文章编号: 1001-0920 (2010) 08-1195-06

三级物流网络选址-路径问题建模与求解算法研究

金 莉[1,2], 朱云龙[1], 申 海[1,2]

(1. 中国科学院 沈阳自动化研究所，沈阳 110016；2. 中国科学院 研究生院，北京 100039)

摘 要: 鉴于固定费用选址问题的一个重要局限是在运输成本的计算中, 假设采用整车运输方式, 其运输成本与考虑运输路径时的成本不同会影响选址决策. 针对一个钢材销售企业的三级物流网络中的两级设施进行选址, 采用多站式运输方式计算运输成本, 问题为三级物流网络选址-路径问题. 采用嵌入拉格朗日启发式算法的分枝定界方法来求解, 并对该方法进行了实验测试, 测试结果表明该方法是有效的.

关键词: 三级物流网络；选址路径问题；拉格朗日启发式算法；分枝定界法

中图分类号: TP18; C934 **文献标识码:** A

Research on modeling and algorithm for three-layer distribution network location-routing problem

JIN Li[1,2], *ZHU Yun-long*[1], *SHEN Hai*[1,2]

(1. Shenyang Institute of Automation，Chinese Academy of Sciences，Shenyang 110016，China；2. School of Graduate，Chinese Academy of Sciences，Beijing 100039，China．Correspondent：JIN Li，E-mail：jinli@sia.cn)

Abstract: An important limitation of the fixed charge location model is the assumption that full truckload quantities are shipped from a distribution center to a customer. The different location decision is made from the different cost of delivery. A particular three-layer steel product distribution network is investigated, in which two echelons of facilities are located. To make the location decision, the delivery cost of a multiple-stop route is calculated. A Lagrangian relaxation-based branch and bound algorithm is used to solve this location-routing problem. The numerical results for various size test problems are presented, and the results show the effectiveness of the method.

Key words: Three-layer distribution network；Location-routing problem；Lagrangian heuristic；Branch and bound

1 引 言

选址-路径问题 (LRP) 是设施选址问题和车辆路径问题的集成, 两种问题之间存在相互依赖的关系[1]. 在进行设施选址决策时, 应考虑运输成本. 目前, 关于选址问题的研究大多假设从供货点到客户进行整车运输, 供货后直接返回供应点. 但在企业的实际运营中,尤其是零售业, 由供货点出发的运输车辆要为多个客户送货后再返回出发点, 即运输是沿着多站路径的零担运输. 整车运输的运输成本独立于其他运输, 而零担运输的运输成本依赖运输路径上的其他客户和客户访问顺序. 研究者指出, 采用不同的运输路径作出的选址决策是不同的[2]. 目前, 已有较多的文献研究了选址与路径集成问题.

LRP 问题按照设施的级数进行分类, 典型的有从工厂到分销中心再到客户的三级物流问题和只关注由分销中心到客户的两级物流问题. 按照求解方法分类, 可分为精确性方法和启发式方法. 由于选址问题和车辆路径问题是 NP 难问题, 集成模型更加复杂. LRP 问题的精确求解算法十分有限, 较早的算法是单供应点问题, 采用松弛技术求解整数线性规划模型, 利用分枝定界法保证解的整数性[3]. Laporte[4,5]采用相似方法求解零售点到客户的两级 LRP 问题. 拉格朗日松弛方法是提供下界最成功的方法之一. Beasly[6]实现了拉格朗日松弛在选址问题中的成功应用. 李志华等[7]采用增强型分枝定界方法求解带固定费用有能力约束的网络设计问题. 但更多文献采用了启发式方法求解LRP问题. Perl 等[8]研究了具有产品供应点、分销中心和客户的三级选址路径问题,

收稿日期: 2009-06-29；**修回日期:** 2009-10-09.

基金项目: 国家科技计划项目(2007AA04Z189)；国家科技支撑计划项目(2006BAH02A07).

作者简介: 金莉(1976−), 女, 沈阳人, 博士生, 从事物流控制与优化、智能算法的研究；朱云龙(1967−), 男, 江苏南通人, 研究员, 博士生导师, 从事先进制造、信息集成等研究.

供应点固定, 供应点到分销中心的运输采用整车运输, 分销中心采用访问多个客户节点的多站式运输方式. 模型中同时求解仓库到分销中心的产品运输数量, 因此为混合整数线性规划问题. Perl用三阶段启发式算法解决问题[8]. Wu等[9]提出一个简单二阶段启发式算法来解决该问题, 并在具有150个节点的问题中进行了测试.

