题的改进伊藤算法。采用冷却进度表来作为控制温度变化的参数,并将漂移和波动过程同步进行,实验结果表明改进后的算法能够取得更为理想的效果。

2 VRP 问题模型描述

基本 VRP 问题的约束条件分为对车辆的约束和对客户需求点的约束。对车辆的约束主要体现在容量上,每个车辆访问客户服务点的过程中,只能对剩余载货量可以满足的客户点提供配送服务,并且只能访问客户服务点一次,不允许分批多次配送。本文研究的 VRP 不对车辆数做出约束,评价配送策略成功与否的唯一依据就是所有车辆行驶的路径总长

到稿日期:2012-11-18 返修日期:2013-03-13 本文受国家自然科学基金项目(61170305),广西自然科学基金项目(2011GXNSFB018074),广西教育厅科研项目(2009111x406,200103YB136),河池学院自然科学基金项目(2008B-N005),广西新世纪教改工程立项项目(2012JGA198)资助。

易云飞(1981-),男,博士,讲师,主要研究方向为机器学习、智能计算等,E-mail:gxxyyf@163.com;蔡永乐(1991-),男,主要研究方向为智能计算;董文永(1973-),男,博士,教授,博士生导师,主要研究方向为演化计算、机器学习等;林郭隆(1990-),男,主要研究方向为智能计算。

度,长度越小越符合预期,每个车辆必须保证从仓库出发,最后配送完毕后回到出发仓库。某中心仓库有若干辆货车(运载能力为 Q) 为 N 个客户送货,货车要求从中心仓库出发,完成客户点的运送任务后最终回到中心仓库,客户点 i 的货物需求量是 $q_i (i=1,2,\dots,n)$,且 $q_i < Q$,记配送中心编号为 0,各客户编号为 $i (i=1,2,\dots,n)$, c_{ij} 表示客户 i 到客户 j 的距离。要求以最少的车辆数、最少的运送成本来完成货物的派送任务。

各变量定义如下:若车辆 k 由客户点 i 到 j ,则 $x_{ijk} = 1$;否则 $x_{ijk} = 0$ 。若车辆 k 负责客户点 i 的货物配送,则 $y_{ik} = 1$;否则 $y_{ik} = 0$ 。数学模型可描述为:

$$\text{Min } z = \sum_{i=0}^n \sum_{j=0}^n \sum_{k=0}^M c_{ij} x_{ijk} \quad (1)$$

其中,约束条件:

$$\sum_{k=0}^M y_{ik} = 1 (i=1,2,\dots,n) \quad (2)$$

$$\sum_{i=0}^n x_{ijk} = y_{jk} (j=0,1,\dots,n, k=1,2,\dots,m) \quad (3)$$

$$\sum_{j=0}^n x_{ijk} = y_{ik} (j=0,1,\dots,n, k=1,2,\dots,m) \quad (4)$$

$$\sum_{i=0}^n q_i y_{ik} \leq Q (k=1,2,\dots,m) \quad (5)$$

在上述模型中,式(1)是问题的目标函数,表示完成所有客户点的配送任务花费的路径长度。式(2)一式(5)是问题的约束条件:式(2)表示每个客户点的运输任务仅由一辆车来完成;式(3)、式(4)表示到达和离开某个客户点的货车有且仅有一辆;式(5)表示每辆车承载的货物总量不超过其最大容量。

3 伊藤算法求解 VRP 问题

伊藤算法自提出以来,董文永教授等人已经将伊藤算法应用到了 TSP 问题等组合优化问题的求解当中,取得了比蚁群算法更强的竞争力,本文尝试使用伊藤算法去求解 VRP 问题。

3.1 粒子初始化

初始时刻,各条边权重相等。设边权重 $\tau(i,j) = c$ (c 为常数),粒子 k 在搜索过程中沿边权重决定移动方向,选用轮盘赌的方法进行选择,公式如下:

$$p^k(i,j) = \begin{cases} \frac{\tau(i,j)}{\sum_{l \in \text{tabu}_k} [\tau(i,l)]}, & i \in \text{tabu}_k \cap j \notin \text{tabu}_k \\ 0, & \text{其他} \end{cases} \quad (6)$$

