

多物流配送中心选址及求解

李艳冰, 徐克林, 朱 伟

(同济大学 机械工程学院, 上海 201804)

摘要: 经典蚁群算法不能直接用于求解多配送中心选址问题(MDLP), 据此, 将 MDLP 映射为扩展 K-TSP 过程并设计了改进的蚁群算法. 改变了经典蚁群算法禁忌表的设置方式, 算法运行时, 给蚁群建立一个共享禁忌表, 里面存放所有蚂蚁访问过的客户点, 任何蚂蚁只能选择共享禁忌表未曾记录的客户点, 从而增强蚂蚁间的信息交流, 促进它们的分工与协作, 使蚂蚁无遗漏无重复地遍历各配送点并找出问题的最优解. 为提高算法的求解性能, 在蚂蚁的选择规则里加入了代价引导函数, 使用 2-opt 策略优化可行解并优化了信息素的更新方式. 仿真算例及算法对比表明, 模型和算法可以有效地表达和求解 MDLP.

关键词: 多配送中心选址问题; 扩展 K-TSP; 共享禁忌表; 2-opt; 信息素更新

中图分类号: TP301

文献标识码: A

Multi-distribution Center Location Problem and Its Resolution

LI Yanbing, XU Kelin, ZHU Wei

(College of Mechanical and Electrical Engineering, Tongji University, Shanghai 201804, China)

Abstract: To solve multi-distribution center location problem (MDLP) directly, MDLP was mapped into the process of expanded K-TSP and improved ants colony algorithm was designed. The setting mode of tabu for classic ants colony algorithm was changed, specifically, when algorithm run, a sharing tabu for ants colony was constructed and all the distribution points the ants had traversed were put into it, any ants could only choose the customer point unrecorded in it, thereby this intensified exchange of information among ants and promoted their division and cooperation, so that the ants visited all the distribution sites exhaustively without repetition, and finally found the optimal solution. In order to improve the algorithm's performance, price guided function was added to ants' selection rule, and 2-opt strategy was used to optimize the feasible solution, and optimized pheromone

update policy was designed. The simulation example and the algorithm comparison show that the model and the algorithm may express and solve MDLP effectively.

Key words: multi-distribution center location problem; extended K-TSP; sharing tabu; 2-opt; pheromone update

物流系统中, 配送中心是连接物流上下游的枢纽, 对促进生产与消费的协调与配合、保证物流系统的平衡发展起着重要作用. 在各企业努力降低成本、增加利润和增强竞争力的今天, 配送中心的选址问题尤为受人关注. 多配送中心选址问题 (multi-distribution center location problem, MDLP) 的求解具有 NP(非确定多项式) 难的性质, 近年来, 各种启发式算法^[1-3] 和智能算法^[4-5] 的研究成为该领域热点.

蚁群算法采用正反馈并行自催化机制, 具有较强的顽健性及分布计算能力, 容易与其他优化算法相融合, 尤适于求解复杂的组合优化问题^[6], 但因其不能直接求解 MDLP 等原因, 目前蚁群算法在这方面的研究成果不多. 为此, 通过改变禁忌表设置方式、改进转移规则、2-opt 优化及信息素更新等策略, 本文提出了改进的蚁群算法求解多配送中心选址问题并通过数值实验验证了模型和算法的有效性.

1 MDLP 模型

多配送中心选址问题可描述为: 在给定的 P 个候选位置中选择 k 个点, 以合理的规模建立配送中心, 服务 l 个客户点的配送需求. 如何选择 k 个位置, 使得在满足客户需求及配送中心供应能力前提下, 服务成本(运输成本、配送中心的可变成本和固定成本之和)最小.

