

基于自适应免疫遗传算法的 VRP 问题的研究

檀庭方

(重庆大学, 重庆 400044)

摘要:物流配送车辆路径优化问题是近年来物流领域中的研究热点,该问题属于 NP 难题,当问题规模较大,很难得到问题的最优解和满意解。应用遗传算法是被认为求解 NP 难题的有效手段之一,文中在求解物流配送车辆路径优化问题时,在传统遗传算法的基础上,加入自适应算子,并引入了免疫算法的思想,实验结果表明该算法具有更好的全局和局部搜索能力和收敛速度,可有效地解决物流配送车辆路径优化问题。

关键词:车辆调度;遗传算法;免疫算法;物流配送

中图分类号:TP301.6

文献标识码:A

文章编号:1673-629X(2007)06-0074-03

Study on Optimization of Logistics Distribution VRP Based on Self-Adaption Immune-Genetic Algorithm

TAN Ting-fang

(Chongqing University, Chongqing 400044, China)

Abstract: The logistics distribution VRP is a hot research problem in logistics area. It is an NP problem and difficult to get an optimal and satisfied solution. Using genetic algorithm is considered to be one of the most efficient way to solve the NP Problem. While adopting the traditional genetic algorithm to solve the logistics distribution VRP, adds the self-adaption operator and inducts the ideas of immune algorithm, experimental results present that the new algorithm in the paper performs better in aspect of global and local search ability and convergence speed, and it is proved to be a more efficient way to solve the logistics distribution VRP.

Key words: vehicle scheduling problem; genetic algorithm; immune algorithm; logistics distribution

0 引言

物流配送是现代物流管理中的一个重要环节,它是指按用户的订货要求,在配送中心进行分货、配货,并将配好的货物及时送交收货人的活动。随着市场竞争的加剧和各企业 JIT(Just In Time)策略的实施,配送这一环节就显得尤为重要,文中着重讨论配送过程中的路径优化问题。车辆路径问题(Vehicle Routing Problem, 简称 VRP),是目前在物流系统中较受关注的一个方面,它是指在客户需求位置已知的情况下,确定车辆的行驶路线,使得运输线路最短或运输成本最低。

遗传算法是模拟生物的遗传进化过程而形成的全局优化概率搜索算法,它最早由美国 Michigan 大学的 John Holland 教授于 20 世纪 70 年代提出,其基本思想是将某种编码技术作用于称为个体的数串,然后模拟

由这些串组成的群体的进化过程,对这个群体进行选择操作、交叉操作、变异操作,通过寻求群体进化中的最优个体,得到所求问题的最优解^[1]。它是一种新兴的自适应随机搜索方法,它对优化对象既不要求连续,也不要求可微,并具有极强的鲁棒性和内在的并行计算机制,特别适合求解复杂的多极值问题和组合最优化问题。然而经典遗传算法容易陷入局部最优解,在进化过程中不可避免地产生了退化的可能,从而导致了进化后期的波动现象,使迭代次数过大和准确率不高。

免疫系统是生物体的一个高度进化、复杂的功能系统,它能自适应地识别和排除侵入机体的抗原性异物,并具有学习记忆和自适应调节能力,维护机体内环境的稳定,免疫算法正是模仿这一功能的仿生算法^[2]。抗原对应于所求问题的目标函数,抗体对应于问题的解,根据目标函数决定抗原与抗体的亲和力,而抗体之间的亲和力取决于相应解的相似程度,利用这两种亲和力对备选解进行评价和选择,通过抗体之间的相互激励作用提高最优点附近的搜索效率,通过记忆细胞

收稿日期:2006-08-03

作者简介:檀庭方(1982-),男,安徽安庆人,硕士研究生,研究方向为电子商务与现代物流;导师:刘卫宁,博士,教授,CCF 会员,研究方向为计算机网络通信、网络信息系统、电子商务与现代物流、智能交通系统。

