

基于禁忌搜索算法求解定位—运输路线安排问题

林晖钢¹, 胡大伟¹, 徐丽蕊^{1, 2}

1 长安大学汽车学院, 西安 (710064)

2 陕西工业职业技术学院工商管理系, 陕西咸阳 (712000)

E-mail: lhgzhl@163.com

摘 要: 定位—运输路线安排问题 (location routing problems, LRP) 是集成化物流系统分销网络设计和管理决策中的难题, 也是任何一个大型物流配送企业必须要面对的问题。由于 LRP 的 NP—HARD 属性, 其求解方法目前大多局限在将定位—配给问题 (location allocation problems, LAP) 的输出作为车辆路线安排问题 (vehicle routing problems, VRP) 的输入而求解。然而, 在 LAP 最优的前提下求出的 VRP 的最优并不一定就是 LRP 的最优解, 从而导致这样的处理方式不可避免的会陷入局部最优解的状态。本文针对多站点定位—运输路线安排问题 (multi-depot location routing problems, MDLRP) 数学模型, 用 Lingo 软件对小规模测试数据情形进行了验证, 然后采用禁忌搜索法 (TS) 分别求解 LAP 和对应的每一个设施的 VRP, 并将 VRP 的结果作为 LAP 的输入, 再将 LAP 解及其邻域解作为 VRP 输入不断反复循环求解 MDLRP, 并在此基础上对较大规模测试数据进行了仿真运算。结果表明采用禁忌搜索方法求解一定规模的 MDLRP 快速有效。

关键词: 定位—运输路线安排问题; 定位—配给问题; 车辆路线问题; 禁忌搜索

1. 引言

设施定位、车辆路线各自作为单独的问题, 国内外已有较多学者进行了研究, 其理论也已比较完善, 这些研究为物流管理的优化决策奠定了坚实的基础。国外一批学者对 LRP 进行了一系列的研究^[1]。LRP 的概念认为: 在设施 (制造厂、库存点或分销中心) 相对于客户的位置、货物的配给、货物运输的车辆路线安排之间存在相互依赖的关系, 根据这种关系来进行综合优化与管理; 相比单一的物流系统优化问题, LRP 更加贴近目前物流系统复杂的实际特征, 对进一步优化整个物流系统, 降低系统成本具有一定的理论价值和现实意义。而当前大部分学者对 LRP 的研究都局限在将 LAP 的输出作为 VRP 的输入而进行求解, 这种方法得出的 LRP 解往往并不是最优的。基于这样一个现实情况, 本文将 LAP 和 VRP 综合考虑, 得出了相应的 MDLRP 优化解, 首先用 C++ 编程软件与 Lingo 软件对小规模数据集 (8 点数据集) 进行了计算及其对比验证本文算法计算结果的精确性, 然后对较大规模数据 (25 点、50 点、100 点数据集) 用本文算法求出了相应的优化解, 证明了本文算法对求解 MDLRP 问题的有效性和准确性。

2. MDLRP 参数定义及其数学模型与验算

本文研究的 LRP 模型基于如下假设:

- ① 客户数量、位置及需求已知;
- ② 候选设施位置及容量已知;
- ③ 各客户需求均能得到满足且每个客户只由一辆车服务;
- ④ 每辆车在完成全部运输任务后回到出发点;
- ⑤ 各线路的总需求小于或等于车辆的容量;
- ⑥ 车辆类型给定。

2.1 参数定义

x_{ijk} 在路线 k 上, 如果点 i 在 j 之前值为1, 否则为0;

y_i 如果建立 i 设施则值为1, 否则为0;

z_{ij} 如果客户 j 由设施 i 来服务则值为1, 否则为0;

R 配送设施点潜在位置的集合;

J 所有客户点的集合;

V 所有车辆集合;

N 代表客户的数量;

C_{ij} i, j 两点之间的距离 (作为 i, j 两点间的运输费用);

$S = R \cup J$ 所有客户点和潜在设施点的集合;

G_r 设施 r 建设和运行的固定成本;

F_k 使用车辆 k 的固定成本;

V_r 设施 r 的最大容量;

q_j 客户 j 的需求量;

Q_k 车辆 k 的容量;

