第 39 卷第 2 期　　机 械 工 程 学 报　　Vol.39 No.2
2003 年 2 月　　CHINESE JOURNAL OF MECHANICAL ENGINEERING　　Feb. 2003

物流网络设计建模与求解算法研究*

李志华　王启富　钟毅芳　周亦波
(华中科技大学 CAD 中心　武汉　430074)

摘要：研究了制造系统中物流网络设计问题，构建了一个带固定费用的容量受限的网络设计模型，提出了一种求解该问题的基于拉格朗日启发式算法的增强型分枝定界方法。通过大量的试验测试，结果表明该算法能有效地解决大型的具有NP-hard特性的网络设计问题。

关键词：制造系统　物流网络设计　拉格朗日启发式算法　增强型分枝定界算法
中图分类号：TP391.7

0　前言

物流网络设计是制造系统设计中的关键问题。但由于该问题的复杂性，以往的研究大都只集中在布局设计方面[1,2]，而对物流网络设计重视不够。此外，Chhajed 等[3]所建立的网络设计模型存在一定的问题，因为该模型允许物流路径穿越制造资源所占用的区域，这显然是不合理的。

本文构建了一种带固定费用的容量受限的网络设计模型。与已有的模型相比，它能更全面更真实地反映生产实际。但由于该模型为大型 NP-hard 问题[4]。已有的求解方法[5～8]都存在某些不足。因此，提出了一种新的求解方法：基于拉格朗日启发式的增强型分枝定界方法。其基本思想是首先利用拉格朗日松弛法将难解的原问题松弛为容易求解的子问题，以求得原问题的下界；然后利用某种启发式算法求得原问题的上界；最后利用分枝定界法的全局搜索功能，不断地改进原问题的上、下界。为了提高算法的效率，在分枝定界过程中，充分利用已有的信息，通过构造多个剪切准则、惩罚性检测规则和变量固定法则，极大地降低了分枝定界树的规模，从而将传统的分枝定界算法改进为能求解大型网络设计问题的增强型分枝定界算法。

1　网络设计模型

所构建的网络设计模型适用于平面式的物料处理设备。它们沿着固定的路线输送物料。

如图 1 所示，假设在车间平面内已布置了多个制造资源。现将一单位正交网格加在其上，网格的每一交叉点代表无向图中的一个顶点。但是，制造资源内部和限制区域内部所占用的网格点(如图中的 A 点)不能作为无向图中的顶点(因为物流路径不能穿越制造资源和限制区域)，唯一例外的是制造资源的输入/输出口所在的网格点(即图中标为 I、O 的点)，这些 I、O 点应作为无向图中的顶点，因为物料正是通过它们进入或离开制造资源的。

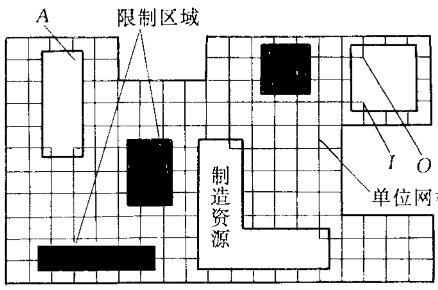

图 1　车间平面及制造资源的离散化

用 N 表示有效的网格点集，用 E 表示连接这些网格点的无向边集，用 A 表示有向弧集，即对于 $\{i,j\}\in E$，有 $(i,j),(j,i)\in A$，用 $G=(N,E)$ 表示相应的无向图。此外，用 T 表示制造资源间的物流集，每一物流 k 有一个源点(对应某个制造资源的出口)和一个终点(对应另一制造资源的入口)，分别记为 $o(k)$ 和 $d(k)$，其流量大小记为 r^k。用 f_{ij} 表示网络边 $\{i,j\}$ 的构建费用，用 c_{ij}^k 表示单位物流 k 在有向弧 (i,j) 上的运输费用，用 u_{ij} 表示无向边 $\{i,j\}$ 的容量。最后，引进两组变量：流量变量 x_{ij}^k 和设计变量 y_{ij}，其意义如下

$$x_{ij}^k = \text{物流 } k \text{ 在有向弧}(i,j)\text{上的流量}$$

$$y_{ij} = \begin{cases} 1 & \text{若无向边}\{i,j\}\text{为物流网络边} \\ 0 & \text{否则} \end{cases}$$

