```
    >= number aircraft departed thus far;
   OVNITE( I, J) +
    @SUM( FXCXC( N1, J1, K1)| K1 #EQ# J #AND#
   ARVAT( N1, J1, K1) #LT# DEPAT( N, J, K):
   X( I,N1,J1,J))>=@SUM( FXCXC( N1,J1,K1)| J1 #EQ# J #AND#
   DEPAT( N1, J1, K1) #LE# DEPAT( N, J, K):
   X( I, N1, J, K1))));
! This model does not allow deadheading, so at the end of the day,
    arrivals must equal departures;
   @FOR( ACRFT( I):
          @FOR( CITY( J):
          @SUM( AXF( I, N, J1, J): X( I, N, J1, J)) =
          @SUM( AXF( I, N, J, K): X( I, N, J, K))));
   ! Each flight must be covered;
   @FOR( FXCXC( N, J, K):
          @SUM( AXF( I, N, J, K): X( I, N, J, K)) = 1);
  ! Fleet size limits;
   @FOR( ACRFT( I): @SUM( AXC( I, J):
    OVNITE( I, J)) <= FSIZE( I));
  ! Fractional planes are not allowed;
  @FOR( AXF: @GIN( X); );
END
```

Đôi khi, đặc biệt là trong vận tải đường bộ, đã có một trong các tùy chọn của việc sử dụng xe được thuê chỉ để thực hiện các chuyến đi đã được lựa chọn. Đối với mô hình này, việc sửa đổi chính là các xe được thuê không cần phải tôn trọng bảo tồn các hạn chế dòng chảy. Các chi tiết khác, đôi khi có liên quan đến việc bảo trì phương tiện vận chuyển. Ví dụ, với máy bay, một máy bay cụ thể phải được thực hiện dịch vụ bảo dưỡng sau khi đã hạ cánh một số thời gian quy định, hoặc sau một số giờ bay quy định, hoặc sau một thời gian nhất định đã qua. Đó không phải là quá khó khăn để kết hợp các chi tiết đó, mặc dù mô hình sẽ trở nên lớn hơn đáng kể.

8.9.6. Mô hình dòng chảy Leontief

Trong một mô hình dòng chảy Leontief, mỗi hoạt động sản xuất một đầu ra. Tuy nhiên, nó có thể không sử dụng hoặc sử dụng nhiều đầu vào. Ví dụ sau đây minh họa.

Ví dụ: Mô hình đầu vào-đầu ra Islandia.

Các nước của Islandia có bốn ngành công nghiệp xuất khẩu chính: thép, máy móc tự động, điện tử, và nhựa. Bộ trưởng kinh tế của Islandia muốn tối đa hóa xuất khẩu, nhập khẩu. Đơn vị trao đổi trong Islandia là klutz. Giá tính bằng đơn vị klutzes trên thị trường thế giới của một đơn vị thép, máy tự động, điện tử, và nhựa, tương ứng là: 500, 1500, 300, và 1200. Sản xuất một đơn vị thép đòi hỏi 0.02 đơn vị máy tự động; 0.01 đơn vị của nhựa; 250 klutzes để mua nguyên liệu trên thị trường thế giới, cộng với một nửa ngày công lao động của một người trong 1 năm lao động. Sản xuất một đơn vị tự động đòi hỏi 0.8 đơn vị thép, 0.15 đơn vị điện tử, 0.11 đơn vị bằng nhựa, số công lao động của một người lao động trong 1 năm, và 300 klutzes vật liệu nhập khẩu. Sản xuất một đơn vị thiết bị điện tử đòi hỏi 0.01 đơn vị thép, 0,01 đơn vị máy tự động, 0,05 đơn vị bằng nhựa, một nửa số ngày công lao động của một người trong 1 năm, và 50 klutzes vật liệu nhập khẩu. Sản xuất máy tự động được giới hạn ở 650000 đơn vị. Sản xuất một đơn vị của nhựa cần 0,03 đơn vị máy tự động, 0,2 đơn vị thép, 0,05 đơn vị điện tử, số ngày công của 2 người lao động trong 1 năm, cộng với 300 klutzes nguyên vật liệu nhập khẩu. Các giới hạn trên nhựa là 60000 đơn vị. Tổng số nhân lực có sẵn trong Islandia là 830.000 người mỗi năm. Không có thép, máy tự động, điện tử, hoặc các sản phẩm nhựa được nhập khẩu.