2.2 多站点定位—配给问题的数学模型

$$\text{Min} \quad \sum_{r \in R} G_r y_r + \sum_{i \in S} \sum_{j \in S} \sum_{k \in V} C_{ij} x_{ijk} + \sum_{k \in V} F_k \sum_{i \in R} \sum_{j \in J} x_{ijk}$$

s.t :

$$\sum_{k \in V} \sum_{i \in S} x_{ijk} = 1, \quad j \in J, \quad (1)$$

$$\sum_{i \in S} \sum_{j \in J} q_j x_{ijk} \leq Q_k, \quad k \in V, \quad (2)$$

$$\sum_{j \in J} q_j z_{rj} - V_r y_r \leq 0, \quad r \in R, \quad (3)$$

$$\sum_{i \in S} x_{ipk} - \sum_{j \in S} x_{pj k} = 0, \quad k \in V, \quad p \in S, \quad i \neq j \quad (4)$$

$$\sum_{r \in R} \sum_{j \in J} x_{rjk} \leq 1, \quad k \in V, \quad (5)$$

$$\sum_{l \in S} (x_{rlk} + x_{ljk}) \leq 1 + z_{rj}, \quad r \in R, \quad j \in J, \quad k \in V \quad (6)$$

$$x_{ijk} = 0 \text{ 或 } 1, \quad i, j \in S, k \in V, i \neq j, \quad (7)$$

$$y_r = 0 \text{ 或 } 1, \quad r \in R, \quad (8)$$

$$z_{ij} = 0 \text{ 或 } 1, \quad i \in R, j \in J \quad (9)$$

在上述这个模型中, 目标函数为总成本 (包括设施建设和运作成本、运输成本以及车辆投入及其使用成本) 最小。约束 (1) 保证每一个客户只由一个运输车辆服务; 约束 (2) 为运输车辆容量约束条件; 约束 (3) 为仓库的容量限制条件; 约束 (4) 保证路线连续; 约束 (5) 保证每一个运输工具的路线只能从一个设施驶出; 约束 (6) 保证客户只能由选中的设施提供服务; 约束 (7) — (9) 表示 0、1 变量约束。

2.3 模型检验

由于目前学术界并不存在任何定位一路线问题的测试数据。为了检验此模型的正确性,本文以 VRP 的 Solomon 测试数据源 R101 数据系列为基础随机产生包含 3 个候选设施点和 5 个客户需求点的数据组,对定位一路线问题数学模型进行检验。通过产生 1~100 范围内的 8 个随机数对应于 100 个客户中的各个客户编号,将其中三个作为本算例中候选设施点的数据,其余 5 个作为客户点的数据。这样随机产生的一组候选设施点和客户,潜在设施点的坐标、容量和建设成本如表 1 所示;客户点的坐标和需求量的数据如表 2 所示。车辆容量取 40,车辆固定成本取 30。

表1 潜在设施点的坐标、容量和建设成本

	D1	D2	D3
X坐标	35	25	10
Y坐标	35	30	43
设施点的容量	70	70	70
建设成本	200	200	200

表2 客户点的坐标和需求量

	C1	C2	C3	C4	C5
X坐标	30	50	10	45	45
Y坐标	60	35	20	65	10
客户需求量	16	19	19	9	18

用 Lingo10.0 对 MDLRP 模型进行编程,计算时间为 156s,结果表明两个设施被选中,对应的目标函数值为 666.406,从而证明了本文带容量约束的 MDLRP 数学模型的正确性。

3. 禁忌搜索算法介绍

TS (Tabu Search) 算法是近年来受到普遍关注的一种高效率的现代启发式优化算法,该算法由 F.Glover^[2]于 1986 年首先提出,进而形成一套完整算法^{[3][4]}。并随着计算机技术的发展而成功的应用于各个领域,解决了大量复杂的优化问题。近几年,该算法被广泛领域引入,如水电火电联合经济调度^[3]、电力系统无功优化^[4]、输电系统最优规划^[5]以及交通运输系统规划^[5]等领域,并取得了一定研究成果。部邻域搜索陷入局部最优的不足,禁忌搜索算法用一个禁忌表记录下已经到达过的局部最优点,在下一搜索中,利用禁忌表中的信息不再或有选择地搜索这些点,以此来跳出局部最优点。

本文利用禁忌搜索算法求解组合优化问题时,首先产生一个初始解作为当前解,然后在当前解的邻域中搜索若干邻域解,取其中的最好解作为新的当前解。为了避免对已搜索过的局部最优解的重复,禁忌搜索算法使用禁忌表记录已搜索的局部最优解的历史信息,这使得算法可在一定程度上避开局部最优点,从而开辟新的搜索区域。本文设计的禁忌搜索算法流程图如图 1 所示,

