

文章编号:1003 - 207(2003)05 - 0056 - 04

横向并购初期固位策略销售服务网络结构优化

沈凤武,徐寅峰

(西安交通大学管理学院管理科学系,西安 710049)

摘要:本文着眼于当今世界企业经营领域的前沿,对企业横向购并初期的销售服务网络的集成优化进行了深入研究,本文基于固定位置策略建立了优化模型,给出了模型的求解算法与计算复杂性。同时,本文以合并后 5 个销售服务中心 30 个客户为背景,给出了算法的仿真计算结果。

关键词:Voronoi 图;算法;计算复杂性;并购

中图分类号:F272;O221 **文献标识码:**A

1 引言

在两个同行业公司发生兼并、或收购活动后(即横向并购)的初期,对于面对相同的客户群体或运作服务区域的两个公司的销售服务部门很可能会出现服务对象交叠而导致销售服务中心发生冗余的问题,而且分支过密是导致冗员问题的一个非常重要的原因。针对这一问题,传统的措施一般是将被并公司作为拆剪的对象(这是一种普遍的做法,除非目标公司良好的销售服务机构是主购公司施行并购的主要动因)。这往往是从公司人事管理以及企业文化与经营战略的有效性的角度来考虑的。我们知道对于公司的并购活动而言,这是一个系统的集成过程,那么仅仅从上述几个角度去分析实施并购后的系统集成是不够的。而且在考虑上述问题的时候应该是建立在公司合并后的整体系统结构得到优化的基础上,对于一个不是最优化的系统结构来说,围绕系统的所有工作都要参与分担一部分用来克服因系统缺欠所产生的成本。可见系统的优化在系统集成的过程中是一个非常重要的基础工作。以往类似的研究大都是在给定需求点集合 $D = \{d_1, d_2, \dots, d_n\}$ 的情形下,求解供应点的集合 $S = \{s_1, s_2, \dots, s_p\}$, 并使得 $r = \max_i \left\{ \min_j \left\{ d(d_i, s_j^*) \right\} \right\}$ 最小, Drezner (1984) 给出了计算复杂性为 $O(n^{2P+1} \log n)$ 的近

似算法^[3]; R. Z. Hwang, R. C. T. Lee 和 R. C. Chang 提出的 Slab 划分法来解决 EPC 问题需花费 $O(n^{O(\sqrt{P})})$ 时间^[4]。另外,文献[5]的关于 k -partition 的研究与销售服务网络的结构优化有着较强的相关性,徐寅峰教授证明了 $k=3$ 时, k -partition 问题提 NP-c 问题,并给出了具体的算法^[6]。比较而言本文的优化目标变化了,本文研究的背景是需求点集合 $D = \{d_1, d_2, \dots, d_n\}$ 与供应点集合 $S = \{s_1, s_2, \dots, s_p\}$ 均已给定,我们的目标是求解满足条件的 $\min |S|$ 。

2 模型的建立

现以两个公司的购并为例来介绍一下这类问题的优化过程。公司 1 的销售服务中心集合用 $S_1 = \{a_1, a_2, \dots, a_n, i = 1, \dots, n\}$ 来表示,公司 2 的销售服务中心的集合用 $S_2 = \{b_1, b_2, \dots, b_m, j = 1, \dots, m\}$ 来表示,客户的集合用 $C = \{c_1, c_2, \dots, c_k, l = 1, \dots, k\}$ 表示。销售服务机构的集合 S_1 与 S_2 在市场中共同的客户集合 C ,当两个公司合并后,原来分别属于两个独立公司的销售服务网络中的销售服务点所对应的服务区域必然会发生交叉或重叠。我们研究的目的是对发生交叉、重叠区域所对应的销售服务网点进行优化评价,删除冗余的销售服务网点,使得两个公司合并后的销售服务分支机构数量最少且效率极大的高于合并前任一公司的效率。

固位策略的含义是指:我们只可以对原有的销售服务网点进行删减却不可以改变他们原来的位置。

模型 5 - 1 给定三个集合 S_1, S_2 与 $C, S_1 =$

收稿日期:2003 - 01 - 10;修订日期:2003 - 08 - 08

基金项目:国家自然科学基金会优秀创新研究群体基金资助项目(70121001)