Hỏi có bao nhiêu số lượng các loại sản phẩm phải được sản xuất và xuất khẩu?

Xây dựng mô hình bài toán và tìm giải pháp tối ưu về vấn đề của Islandia.

Việc xây dựng một mô hình đầu vào - đầu ra nên làm theo quy trình hai bước tương tự cho xây dựng bất kỳ mô hình QHTT nào. Cụ thể, (1) xác định các biến quyết định và (2) xác định các ràng buộc (hạn chế hay khó khăn). Chìa khóa để xác định các biến quyết định cho vấn đề này là làm cho sự phân biệt giữa số lượng hàng hóa sản xuất và số lượng xuất khẩu. Một khi điều này được thực hiện, các biến quyết định có thể được biểu như:

PROD(STEEL) là số lượng đơn vị thép được sản xuất,
PROD(AUTO) là số lượng đơn vị máy tự động được sản xuất,
PROD(PLASTIC) là số lượng đơn vị nhựa được sản xuất,
PROD(ELECT) là số lượng đơn vị thiết bị điện tử được sản xuất,
EXP(STEEL) là số lượng đơn vị thép được xuất khẩu,
EXP(AUTO) là số lượng đơn vị máy tự động được xuất khẩu,
EXP(PLASTIC) là số lượng đơn vị nhựa được xuất khẩu,
EXP(ELECT) là số lượng đơn vị thiết bị điện tử được xuất khẩu,

Các mặt hàng có thể được xác định thẳng như thép, máy tự động, điện tử, nhựa, nhân lực, công suất máy tự động, và năng lực nhựa. Như vậy, sẽ có bảy hạn chế. Việc xây dựng toàn bộ mô hình của bài toán này là:

```
MODEL: ! Islandia Input/output model;
SETS:
        COMMO: PROD, EXP, REV, COST, MANLAB, CAP;
                CXC(COMMO, COMMO): USERATE;
ENDSETS
DATA:
        COMMO = STEEL, AUTO, PLASTIC, ELECT;
        COST = 250   300    300    50;
        REV =        500    1500   1200   300;
```

```
        MANLAB = .5          1       2        .5;
        ! Amount used of the column commodity per unit of the row
        commodity;
                USERATE=   -1       .02      .01      0
                           .8       -1       .11      .15
                           .2       .03      -1       .05
                      .01  .01      .05      -1;
        MANPOWER = 830000;
        CAP = 999999 650000 60000 999999;
    ENDDATA
    [PROFIT] MAX=@SUM(COMMO: REV * EXP - PROD * COST);
        @FOR( COMMO( I):
    [ NETUSE] ! Net use must equal = 0;
        EXP(I) + @SUM(COMMO(J):
        USERATE(J,I)* PROD(J)) = 0;
    [CAPLIM] PROD( I) <= CAP( I));
    [MANLIM] @SUM(COMMO:PROD*MANLAB)<MANPOWER;
    END
```

Chú ý là, mô hình này có các tính năng của dòng chảy Leontief. Cụ thể, mỗi biến quyết định có ít nhất một hệ số hạn chế mang dấu âm.

Giải pháp của mô hình này là:

Variable	Value	Reduced Cost
PROD(STEEL)	393958.3	0.0000000
PROD(AUTO)	475833.3	0.0000000
PROD(PLASTIC)	60000.0	0.0000000
PROD(ELECT)	74375.0	0.0000000
EXP(STEEL)	547.9167	0.0000000
EXP(AUTO)	465410.4	0.0000000
EXP(PLASTIC)	0.0	2096.875
EXP(ELECT)	0.0	121.8750

Row	Slack or Surplus	Dual Price
PROFIT	0.4354312E+09	1.000000
NETUSE(STEEL)	0.0000000	500.0000
CAPLIM(STEEL)	606040.7	0.0000000
NETUSE(AUTO)	0.0000000	1500.000
CAPLIM(AUTO)	174166.7	0.0000000
NETUSE(PLASTIC)	0.0000000	3296.875
CAPLIM(PLASTIC)	0.0000000	2082.656
NETUSE(ELECT)	0.0000000	421.8750
CAPLIM(ELECT)	925624.0	0.0000000
MANLIM	0.0000000	374.0625

Giải pháp này chỉ ra cách tốt nhất để bán thép, máy tự động, điện tử, nhựa, tài nguyên của Islandia và nguồn lực là ở dạng của máy tự động.