于是，带固定费用的容量受限的网络设计模型可用数学形式描述如下

* 国家自然科学基金资助项目(59905009)。20010814 收到初稿，20011220 收到修改稿

【CNDP】

$$v^*=\min\sum_{k\in T}\sum_{\{i,j\}\in E}(c_{ij}^k x_{ij}^k+c_{ji}^k x_{ji}^k)+\sum_{\{i,j\}\in E}f_{ij}y_{ij} \tag{1}$$

$$\text{s.t.}\quad \sum_{j:(i,j)\in A}x_{ij}^k-\sum_{j:(j,i)\in A}x_{ji}^k=b_i^k\quad \forall i\in N,\ \forall k\in T \tag{2}$$

$$\sum_{k\in T}(x_{ij}^k+x_{ji}^k)\leqslant u_{ij}y_{ij}\quad \forall\{i,j\}\in E \tag{3}$$

$$x_{ij}^k+x_{ji}^k\leqslant d_{ij}^k y_{ij}\quad \forall\{i,j\}\in E,\ \forall k\in T \tag{4}$$

$$x_{ij}^k\geqslant 0,\quad x_{ji}^k\geqslant 0\quad \forall\{i,j\}\in E,\ \forall k\in T \tag{5}$$

$$y_{ij}\in\{0,1\}\quad \forall\{i,j\}\in E \tag{6}$$

式中 $b_i^k=\begin{cases}r^k & 当 i=o(k)\\ -r^k & 当 i=d(k)\\ 0 & 其他\end{cases}$，$d_{ij}^k=\min(r^k,\ u_{ij})$

目标函数式(1)包含两种费用：物料运输总费用和网络构建总费用。这两种费用是相互矛盾的，网络设计的目标就是要求得这二者之间的最佳平衡。约束式(2)是流量守恒等式，它使每一物流路径保持连续。约束式(3)为容量限制约束，它使通过每一无向边的流量不超过其最大值，以避免物流阻塞。约束式(4)禁止物流通过物流网络以外的边。约束式(5)和式(6)限制变量的取值范围。

2　拉格朗日启发式算法

拉格朗日启发式算法包括三部分：原问题的拉格朗日松弛、求解拉格朗日对偶问题的次梯度优化和产生原问题可行解的启发式算法。

2.1　拉格朗日松弛

用拉格朗日乘子 w_i^k 对 CNDP 的约束式(2)进行松弛，得到拉格朗日对偶问题

【LD】　$v_{\mathrm{L}}=\max\varphi(w)$

拉格朗日对偶问题与乘子 w 有关。对于任一固定的乘子值 $w=\overline{w}$，拉格朗日松弛问题为

【DS】

$$\varphi(\overline{w})=\min\sum_{k\in T}\sum_{\{i,j\}\in E}(c_{ij}^k x_{ij}^k+c_{ji}^k x_{ji}^k)+\sum_{\{i,j\}\in E}f_{ij}y_{ij}+\sum_{k\in T}\sum_{i\in N}\overline{w}_i^k\Big(b_i^k-\sum_{j:(i,j)\in A}x_{ij}^k+\sum_{j:(j,i)\in A}x_{ji}^k\Big)=$$

$$\min\sum_{k\in T}\sum_{\{i,j\}\in E}(\tilde{c}_{ij}^k x_{ij}^k+\tilde{c}_{ji}^k x_{ji}^k)+\sum_{\{i,j\}\in E}f_{ij}y_{ij}+\sum_{k\in T}\sum_{i\in N}\overline{w}_i^k b_i^k$$

$$\text{s.t.}\quad \sum_{k\in T}(x_{ij}^k+x_{ji}^k)\leqslant u_{ij}y_{ij}\quad \forall\{i,j\}\in E$$

$$x_{ij}^k+x_{ji}^k\leqslant d_{ij}^k y_{ij}\quad \forall\{i,j\}\in E,\ \forall k\in T$$

$$x_{ij}^k\geqslant 0,\quad x_{ji}^k\geqslant 0\quad \forall\{i,j\}\in E,\ \forall k\in T$$

$$y_{ij}\in\{0,1\}\quad \forall\{i,j\}\in E$$

式中 $\tilde{c}_{ij}^k=c_{ij}^k+\overline{w}_j^k-\overline{w}_i^k$　$\tilde{c}_{ji}^k=c_{ji}^k+\overline{w}_i^k-\overline{w}_j^k$