式中, tabu_k 记录粒子 k 当前所走过的城市,称为禁忌表。

根据该方法每个粒子得到一个初始解,完成一次迭代。当所有粒子完成一次迭代时,计算各粒子走过的路径长度,得到本次迭代的最优解。本次迭代最优解与当前最优解相比较,如果比当前最优解小,即粒子走过的总路径长度更短,则更新当前最优解。

本文决定采用线性变换法来更新半径,假设当前种群中包含的粒子为 x_1, x_2, \dots, x_m ,按照粒子适应度值选取出适应度值最大的粒子分别为 x_{upper}, x_{low} 。

$$r(x_i) = r_{\min} + \frac{f(x_{upper}) - f(x_i)}{f(x_{upper}) - f(x_{low})} (r_{\max} - r_{\min}) \quad (7)$$

式中, $f(x)$ 为计算适应度值的函数,也就是粒子的总路径长

度。

3.2 冷却进度表的设计

环境温度的变化可以采用模拟退火算法中的设计方法,采用冷却进度表来作为控制温度变化的参数。冷却进度表是影响算法实验性能的重要因素,合理选取是算法应用的关键。一般情况下,退火表长度可以设置成与问题相关。退火函数设置如下:

$$T_i = \rho \cdot T_{i-1} \quad (8)$$

式中, $0 < \rho \leq 1$,默认情况下 $\rho = 0.99$ 。

3.3 波动算子的设计

波动算子是用来控制粒子的探索特性的,它可以使粒子在邻域内局部扰动。一般波动算子的设计需要考虑两个方面:一个是波动强度,即波动的系数 γ ,另一个是波动的过程,即,粒子如何运动,如何按照波动强度完成邻域内的波动。波动系数 γ 与粒子的半径、当前的环境温度等众多因素有关,因此可以设计一个关于粒子半径 r 和环境温度 T 的函数,为了简化起见,假设函数 $\gamma = f(r, t)$ 是可分离变量的,即波动的速率是由粒子的当前半径 r 的函数和当前温度 T 的函数共同决定的:

$$f(r, t) = \gamma_{\min} + f_1(r) \cdot f_2(t) \cdot (\gamma_{\max} - \gamma_{\min}) \quad (9)$$

式中, $\gamma_{\max}, \gamma_{\min}$ 分别表示粒子的最大波动强度和最小波动强度,一般设定 $\gamma_{\max} = 1 - \gamma_{\min}$;本文取 $\gamma_{\max} = 1, \gamma_{\min} = 0$,将式(9)简化为:

$$f(r, t) = f_1(r) \cdot f_2(t) \quad (10)$$

式中, f_1 是粒子半径对漂移强度的影响。

$$f_1(r) = \frac{(e^{-\gamma r} - e^{-\gamma r_{\max}})}{(e^{-\gamma r_{\min}} - e^{-\gamma r_{\max}})} \quad (11)$$

函数 $f_2(T)$ 表示环境温度对粒子漂移强度的影响,参照模拟退火算法中 Metropolis 准则设计如下一种可行函数,其结构如下:

$$f_2(T) = \exp(-1/T) \quad (12)$$

有了波动强度,粒子根据式(13)设定边权重,其中 σ 为当前粒子的路径。

$$\tau(c_i^{(0)}, c_{i+1}^{(k)}) = \begin{cases} 1 - \gamma, & \text{if } e(i, j) \in \sigma \\ \gamma / (n_{i+1} - 1), & \text{其它} \end{cases} \quad (13)$$

得到边权重之后,粒子根据初始化时提到的轮盘赌方法得到一个新解,如果该新解优于当前解,则更新当前解,完成波动过程。

3.4 漂移算子的设计

漂移算子是用来控制粒子宏观上朝着吸引元的方向移动,和波动算子一样,它也由两部分组成:一个是漂移强度,也即漂移的系数 μ ,另一个是漂移的过程,即如何按照漂移强度完成在领域内的漂移。漂移系数 μ 同样受到粒子半径和环境温度的影响,所以也可以将其设计成关于这两个因素的函数表示。为了简化起见,漂移强度的计算方法和波动强度一样。

有了漂移强度,根据式(14)更新边权重,其中 σ 为当前粒子的路径,而 σ' 为相对应的吸引元的路径(即当前最优解路径):

$$\tau(i, j) = \begin{cases} 1, & \text{if } e(i, j) \in \sigma \cap e(i, j) \in \sigma' \\ 1 - \mu, & \text{if } e(i, j) \in \sigma' \\ \mu, & \text{if } e(i, j) \in \sigma \\ 0, & \text{else} \end{cases} \quad (14)$$