三级物流网络规划中, 仓库的选址与零售点的选址同为战略级决策, 应该考虑它们之间的关联关系. 基于此, 本文构建了一个三级物流网络LRP模型, 同时进行仓库及零售点两级设施的选址. 仓库及零售点之间采用整车运输, 零售点到客户的运输采用多站式运输. 考虑到问题的复杂性, 提出一个三阶段确定性求解算法, 采用分枝定界方法分阶段求解, 并利用拉格朗日松弛提供的下界加快求解速度, 极大地降低了分枝定界树的规模, 从而使传统的分枝定界算法能够应用于求解大型集成网络规划问题.

2 问题描述与模型

某钢材加工销售企业规划具有仓储加工中心、零售店的配送网络, 根据对客户需求量的预测, 需要从备选仓库地点中选建仓库和从备选零售点中选建零售店, 并从备选车辆中确定车辆的配送方案, 使得整个网络的成本最小. 模型的建立基于如下假设: 1) 仓储加工中心的商品满足所有客户的需要; 2) 商品为单种类商品; 3) 有多个备选仓库和零售点; 4) 运输工具为多种车型, 车辆足够多, 可完全满足客户服务需要, 每辆车有能力约束, 完成全部运输任务后返回出发点; 5) 客户和零售点均为单源供应; 6) 零售点具有能力约束.

三级物流网络集成规划模型(模型SP)构建如下:

$$\min \quad \sum_{s\in S} g_sT_s+\sum_{s\in S}\Big(\sum_{j\in J}c_{sj}X_{sj}\Big)+\sum_{j\in J}f_jR_j+\sum_{k\in K}\tau_k\Big(\sum_{i\in P}\sum_{j\in P}d_{ij}Z_{ijk}\Big). \tag{1}$$

$$\text{s.t.} \quad \sum_{s\in S}X_{sj}=R_j,\ \forall j\in J; \tag{2}$$

$$\sum_{k\in K}\sum_{j\in P}Z_{ijk}=1,\ \forall i\in I; \tag{3}$$

$$\sum_{i\in I}h_iY_{ij}\leqslant n_j,\ \forall j\in J; \tag{4}$$

$$\sum_{i\in I}h_i\Big(\sum_{j\in P}Z_{ijk}\Big)\leqslant\sigma_k,\ \forall k\in K; \tag{5}$$

$$\sum_{j\in P}Z_{ijk}-\sum_{j\in P}Z_{jik}=0,\ \forall i\in P,\ \forall k\in K; \tag{6}$$

$$\sum_{j\in J}\sum_{i\in I}Z_{ijk}\leqslant 1,\ \forall k\in K; \tag{7}$$

$$\sum_{u\in P}Z_{iuk}+\sum_{v\in P}Z_{jvk}-Y_{ij}\leqslant 1,\ \forall j\in J,\ \forall i\in I,\ \forall k\in K; \tag{8}$$

$$\sum_{j\in J}Y_{ij}=1,\ \forall i\in I; \tag{9}$$

$$X_{sj}\leqslant T_s,\ \forall s\in S,\ \forall j\in J; \tag{10}$$

$$Y_{ij}\leqslant R_j,\ \forall j\in J,\ \forall i\in I; \tag{11}$$

$$R_j,T_s,X_{sj},Y_{ij},Z_{ijk}\in\{0,1\}. \tag{12}$$

其中: $S=\{s|s=1,2,\cdots,m\}$ 为备选仓库地点集合; $J=\{j|j=1,2,\cdots,n\}$ 为备选零售点集合; $I=\{i|i=1,2,\cdots,r\}$ 为客户集合; $K=\{k\mid k=1,2,\cdots,t\}$ 为备选车辆集合; 集合 $P=I\bigcup J=\{p\mid p=1,2,\cdots,m+n\}$; g_s 为备选仓库 s 的建设成本; f_j 为备选零售点 j 的建设成本; h_i 为客户 i 的需求量; d_{ij} 为节点 $i\in P$ 与 $j\in P$ 之间的距离; n_j 为备选零售点 j 的最大吞吐量; c_{sj} 为备选仓库 s 到备选零售点 j 的运输成本; τ_k 为车辆 k 的单位距离运输成本; σ_k 为车辆k 的运输能力. 模型中采用的决策变量均为0-1变量: T_s 表示在 s 点建立仓库; R_j 表示在 j 点建立零售店; X_{sj} 表示由第 s 个仓库为第 j 个零售点供货; Y_{ij} 表示由第 j 个零售点为第 i 个顾客送货; Z_{ijk} 表示车辆 k 由点 $i\in P$ 直接到达点 $j\in P$.