禁忌搜索算法的特点是使用了禁忌技术。所谓禁忌就是禁止重复前面搜索,为了回避局

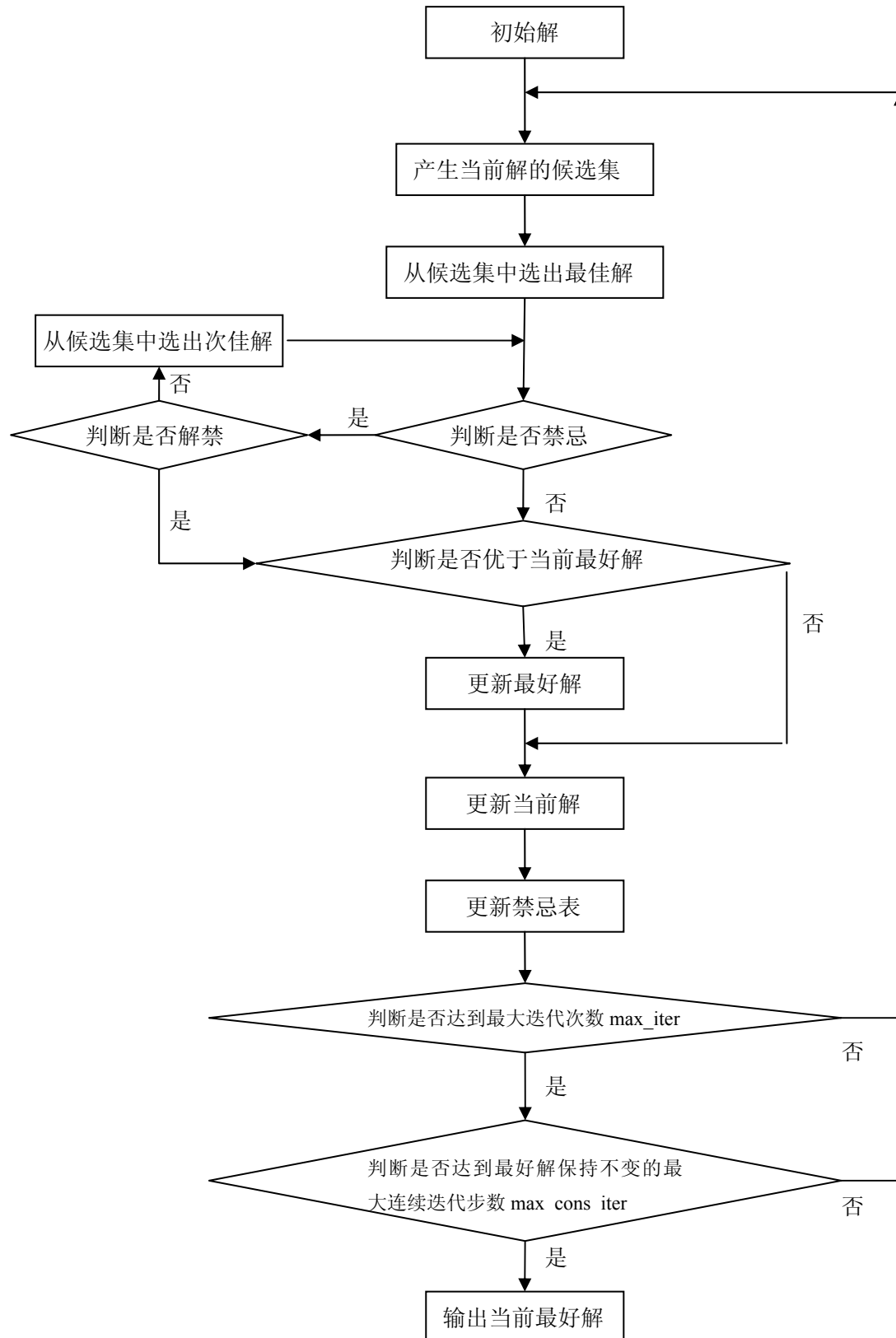


图1 禁忌搜索算法流程图

禁忌搜索算法作为一种人工智能算法，涉及解的表示、解的评价、邻域解的构造及算法终止准则，并且实现的技术问题是算法的关键，不同的算法策略将会对解的好坏产生重要影响。禁忌搜索算法的主要策略有：禁忌对象和禁忌长度的确定、候选集的产生方法、特赦规

则以及终止规则。

(1)解的表示^{[8][9]}

不同编码方式的运行时间和适应度函数值以及目标函数值计算的复杂程度不同,并且编码方式也影响着解空间的大小。一般来说,在与车辆路线问题有关问题的求解中都采用自然数编码的方式,而自然数编码方式也存在着如下三种不同形式:

第一种,客户和配送中心共同排列的编码方式。

第二种,车辆和客户对应排列的编码方式。

第三种,客户直接排列的编码方式。

根据本文研究问题的特征,本文采用第三种编码方式,即客户直接排列的编码方式确定一组 MDLRP 的初始可行解。

(2)禁忌对象的确定

禁忌对象是指禁忌表中被禁的元素。由于解状态的变化可分为解的简单变化、解向量分量的变化和值变化三种情况,则在确定禁忌对象时也有对解的简单变化进行禁忌、对解的分量变化进行禁忌及对解的值变化进行禁忌三种情况。一般来说,对解的简单变化进行禁忌比对解的分量变化进行禁忌和对解的值变化进行禁忌的受禁范围要小,因此可能造成计算时间的增加,但其优点是提供了较大的搜索范围。解分量的变化和值变化的禁忌范围要大,这减少了计算的时间,可能引发的问题时禁忌的范围增大以至陷入局部最优解。

根据定位配给问题和车辆路线问题的特点,可采用对解的简单变化进行禁忌的方法。其原理是:当解从 x 变化到 y 时, Y 可能是局部最优解,为了避开局部最优解,禁忌 y 这一解再度出现,可采用如下禁忌规则:当 y 的邻域中有比它更优的解时,选择更优的解;当 Y 为 $N(y)$ 的局部最优解时,不再选 Y ,而选比 y 稍差的解。

(3)禁忌长度的确定

禁忌长度是指被禁对象不允许被选取的迭代步数,一般是给被禁忌对象 x 一个数 L (称为禁忌长度),要求禁忌对象 x 在 L 步迭代内被禁,在禁忌表中采用 $\text{Tabu}(x)=L$ 记忆,每迭代一步,该项指标做运算 $\text{Tabu}(x)=L-1$,直到 $\text{Tabu}(x)=0$ 时解禁。禁忌长度在同一问题求解过程中是固定的,但是随问题规模的不同取不同的值,一般来说,问题规模越大,禁忌长度越长。禁忌长度短会造成循环,也可能在一个局部最优解附近循环。禁忌长度长会造成算法的记忆存储量增加,使得算法计算时间增加,还可能造成算法无法继续计算下去。本文 L 取为常数, $L=\sqrt{n}$ (n 为邻域中邻居的总个数),这种规则容易在算法中实现。

(4)候选集的产生方法

候选集合由邻域中的邻居组成。常规的方法是从邻域中选择若干个目标值或评价值最佳的邻居入选。有时这种计算量较大,而且容易导致局部收敛的情况,所以本文采用不在邻域的所有邻居中选择,而是在邻域中的一部分邻居中随机选择若干个目标值或评价值最佳的状态实现部分邻居的选取。由于考虑到计算时间的复杂度问题,本文设计每次对当前解随机进行 5 次邻域解产生操作,将其作为当前解的候选集。

在定位阶段既要确定配送中心数目,又要选择具体的配送中心,因此设计两种邻域搜索方法,分别是“客户交换”和“客户插入”操作。

根据本文特点,在路线优化阶段,设计随机从插入法、2-opt、路线内 2swap 和路线间 2-swap 四种邻域搜索方法中选择其中一种来产生邻域解,这样增强了禁忌搜索算法的搜索

性能，扩大了搜索范围，以便能在车辆行驶费用方面得到更好的改进。

①插入法

插入法是考虑一条路线中任意一个点插入到其他路线中的方法，即从一条路线中随机选择一个点，插入到另一条路线中随机选择的两个点之间，如图 2，从一条路线中随机选择点 4，另一条路线中随机选择点 7，通过插入法，则路线由 0-1-2-3-4-0 和 0-5-6-7-8-0 变为 0-1-2-3-0 和 0-5-6-7-4-8-0。

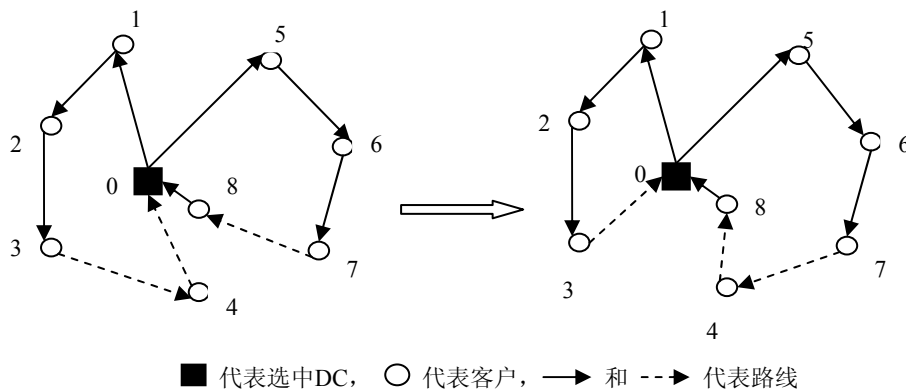


图2 插入法示意图

②2-opt

将当前路线中的 k 条边用另外 k 条边代替，这种变换称为 k -opt^[10]，最为常见的是 2-opt，主要思想是随机选择两个点 i 和 j ，将边 $(i, i+1)$ 、 $(j, j+1)$ 用 (i, j) 、 $(i+1, j+1)$ 代替，如图 3 所示，随机选择 2 和 5，进行 2-opt 交换，则路线由 0-1-2-3-4-5-6-0 变为 0-1-2-5-4-3-6-0。