$\forall\{i,j\}\in E$　$\forall k\in T$

DS 问题可以进一步分解成 $|E|$ 个子问题，每一个子问题对应一条无向边 $\{i,j\}$。

【DS_{ij}】

$$g_{ij}(\overline{w})=\min\sum_{k\in T}(\tilde{c}_{ij}^k x_{ij}^k+\tilde{c}_{ji}^k x_{ji}^k)+f_{ij}y_{ij}$$

$$\text{s.t.}\quad \sum_{k\in T}(x_{ij}^k+x_{ji}^k)\leqslant u_{ij}y_{ij}$$

$$x_{ij}^k+x_{ji}^k\leqslant d_{ij}^k y_{ij}\quad \forall k\in T$$

$$x_{ij}^k\geqslant 0,\quad x_{ji}^k\geqslant 0\quad \forall k\in T$$

$$y_{ij}\in\{0,1\}$$

注意，DS_{ij} 问题与运筹学中典型的背包问题相类似，因此可以用贪婪算法原理求解 DS_{ij}。下面给出具体的过程。

(1) 置 $\hat{g}_{ij}=f_{ij}$，$x_{ij}^k=0$，$x_{ji}^k=0, \forall k\in T$。

(2) 各 $\tilde{c}_{ij}^k$，$\tilde{c}_{ji}^k$，$\forall k\in T$ 由小到大排序，取第一个元素,假设为 i'，j'，k'。

(3) 若 $\tilde{c}_{i'j'}^{k'}>0$，则转(6)。

(4) 若 $u_{ij}>r^{k'}$，则置 $x_{i'j'}^{k'}=r^{k'}$，$u_{ij}=u_{ij}-r^{k'}$，$\hat{g}_{ij}=\hat{g}_{ij}+\tilde{c}_{i'j'}^{k'}x_{i'j'}^{k'}$，$\tilde{c}_{i'j'}^{k'}=+\infty$ 和 $\tilde{c}_{j'i'}^{k'}=+\infty$。否则，置 $x_{i'j'}^{k'}=u_{ij}$，$u_{ij}=0$ 和 $\hat{g}_{ij}=\hat{g}_{ij}+\tilde{c}_{i'j'}^{k'}x_{i'j'}^{k'}$。

(5) 若 $u_{ij}>0$，则转(2)。

(6) 若 $\hat{g}_{ij}\leqslant 0$，则置 $y_{ij}=1$。否则，置 $y_{ij}=0$，$x_{ij}^k=0$，$x_{ji}^k=0$，$\forall k\in T$。

在求解完所有的 DS_{ij} 后，再求解 DS 问题则很容易，只需 $\varphi(\overline{w})=\sum_{\{i,j\}\in E}\hat{g}_{ij}\overline{y}_{ij}+\sum_{k\in T}\sum_{i\in N}\overline{w}_i^k b_i^k$。

以上只是求出了原问题的一个下界值 $\varphi(\overline{w})$。要想得到最优的下界 v_{L}（与乘子 w 有关），则还必须求解拉格朗日对偶问题 LD。这是通过次梯度优化实现的。

2.2　次梯度优化

为了得到原问题的最佳下界 v_{L}，次梯度优化过程就必须求出一系列的乘子值 w，使下界 $\varphi(w)$ 沿着上升的方向渐近 v_{L}。记 $\overline{w}_l$ 为第 l 次循环时的拉格朗日乘子值，则在下一次循环时乘子值为：$\overline{w}_{l+1}=\overline{w}_l+t_l d_l$。乘子的初始值为 0。

上式中，t_l 和 d_l 分别为第 l 次循环时的搜索步长和搜索方向。搜索方向是按 $\varphi(w)$ 的上升方向构造的，一般取为 $d_l=\xi_l$。其中 ξ_l 为次梯度。

$\varphi(\overline{w})$ 的次梯度 $\xi_i^k=b_i^k-\sum_{j:(i,j)\in A}\overline{x}_{ij}^k+\sum_{j:(j,i)\in A}\overline{x}_{ji}^k$，

$\forall i \in N$, $\forall k \in T$。其中 $\bar{x}$ 为 DS 在 $\overline{w}$ 处的优化解。

搜索步长 $t_l = \lambda_l (\tilde{v} - \varphi(\overline{w}_l)) / \|\xi_l\|^2$，其中，$\tilde{v} = (\eta \bar{v} + \varphi(\overline{w}_l))/2$，一般取 $\lambda_l = 2$ 和 $\eta = 1.05$。$\bar{v}$ 为迄今为止得到的最好上界。

次梯度优化的终止准则：① $\xi_l = 0$，这是最理想的情况，但这种情况很少发生，实际中常用下面的终止准则来代替。② $\|d_l\| \leqslant \varepsilon$，或 $t_l \leqslant \varepsilon$，或 $\bar{v} - \varphi(\overline{w}_l) \leqslant \varepsilon$，($\varepsilon$ 为一给定的极小正数)。③为了加快速度，有时还采用终止准则 $l > M$。M 为一给定的迭代次数，当循环超过这个次数时就强行 终止。