Bài toán này sẽ phù hợp với định dạng đầu vào - đầu ra của mô hình Leontief cổ điển nếu, thay vì tối đa hóa lợi nhuận, mức mục tiêu đã đặt ra cho xuất khẩu (hoặc tiêu thụ) của thép, máy tự động, và nhựa. Vấn đề sau đó sẽ để xác định mức độ sản xuất cần thiết để hỗ trợ các mức xuất khẩu / tiêu thụ nhất định. Trong trường hợp này, hàm mục tiêu là không thích hợp.

Một khái quát tự nhiên là để cho phép thay thế công nghệ sản xuất hàng hóa khác. Những công nghệ này có thể khác nhau tương ứng với mức độ cơ giới hóa hoặc hình thức tiêu thụ năng lượng (ví dụ, khí đốt, than đá, hoặc thủy điện).

8.9.7. Biểu đồ hoạt động (Biểu đồ nguồn lực hay tài nguyên)

Phương pháp đồ họa để mô tả một mô hình có thể được mở rộng cho mô hình QHTT tùy ý. Giá cả của một trong những nguồn lực phải trả tiền để đại diện cho một đồ họa QHTT chung là một trong những thành phần phải giới thiệu bổ sung vào sơ đồ mạng. Có hai loại thành phần trong sơ đồ như vậy: (1) hoạt động, tương ứng với các biến và được biểu thị bởi một hình vuông, và (2) nguồn tài nguyên, tương ứng với các hạn chế và được biểu hiện bằng một vòng tròn. Mỗi ràng buộc có thể được xem như tương ứng với một số hàng hóa, và trong lời nói, như nói rằng "(sử dụng hàng hóa) <= (nguồn của hàng hóa)". Đường mũi tên đến một hình vuông tương ứng với các nguồn tài nguyên, hàng hóa, hoặc ràng buộc mà các biến có một sự tương tác. Đường mũi tên đến một vòng tròn tương ứng với các hoạt động hay các biến quyết định mà ràng buộc có một sự tương tác.

Ví dụ: Tính toán tổng hợp của người nông dân

Một nông dân có 120 mẫu Anh có thể được sử dụng để trồng lúa mì hoặc ngô. Năng suất là 55 giạ lúa mì hoặc 95 giạ ngô mỗi mẫu Anh mỗi năm. Bất kỳ phần nhỏ trong số 120 mẫu Anh có thể được dành cho việc trồng lúa mì hoặc ngô. Yêu cầu lao động là 4 giờ mỗi mẫu Anh mỗi năm, cộng với 0.15 giờ mỗi giạ lúa mì, và 0.70 giờ mỗi giạ ngô. Chi phí phân bón, hạt giống, vv, là 20 cent cho mỗi giạ lúa mì sản xuất và 12 cent cho mỗi giạ ngô sản xuất. Lúa mì có thể được bán với giá 1.75 USD mỗi giạ và ngô với giá 0.95 USD mỗi giạ. Lúa mì có thể được mua với giá 2.50 USD mỗi giạ và ngô với giá 1.50 USD mỗi giạ.

Ngoài ra, nông dân có thể nuôi lợn và / hoặc gia cầm. Người nông dân bán lợn, gia cầm khi nuôi chúng được một năm. Một con lợn bán với giá 40 USD. Ông nhốt gia cầm vào trong các lồng. Một lồng gia cầm ông bán được với giá 40 USD. Nuôi một con lợn đòi hỏi chi phí 25 giạ lúa mì hoặc 20 giạ ngô, cộng với 25 giờ công lao động và 25 feet vuông không gian sàn. Một chuồng gia cầm đòi hỏi 25 giạ ngô hoặc 10 giạ lúa mì, cộng với 40 giờ công lao động và 15 feet vuông không gian sàn.