收稿日期: 2011-03-09

第一作者: 李艳冰(1976—), 男, 博士研究生, 主要研究方向为工业工程、现代物流与供应链管理, E-mail: liyanbing05@gmail.com

通讯作者: 徐克林(1945—), 女, 教授, 博士生导师, 主要研究方向为现代工业工程. E-mail: tjklxu@163.com

假定每个客户点需求只由一个配送中心服务,记 $C_i (1 \leq i \leq p)$ 为中心 i 的供应能力, $q_j (1 \leq j \leq l)$ 为客户点 j 的需求量, d_{uv} 为网络中节点 u 到 v 的距离, r_w 是节点 u 到 v 的运输费率, N_i 是配送中心 i 服务的客户点数, 配送中心 i 服务的第 j 个客户记为 s_i^j , f_i 为中心 i 的固定费用, 则 MDLP 模型如下:

$$F_{\min} = \sum_{i=1}^p z_i \left\{ (d_{i, s_i^1} + d_{s_i^1, i} + \sum_{k=1}^{N_i-1} d_{s_i^k, s_i^{k+1}}) + f_i \right\} \quad (1)$$

$$\text{s. t. } C_i \sum_{j=1}^p z_i \geq \sum_{j=1}^l q_j \quad (2)$$

$$\sum_{j=1}^l q_j y_{ij} \leq C_i, \quad i = 1, 2, \dots, k \quad (3)$$

$$\sum_{i=1}^k y_{ij} = 1, \quad j = 1, 2, \dots, l \quad (4)$$

$$\sum_{j=1}^l y_{ij} = N_i, \quad i = 1, 2, \dots, p \quad (5)$$

$$\sum_{i=1}^p z_i = k \quad (6)$$

$$z_i = \begin{cases} 1 & \text{候选点 } i \text{ 被选中} \\ 0 & \text{否则} \end{cases} \quad (7)$$

$$y_{ij} = \begin{cases} 1 & \text{客户点 } j \text{ 由中心 } i \text{ 服务} \\ 0 & \text{否则} \end{cases} \quad (8)$$

其中, 式(1)为目标函数, 表示系统服务成本最低; 式(2)表示所选配送中心总的服务能力不小于所有客户总的需求量; 式(3)表示每个被选中心的服务能力不小于其服务客户的总需求量; 式(4)表示客户需求由一个配送中心服务; 式(5)为配送中心 i 服务的客户数量表达式; 式(6)为配送中心的数量约束; z_i, y_{ij} 为决策变量。

2 MDLP 与扩展 K-TSP 的映射

在研究 MDLP 时, 发现它与扩展 K-TSP 有诸多相似点. 为此, 把扩展 K-TSP 作为 MDLP 的数学原型, 下面考察 K-TSP 与 MDLP 的映射关系。

2.1 扩展 K-TSP 问题

扩展 K-TSP 问题可描述为: k 个人从 k 个城市出发分头去访问 $n+1$ 个城市, 每个城市有且仅有一个人到达, 最后 k 个人都回到原出发城市, 问怎样安排使得 k 个人的总访问路线最短^[7]?

该问题的数学语言表示为: 设 $V = (v_1, v_2, \dots, v_n, v_{n+1})$ 为平面上 $n+1$ 个点的集合; $G = (V, E)$ 是 V 上的完全图; $W: E \rightarrow R$ 为权函数. 称 H 为图 $G = (V, E)$ 的 k -周游路, 如果它是 k 条子周游路的集合

$H = (H_1, H_2, \dots, H_k)$, 这里: ① H_i 至少包含 2 条边, $i = 1, 2, \dots, k$; ② H_i 经过顶点 $v_i, i = 1, 2, \dots, k$; ③ 任给 $v \in V / \bigcup_{i=1}^k \{v_i\}$, 存在唯一的子周游路 $H_j (j = 1, 2, \dots, k)$ 经过 v .

k -周游路 H 的长度记为 $L(H)$, 即

$$L(H) = \sum_{i=1}^k L(H_i) = \sum_{i=1}^k \sum_{E \in H_i} L(E) \quad (9)$$

2.2 K-TSP 与 MDLP 映射过程

由对 MDLP 的描述可知, 在 MDLP 中, 一组 (k 个) 配送中心被选定, 这些配送中心可以映射为 K-TSP 中的 K 个出发城市, l 个需求点相当于 K-TSP 中被访问的 $n+1$ 个城市, 运输工具从选定的 k 个配送中心出发给 l 个需求点的配送过程可映射为 K-TSP 中旅行商从给定的 k 个城市出发分头去访问 $n+1$ 个城市的过程, MDLP 的总服务成本最小可以映射为 K-TSP 的总访问路线最短. 二者的映射关系如图 1 所示。