目标函数(1)包含了仓库建设成本、零售店建设成本, 以及从仓库到零售店和从零售店到客户的运输成本, 在满足约束条件(2)∼(12)下使之最小; 约束(2)为单源约束; 约束(3)要求一个客户只能被访问1次; 约束(4)要求每个零售店满足能力约束; 约束(5)要求每辆车满足能力约束; 约束(6)要求到达任何节点的车辆必须离开该节点; 约束(7)要求一辆车的运输路径最多经过一个零售店; 约束(8)说明若一条运输路径经过客户节点 i 及零售店 j, 则客户 i 由零售店 j 供应; 约束(9)为单源约束; 约束(10)和(11)要求零售店及客户只能由建立的设施供应; 约束(12)是0-1约束.

3 拉格朗日启发式算法

拉格朗日松弛是指将一组约束从原问题移走, 以拉格朗日乘子为权重吸收到目标函数中, 目的是利用剩余约束的特殊结构, 获得一个比原问题更易于求解的拉格朗日对偶问题. 采用次梯度优化方法求解拉格朗日对偶问题来获得最好下界, 同时, 可以利用在拉格朗日松弛中获得的信息构造一个原问题的可行解来获得原问题最优解的上界.

3.1 拉格朗日松弛

对原问题增加以下2个约束来改善松弛:

$$\sum_{s\in S}T_s\geqslant 1, \tag{13}$$

$$\sum_{j\in J} n_j R_j \geqslant \sum_{i\in I} h_i. \tag{14}$$

其中: 约束 (13) 保证至少建立一个仓库, 约束 (14) 保证建立的零售店的总能力足够满足所有客户的需求. 加入以上约束后, 依据 Tragantalerngsak[10]对几种拉格朗日松弛效果的研究, 用拉格朗日乘子 λ_{sj} 和 γ_{ij} 分别松弛约束集 (10) 和 (11), 得到拉格朗日对偶问题为

$$\begin{aligned}\min &\sum_{s\in S}\Big(g_s-\sum_{j\in J}\lambda_{sj}\Big)T_s+\sum_{j\in J}f_jR_j-\\ &\sum_{i\in I}\sum_{j\in J}\gamma_{ij}R_j+\sum_{s\in S}\sum_{j\in J}\Big(c_{sj}+\lambda_{sj}\Big)X_{sj}+\\ &\sum_{k\in K}\tau_k\Big(\sum_{i\in P}\sum_{j\in P}d_{ij}Z_{ijk}\Big)+\sum_{i\in I}\sum_{j\in J}\gamma_{ij}Y_{ij};\\ \text{s.t. }&式(2)\sim(9),\ (12)\sim(14).\end{aligned} \tag{15}$$

问题 (15) 可以分解为如下 3 个子问题来求解:

$$\begin{aligned}\min &\sum_{s\in S}\Big(g_s-\sum_{j\in J}\lambda_{sj}\Big)T_s;\\ \text{s.t. }&式(12),\ (13).\end{aligned} \tag{16}$$

$$\begin{aligned}\min &\sum_{k\in K}\tau_k\Big(\sum_{i\in P}\sum_{j\in P}d_{ij}Z_{ijk}\Big)+\sum_{i\in I}\sum_{j\in J}\gamma_{ij}Y_{ij};\\ \text{s.t. }&式(3)\sim(9),\ (12).\end{aligned} \tag{17}$$

$$\begin{aligned}\min &\sum_{j\in J}f_jR_j-\sum_{i\in I}\sum_{j\in J}\gamma_{ij}R_j+\\ &\sum_{s\in S}\sum_{j\in J}(c_{sj}+\lambda_{sj})X_{sj};\\ \text{s.t. }&式(2),\ (12),\ (14).\end{aligned} \tag{18}$$