对抗体的抑制作用摆脱局部最优点,以达到最终收敛到全局最优点的目的。

1 问题描述

物流配送车辆路径优化问题一般可以描述为:从某物流配送中心用多辆配送车辆向多个客户送货。每个客户的位置和货物需求量一定,每辆车的载重量一定,其一次配送的最大行驶距离一定。要求合理安排车辆配送路线,使目标函数得到最优。并满足以下条件:

(1)每条配送路径上各客户需求量之和不超过配送车辆的载重量;

(2)每条配送路径的长度不超过配送车辆一次配送的最大行驶距离;

(3)每个客户的需求必须满足,且只能由一辆配送车送货^[3]。

设配送中心需要向 k 个客户送货,每个客户的货物需求量是 $g_i (i = 1, 2, \dots, k)$, 每辆配送车的载重量是 q , 且 $g_i < q$, 其一次配送的最大行驶距离为 D 。首先为了安排路线需要对要使用的车辆数有一个估计。在现实情况中,货物装(卸)车越复杂,约束条件越多,一辆车的实际载货量就越小。在文中使用文献[4]的公式来确定需要的车辆数:

$$m = \lfloor \sum_{i=1}^k g_i / \alpha q \rfloor + 1$$

其中, m 为所需车辆数, α 为参数, $0 < \alpha < 1$ 。约束条件越多,货物装(卸)越复杂, α 值越小。文中取 α 为 0.95。

下面建立此问题的数学模型: c_{ij} 表示点 i 到点 j 的运输成本,可以是时间、路程、费用等,为计算方便,文中就表示路程; M 表示当一辆车的货运量超过最大载重量时的惩罚系数,为一个很大的整数;配送中心编号为 0,各客户编号为 $i (i = 1, 2, \dots, k)$, 定义变量如下:

$$x_{ijp} = \begin{cases} 1, & \text{车 } p \text{ 由 } i \text{ 向 } j \text{ 行驶} \\ 0, & \text{否则} \end{cases}$$

$$y_{ip} = \begin{cases} 1, & \text{客户 } i \text{ 的货物由 } p \text{ 车完成} \\ 0, & \text{否则} \end{cases}$$

得到的数学模型如下所示:

$$\min Z = \sum_{i=0}^k \sum_{j=0}^k \sum_{p=1}^m c_{ij} x_{ijp} + M \sum_{p=1}^m \max \left(\sum_{i=0}^k g_i y_{ip} - q, 0 \right) \quad (1)$$

$$\text{s. t.} \quad \sum_{i=0}^k g_i y_{ip} \leq q, \quad p = 1, 2, \dots, m \quad (2)$$

$$\sum_{i=0}^k \sum_{j=0}^k c_{ij} x_{ijp} \leq D, \quad p = 1, 2, \dots, m \quad (3)$$

$$\sum_{p=1}^m y_{ip} = \begin{cases} 1, & i = 1, 2, \dots, k \\ m, & i = 0 \end{cases} \quad (4)$$

$$\sum_{i=0}^k x_{ijp} = y_{jp}, \quad j = 1, 2, \dots, k; p = 1, 2, \dots, m \quad (5)$$

$$\sum_{j=1}^k x_{ijs} = y_{is}, \quad i = 1, 2, \dots, k; s = 1, 2, \dots, k \quad (6)$$

上述模型中,式(2)为汽车容量约束;式(3)为最大行驶距离约束;式(4)保证了每个客户的运输任务仅由一辆车完成,所以运输任务则由 m 辆车协同完成;式(5)和式(6)限制了到达和离开某一客户的汽车有且仅有一辆^[4]。

2 算法设计

2.1 编码

为提高效率,采用非负整数的序数编码,用 0 表示配送中心,用 $1, 2, \dots, k$ 表示各客户需求点,因为共有 m 辆车,最多存在 m 条配送路径,一条 VRP 的可行路线可编码成为一个长为 $k + m + 1$ 的染色体 $(0, p_{11}, p_{12}, \dots, p_{11}, 0, p_{21}, p_{22}, \dots, p_{2s}, 0, \dots, 0, p_{m1}, p_{m2}, \dots, p_{mt}, 0)$, 染色体中相邻两个 0 之间表示一个子路径,第 $i (i = 1, 2, \dots, m)$ 辆车的子路径为 $0, p_{i1}, p_{i2}, \dots, p_{it}, 0$, 其具体含义为:第 i 辆车从配送中心出发,经过 $p_{i1}, p_{i2}, \dots, p_{it}$, 再回到配送中心。如一个有 8 个客户需求点的染色体(抗体)012304506780 所表示的行驶路径为:

子路径 1: 配送中心 0 → 1 号客户点 → 2 号客户点 → 3 号客户点 → 配送中心 0

子路径 2: 配送中心 0 → 4 号客户点 → 5 号客户点 → 配送中心 0

子路径 3: 配送中心 0 → 6 号客户点 → 7 号客户点 → 8 号客户点 → 配送中心 0

染色体内部的子路径之间是无序的,子路径内部是有序的,子路径的全正序与全反序排列不受影响,即不改变目标函数的值^[5]。

2.2 算法设计

抗原对应于所优化的目标函数和约束条件,抗体对应于问题在给定范围内的可行解。

Step1 设定遗传种群的大小为 n , 记忆库的大小为 l , 抗体亲和力阈值为 T_{c1} , 记忆抗体免疫亲和力阈值为 T_{c2} , 遗传种群和记忆库共同组成初始群体, 记为 $p(t) (t = 0)$, 初始交叉、变异概率分别为 PC 、 PM , 中止条件为 S_c 。

对于中止条件 S_c , 由于遗传算法具有较大的随机性, 一般给出一条或几条启发式的中止条件, 只要有一条满足, 就认为算法收敛了, 如, 迭代次数达到规定值时, 可停止计算, 计算每代群体中染色体适应度的方差, 当方差小于一给定值 ϵ 时, 可认为算法收敛。

Step2 初始化初始种群 $p(t)$ 。

抗体的产生如下:随机产生 k 个配送客户点的一个全排列,如 i_1, i_2, \dots, i_k , 若 $\sum_{j=i_1}^{s-1} g_{ij} \leq q$, 且 $\sum_{j=i_1}^s g_{ij} > q$, 则在原排列的 i_{s-1} 与 i_s 之间插入一个 0, 再从新排列的 i_s 开始, 若 $\sum_{j=i_s}^{t-1} g_{ij} \leq q$, 且 $\sum_{j=i_s}^t g_{ij} > q$, 则在新排列的 i_{t-1} 与 i_t 之间插入一个 0, 依次类推, 直到在排列中插入了 $m-1$ 个 0, 然后在最后再插入一个 0, 就构成了一个抗体。

Step3 计算各抗体之间的亲和力, 记抗体 g 和 h 的亲和力为 S_{gh} 。

抗体的亲和力反映了抗体之间的相似度, 文中采用文献[6]的抗体之间的亲和力计算方法, 即基于分组匹配的亲和力计算方法, 将其中一个抗体按字路线分成若干组, 把每一组与另一个抗体进行逐个比对, 得到每组匹配数, 将各组匹配数之和除以抗体的长度, 得到这两个抗体的亲和力。

Step4 计算各抗体与抗原的亲和力。

文中将抗体 g 与抗原的亲和力定义为抗体 g 的适应度, 将运输成本作为目标函数, 为迅速改进适应度的值, 缩短算法运行时间, 适应度函数采用一种加速适应度函数, 抗体 g 的亲和力定义为:

$$A(g) = \exp\left(-\frac{f(g) - f_{\min}}{g}\right)$$

Step5 计算各抗体的浓度。

抗体 g 的浓度定义为:

$$C(g) = \frac{\sum_{i=1}^{n+l} R_{gi}}{n+l}, R_{gi} = \begin{cases} 1, & S_{gi} \geq T_{c1} \\ 0, & \text{否则} \end{cases}$$

Step6 生成记忆库的抗体。

将抗体浓度按降序排列, 选择其前 l 格作为记忆库抗体。因为浓度较大表明抗体在群体中占据了较大优势, 达到了一个相对最优点, 此时生成一个记忆抗体, 以记录此局部最优解。若记忆库已满, 则与抗原亲和力最低的记忆抗体细胞被新产生的有较高抗原亲和力的记忆抗体所取代。