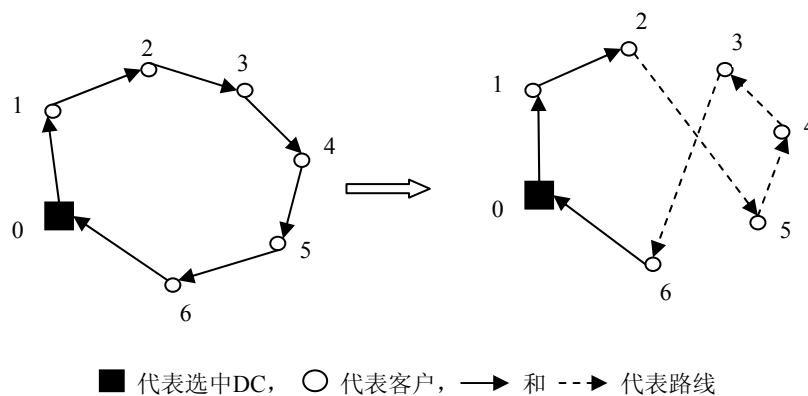


图3 2-opt交换法示意图[11]

③路线内2-Swap

路线内 2-swap 主要思想是交换一条路线上两个点的位置，如图 4 所示，随机选择 2 和 5，进行路线内 2-Swap，则路线由 0-1-2-3-4-5-6-0 变为 0-1-5-3-4-2-6-0。

④路线间 2-Swap

路线间 2-swap 主要思想是交换不同路线上两个点的位置，如图 5 所示，随机选择两条路线上的两个点 2 和 5，进行路线间 2-Swap，则路线由 0...1-2-3...0 和 0...4-5-6...0 变为 0...1-5-3...0 和 0...4-2-6...0。这种思想在车辆路线问题的客户分配中可以起到优

化分组的作用。

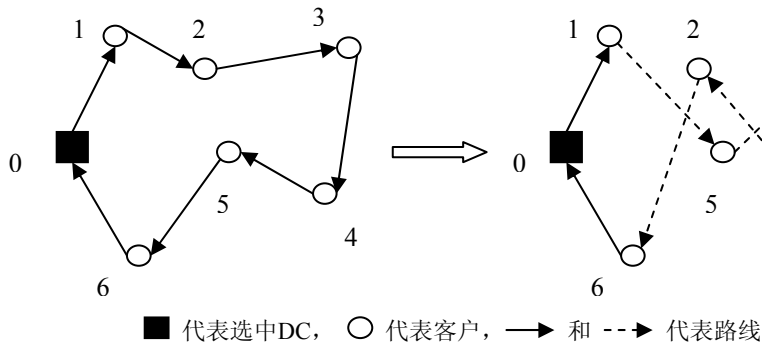


图4 2-Swap路线内交换法示意图

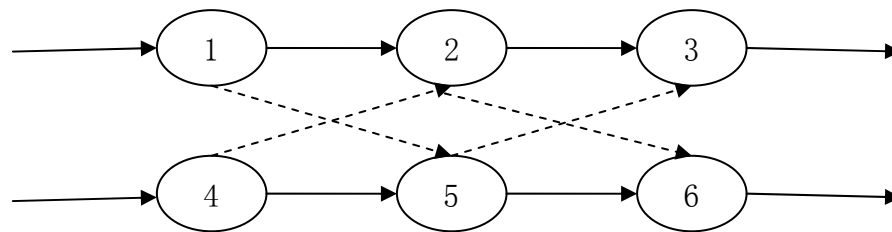


图5 2-swap路线间交换示意图

(5) 特赦规则

在禁忌搜索算法的迭代过程中, 会出现候选集中的全部对象都被禁忌, 或某一对象被禁, 但若解禁后其目标值将下降情况。在这样的情况下, 为了达到全局的最优, 让一些禁忌对象重新可选, 这种方法称为特赦, 相应的规则称为特赦规则, 也即解禁原则。

本文采用基于评价值的特赦规则。即在整个计算过程中, 记忆已出现的最好解 x^{best} , 当候选集中被禁忌解的邻域中, 出现一个解 x^{now} 优于 x^{best} , 则解禁 x^{now} 使其自由, 并更新为当前解和最好解。