2.3 解的可行化

为了得到 CNDP 的可行解，考虑一个给定的无向边子集：$\overline{E} \subseteq E$，其中子集中的边都是物流网络边，即 $\overline{E} = \{\{i,j\} \in E : y_{ij} = 1\}$。这样，原问题的子问题为

【PS】

$$v(\overline{E}) = \min \sum_{k \in T} \sum_{\{i,j\} \in \overline{E}} (c_{ij}^k x_{ij}^k + c_{ji}^k x_{ji}^k) + \sum_{\{i,j\} \in \overline{E}} f_{ij}$$

$$\text{s.t.} \quad \sum_{j:(i,j) \in \overline{A}} x_{ij}^k - \sum_{j:(j,i) \in \overline{A}} x_{ji}^k = b_i^k \quad \forall i \in N, \forall k \in T$$

$$\sum_{k \in T} (x_{ij}^k + x_{ji}^k) \leqslant u_{ij} \quad \forall \{i,j\} \in \overline{E}$$

$$x_{ij}^k \geqslant 0, x_{ji}^k \geqslant 0 \quad \forall \{i,j\} \in \overline{E}, \forall k \in T$$

PS 问题是一个多物流的网络流问题，可通过下面的启发式增量算法求解。

(1) 在未选取的物流中(初始值为 T)，随机地选取某一物流，设为 k。若所有的物流都已完成，则转(8)。

(2) 对于第 k 个物流，令 $x_{ij}^k = 0$, $x_{ji}^k = 0$。

(3) 在图 $G = (N, \overline{E})$ 中，对 $\overline{E}$ 的所有边赋以权值 1。

(4) 用 Dijkstra 算法在图 G 中寻找从源点 $o(k)$ 到终点 $d(k)$ 的最短路径。若没有路径存在，则停止，此时 PS 无解。

(5) 求该路径上流量可能增加的最大值 $f_m = \min\{$路径上各无向边的容量$\}$。

(6) 求该路径上流量的净增量 $f_a = \min\{f_m, r^k\}$

修改流经路径上各弧的流量 $x_{ij}^k = x_{ij}^k + f_a$；修改路径上各无向边的容量 $u_{ij} = u_{ij} - f_a$；修改第 k 个物流的流量值 $r^k = r^k - f_a$；修改图 G，即去除 $u_{ij} = 0$ 的无向边。

(7) 若 $r^k = 0$，则转(1)。否则，转(4)。

(8) 计算上界值 $v(\overline{E})$。

3 分枝定界过程

3.1 基本策略

分枝是对设计变量 y 进行的，每次得到两枝：$y_{ij} = 1$ 和 $y_{ij} = 0$。在分枝树的每一结点处，拉格朗日启发式算法首先产生该结点的一个可行解，然后执行一系列的次梯度搜索。在搜索过程中，基于求解拉格朗日对偶问题所产生的信息，启发式增量算法被定期地用来产生新的可行解。为此，将求解 DS 时所得到的 y 值用来构成 PS 的 $\overline{E}$ 集。记 $E(\overline{w}_l)$ 为求解 DS 时第 l 次循环所得到的所有 $y_{ij} = 1$ 的边集，则 $\overline{E} = E(\overline{w}_l)$。

此外，通过构造惩罚性检测规则，用来在次梯度搜索过程中确定某些变量的正确取值。经过一定次数的循环后，拉格朗日启发式算法将能检测出某结点或者可以被剪切掉或者可以进一步分枝。在分枝之前，还可以固定某些变量的取值，以使算法只集中于很有希望的分枝。

3.2 剪切和停止准则

标准的剪切准则如下。

(1) 求得结点的优化解(即 $\xi_l = 0$)。

(2) 结点无可行解。

(3) 求解拉格朗日对偶问题时，下界超过迄今为止获得的最好上界，即 $\varphi(\overline{w}_l) \geqslant \bar{v}$。

由于收敛速度、解的质量等多种因素的影响，上述 3 条剪切准则在次梯度搜索过程中有可能不能满足。下面，将导出另外的一条准则。

记 $c(E)$ 为物料运输费用的最优值，其中 E 的所有边都可用作物流网络边，并且记 $PS(E)$ 为相应的原问题的子问题。

定理 1 $c(E) \leqslant c(\overline{E})$, $\forall \overline{E} \subseteq E$

定理 2 $c(E_1 \cup E_*) \leqslant c(E_1' \cup E_*')$

其中 $E_1 = \{\{i,j\} \in E : y_{ij} = 1\}$, $E_0 = \{\{i,j\} \in E :, y_{ij} = 0\}$，$E_* = E \setminus (E_0 \cup E_1)$，并且节点 (E_0', E_1', E_*') 是由结点 (E_0, E_1, E_*) 进一步分枝得到的子结点。