问题 (16) 采用背包问题的解决方法求解[11]. 问题 (17) 是个多车场车辆路径问题, 可以采用启发式方法分步求解: 首先根据客户与零售点的距离分配客户[12]; 然后为每个零售点 j' 的客户确定车辆运输路径, 采用节约/插入算法[13]. 假设零售店 j 建立, 即 $R_j = 1$, 问题 (18) 最优的目标函数值中包括如下定义的值 v_j:

$$\begin{aligned}&v_j=\min\Big\{f_j-\sum_{i\in I}\gamma_{ij}+\sum_{s\in S}(c_{sj}+\lambda_{sj})X_{sj}\Big\};\\ &\text{s.t. }\sum_{s\in S}X_{sj}=1,\ X_{sj}\in\{0,1\},\ \forall s\in S,\ \forall j\in J.\end{aligned} \tag{19}$$

则问题(18)可通过以下启发式方法求解: 令

$$v_j=\min\Big\{f_j-\sum_{i\in I}\gamma_{ij}+s_j\Big\},$$

$$s_j=\min_s\{c_{sj}+\lambda_{sj},s\in S\},\ \forall j\in J.$$

问题 (18) 可以表示为

$$\min\sum_{j\in J}v_jR_j;\ \text{s.t. 式}(12),\ (14). \tag{20}$$

问题 (20) 类似背包问题, 可用贪婪算法原理求解. 记 $v(X)$ 为问题 X 的目标值, 则模型 SP 目标值的下界可由$v(式(16))+v(式(17))+v(式(18))$获得.

3.2 次梯度优化

次梯度方法通过在次梯度方向执行搜索, 以获得原问题的最佳下界. 令 t 为迭代次数, d^t 和 ξ^t 分别为在 t 步迭代时的步长和次梯度参数. 记 $\left(\lambda_{sj}^t,\gamma_{ij}^t\right)$ 为第 t 步迭代时的拉格朗日乘子值, 则下一次迭代时的乘子值为

$$\left(\lambda_{sj}^{t+1},\gamma_{ij}^{t+1}\right)=\left(\lambda_{sj}^t,\gamma_{ij}^t\right)+d^t\xi^t.$$

令 λ_{sj}^{t+1} 和 γ_{ij}^{t+1} 取正值, 有

$$\lambda_{sj}^{t+1}=\max\left\{\lambda_{sj}^{t+1},0\right\},\ \gamma_{ij}^{t+1}=\max\left\{\gamma_{ij}^{t+1},0\right\}.$$

次梯度计算如下:

$$\xi^{\mathrm{T}}=((X_{sj}^t-T_s^t)^{\mathrm{T}},(Y_{ij}^t-R_j^t)^{\mathrm{T}}),$$

$$\forall i\in I,\ \forall j\in J,\ \forall s\in S.$$

步长可以通过多种方法定义, 实际应用中常用

$$d^t=\beta^t(\eta\bar{v}-v\left(LR_{\lambda^t,\gamma^t}\right))/\left\|\xi^t\right\|^2.$$

其中: $\bar{v}$ 是最优目标值的上界; 一般取 $\beta^t = 2$, $\eta = 1.05$[6]. 在经过一定次数的连续迭代后, 如果下界没有改进, 则 β^t 的值减半.

次梯度优化的终止准则为: 1) $\|\xi^t\| < \varepsilon$, ε 取为 0.01; 2) 当 $t = t_{\max}$时停止, $t_{\max}$为最大迭代次数.

3.3 生成可行解

确定选建的零售点集合 J_1 及未被选建的零售点集合 J_0 分别为

$$J_1=\{j\mid R_j=1,X_{sj}=1,\ T_s=1\},$$

$$J_0=J/J_1.$$

将决策变量的值按照这些集合调整, 检查是否存在将客户分配给集合 J_0 中的零售点, 若存在, 则将客户重新分配, 详细过程如下:

Step 1: 对于零售点 $\tilde{j}\in J_0$, 置 $R_{\tilde{j}}=0$, $X_{s\tilde{j}}=0$, $\forall s\in S$. 若有 $Y_{i\tilde{j}}=1, \forall i\in I$, 则将零售点 $\tilde{j}$ 的客户重新分配, 并且仍由原车辆为客户服务, 转至 Step 2; 若不存在这样的零售点, 则转至 Step 6.

Step 2: 对于车辆 $k\in K_{\tilde{j}}$, $K_{\tilde{j}}=\left\{k\left|Z_{i\tilde{j}k}=1\right.\right\}$, 找到集合 J_1 的子集 J_{1_k}, J_{1_k} 中每个成员的剩余能力均满足车辆 k 所服务客户的总需求. 若 $J_{1_k}=\varnothing$, 则转至 Step 5.