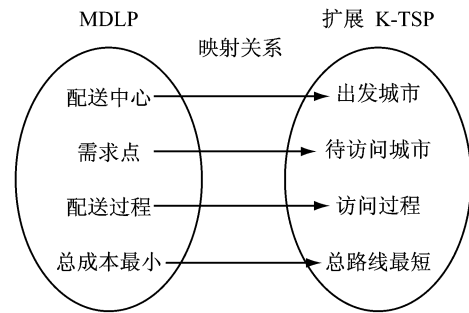


图 1 MDLP 与扩展 K-TSP 映射过程

Fig.1 Mapping between MDLP and expended K-TSP

由此可知, 求解扩展 K-TSP 的算法可应用于 MDLP。

3 求解 MDLP 的改进蚁群算法

3.1 蚁群算法

蚁群算法 (ant colony algorithm, ACA) 是受启于蚂蚁觅食行为的一种仿生算法. 在蚁群算法中, 为每只蚂蚁设置一个禁忌表 t_{abu} , 其中存放已经访问过的节点. 当一只蚂蚁在 t 时刻从节点 i 向节点 j 转移时, 按式(10)确定的概率选择转移目标^[8].

$$p_{ij}(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}(t)]^\beta}{\sum_{s \notin t_{abu}} [\tau_{is}(t)]^\alpha \cdot [\eta_{is}(t)]^\beta}, & j \notin t_{abu} \\ 0, & \text{其他} \end{cases} \quad (10)$$

式中: $\tau_{ij}(t)$ 为 t 时刻节点 i, j 间路径上的信息素量; $\eta_{ij}(t) = 1/d_{ij}$ 为启发函数; α 为信息素启发因子; β 为

期望函数启发因子。

为避免信息素积累淹没启发信息,当蚁群完成一次遍历后,按式(11)更新信息素。

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \Delta\tau_{ij}(t) \quad (11)$$

式中: $\rho(0<\rho<1)$ 为信息素挥发因子; $\Delta\tau_{ij}(t)$ 为 t 时刻蚂蚁遍历节点 i, j 间路径时留下的信息素量, $\Delta\tau_{ij}(t)$ 的初始化为零。

蚁群算法(ACA)能有效求解旅行商问题(travelling salesman problem, TSP),但它不能直接求解 K-TSP,因而更不能直接求解扩展 K-TSP。ACA求解 TSP 问题时,每条蚁路都是一个完全路径,都是一个可行解;在求解 K-TSP 问题时,每只蚂蚁只形成子路径,因而存在选择子路径构造可行解的问题。蚂蚁的子路径存在下述几种情况:(1)蚂蚁的所有子路径恰好构成一个哈密尔顿通路,只是这种概率比较小;(2)蚂蚁的子路径没有覆盖所有节点,则子路径不能构造可行解;(3)子路径覆盖的节点有重复,则子路径也不能构造可行解;而在求解扩展 K-TSP 问题时,除以上原因外,还有蚂蚁从不同的城市出发,最后回到原出发点的问题,不像求解 TSP 问题那样,所有蚂蚁有相同的出发点。

ACA不能直接求解扩展 K-TSP,因而也不能直接求解 MDLP。为此,本文设计了改进的蚁群算法,该算法不仅可以直接求解 MDLP 问题,而且具有较好的求解性能及收敛速度。具体措施为:(1)改进了禁忌表的设置方式,所有蚂蚁共享一个禁忌表,而不是给每只蚂蚁单独设立禁忌表,确保蚂蚁的子路径合成一个哈密尔顿通路,从而形成问题的可行解;(2)修改了概率选择规则,在概率选择规则中加入代价引导函数。

为提高求解性能,算法使用 2-opt 策略优化蚂蚁子路径并设计了信息素更新规则。

3.2 改进蚁群算法(IACA)

3.2.1 蚁群共享禁忌表

蚂蚁从拟选的 k 个配送中心出发,为所有蚂蚁建立一个空白共享禁忌表并在其中放入 k 个配送中心。发生转移时,每只蚂蚁依据转移策略选择共享禁忌表里未曾记录的配送点后,随之将该配送点放入禁忌表,其他蚂蚁就不能再选择该配送点。在满足服务能力约束的情况下,蚂蚁继续选择下一个配送点,否则返回其出发的配送中心。当所有蚂蚁都返回各自的出发点后,配送结束,使用 2-opt 优化配送顺序并更新信息素,计算解的目标值。多次运行算法,直到目标值趋于稳定,对一个选择方案的评估结束。比