作者简介:沈凤武(1968 -),男(汉族),辽宁人,西安交通大学管理学院管理科学与工程博士生,研究方向:工商管理。

$\{a_1, a_2, \dots, a_i, i = 1, \dots, n\}$, $S_2 = \{b_1, b_2, \dots, b_j, j = 1, \dots, m\}$, $C = \{c_1, c_2, \dots, c_l, l = 1, \dots, k\}$, 另有集合 $S = S_1 \cup S_2$ 。定义函数:

$$f_1(S_1, C) = \min_{a_i, c_{il}} d(a_i, c_{il})$$

$$f_2(S_2, C) = \min_{b_j, c_{jl}} d(b_j, c_{jl})$$

$$f_3(S, C) = \min_{s_h, c_{hl}} d(s_h, c_{hl})$$

其中, $d(a_i, c_{il})$, $d(b_j, c_{jl})$ 分别表示平面内两点间的欧氏距离, c_{il}, c_{jl}, c_{hl} 分别表示点 a_i, b_j, s_h 所在 Voronoi 区间内的点。求:

$$\left\{ \begin{array}{l} \min \{ f_1(S_1, C), f_2(S_2, C) \} \\ f_3(S, C) < \min \{ f_1(S_1, C), f_2(S_2, C) \} \\ \max_{s_h, c_{hl}} d(s_h, c_{hl}) < \max \left\{ \max_{a_i, c_{il}} d(a_i, c_{il}), \max_{b_j, c_{jl}} d(b_j, c_{jl}) \right\} \end{array} \right.$$

3 算法设计

3.1 求解思路:

运用最近点 Voronoi 图量化优化指标,通过脱机减量法构造最近点意义下的 Voronoi 图来评价各销售服务网点,将冗余点删除。

3.2 算法

第一步 分别作平面点集 S_1 和 S_2 的最近点意义下的 Voronoi 图

第二步 计算出每个点(销售服务中心)所对应的 Voronoi 区域 $d(a_i, c_{il})$ 和 $d(b_j, c_{jl})$ 。

第三步 求解 $f(S_1)$ 和 $f(S_2)$

第四步 计算 $\min\{f(S_1), f(S_2)\}$

第五步 作合并后的点集 $S = S_1 \cup S_2 = \{s_1, s_2, \dots, s_h, h = 1, \dots, (n+m)\}$ 的最近点意义下的 Voronoi 图。

第六步 减量法构造点集 $S = \{s_1, s_2, \dots, s_h, h = 1, \dots, (n+m)\}$ 的最近点意义下的 Voronoi 图。减量算法^[7]。

第七步 比较在减掉一个点后,构造出新的 Voronoi 图中, $\max_{s_h, c_{hl}} d(s_h, c_{hl})$ 是否小于 $\max\{\max_{a_i, c_{il}} d(a_i, c_{il}), \max_{b_j, c_{jl}} d(b_j, c_{jl})\}$, 如果小于则进行下一步,否则将改减掉点恢复保留,并返回到第六步删减其他的点(S/s_h)。

第八步 判断 $f(S, C)$ 是否大于 $\min\{f_1(S_1, C), f_2(S_2, C)\}$

if $f(S, C) > \min\{f_1(S_1, C), f_2(S_2, C)\}$

then 输出上一次的结果即 s_{h-1} 所对应的结果。否则,返回第五步。

计算结束。

4 算法复杂性分析

下面我们来分析一下上述算法的计算复杂性。

第一步的时间复杂性为 $O(n)^{[2]}$ 。

第二步的时间复杂性:计算 $d(a_i, c_{il})$ 的时间为 $O(l)$, 由于计算 $d(a_i, c_{il})$ 的过程与计算 $f(S_1)$ 和 $f(S_2)$ 均可在常数时间内完成,故第二步的整体计算时间复杂性为 $O(n)$ 。

第三步计算 $f(S_1)$ 和 $f(S_2)$ 需花费 $O(\log n)$ 时间完成,所第一至三步共需花费 $O(n \log n)$ 时间。

第四步的时间复杂性为常数可得。

第五步的时间复杂性为 $O(n)$ 。

第六步的时间复杂性:判断 p_i 是否为凸壳顶点或在凸壳内,这需要先求出 $CH(S)$ 然后在进行比较,其耗费时间为 $O(n \log n)$ 。如果点 p_{i-1}, p_i, p_{i+1} 是 $BCH(S)$ 上的连续的3个点,删去 p_i 之后,需耗费 $O(\log^2 n)$ 的时间来重新构造 $BCH(S - \{p_i\})$, 即判定 p_{i-1} 与 p_{i+1} 之间是否有新的凸壳顶点。删去点 p_i 及 p_i 关联的 Voronoi 多边形的边和顶点,修改相应的 Voronoi 多边形的边和顶点,耗费常数时间。因此这一过程的整体时间复杂性为 $O(n \log n)$ 。