(6) 评价函数

评价函数赋予候选集中每一个元素一个实数值, 通过评价函数值来选取下一步计算的替代点。

领域搜索中需要对在得到的几个解中选择质量最优的解作为新的当前解, 为了提高算法效率, 在定位阶段评价解的优劣时, 我们通过目标函数预测值来比较这几个解的质量。预测方法是: 根据求解的设施定位方案, 在满足设施容量约束的前提下, 将客户分配至距离最近的设施点, 假设设施与客户间的运输路线为直线距离, 并且不考虑车辆路线, 而采用各客户与设施点距离总和作为目标函数预测值。

而在路线优化阶段, 则用配送路线方案的目标函数值作为评价函数, 对候选解进行评价和选取最优值。其目标函数值越优, 则解的质量越高。

(7) 终止准则^{[6][7]}

禁忌搜索是一种启发式算法, 希望在可接受的时间里能得到一个满意的解。终止规则就是确定什么时候终止算法, 通常使用的有以下几种方法:

① 确定步数终止;

- ②频率控制原则；
- ③目标值变化控制原则；
- ④目标值偏离程度原则。

为便于算法实现，本文综合方法①和方法③，采用事先确定最大迭代代数和最好解保持不变的最大连续迭代步数作为终止规则。

4. 实例计算及结果分析

本文中所有计算实例中的单位距离运输费用均取为 1；对于 8 点数据由于有可比对象（LINGO 计算结果），为了对比方便程序当中车辆费用取为 30；而对于 25 点、50 点和 100 点的数据由于没有可比对象，所以在程序中车辆容量暂时设定为：50，90，150；车辆费用暂时分别设定为 30，0，0；当然也可以根据需要自行设定。

根据模型检验中的 8 点数据，对本文禁忌搜索算法运用 C++ 语言编程计算，对应的参数设置如下：禁忌长度取为 3，最大迭代次数 max_cons_iter 取为 800，候选解数量取为 5，LRP 迭代次数取为 20 时，得出目标函数的结果如表 3 所示。

表3 计算结果

选中的设施	对应的车辆路线	目标函数值	CPU 运行时间
1	0-4-1-0 , 0-2-5-0	666.406	0.0922 秒
2	0-3-0		

根据上表可以看出，由本文算法模拟计算，得出的结果与我们应用 Lingo10.0 对 MDLRP 模型 8 点数据进行编程计算得出的目标函数值完全相同，并且计算 20 次得出该最优解的频率为 85%以上，计算机运行时间也很快。从而验证了本文程序的有效性。

在 solomon 数据集中 R101 数据集合（见表 4）中随机选取其中 5 个客户点作为设施节点，其余 20 个作为客户点，用本文算法进行编程计算在相应的参数设置为：禁忌长度取为 5，最大迭代次数 max_cons_iter 取为 8000，候选解数量取为 25，LRP 迭代次数取为 50 时，计算 20 次得出的最好结果如表 5 所示，最好结果出现的频率为 75%以上。

表4 设施点坐标、容量、建设费用和客户点坐标、需求量

潜在设施				X		Y		容量		建设费用	
1				15		30		100		100	
2				30		60		100		100	
3				50		35		100		100	
4				30		25		100		100	
5				40		20		100		100	
客	X	Y	需	客	X	Y	需	客	X	Y	需求
户			求	户			求	户			量
			量				量				
1	41	48	10	8	55	60	16	15	15	60	17
2	35	17	7	9	20	65	12	16	45	65	9
3	55	45	13	10	15	10	20	17	45	10	18
4	55	20	19	11	30	5	8	18	55	5	29

5	25	30	3	12	10	20	19	19	65	30	3
6	20	50	5	13	5	30	2	20	65	20	6
7	10	43	9	14	20	40	12				

表5 计算结果

选中的设施	对应的车辆路线	目标函数值	CPU 运行时间
1 2 5	0-5-1-7-13-0 , 0-12-10-0 0-6-15-9-0 , 0-1-3-8-16-0 0-17-18-0 , 0-2-11-4-20-19-0	832.557	11.514 秒

在 solomon 数据集中 R101 数据集合（见表 6）中随机选取其中 8 个客户点（根据该数据集中的客户需求总量确定）作为设施节点，其余 42 个作为客户点，用本文算法进行编程计算在相应的参数设置为：禁忌长度取为 5，最大迭代次数 max_cons_iter 取为 4000，候选解数量取为 20，LRP 迭代次数取为 30 时，计算 20 次得出的最好结果如表 7 所示，该最好结果出现的频率为 80%以上。