Step 3: 从车辆 k 的路径中移出零售点 $\tilde{j}$, 用插入/节约算法分别插入零售点 $j^*\in J_{1_k}$. 选择将客户分配给使运输成本增加得最小的零售点, 更新相应的 Y 和 z 值. 置$K_{\tilde{j}}=K_{\tilde{j}}/\{k\}$.

Step 4: 若 $K_{\tilde{j}}$=∅, 则转至 Step1; 否则, 转至 Step 2.

Step 5: 建立零售点 $\tilde{j}$, 相应地将 R_j, X_{sj}, T_s置 1. $J_1=J_1\bigcup\{\tilde{j}\}$, $J_0=J_0/\{\tilde{j}\}$, 转至 Step 1.

Step 6: 关闭未被分配客户的零售点.

Step 7: 关闭未被分配零售点的仓库.

4 分枝定界过程

4.1 剪切策略

当某一分枝的下界大于当前的最好可行解时, 这一分枝不会再获得更好的解, 可以剪切. 当上、下界的相对差值(上界与下界之差除以下界)小于一个很小值时, 可以认为获得了当前节点的最优解, 该分枝也可以剪切, 不再进一步考虑.

4.2 分枝定界算法

模型SP有5组决策变量, 其中设施的位置和数量的不同决策会对运作成本有较大影响. 因此, 算法按照决策变量分阶段进行分枝过程[14]. 搜索策略选择1-分枝及LIFO标准组合.

第1阶段只对变量T_s采用分枝策略构造分枝定界树. 树中所有T_s变量均已固定的节点称为终端节点, 到达终端节点时, 如果分枝没有全部被剪切, 则第2阶段继续从这些未被剪切的终端节点开始, 根据变量R_j和X_{sj}的关联关系, 对这两个变量采用分枝策略. 第3阶段从第2阶段未被剪切的终端节点开始, 这时问题转变为分派和车辆路径问题. 对变量Y_{ij}采用分枝策略后, 采用启发式算法求解.

4.2.1 第1阶段

分枝定界树的根节点定义为模型SP. 在根节点求解拉格朗日对偶问题, 得到模型SP的下界. 采用第3.3节的方法生成初始可行解, 计算得到的目标值v^*作为当前上界. 检查对偶问题的解是否是最优解, 如果不是, 则开始执行分枝过程.

在对变量T_s进行分枝的过程中, 分枝策略选择最容易确定取值的T_s变量进行分枝. g_s越小, T_s越容易置1, 先对具有最小g_s值的变量进行分枝. 在分枝过程中, 求解拉格朗日对偶问题, 计算可行解, 进行剪切测试并更新最优目标值v^*. 当回溯到根节点时, 即完成了第1阶段.

4.2.2 第2阶段

对应第1阶段未被剪切的每个终端节点, 令S'为该节点确定建立的仓库集合, 则该节点确定的问题为

$$\min\sum_{s\in S'}\Big\{\sum_{j\in J}c_{sj}X_{sj}\Big\}+\sum_{j\in J}f_jR_j+\sum_{k\in K}\tau_k\Big\{\sum_{i\in P}\sum_{j\in P}d_{ij}Z_{ijk}\Big\};$$

$$\text{s.t.}\ \sum_{s\in S'}X_{sj}=R_j,\ \forall j\in J,\ \text{式}(3)\sim(12). \tag{21}$$

利用拉格朗日乘子γ_{ij}松弛约束集(11), 同时用次梯度优化确定问题(21)的下界, 得到的松弛问题可分解为以下两个子问题:

$$\min\sum_{j\in J}f_jR_j-\sum_{i\in I}\sum_{j\in J}\gamma_{ij}R_j+\sum_{s\in S'}\sum_{j\in J}c_{sj}X_{sj};$$

$$\text{s.t.}\ \sum_{s\in S'}X_{sj}=R_j,\ \forall j\in J,\ \text{式}(12),\ (14). \tag{22}$$

$$\min\sum_{k\in K}\tau_k\Big(\sum_{i\in P}\sum_{j\in P}d_{ij}Z_{ijk}\Big)+\sum_{i\in I}\sum_{j\in J}\gamma_{ij}Y_{ij};$$

$$\text{s.t.}\ \text{式}(3)\sim(9),\ (12). \tag{23}$$

问题(22)和(23)的求解方法与(17)和(18)的解法相似.