第七步的时间复杂性:构造新的 Voronoi 图,需要耗费 $O(n)$ 时间。比较判断 $\max_{s_h, c_{hl}} d(s_h, c_{hl})$ 是否小于 $\max\{\max_{a_i, c_{il}} d(a_i, c_{il}), \max_{b_j, c_{jl}} d(b_j, c_{jl})\}$, 需耗费常数时间。在该步中含有一个循环,即第五步,故该步的整体时间复杂性为 $O(n^2 \log n)$ 。

第八步的时间复杂性:判断 $f(S, C)$ 是否大于 $\min\{f_1(S_1, C), f_2(S_2, C)\}$, 需耗费常数时间。如果 $f(S, C) > \min\{f_1(S_1, C), f_2(S_2, C)\}$, 该步共需耗费常数时间。

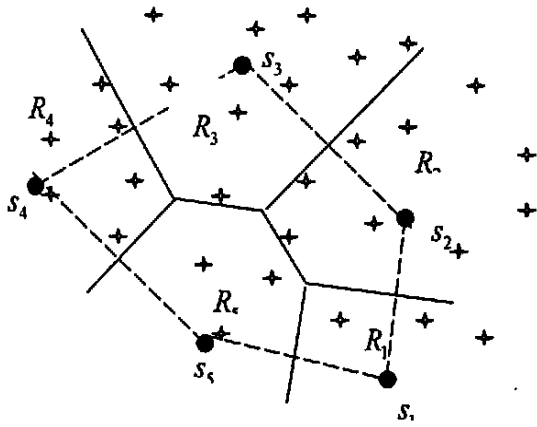
如果 $f(S, C) < \min\{f_1(S_1, C), f_2(S_2, C)\}$, 则需要第五步参与循环,那么,该步的整体耗费时间为 $O(n \log n)$ 。

故上述算法的时间复杂性为 $O(n^2 \log n)$ 。

5 算例

这里我们给出一个以合并后共有5个销售服务中心为例的固为策略的优化过程。

给定合并后的需求点集合 $D = \{d_1, d_2, \dots, d_n\}$ 和供应点的集合 $S = \{s_1, s_2, \dots, s_5\}$, 省略合并前的过程, 执行算法第五步可得到如图 4.1 所示的结果。



表示需求点(客户)
表示供应点(销售服务中心)

图 4.1 合并后销售服务网络

执行算法第六步, 判定 s_5 不满足约束条件其删除得到如图 4.2 所示的结果。

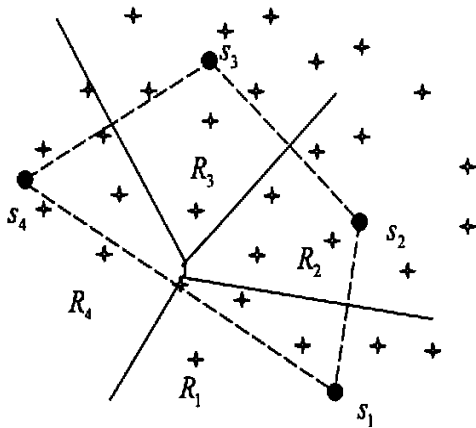


图 4.2

继续执行第七、八两步的结果需返回第六步, 判定结果为 s_1 不满足约束条件, 将该销售服务中心删除, 得到如图 4.3 的结果。

继续下一步计算直到第八步, 该结果满足条件, 输出。

这里给出的例子是一个所有销售服务中心都分布在凸壳上的情形, 下面给出一个凸壳内含有供应点(销售服务中心)的情况, 如图 4.4 所示。

此时, 若 s_5 为不满足约束条件的供应点, 构造新的 Voronoi 图, 只需删除 s_5 点及其关联的 Voronoi 多边形的边和顶点, 并修改其它各点关联的 Voronoi

多边形的边和顶点, 形成图 4.4 的虚线部分。图中可以看出其它各点关联的 Voronoi 多边形与 s_5 关联的 Voronoi 多边形有共同的边, 虚线即为删除 s_5 点后构成的新的 Voronoi 多边形。其他步骤转变为凸壳内不含点的计算过程。