基于上述两条定理(限于篇幅，定理的推导证明省略)，有：

定理 3 若结点 (E_0, E_1, E_*) 满足 $c(E_1 \cup E_*) + \sum_{(i,j) \in E_1} f_{ij} \geqslant \bar{v}$，那么该结点将不可能导出一个使目标函数值比 $\bar{v}$ 更小的可行解。

由定理 3，可以得到下面的剪切准则。

(4) 若 $c(E_1 \cup E_*) + \sum_{(i,j) \in E_1} f_{ij} \geqslant \bar{v}$，则结点

(E_0, E_1, E_*) 可被剪切。

如果以上剪切准则都不满足，那么就必须在某一时刻终止拉格朗日启发式算法，然后再进行分枝。终止准则见 2.2 部分。

3.3　惩罚性检测

对任一无向边 $\{i,j\}\in E_*$，由 2.1 部分的贪婪算法知，若 $\hat{g}_{ij}>0$，则 $y_{ij}=0$。因此，下界 $\varphi(\overline{w})$ 中不包含该 $\hat{g}_{ij}$ 项。如果此时将该设计变量的值固定为 1(即取 $y_{ij}=1$ 这一分枝)，那么下界值应为 $\varphi(\overline{w})+\hat{g}_{ij}$。如果该下界值又恰好大于上界值，即 $\varphi(\overline{w})+\hat{g}_{ij}\geqslant\overline{v}$，则根据剪切准则(3)，$y_{ij}=1$的这一分枝可被剪切。因此，我们只需考虑 $y_{ij}=0$的这一分枝。同样，若 $\hat{g}_{ij}<0$，则 $y_{ij}=1$，此时下界 $\varphi(\overline{w})$ 中包含该 $\hat{g}_{ij}$ 项。如果固定 $y_{ij}=0$，则下界值应为 $\varphi(\overline{w})+\left|\hat{g}_{ij}\right|$。如果该下界值又恰好大于上界值，即 $\varphi(\overline{w})+\left|\hat{g}_{ij}\right|\geqslant\overline{v}$，则 $y_{ij}=0$的这一分枝可被剪切。因此，只需考虑 $y_{ij}=1$ 的这一分枝。

综上所述，在每次求解 DS 时，都可以通过下面惩罚性检测规则来判断不好的分枝。即对于每一无向边 $\{i,j\}\in E_*$ 有：

(1) 若 $\hat{g}_{ij}>0$，并且 $\varphi(\overline{w})+\hat{g}_{ij}\geqslant\overline{v}$，则固定 $y_{ij}=0$。

(2) 若 $\hat{g}_{ij}<0$，并且 $\varphi(\overline{w})+\left|\hat{g}_{ij}\right|\geqslant\overline{v}$，则固定 $y_{ij}=1$。

3.4　启发式变量固定法

启发式变量固定法(β 固定法)用来加速分枝定界过程。它考察求解 DS 时所得到的 $\hat{g}_{ij}$ 解的情况。由 2.1 和 3.3 部分知：y_{ij} 的值取决于 $\hat{g}_{ij}$ 的值，并且当 $\hat{g}_{ij}>>0$(或 $\hat{g}_{ij}>>0$)时，y_{ij} 在最后的优化解中应取值 0(或 1)。因此，可以用 $\left|\hat{g}_{ij}\right|$ 的值来构造 β 固定法：取 $\beta\in[0,1]$，并记 n 为初始的未固定的边数(一般为 $|E|$)。则每次需要固定的边数为 $\lceil\beta n\rceil$ 或者为 $|E_*|$ (当 $|E_*|<\lceil\beta n\rceil$ 时)。为了选择待固定的设计变量，将 $\left|\hat{g}_{ij}\right|$ 的值由大到小排序，然后依次选取相应的设计变量。为了综合考虑次梯度循环过程中所得到的多个 $\hat{g}_{ij}$ 的效果，令 $R_{ij}=\hat{g}_{ij,l}$。然后在次梯度优化过程中，如果第 l 次循环得到一个较好的下界，就使 $R_{ij}=r*R_{ij}+\hat{g}_{ij,l}$，其中 $\gamma\in[0,1]$，一般取 $\gamma=0.5$。这样，当拉格朗日启发式算法终止时，就可以用下面的启发式算法固定一些设计变量。