较多个选择方案的目标值,目标值小者即为 MDLP 的解。

3.2.2 蚂蚁选择策略

为提高算法的求解性能,在蚂蚁的选择策略中加入了代价引导函数,修改后的概率选择规则如下:

$$p_{ij}(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}(t)]^\beta \zeta_{ij}(t)}{\sum_{s \in t_{abu}} [\tau_{is}(t)]^\alpha \cdot [\eta_{is}(t)]^\beta}, & j \notin t_{abu} \\ 0, & \text{其他} \end{cases} \quad (12)$$

式中: $\tau_{ij}(t)$ 为 t 时刻节点 i, j 间路径上的信息素量; $\eta_{ij}(t) = 1/c_{ij}$ 为启发函数, c_{ij} 为从节点 i 到节点 j 的配送代价; $\zeta_{ij}(t) = 1/c_j$ 为代价引导函数, c_j 为贪婪算法得出的从各配送中心到节点 j 的配送成本的平均值; α 为信息素启发因子; β 为期望启发因子。

3.2.3 信息素更新规则

因为各子路径均可构造可行解,故对 2-opt 优化后的子路径的每条边均增加信息素量

$$\Delta\tau_{ij}(t) = \Delta\tau_{ij}(t) + 1/c_m \quad (13)$$

式中: $c_m(m=1,2,\dots,k)$ 为各条子路径的配送代价。

比较各可行解的目标函数值 F ,设当前最好解的目标值为 F_b ,如果 $(F-F_b)/F_b < \epsilon$ 成立(ϵ 是一个较小的正常量),则给最好解的各边增加信息素量

$$\Delta\tau_{ij}(t) = \Delta\tau_{ij}(t) + 1/F_b \quad (14)$$

同时,按式(11)更新信息素。为防止信息素过分悬殊导致算法陷入局部最优,依照最大最小蚂蚁策略修改各边的信息素^[9],即令 $\tau_{\min} \leq \tau_{ij}(t) \leq \tau_{\max}$,若 $\tau_{ij}(t) < \tau_{\min}$,则 $\tau_{ij}(t) = \tau_{\min}$;若 $\tau_{ij}(t) > \tau_{\max}$,则 $\tau_{ij}(t) = \tau_{\max}$ 。

3.2.4 MDLP-IACA 执行过程

- (1)初始化各关键参数;
- (2)清空蚂蚁的共享禁忌表 t_{abu} ;
- (3)将蚂蚁置于拟选各配送中心位置上,并将这些配送中心加入共享禁忌表;
- (4)按式(12)执行蚂蚁转移策略,直到全部蚂蚁形成自己的子路径;
- (5)对各子路径使用 2-opt 反复优化,并按式(13)更新可行解各边的信息素;
- (6)按式(14)更新当前最好解的信息素;
- (7)按式(11)更新信息素,并置所有边上的信息素增量为零;
- (8)若不满足结束条件,转入(2),否则,结束。

改进蚁群算法中,参数 α, β, ρ 对算法的求解性能及收敛情况有很大的影响。其中, α 值的大小反映了配送点到配送中心的信息量受重视的程度,可理解为信息量的指数加权。 $\alpha=0$,算法性能接近于贪婪算法, α 值越大,蚂蚁选择以前经过的路线的可能性

越大,但过大会使算法陷于局部最小解; β 的大小表明启发式信息受重视的程度, β 值越大,蚂蚁选择离它近的城市的可能性也越大,但过大也会使算法陷入局部最小解; ρ 反映信息素的挥发程度, ρ 过小会使算法过早收敛(信息素积累快),过大则导致前面

蚂蚁好的搜索结果不能被后来蚂蚁充分利用. 本文通过大量的仿真实验,得出 α, β, ρ 对算法结果的影响曲线,分别如图 2~图 4 所示. 从曲线图看出 $\alpha=1.1, \beta=0.9, \rho=0.15$ 是较好的配置组合.