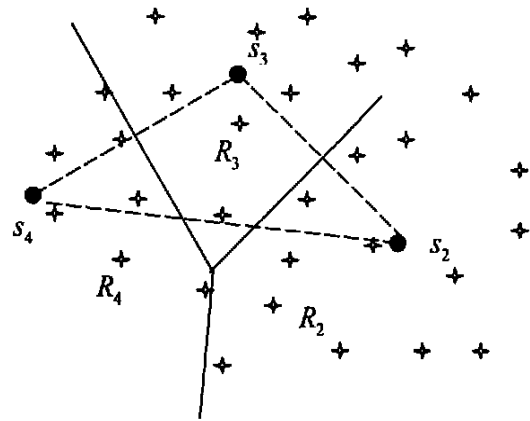


图 4.3

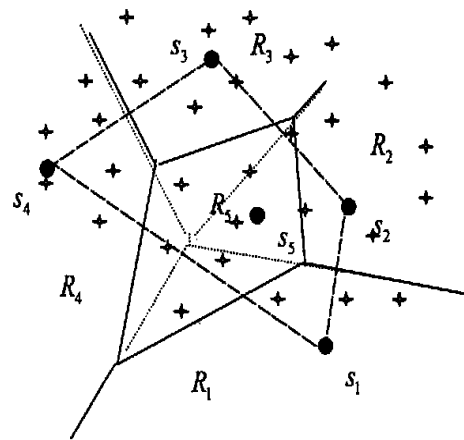


图 4.4 内含供应点的 Voronoi 图

6 结论

本文针对企业横向并购中的实际问题, 建立了网络优化模型, 并设计给出了计算复杂性为 $O(n^2 \log n)$ 的算法。由于本文仅考虑了销售服务网点只可删减不能移动的情况所以说本文的结果的实用性还有很大的限制, 在企业实际的兼并、收购、联盟等经营活动中有很多情况与本文所考虑的问题不同。在我们后续的研究中, 基于可变策略(销售服务中心位置可变迁的策略)的销售服务网络优化已经取得一些结果。

参考文献:

- [1] G. J. Werden. Horizontal mergers: Comments[J], Amer. Econ. Rev, 1991, 81:1002 - 1006.
- [2] Nimrod Megiddo. Linear - time algorithms for linear programming in R3 and related problems. Franz Aurenhammer. Voronoi diagrams - a survey of a fundamental geometric data structure[J]. ACM Computing Surveys, 1991, 23: 345 - 406.
- [3] Drezner, Z., The P - center problem - Heuristics and optimal algorithms [J]. Oper. Res. Soc., 1984, 35 (8) : 741 - 745.
- [4] R. Z. Hwang, R. C. T. Lee, and R. C. Chang., The Slab Dividing Approach To Solve the Euclidean P - Center Problem[J]. Algorithmica 1993, 9:1 - 22.
- [5] Hwang, F. K., Optimal partitions[J]. Optimization Theory and Applications 1981, 34:1 - 10.
- [6] 徐寅峰, 刘自成. 关于最大权 k 子集分拆问题[J]. 高效应用数学研究, 1994, 9(4).
- [7] 周培德. 计算几何——算法分析与设计[M]. 清华大学出版社.
- [8] Franz Aurenhammer Voronoi diagrams - a survey of a fundamental geometric data structure [J]. Computing Surveys 1991, 23:345 - 406.
- [9] S. Voronoi diagrams and Delaunay triangulations [M]. In Du D Z(ed). Computing in Euclidean geometry, Word Scientific Publishing Co. Pte. Ltd, 1995.
- [10] Gold, C. M. Remmele P. R. Roos, T. Fully Dynamic and Kinematic Voronoi Diagrams in GIS [J]. Algorithmica 1998.
- [11] Kenny Wong An Efficient Implementation of Fortune's Plane - Sweep Algorithm for Voronoi Diagrams [Z]. <http://www.cs.ualberta.ca/~kenw/publications.html>.
- [12] F. Aurenhammer and R. Klein. Voronoi diagrams [M]. In J. Sack and G. Urrutia, editors. Handbook of Computational Geometry, pages 201 - 290, Elsevier Science Publishing, 2000.
- [13] 谢金星, 邢文训. 网络优化 [M]. 清华大学出版社, 2000.
- [14] 谢金星, 邢文训. 现代优化计算方法 [M]. 清华大学出版社, 2001.

Sales Service Net Structure Optimization of Fixed Strategy for Horizontal Mergers

SHEN Fengwu, XU Yin-feng

(Dept. of Management Science, Management School, Xi'an Jiaotong University, Xi'an 710049, China)

Abstract: From the fore - field of enterprises operation, an optimized model was established under fixed strategy about horizontal mergers, further the algorithm for the model and its complexity were given about integration and optimization of enterprises service net. In this paper, the corresponding simulation is given based on 5 supply points and 30 demand points.

Key words: Voronoi graph; algorithm; computing complexity; merger