(1) 置 $m=1$。

(2) 按 $\left|R_{ij}\right|$ 由大到小的次序，选择一条边，设为 $\{i',j'\}$。这里 $\{i,j\}\in E_*$。

(3) 若 $R_{i'j'}<0$，则固定 $y_{i'j'}=1$；否则，固定 $y_{i'j'}=0$。置 $E_*=E_*\setminus\{i',j'\}$ 和 $m=m+1$。

(4) 若 $m\geqslant\min(|E_*|,\lceil\beta n\rceil)$，则算法停止。否则，转(2)。

β 固定法的最大特点是，一旦确定了 β 的取值便能知道分枝定界树的最大规模。因此，可以通过改变 β 的值来方便地控制算法的计算时间和计算质量。

3.5　分枝和搜索策略

分枝策略是指选择什么样的设计变量进行分枝的问题。一种较好的选择是对 y_{ij} 取值最不确定的设计变量进行分枝。y_{ij} 的值取决于 $\hat{g}_{ij}$ 的值，$\left|\hat{g}_{ij}\right|$ 越大，y_{ij} 的取值也就越确定。因此，在使用 β 固定法之后，则需在剩余的设计变量中选择具有最小 $\left|\hat{g}_{ij}\right|$ 值的那个变量进行分枝。

搜索策略是指两个分枝中，优先搜索哪个分枝的问题，以及按何种方式搜索整个分枝树的问题。假设有两个分枝 $y_{ij}=0$ 和 $y_{ij}=1$；又假设在最后一次求解 DS 时得 $y_{ij}=\tilde{y}_{ij}$。于是有四种可供选择的搜索策略：① 优先搜索 $y_{ij}=\tilde{y}_{ij}$ 的分枝。② 优先搜索 $y_{ij}\neq\tilde{y}_{ij}$ 的分枝。③ 优先搜索 $y_{ij}=0$ 的分枝。④ 优先搜索 $y_{ij}=1$ 的分枝。为了充分利用已有的信息，节省计算时间，可选择第一种搜索策略，并用深度优先法存储和搜索分枝定界树中的结点。

3.6　总体算法

记分枝定界树中未访问的结点集为 Φ，则总体算法如下。

(1) 初始化 B&B。置 $E_0=E_1=\phi$，$E_*=E$，$\Phi=\{(E_0,E_1,E_*)\}$ 和 $\overline{v}=\infty$。

(2) 初始化结点。

a) 根据深度优先策略，从 Φ 中选取一结点，更新 E_0,E_1,E_* 的值，并为当前结点获得初始乘子值 $\overline{w}_1$。(对于根结点：$\overline{w}_1=0$；对于其他结点：$\overline{w}_1$ 为最近的乘子值)。

b) 用增量算法求解 PS $(E_1\cup E_*)$。若 $c(E_1\cup E_*)+\sum_{(i,j)\in E_1}f_{ij}\geqslant\overline{v}$，或无可行解，则转(9)。否则，进入下一步。

c) 若 $v(E_1\cup E_*)<\overline{v}$，则置 $\overline{v}=v(E_1\cup E_*)$，并记录变量 y_{ij} 的值。若 $E_*=\phi$，或者 $E_*\neq\phi$，但求解 PS 时，没有使用到 E_* 中的任何边，则转(9)。否则，进入下

一步。

d) 置 $\lambda_1=\lambda_0$, $l=1$ 和 $\underline{v}=$ 父结点的最好下界(若此结点为根结点，则 $\underline{v}=0$)。

(3) 对偶子问题。用 $\overline{w}_l$, E_0, E_1 和 E_* 求解 DS。若 $\varphi(\overline{w}_l)>\underline{v}$ ，则置 $\underline{v}=\varphi(\overline{w}_l)$ 和 $\lambda_{l+1}=\lambda_l$ 。若 $\underline{v}\geqslant\overline{v}$ ，则转(9)。否则，若 $\underline{v}$ 经过连续 K 次循环后还未得到改善，则置 $\lambda_{l+1}=\lambda_l/2$ 。计算次梯度 ξ_l 。