选择第1阶段未被剪切的终端节点中, 具有最小下界且下界小于当前值v^*的节点, 将符合条件的节点作为第2阶段的根节点. 求解拉格朗日对偶问题, 下界可由$\sum_{s\in S'}g_s+v(\text{式}(17))+v(\text{式}(18))$获得. 求得可行解, 更新$v^*$. 利用上下界判断该节点能否被剪切, 若不能剪切, 则以这个节点作为一个新树的根节点, 进行第2阶段的分枝过程. 由问题(22)的求解过程可知, 若固定所有变量R_j, 则X_{sj}, $\forall s\in S'$的值也可确定, 因此, 本阶段对变量R_j进行分枝, 在分枝过程中同时确定X_{sj}的值.

分枝策略选择具有最小$\hat{v}_j$($\hat{v}_j=v_j/n_j$)值的变量R_j进行分枝. 在检验0-分枝时, 如果固定为1的零售点的能力加上自由零售点的能力之和小于总需求, 则节点被剪切. 在第2阶段的终端节点, 所有的仓库和零售点均已固定, 如果存在没有剪切的终端节点, 则执行第3阶段的分枝定界.

4.3 第3阶段

设每个终端节点确定建立的零售点集合为J', 则该节点确定的问题为

$$\min\sum_{k\in K}\tau_k\Big\{\sum_{i\in P}\sum_{j\in P}d_{ij}Z_{ijk}\Big\},$$

$$\text{s.t.}\ \text{式}(3)\sim(9),\ (12). \tag{24}$$

其中$P=I\bigcup J'$. 问题(24)与(20)基本相同, 可以采用相同的方法求解. 本阶段对变量Y_{ij}按照$\hat{v}_j=v_j/n_j$由小到大的顺序进行分枝, 在分枝过程中更新上界, 记录上界对应解. 对于所有Y_{ij}变量分枝后得到的节点, 对应问题转化为单仓库的车辆路径问题. 根据问题以及求解过程特点, 不再采用分枝定界方法求解, 而以对所有Y_{ij}变量分枝后得到的节点为问题的终端节点. 分枝过程结束后取上界对应解为最优解.

5 计算实例

为了检验算法的有效性, 构造一组测试问题, 运用C++在Pentium 4 PC机上实现上述算法, 并得出计算结果. 根据问题的节点数目, 构造4组不同规模的测试问题, 问题规模见表1. 每组零售点设3种不同

的吞吐能力, 仓库成本分别为 100, 200, 300. 各客户及候选设施点的坐标依均匀分布在 $[0,100]^2$ 区间内随机生成, 客户需求依均匀分布在区间 $[1,50]$ 内随机生成, 其他参数如表 2 所示. 表 2 设定两种车型, 每辆车的运输能力至少能满足 2 个客户的需求. 测试问题共 36 个实例. 为了适应问题规模, 提高搜索效率, 测试了多个不同的初始步长值、步长减半的阈值和最大次梯度迭代数. 各节点初始参数 β^t 设为 2, 若下界在 4 次迭代后没有改善, 则步长变为原来的一半.

表 1　每组问题规模

问题集	问题规模 ($m \times n \times r$)	变量数量	约束数量	总需求/总车辆能力	零售点能力	总需求/零售点总能力
1	5×10×20	11 910	45 265	0.07	80	0.74
					150	0.39
					250	0.23
2	5×10×50	28 770	180 565	0.19	250	0.57
					350	0.41
					500	0.28
3	5×10×100	56 870	606 065	0.39	350	0.83
					500	0.58
					750	0.39
4	5×10×200	113 070	2 207 065	0.76	750	0.76
					1 000	0.57
					1 500	0.38

表 2　各实例参数范围

参　数	范　围
选址范围	$U[0,\ 100]$
客户需求	$U\,[1,\ 50]$
仓库固定成本	100,200,300
零售点固定成本	100
仓库到零售点单位运输成本	4
车辆数	50
零售点车辆单位运输成本	1,2
车辆运输能力	100,200

表 3~表 5 分别为仓库固定成本为 100, 200 和 300 时的测试结果. 其中启发式上限值是指原问题最好的上限值, 而下限值是通过求解 Lagrangian 松弛问题获得的. 对于 NP-hard 难题, 运用传统分枝定界法求解最优值是非常困难的, 因此利用上界和下界的相对对偶间隙 $((\mathrm{UBD}-\mathrm{LBD})/\mathrm{LBD}) \times 100\%$ 来评价算法. 表中第 2~9 列为设置迭代次数为 800 时的计算结果, 后 4 列是迭代次数为 400 时的计算结果.