更新完权重之后,粒子根据初始化时提到的轮盘赌方法更新当前解,完成漂移过程。

4 改进伊藤算法求解车辆路径问题

4.1 伊藤算法中下一个客户点的选取规则

在轮盘赌的规则选取过程中,只考虑了各个客户点之间的权重,这种设计是有缺陷的,跟实际的问题不太符合,而且很难求出满意解。首先各条边的初始权重是相同的,这样刚开始微粒的运动就带有很大的盲目性,一般迭代初始阶段微粒的运动对最终结果会有很大影响,因为后续边权重的变更都是根据上一次迭代的结果自适应改变的;另外对于算法结果好坏评判的唯一依据是车辆行驶的总的距离,式(6)的客户点选择规则没有体现出边的长度因素。本文参照蚁群算法状态转移规则的设计,将各条边的长度也考虑进去,最终的概率选取公式设计如下:

$$p^k(i, j) = \begin{cases} \frac{[\tau(i, j)]^\alpha [\eta(i, j)]^\beta}{\sum_{l \in \text{tabu}_k} [\tau(i, l)]^\alpha [\eta(i, l)]^\beta}, & i \in \text{tabu}_k \cap j \notin \text{tabu}_k \\ 0, & \text{else} \end{cases} \quad (15)$$

式中, $\eta(i, j)$ 为距离的倒数,即 $1/d_{ij}$,称为经验知识,也可称为能见度。 tabu_k 是一个禁忌表,存储车辆 k 已经搜索过的客户需求点。 α 为控制边权重在概率选择中重要程度的因子,而 β 为控制能见度(即,边长因素)在概率选择中重要程度的因子。 $\tau(i, j)$ 表示在边 (i, j) 上的权重,类似于蚁群算法中的信息素浓度,但是关于边权重变化的映射方法不同,它采用伊藤算法的思想进行设计,更能体现出布朗运动这样的事物运动规律,其值通过粒子漂移和波动的过程动态设定。

4.2 漂移和波动同步进行

在传统的伊藤算法中,漂移过程和波动过程单独进行,先是粒子根据波动策略更新边权重,并根据更新后的边权重更新当前解,完成波动过程,然后粒子根据漂移策略更新边权重,再更新当前解,完成漂移过程。而在实际的布朗运动中,漂移和波动往往是同步进行的,基于此本文决定将漂移和波动过程综合起来考虑。这样得到的边权重的更新方法如下,其中 σ 为当前粒子的路径, σ' 为对应吸引元的路径:

$$\tau(i, j) = \begin{cases} 1 + \gamma + \mu, & \text{if } e(i, j) \in \sigma \cap e(i, j) \in \sigma' \\ \gamma + \mu, & \text{if } e(i, j) \in \sigma \\ 1, & \text{if } e(i, j) \in \sigma' \\ \gamma, & \text{其他} \end{cases} \quad (16)$$

得到边权重之后,按照上面提到的下一客户点的选取规则,更新当前解。在后面的实验中会发现,这是一种有效的设计。

4.3 改进算法求解 VRP 问题执行步骤

Step1 使用要求解的 VRP 问题算例初始化算法中的各个变量,客户需求点个数为 n ,车辆的最大载重量为 C ,算法允许最大连续未更新迭代次数为 N_c ,每个客户需求点的坐标位置为 x_i, y_i 以及需求量为 $demand_i$,根据客户需求点坐标计算

出各个客户需求点之间的距离 d_{ij} ,初始化边权重影响因子 α 和距离影响因子 β ,每条边的初始权重设为 1。给出算法的种群粒子个数 m ,每个粒子都从仓库点开始进行漂移和波动。

Step2 粒子开始搜索路径,当某个客户需求点没有被粒子搜索过,即不在禁忌表中,并且它的货物需求量小于目前车辆的剩余载重量,那么此客户需求点就列为粒子的可选点,根据式(15)计算所有在可选点列表内的点的概率,再采用轮盘赌策略选择下一个访问点,这样既可选择概率大的点,又能确保一定概率内大范围搜索其它点,避免局部收敛,然后将访问的点放入禁忌表中,同时将车辆的剩余载重量减去刚访问过的客户的需求量。如果剩余的客户需求点中没有符合条件的,就将车辆返回仓库,重新赋值为最大载重量继续搜索剩余客户点,如此反复,直到所有客户需求点都被访问过,车辆回到配送中心。