表6 设施点坐标、容量、建设费用和客户点坐标、需求量

潜在设施		X		Y		容量		建设费用			
1		42		66		200		200			
2		38		70		200		200			
3		28		55		200		200			
4		23		55		200		200			
5		20		80		200		200			
6		25		85		200		200			
7		5		35		200		200			
8		30		32		200		200			
客 户	X	Y	需 求 量	客 户	X	Y	需 求 量	客 户	X	Y	需 求 量
1	45	68	10	15	18	75	20	29	10	40	30
2	45	70	30	16	15	80	10	30	2	40	20
3	42	68	10	17	30	50	10	31	0	40	30
4	42	65	10	18	30	52	20	32	0	45	20
5	40	69	20	19	28	52	20	33	35	30	10
6	40	66	20	20	25	50	10	34	35	32	10
7	38	68	20	21	25	52	40	35	33	35	10
8	35	66	10	22	25	55	10	36	32	30	10
9	35	69	10	23	23	52	10	37	30	30	10
10	15	75	20	24	20	50	10	38	5	45	10
11	22	75	30	25	20	55	10	39	30	35	10
12	22	85	10	26	10	35	20	40	28	30	10
13	33	32	20	27	8	40	40	41	28	35	10
14	20	85	40	28	8	45	20	42	26	32	10

表7 计算结果

选中的设施	对应的车辆路线	目标函数值	CPU 运行时间
1	0-1-2-3-5-9-0, 0-7-8-22-19-0 0-18-4 -0, 0-6-0	1154.95	101.771 秒
5	0-12-14-0, 0-16-10-15-23-11-0		
7	0-26-0, 0-27-28-38-30-0, 0-31-32-25-29-0		
8	0-39-41-20-17-35-34-33-13-0, 0-21-24-42-40-37-36-0		

在 solomon R101 数据集（见表 8）中随机选取其中 15 个客户点（根据该数据集中的客户需求总量确定）作为设施节点，其余 85 个作为客户点，用本文算法进行编程计算在相应的参数设置为：禁忌长度取为 5，最大迭代次数 max_cons_iter 取为 4000，候选解数量取为 35，LRP 迭代次数取为 40 时，计算 20 次得出的最好结果如表 9 所示，该最好结果出现的频率为 80%以上。

表8 设施点坐标、容量、建设费用和客户点坐标、需求量

潜在设施				X		Y		容量		建设费用	
1				42		68		500		200	
2				35		66		500		200	
3				30		50		500		200	
4				28		52		500		200	
5				10		40		500		200	
6				44		5		500		200	
7				15		75		500		200	
8				40		5		500		200	
9				66		55		500		200	
10				72		55		500		200	
11				85		25		500		200	
12				92		30		500		200	
13				63		58		500		200	
14				8		45		500		200	
15				45		35		500		200	
客户	X	Y	需求量	客户	X	Y	需求量	客户	X	Y	需求量
1	45	68	10	30	2	40	20	59	47	40	10
2	45	70	30	31	0	40	30	60	45	30	10
3	42	66	10	32	0	45	20	61	95	30	30
4	42	65	10	33	35	30	10	62	95	35	20
5	40	69	20	34	35	32	10	63	53	30	10
6	40	66	20	35	33	32	20	64	45	65	20
7	38	68	20	36	33	35	10	65	90	35	10

8	38	70	10	37	32	30	10	66	88	30	10
9	35	69	10	38	30	30	10	67	88	35	20
10	25	85	20	39	30	32	30	68	87	30	10
11	22	85	10	40	30	35	10	69	85	35	30
12	20	80	40	41	28	30	10	70	75	55	20
13	20	85	40	42	28	35	10	71	70	58	20
14	18	75	20	43	26	32	10	72	68	60	30
15	15	80	10	44	25	30	10	73	65	55	20
16	22	75	30	45	25	35	10	74	65	60	30
17	30	52	20	46	48	40	10	75	5	45	10
18	28	55	10	47	42	10	40	76	60	55	10
19	25	50	10	48	42	15	10	77	67	85	20
20	25	52	40	49	60	60	10	78	65	85	40
21	25	55	10	50	40	15	40	79	65	82	10
22	23	52	10	51	38	5	30	80	62	80	30
23	23	55	20	52	38	15	10	81	60	80	10
24	20	50	10	53	35	5	20	82	60	85	30
25	20	55	10	54	50	30	10	83	58	75	20
26	10	35	20	55	50	35	20	84	55	80	10
27	8	40	40	56	50	40	50	85	55	85	20
28	53	35	50	57	48	30	10				
29	5	35	10	58	47	35	10				