表 3　算法计算结果 (1)

问题集	零售点能力	仓库数	零售店数	派出车辆数	运输成本	下限值	启发式上限值	相对对偶间隙	计算时间/s	下限值*	启发式上限值*	相对对偶间隙*	计算时间*/s
1	80	2	8	8	1 492	3 636.69	3 648	0.31	5	3 636.69	3 648	0.31	53
	150	2	5	9	1 414	2 825.61	2 830	0.15	3	2 825.61	2 830	0.15	2
	250	2	3	8	1 678	2 542.71	2 546	0.12	3	2 542.71	2 578	1.25	2
2	250	2	6	18	2 502	4 231.9	4 254	0.52	26	4 231.9	4 254	0.52	12
	350	2	5	17	2 892	4 308	4 308	0	5	4 308	4 308	0	2
	500	2	3	18	3 318	4 182.95	4 186	0.07	4	4 186	4 186	0	2
3	350	3	9	30	5 718	8 260	8 260	0	15	8 260	8 260	0	0
	500	2	6	31	5 336	7 088	7 088	0	15	7 088	7 088	0	0
	750	2	4	29	6 026	7 178	7 178	0	15	7178	7 178	0	7
4	750	2	8	44	9 576	11 892	11 892	0	361	11 861.5	11 892	0.25	25
	1 000	2	6	45	11 144	11 827.3	11 836	0.07	133	11 824.6	11 836	0.09	28
	1 500	2	4	44	10 710	11 862	11 862	0	56	11 862	11 862	0	27

表 4　算法计算结果 (2)

问题集	零售点能力	仓库数	零售店数	派出车辆数	运输成本	下限值	启发式上限值	相对对偶间隙	计算时间/s	下限值*	启发式上限值*	相对对偶间隙*	计算时间*/s
1	80	2	8	8	1 388	3 904	3 904	0	0	3 904	3 904	0	0
	150	2	5	9	1 468	3 047.26	3 084	1.2	6	3 047.26	3 084	1.2	6
	250	2	3	8	1710	2 778	2 778	0	0	2 778	2 778	0	0
2	250	2	6	18	2 502	4 406.94	4 462	1.24	37	4 400.53	4 462	1.39	16
	350	1	5	17	2 892	4 580	4 580	0	39	4 580	4 580	0	17
	500	2	3	18	3 318	4 298.58	4 386	2.03	45	4 331.3	4 386	1.26	19
3	350	3	9	30	5 788	8 474.21	8 560	1.01	118	8 474.21	8 560	1.01	59
	500	2	6	31	5 336	7 288	7 288	0	15	7 288	7 288	0	8
	750	2	4	29	6 842	7 378	7 378	0	16	7 378	7 378	0	8
4	750	1	8	44	11 028	12 180	12 180	0	427	1 2091.7	12 180	0.72	217
	1 000	2	6	45	11 144	12 031.6	12 084	0.43	444	12 036	12 036	0	28
	1 500	2	4	44	10 710	11 990.2	12 062	0.59	493	12 062	12 062	0	24

表 5 算法计算结果 (3)

问题集	零售点能力	仓库数	零售店数	派出车辆数	运输成本	下限值	启发式上限值	相对对偶间隙	计算时间/s	下限值*	启发式上限值*	相对对偶间隙*	计算时间*/s
1	80	1	8	8	1 534	4 038.21	4 042	0.09	3	4 038.21	4 042	0.09	1
	150	1	5	9	1 468	3 237.81	3 256	0.56	6	3 237.81	3 256	0.56	4
	250	1	3	8	1 710	2 910	2 910	0	0	2 910	2 910	0	0
2	250	1	6	18	2 502	4 523.76	4 634	2.43	38	4 523.76	4 634	2.43	15
	350	1	5	17	2 892	4 680	4 680	0	43	4 680	4 680	0	16
	500	1	3	18	3 318	4 461.28	4 550	1.98	38	4 461.28	4 550	1.98	18
3	350	1	9	30	5 788	8 668	8 668	0	122	8 642.3	8 668	0.29	66
	500	2	6	31	5 336	7 465.77	7 468	0.02	57	7 465.77	7 468	0.02	31
	750	1	4	29	6 026	7 542	7 542	0	14	7 542	7 542	0	8
4	750	2	8	44	11 028	12 280	12 280	0	330	12 280	12 280	0	195
	1 000	1	6	45	10 084	12 215.9	12 216	≈0	192	12 215.9	12 216	≈0	83
	1 500	1	4	44	10 710	12 226	12 226	0	56	12 226	12 226	0	24