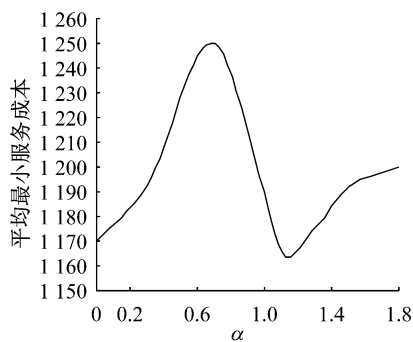


图 2 参数 α 对解值的影响曲线

Fig.2 Curve of parameter α effect on solution value

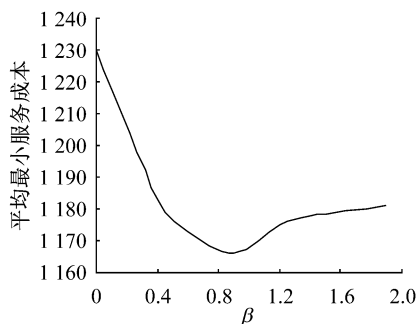


图 3 参数 β 对解值的影响曲线

Fig.3 Curve of parameter β effect on solution value

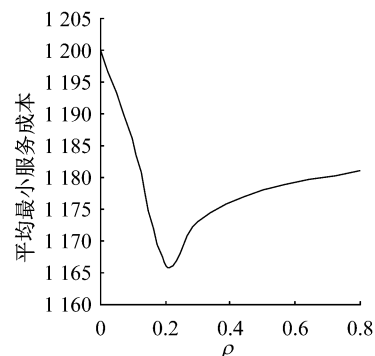


图 4 参数 ρ 对解值的影响曲线

Fig.4 Curve of parameter ρ effect on solution value

4 算例与分析

一物流配送矩形区域,范围为(0,0)到(100,100),其中散布着27个配送点,各配送点的坐标和

需求量见表 1;有 9 个候选配送中心,各中心的坐标、固定费用见表 2;设配送中心到各配送点的运费率均为 1,选取 5 个配送中心,使得各中心的配送总成本及固定费用成本之和最小.

表 1 配送点基本数据

Tab.1 Basic data for distribution points

i	1	2	3	4	5	6	7	8	9
(x_i, y_i)	(39,68)	(40,15)	(70,60)	(42,64)	(87,45)	(39,7)	(28,71)	(55,20)	(19,5)
q_i	1.1	0.6	0.5	0.7	1.4	1.0	2.0	3.0	1.0
i	10	11	12	13	14	15	16	17	18
(x_i, y_i)	(13,47)	(34,78)	(5,16)	(68,45)	(15,16)	(76,35)	(8,5)	(33,62)	(7,71)
q_i	0.8	2.1	0.7	0.6	1.3	0.3	4.0	1.7	1.0
i	19	20	21	22	23	24	25	26	27
(x_i, y_i)	(5,50)	(10,40)	(13,4)	(41,5)	(85,29)	(80,63)	(50,25)	(78,47)	(30,77)
q_i	2.3	1.2	0.4	1.0	1.1	0.9	0.3	0.4	2.5

表 2 配送中心基本数据

Tab.2 Basic data for distribution centers

i	1	2	3	4	5	6	7	8	9
(x_i, y_i)	(7,20)	(80,50)	(23,74)	(45,16)	(34,67)	(4,47)	(56,37)	(10,10)	(37,97)
f_i	14	15	12	14	14	11	13	16	12

利用 Visual C++ 6.0 编程. 参数设置为 $\alpha=1.1, \beta=0.9, \rho=0.15, \epsilon=0.05$, 最大迭代次数 $N_{c, \max}=1000, \tau_{\min}$ 为 0.1, $\tau_{\max}=10$, 蚂蚁数=50, 算法执行 50 次, 算法所得解的情况见表 3.

表 3 中 GA 算法(遗传算法)得到的数据是用不同初始值计算 50 次得到的最优结果. 从表 3 看出, 两种算法得出的配送服务点和配送规模一致, 但选择的配送中心及配送中心服务总成本不一样, 改进

蚁群算法得到的解值明显优于遗传算法. 证明改进蚁群算法在求解多配送中心选址问题是可行的.

5 结论

本文将多物流配送中心选址问题映射成扩展 K-TSP 过程, 通过设置共享禁忌表、改进转移规则、
(下转第 799 页)