Step7 计算抗体的生存力。

抗体 g 的生存力定义为: $e(g) = \frac{A(g)}{C(g) \sum_k A(k)}$ 。

$\prod_s (1 - L_{gs})$, 其中 s 为记忆库中的抗体, $L_{gs} = \begin{cases} 1, & S_{gs} \geq T_{c2} \\ 0, & \text{否则} \end{cases}$, S_{gs} 即为抗体 g 与记忆库抗体 s 的亲和力, T_i 为记忆库抗体免疫选择阈值。

Step8 选择产生遗传群体。

按照抗体生存力的大小, 将 $n+l$ 个抗体分成 $n+$

$l+1$ 个区间, 按照轮盘赌法选择 n 个抗体组成遗传群体。

Step9 对遗传种群进行交叉操作。

当抗体与抗原的亲和力, 即其适应度比较集中或分散时, 可适当增大或减小其交叉概率, 交叉概率 P_c 值由下式确定:

$$P_c = PC \frac{f_{\max} - f'}{f_{\max} - f_{\text{mean}}}, f' \geq f_{\text{mean}} \\ P_c = PC, f' < f_{\text{mean}}$$

其中, f_{\max} 为群体的最大适应度值, f_{mean} 为群体的平均适应度值, f' 为交叉产生的两个新个体中适应度大的个体的适应度值[6]。

Step10 对遗传群体进行变异操作。

当抗体与抗原的亲和力, 即其适应度比较集中或分散时, 可适当增大或减小其变异概率, 变异概率 P_m 值由下式确定:

$$P_m = PM \frac{f_{\max} - f}{f_{\max} - f_{\text{mean}}}, f \geq f_{\text{mean}} \\ P_m = PM, f < f_{\text{mean}}$$

其中, f_{\max} 为群体的最大适应度值, f_{mean} 为群体的平均适应度值, f 为变异个体的适应度值。

Step11 判断是否满足中止条件 S_c , 若满足, 则结束算法, 输出结果, 否则, $t = t + 1$, 以新产生的遗传群体和记忆库组成新的初始群体 $p(t+1)$, 转 Step3。

3 实例分析

为与其他算法进行比较, 对文献[5]的算例进行实验分析。该算例由一个配送中心向 7 个需求点进行送货, 每辆车的载重量为 1, 每次配送的最大行驶距离为 300, 配送中心与各客户点的坐标以及各客户点的需求量如表 1 所示(其中 0 表示配送中心)。

表 1 配送中心与客户需求点的坐标及其需求量

序号	0	1	2	3	4	5	6	7
X 坐标	18	22	58	71	83	91	24	18
Y 坐标	54	60	69	71	46	38	42	40
需求量		0.89	0.14	0.28	0.33	0.21	0.41	0.57

参数设置如下:

设车辆容量为 1.0, 由前文计算公式算出共需三辆车来完成所有任务。算法参数为: 抗体数量 30, 交叉概率 0.6, 变异概率 0.3, 抗体亲和力阈值为 0.75, 记忆抗体免疫亲和力阈值为 0.8, 记忆库大小为 5, 进化代数为 100, 为计算方便, 超过最大载重量时的惩罚系数取值为 100000。

文献[5]的实验结果为: 运行 50 次, 平均解为 224.272, 最优解为 217.8, 搜索到最优解的次数为 25 次, 平均进化代数为 36.44。

(下转第 79 页)

(b,0) 的重要性如下:

$$\frac{|[1]_b \cap [1]_d|}{|[1]_b|} - \frac{|[1]_a \cap [1]_d|}{|[1]_a|} = \frac{1}{3} - \frac{1}{1} = -\frac{2}{3}$$

而属性值(c,0)的重要性如下:

$$\frac{|[1]_c \cap [1]_d|}{|[1]_c|} - \frac{|[1]_a \cap [1]_d|}{|[1]_a|} = \frac{1}{2} - \frac{1}{1} = -\frac{1}{2}$$