表9 计算结果

选中的设施	对应的车辆路线	目标函数值	CPU 运行时间
2	0-5-9-10-13-14-16-0	2030.98	686.3758 秒
5	0-15-11-1-3-6-7-0		
13	0-12-21-18-17-4-64-2-8-0		
15	0-20-19-40-44-26-30-27-0, 0-42-68-0, 0-75-32-31-29-41-43-45-22-23-25-24-0		
	0-74-80-79-72-69-73-0, 0-49-83-84-85-82-78-70-0, 0-81-77-71-67-65-61-66-76-0		
	0-46-36-39-38-37-34-33-50-48-57-0, 0-58-47-53-52-0, 0-63-62-28-0, 0-35-51-60-54-55-56-59-0		

5. 结论

以上实例的计算结果表明, 本文的 TS 算法对求解 MDLRP 是有效的, 对于小规模的问题可以在很短的时间内计算出最优结果; 对于较大规模的问题, 由于没有相关的可对比算例数据, 通过文中计算结果可以看出, TS 算法在设定恰当的参数组合时, 同样可以在较短的

时间得出相对优化的结果,而且解的收敛性比较稳定。文中的各组参数搭配是在经过多次重复性试验的基础上得出来的。结果显示随着数据规模的增大,虽然 TS 算法寻求优化解的时间也是成倍增大的,但是得到优化解的频率基本保持不变,进一步证明了本文算法的稳定性。

6. 未来研究展望

模型中客户的位置、需求量都是事先确定的,而在现实中这些变量可能都是实时变化的,今后可以从这方面入手使 MDLRP 模型更加贴近实际。本文在求解过程中只用了单一车型,与实际当中的多车型还存在着差距;本文只考虑了当单向送货物流没有将回收物流考虑进来,未来可以加入这部分研究内容实现供应链的增值。随着经济和信息技术的发展,物流的需求也向多样化、个性化方面发展,所以对带时间窗的多站点定位——运输路线问题(MDLRPTW)的研究,也是当今市场的迫切需要。

参考文献

- [1] 张潜,集成化物流中的定位——运输路线安排问题模型及优化算法研究[J]. 东北大学博士学位论文
- [2] Glover F. Future Paths for Integer Programming and Links to Artificial Intelligence[J]. Computers and Operations Research, 1986, (13):533~549.
- [3] Glover F. Tabu Search: part. ORSA Journal on Computing [J]. 1989, (1): 190-206.
- [4] Glover F. Tabu Search: part. ORSA Journal on Computing [J]. 1990, (2):4-32.
- [5] Bai X, Shahidehpour S. Hydro. thermal scheduling by tabu search and decomposition method[J]. IEEE PWRS, 1996, 11 (2): 968-974
- [6] Gan D, Qu Z, Cai H. Large scale var optimization and planning by tabu search[J]. Electric Power System Research, 1996, 39 (3): 195-204
- [7] 文福栓, 韩祯祥, 基于 Tabu 搜索方法的输电系统最优规划[J]. 电网技术, 1997, 21 (5): 2-7
- [8] 刑文训, 谢金星, 现代优化计算方法[M]. 清华大学出版社. 2005 年 9 月.
- [9] 郎茂祥, 物流配送车辆调度问题的模型和算法研究[D]. 北京: 北方交通大学博士学位论文, 2002.2
- [10] S.Lin, Computer solutions of the traveling salesman problem. Bell System Tech[J]. 1964:2245-2246.
- [11] 胡大伟, 陈诚, 遗传算法和禁忌搜索算法在配送中心选址和路线问题中的应用[J]. 系统工程理论与实践, 2007.9

Solving Location Routing Problems with Taboo Search

Lin Huigang¹, Hu Dawei¹, Xu Lirui^{1, 2}

1 College of automobile, Chang'an University, xi'an (710061)

2 Department of Business Administration, Shan'xi Polytechnic Institute, xianyang, Shanxi (712000)

Abstract

Location routing problem is one of the hard problems in distribution network designing and logistic management, which has been used solving every logistics distribution corporations. As a NP-hard problem, most of the algorithms for LRP at present confine to use the output of location allocation problems as the input of vehicle routing problems. However, on the condition of the best output of LAP the best output of VRP would be not the best solve of LRP, which absolutely leads to the situation that this solving manner would be limited into the best local solution. The paper uses Lingo software to validate LRP of test data in miniature, then using taboo search (TS) to solve LAP and each hub's VRP, let the output of VRP as the input of LAP, and then let the solution and neighbor solution of LAP as the input of VRP repeat this process continuous to solve MDLRP, based on which large scale test data has been formulated. The result indicates that using TS solving MDLRP with certain scale data is speediness and effective.

Keywords: location routing problems, location allocation problems, vehicle routing problems, taboo search.