Step3 将种群中的所有粒子都按照 Step2 中的方法进行漂移和波动,完成一次迭代,然后遍历各个粒子搜索的路径,找到本次迭代中的最优路径及长度,并与全局最优路径长度进行比较,如果本次迭代的最优路径比全局最优路径更好,那么更新全局最优路径,设置连续未更新迭代次数 $N_c=0$,否则 N_c 加 1。

Step4 一次迭代搜索完成之后,根据式(7)中的半径更新策略更新每个粒子的半径,根据式(8)中的退火表设计更新环境温度,根据式(16)中的漂移算子和波动算子映射模型更新各条边的权值。

Step5 迭代次数加 1,然后判断 N_c 是否大于最大连续未更新迭代次数,如果大于,则算法结束,此时记录的全局最优解就是算法找到的最优解;否则回到 Step2 继续进行搜索。

5 实验及结果分析

为了验证改进后算法的有效性,选用标准测试平台 Augerat et al. Set A 中的 A-n32-k5 测试算例进行测试。实验仿真环境: 2.66GHz 主频 Intel 处理器, 2G 内存, 仿真软件 Eclipse Juno Release。实验中粒子数为 50,边权重重要性因子 $\alpha=5$,客户点之间距离重要性因子 $\beta=3$,最大未更新迭代次数 $N_c=200$,初始温度 $T_0=8000$,退火表长度 $L_k=100$,退火速率 $\rho=0.99$ 。经过多次运行本文设计的伊藤算法,得出的最优路径长度为 817.857,图 1 为该实验结果的仿真,图 2 为该结果下的收敛情况,可以看出在迭代到 370 代时,算法得到该最优解,图 3 为算法连续运行 20 次的结果图,所得最长路径为 834.413。

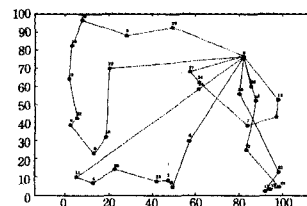


图 1 A-n32-k5 算例结果仿真

标准测试平台 Augerat et al. Set A 中的 A-n32-k5 算例共包含 32 个点,分别为一个仓库点和 31 个客户需求点,整个配送策略使用了 5 个车队来完成,它们访问的客户需求点顺序如下:

- 1) 0→16→13→31→19→17→21→26→0
- 2) 0→20→5→25→10→15→22→9→8→18→29→0
- 3) 0→6→2→3→23→28→4→11→14→0
- 4) 0→27→24→7→1→12→0
- 5) 0→30→0

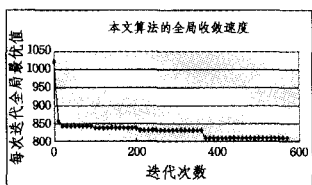


图2 算法的全局收敛速度

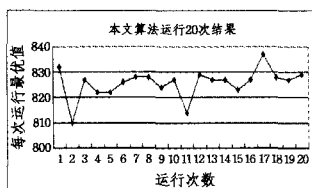


图3 本文算法运行20次结果图

目前,用于求解车辆问题的智能算法种类较多,而且取得了不错的效果,下面是与几种比较典型的蚁群算法的比较。蚁群算法最早是由Dorigo等人于1991年提出来的,它是模拟蚂蚁的运动规律而设计的智能启发式算法,目前比较典型的蚁群算法及其改进模型有AS、ACS和MMAS。下面的实验将它们与本文的ITO算法进行效果比较。

在基本蚁群算法中采用蚁周信息素更新算法,各参数的取值为 $\alpha=1, \beta=5, \rho=0.5, Q=100$; ACS算法所选取的参数为: $\alpha=1, \beta=5, \rho=0.5, \tau_0=0.01, p_0=0.1$; MMAS算法所选取的参数为 $\alpha=1, \beta=5, \rho=0.5, \tau_{\min}=0.01, \tau_{\max}=10$ 。