由于剪切过程有效地减小了分枝定界树的规模, 测试问题至多在分枝定界的第 2 阶段终止. 改变次梯度迭代次数对运算结果的影响不大, 但当问题规模增大时, 会极大地增加计算时间. 对迭代次数为 200 进行测试, 可以获得与迭代次数为 400 基本相同的上限值, 其相对对偶间隙的差别也不超过 6%, 计算时间不超过 80 s. 因此在对偶间隙满足应用需求时, 可以选择较少的迭代次数以提高效率. 当问题规模较大时, 拉格朗日松弛所获得的下界更紧. 当总需求/零售点总能力比率较高时, 计算时间较长, 对偶间隙较大. 这是因为需要建立的零售店较多, 求解过程的分枝定界树的规模随之增大. 若增加零售点能力, 则会减少建立的仓库和零售店数量, 但运输成本会相应增加, 而零售点的建设成本也会随着吞吐能力的增加而增加, 规划中可以根据实际成本选建合适规模的零售店以使总成本最小. 另外, 本文采用启发式方法求解式 (17) 所获得的解为近优解, 当解的质量不好时, 会过早地进行剪切, 影响解的质量. 因此, 采用有效的多车场车辆路径问题求解方法会改善本文的计算结果.

LRP 问题的设施选址子问题和路径子问题的计划周期不匹配. 因为选址决策在一个长的规划周期开始时进行, 期间包含多个路径规划周期, 且若需求是波动的, 则车辆路径随客户需求的变化而变化[15]. 所以, 简单的方法可在求解单周期的LRP问题过程中增加运输成本的权重, 使长计划周期内总成本最小. Salhi等[16]考虑了客户集不变、客户需求变化的多周期动态 LRP 问题, 并研究了多个求解方法.

6 结 论

选址-路径问题集成了设施选址问题和车辆路径问题, 为 NP-hard 问题. 本文研究了具有仓库、零售点和客户的三级物流网络规划问题, 集成仓库和零售点两级设施的选址问题和两级配送路径问题, 构建了 LRP 模型. 模型采用基于拉格朗日松弛的分枝定界算法求解, 基于模型特点, 分 3 个不同的阶段执行分枝定界过程. 通过对不同规模的测试问题进行仿真分析, 数值结果显示该算法求解选址-路径问题是有效的, 且适合较大规模的 LRP 问题. 为适应物流网络规划的实际需要, 进一步研究方向是对动态多周期 LRP 问题的研究.

参考文献(References)

[1] Salhi S, Rand G K. The effect of ignoring routes when locating depots[J]. European J of Operational Research, 1989, 39(2): 150-156.

[2] Maranzana F E. On the location of supply points to minimise transport costs[J]. Operational Research Quarterly, 1964, 15(3): 261-270.

[3] Laporte G, Nobert Y. An exact algorithm for minimizing routing and operating costs in depot location[J]. European J of Operational Research, 1981, 6(2): 224-226.

[4] Laporte G, Nobert Y, Pelletier J. Hamiltonian location problems[J]. European J of Operational Research, 1983, 12(1): 82-89.

[5] Laporte G, Nobert Y, Arpin D. An exact algorithm for solving a capacitated location-routing problem[J]. Annals of Operations Research, 1986, 6(9): 293-310.

[6] Beasley J E. Lagrangian heuristic for location problem[J]. European J of Operational Research, 1993, 65(3): 383-399.

[7] 李志华, 王启富, 钟毅芳, 等. 物流网络设计建模与求解算法研究[J]. 机械工程学报, 2003, 39(2): 84-89.
(Li Z H, Wang Q F, Zhong Y F, et al. Research on modelling and algorithm for material flow network design problem[J]. Chinese J of Mechanical Engineering, 2003, 39(2): 84-89.)

[8] Perl J, askin M S. A warehouse location-routing problem[J]. Transportation Research, 1985, 19(5): 381-396.

(下转第1206页)