(4) 惩罚性检测。对任意 $\{i,j\}\in E_*$，利用惩罚性检测规则来固定变量。更新 E_0, E_1, E_*。若所有 y_{ij} 都被固定，即 $E_*=\phi$，则用增量算法求解 PS(E_1)，若有可行解且 $v(E_1)<\overline{v}$，则置 $\overline{v}=v(E_1)$，然后转(9)。若 $E_*\neq\phi$，则进入下一步。

(5) 原问题的子问题。置 $\overline{E}=E(\overline{w}_l)$ ，并用增量算法求解 PS($\overline{E}$) 。若 $v(\overline{E})<\overline{v}$ ，则置 $\overline{v}=v(\overline{E})$ 和 $\lambda_{l+1}=\lambda_0$ 。若 $\underline{v}\geqslant\overline{v}$ ，则转(9)。

(6) 优化值检验。若 $\|\xi_l\|=0$ ，则转(9)。

(7) 次梯度搜索。计算搜索方向 $d_l=\xi_l$ 和搜索步长 $t_l=\lambda_l(\tilde{v}-\varphi(\overline{w}_l))/\|\xi_l\|^2$ ，计算新的乘子：$\overline{w}_{l+1}=\overline{w}_l+t_l d_l$ 。

(8) 终止检测。若 $\|d_l\|\leqslant\varepsilon$ ，或 $t_l\leqslant\varepsilon$ ，或 $l\geqslant M$，则转(10)。否则，置 $l=l+1$ ，然后转(3)。

(9) 剪切。置 $\Phi=\Phi/\{(E_0, E_1, E_*)\}$ 。若 $\Phi=\phi$ ，则停止。否则，执行一次回溯，然后转(2)。

(10) 变量固定。在剩余的未固定的变量中，使用 β 固定法固定变量。更新 E_0、E_1、E_*的值。若所有 y_{ij} 都已固定，则用增量算法求解 PS(E_1)。若 $v(E_1)<\overline{v}$ ，则置 $\overline{v}=v(E_1)$，然后转(9)。

(11) 分枝。根据分枝和搜索策略选取一变量 y_{ij} 进行分枝，并将产生的两个新的分枝结点加入到 Φ中，然后转(2)。

4 计算实例

用 C 语言在 PentiumIII的微机上实现了上述算法。为了检验算法的有效性，对不同形状和大小的网格图、以及不同的物流数据进行了大量测试。根据图中包含的无向边数，试验数据分成 20、30、40、50 和 60 五大组，每组按照表 1 所示的范围随机产生 20 个实例。

增强型分枝定界算法的参数选择如下：$M=100,\ K=4,\ \lambda_0=2,\ \eta=1.05,\ \varepsilon=0.01$ 。算法的最终结果与最优值和下界值的比较如表 2 所示。对于小规模的问题，将增强型分枝定界算法产生的近优值(v_{EBB})与传统分枝定界算法产生的最优值(v_{OPT})进行比较，20 个实例中有 17 个得到最优解，其余 3 个得到近优解，其目标函数值的平均误差百分比为 2.7%。对于较大规模的问题，由于 NP-hard 特性，用传统分枝定界法求解最优值是非常困难的，因此可用上界(v_{EBB})和下界(v_{LB})的差来评价算法。结果如表 2 中的第三列。可以看出，目标函数值的平均误差百分比最大不超过 6%，比参考文献[6]采用线性松弛法和启发式算法得到的 8%要低得多。这说明增强型分枝定界方法比一般的启发式算法　要好。

表 1　各实例的参数范围

参　数	范　围
物流个数	10～30
流量大小	5～15
网络边构建成本	1～10
网络边容量	40～200
单位运输成本	1～5

表 2　算法结果与最优值和下界值的比较

问题规模 (即无向边数)	与最优值的比较 $\frac{v_{\text{EBB}}-v_{\text{OPT}}}{v_{\text{OPT}}}\times100\,\%$	与下界值的比较 $\frac{v_{\text{EBB}}-v_{\text{LB}}}{v_{\text{LB}}}\times100\,\%$
20	2.7	4.3
30	—	3.9
40	—	5.7
50	—	5.1
60	—	4.6

下面再举一个将设备布局和物流网络集成设计的例子。如图 2 所示，已知六个制造资源的几何尺寸和输入/输出口位置，其相互间的物流大小如表 3 所示；同时已知 $c_{ij}^k=1$， $u_{ij}=60$ 和 $f_{ij}=5$ 。集成设计的方法为：首先由模拟退火法产生各种布局方案，然后调用增强型分枝定界法求解该布局下的物流网络；通过不断循环，最后同时得到较优的布局和较好的物流网络,如图 2 所示。增强型分枝定界算法在此处求得的上、下界分别为 617 和 583，其误差百分比为 5.83%。由图中可以看出，制造资源之间的布局显得非常的整齐、简洁和紧凑，这有利于节省布局空间；同时物流网络的拓扑结构也显得非常的合理，它由主通道和分支通道构成。这种形式的物流网络能简化物流控制，提高物料运输效率。由此可见，所建立的网络设计模型是正确的，其求解算法是有效的。