测试算例选用标准测试平台Augerat et al. Set A中的前10个算例进行测试,最佳结果和平均结果比较如图4和图5所示。

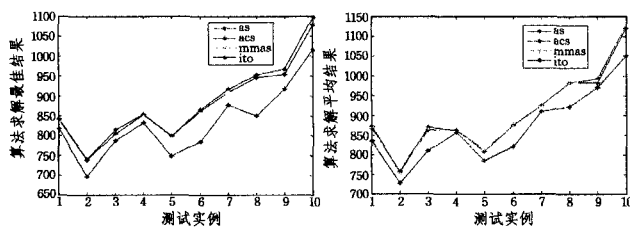


图4 改进ITO算法求解VRP最佳结果比较

从图4和图5可以看出,本文设计的改进伊藤算法在求解车辆路径问题时,不管是最终的全局最优解还是平均解都明显优于AS、ACS、MMAS。

结束语 本文将伊藤算法中的漂移算子和波动算子巧妙地融合了起来,通过多次尝试,得出了有效的算子设计方案,并多次试验边权重重要性因子和距离重要性因子的设定对算法的影响,综合各方面考虑最终得出较优的参数设定,在全局权重的更新上,沿用了伊藤算法中半径、温度的设计,从而体现了出动态自适应的特点,有效地避免了算法的过早收敛,

从而可以得到比其他算法更好的最优解。另外,还借鉴蚁群算法的设计思想,将改进后的伊藤算法用于求解带容量约束的车辆路径问题中,设计了一种实用的映射模型。采用蚁群算法的路径选择策略对城市间距离进行选择,更加符合问题求解的要求。但是,在设定算法的参数时,大多是基于设想和大量的数据分析,这些参数还可以进一步调整,进而提升算法的效率和精度,此外,改进后的算法能否用于更为复杂的VRP模型中,将是进一步研究的方向。

参考文献

- [1] Dantzig G, Ramser J. The truck dispatching Problem [J]. Management Science, 1959(6): 80-91
- [2] 王科峰,叶春明,唐国春. 一类新的车辆路径问题及其两阶段算法[J]. 运筹学学报, 2010, 14(3): 55-63
- [3] 李娅,王东. 基于混沌扰动和邻域交换的蚁群算法求解车辆路径问题[J]. 计算机应用, 2012, 32(2): 444-447
- [4] 骆剑平,李霞,陈泯融. 基于改进混合蛙跳算法的 CVRP 求解[J]. 电子与信息学报, 2011, 33(2): 429-434
- [5] 王君,李波. 带时间窗车辆路径问题的文化基因算法[J]. 计算机工程与应用, 2012, 48(7): 26-29
- [6] 易云飞,阮忠,王国兴,等. 求解车辆路径问题的改进粒子群算法[J]. 计算机科学, 2009, 11A(36): 149-152
- [7] 魏明,靳文舟. 求解车辆路径问题的离散粒子群算法[J]. 计算机科学, 2010, 37(4): 187-191
- [8] 寇明顺,叶春明,陈子皓. 应用蜜蜂繁殖进化型粒子群算法求解车辆路径问题[J]. 工业工程, 2012, 15(1): 23-27
- [9] 董文水,张文生,于瑞国. 求解组合优化问题伊藤算法的收敛性和期望收敛速度分析[J]. 计算机学报, 2011, 34(4): 636-646
- [10] Dong Wen-yong. The Simulation Optimization Algorithm Based on the Ito Process[C]//The 2nd International Conference on Intelligent Computing, 2007: 563-573
- [11] Dong Wen-yong. The Multi-Objective ITO Algorithms [C]//The 2nd International Symposium on Intelligence Computation and Applications, 2007: 21-23
- [12] Dong Wen-yong. Simulation Optimization Based on the Hypothesis Testing and ITO Process [C] // Third International Conference on Natural Computation, 2007: 1210-1221
- [13] Dong Wen-yong. Time Series Modeling Based on ITO Algorithm [C]//Third International Conference on Natural Computation, 2007: 398-402
- [14] Dong Wen-yong, et al. BBOB-benchmarking: A new evolutionary algorithms inspired by ITO process for noiseless function testbed [J]. Journal of Computational Information Systems, 2011, 7(6): 2195-2203
- [15] Dong Wen-yong, Yu Rui-guo, Lei Ming. Merging the Ranking and Selection into ITO Algorithm for Simulation Optimization [C]//5th International Symposium on Intelligence Computation and Applications, 2010: 87-96
- [16] Dong Wen-yong, et al. A New Evolutionary Algorithms for Global Numerical Optimization Based on Ito Process [C]//5th International Symposium on Intelligence Computation and Applications, 2010: 57-67