依据 RAVI 首先考虑属性值最重要的,即(c,0)。依据定义 4,(a,0) 可以做出正确决策,所以把表 2 中第一条决策规则中的属性 c 的值改为“*”,又因属性值(b,0)比属性值(c,0)小,所以第一条规则中的属性 b 的值也改为“*”。

表 2 初步值化简结果

U	a	b	c	d
1	0	?	?	0
2	1	?	*	1
3	?	0	*	1
4	?	1	1	0
5	*	?	2	2
6	*	*	?	2
7	?	*	?	2

逐条决策规则进行考虑,得到表 3。

表 3 进一步处理结果

U	a	b	c	d
1	0	*	*	0
2	1	0	*	1
3	1	0	*	1
4	*	1	1	0
5	*	*	2	2
6	*	*	2	2
7	2	*	2	2

(上接第 76 页)

采用文中的自适应免疫遗传算法的实验结果为:运行 50 次,平均解为 220.674,最优解为 217.8,搜索到最优解的次数 32 次,而搜索到最优解的平均进化代数为 30.53。

通过以上实验结果可以发现,采用自适应遗传算法搜索到最优解或满意解的概率要较大,较好地克服了遗传算法的“早熟”现象,而且,由于采用了自适应的交叉、变异概率,该算法的平均进化代数要少,从而节约了运算时间。实验还进一步发现,随着进化代数的增加,能得到更好的较优解。

4 结 论

文中将自适应免疫遗传算法应用于车辆调度问题。该算法在传统遗传算法的基础上加入了免疫思想,引入了生物免疫机制,还加入了自适应的交叉、变

因为决策规则 5 和 6 重复,所以删除决策规则 6。信息表中决策规则 5 和 7 除属性 a 外其余属性值对应都相等,并且决策规则 5 为“*”,并且依据定义 4 和表 1,可知仅由(c,2)即可做出正确决策,所以删除决策规则 7。最后得到化简结果表 4。

表 4 最终值化简结果

U	a	b	c	d
1	0	*	*	0
2	1	0	*	1
3	*	1	1	0
4	*	*	2	2

4 结 语

利用 Rough 集理论,参考文献[2],[3]等的启发式值约简算法,结合属性值的重要性,提出了一种改进的基于属性值重要性的 Rough 集值约简算法(RAVI)。大大地提高了属性值约简算法的执行效率。

参考文献:

- [1] PAWLAK Z. Rough sets[J]. International Journal of Computer and Information sciences,1982,11(5):341-356.
- [2] 王国胤. Rough 集理论与知识获取[M]. 西安:西安交通大学出版社,2001.
- [3] 林杰斌,刘明德,陈 湘. 数据挖掘与 OLAP 理论与实务[M]. 北京:清华大学出版社,2003.
- [4] 刘 清. Rough 集及 Rough 推理[M]. 北京:科学出版社,2001.
- [5] 胡 斐,张峰筠,刘少辉. 一种基于 Rough 集的属性值约简算法[J]. 计算机工程与应用,2003(31):48-51.

异算子,实验证明,该算法具有较好的全局搜索性能和较快的收敛性,能在较少的进化代数下找到较优解,能较好地避免局部最优解,可有效地解决车辆调度问题。

参考文献:

- [1] 杨 弋,顾幸生. 物流配送车辆优化调度的综述[J]. 东南大学学报:自然科学版,2003,33(9):105-111.
- [2] Dasgupta D. Artificial Immune Systems and Their Applications[M]. Berlin, Heidelberg: Springer-Verlag,1999.
- [3] 章 棘,周 泉. 基于免疫克隆算法的物流配送车辆路径优化研究[J]. 湖南大学学报:自然科学版,2004,31(5):54-58.
- [4] 阎 庆,鲍远律. 新型模拟退火算法求解物流配送路径问题[J]. 计算机应用,2004,24(6):261-263.
- [5] 李 菁,王宗军,蒋元涛,等. 免疫算法在车辆调度问题中的应用[J]. 运筹与管理,2003,12(6):96-100.
- [6] 宋玉林,齐 欢. 基于自适应遗传算法的配送车辆调度聚类分析[J]. 计算机与数字工程,2004,32(2):45-47.