表 **3**　制造资源间的物流大小

	#1	#2	#3	#4	#5	#6
#1	—	—	—	5	—	14
#2	8	—	—	—	15	—
#3	—	12	—	10	7	—
#4	—		15	—	—	—
#5	—	6	—	—	—	—
#6	—	—	5	—	—	—

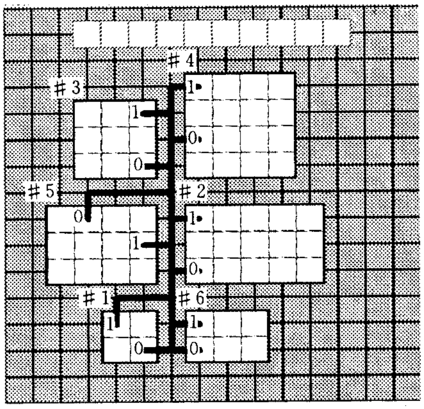

图 2　制造系统集成设计

5　结论

带固定费用的容量受限的网络设计模型克服了已有模型的不足，能更真实更全面地反映生产实际。此外，该模型还可用于其他领域，如交通运输网络的设计、计算机网络和通信网络的设计等。考虑到该模型为大型 NP-hard 问题，传统的优化方法无能为力。因此，提出了一种新的增强型分枝定界方法。该方法在以下几个方面对传统的分枝定界法作了改进：① 分枝定界树的规模。通过构造惩罚性检测规则和启发式变量固定法则，大大降低了分枝定界树的规模。② 分枝策略。对取值最不确定的变量进行分枝，避免了做无谓的分枝计算。③ 剪切准则。除了标准的剪切准则外，还构造了一个附加准则。④ 搜索策略。不是随机地搜索某一分枝，而是有选择地进行搜索。通过大量的试验测试，结果表明所构建的网络设计模型是正确的，其求解算法是有效的。

参　考　文　献

1　Heragu S S. Recent models and techniques for solving the layout problem. European Journal of Operational Research, 1992, 57(2)：136～144

2　Heragu S S, Kusiak A. Efficient models for the facility layout problem. European Journal of Operational Research, 1991, 53(1)：1～13

3　Chhajed D, Montreuil B, Lowe T J. Flow network design for manufacturing systems layout. European Journal of Operational Research, 1992, 57(2)：145～161

4　Johnson D S, Lenstra J K, Rinnooy H G. The complexity of the network design problem. Networks, 1978, 8：279～285

5　Magnanti T L, Wong R T. Network design and transportation planning：models and algorithms. Transportation Science, 1984, 18(1)：1～55.

6　Holmberg K, Yuan D. A lagrangean approach to network design problems. International Transactions in Operational Research, 1998, 5(6)：529～539

7　Herrmann J W. A dual ascent approach to the fixed-charge capacitated network design problem. European Journal of Operational Research, 1996, 95：476～490

8　Sridhar V, Park J S. Benders-and-cut algorithm for fixed-charge capacitated network design problem. European Journal of Operational Research, 2000, 125：622～632

RESEARCH ON MODELING AND ALGORITHM FOR MATERIAL FLOW NETWORK DESIGN PROBLEM

Li Zhihua　Wang Qifu　Zhong Yifang　Zhou Yibo
(*Huazhong University of Science and Technology*)

Abstract：The material flow network design problem in manufacturing systems is investigated. A fixed-charge capacitated network design model is formulated, and an enhanced branch-and-bound algorithm based on a Lagrangian heuristic is proposed to solve the resulting NP-hard problem. The Lagrangian heuristic provides both upper and lower bounds to the problem, and the branch-and-bound procedure is based on the information generated by the Lagrangian heuristic to reduce the size of the branch-and-bound tree. The method is tested on networks of various sizes. Results indicate that the algorithm can provide good solutions in reasonable time.

Key words：Manufacturing systems
Material flow network design
Lagrangian heuristic
Enhanced branch-and-bound algorithm

作者简介：李志华，男，1966 年出生，博士研究生。主要从事 CAD/CAM 和 PDM 的研究工作。