Người nông dân này có 10000 feet vuông không gian sàn. Ông đã có sẵn mỗi năm 2000 giờ công lao động của mình và 2000 giờ công của gia đình mình. Ông có thể thuê lao động ở mức 1.50 USD mỗi giờ. Tuy nhiên, cho mỗi giờ công lao động thuê, người nông dân phải tốn 0.15 giờ công để giám sát. Bao nhiêu đất đai nên được dành cho ngô và lúa mì, và bao nhiêu con lợn và / hoặc gia cầm phải được sản xuất để tối đa hóa lợi nhuận của người nông dân? Vấn đề này được dựa trên một ví dụ trong Hadley (1962).

Bạn có thể tìm thấy nó thuận tiện để sử dụng các biến sau đây cho vấn đề này:

Bảng 8.10

WH	Lúa mì thu hoạch (số giạ) Wheat harvested (in bushels)
CH	Ngô thu hoạch (số giạ) Corn harvested (in bushels)
PH	Lợn được nuôi và bán (số con) Pigs raised and sold
HS	Gà mái nuôi và bán (số lồng) Hens raised and sold (number of coops)
LB	Lao động thuê (số giờ) Labor hired (in hours)
WS	Lúa mì được bán ra thị trường (số giạ) Wheat marketed or sold (in bushels)
CS	Ngô được bán ra thị trường (số giạ) Corn marketed or sold (in bushels)
CH	Ngô dùng để nuôi gà mái (số giạ) Corn used to feed hens (in bushels)
WH	Lúa mì dùng để nuôi gà mái (số giạ) Wheat used to feed hens (in bushels)
CP	Ngô dùng để nuôi lợn (số giạ) Corn used to feed pigs (in bushels)
WP	Lúa mì dùng để nuôi gà mái (số giạ) Wheat used to feed pigs (in bushels)
CB	Ngô mua (số giạ) Corn bought (in bushels)
WB	Lúa mì mua (số giạ) Wheat bought (in bushels)

Biểu đồ hoạt động – nguồn lực của vấn đề trên được chỉ ra trong hình 8.9.

Một vài ghi chú về biểu đồ hoạt động – nguồn lực là:

Mỗi hình chữ nhật trong biểu đồ tương ứng với một biến quyết định trong mô hình.

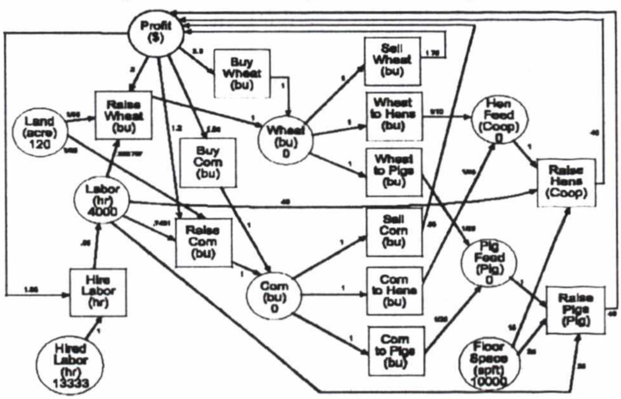
Mỗi vòng tròn trong biểu đồ tương ứng với một hạn chế hoặc mục tiêu.

Mỗi mũi tên trong biểu đồ tương ứng với một hệ số trong mô hình.

Kết hợp với mỗi vòng tròn hoặc hình chữ nhật là một đơn vị đo lường (ví dụ, giờ hoặc giạ).

Các đơn vị hoặc kích thước của mỗi mũi tên là:

"Số đơn vị của vòng tròn" trên một "đơn vị của hình chữ nhật."



Hình 8.9. Biểu đồ hoạt động – nguồn lực

8.9.8. Biểu đồ hình cây bao trùm (Spanning Trees)

Một mạng lưới liên quan đến vấn đề đơn giản khác nhưng quan trọng là vấn đề cây bao trùm. Nó phát sinh, ví dụ, trong việc thiết lập các tiện ích như đường cáp điện, đường giao thông, và hệ thống cống rãnh để cung cấp dịch vụ cho các ngôi nhà trong khu vực mới xây dựng. Cho một tập các ngôi nhà được kết nối, chúng ta muốn tìm một mạng lưới với chi phí tối thiểu, sao cho, mọi gia đình được kết nối vào mạng. Một xấp xỉ với chi phí hợp lý của mạng là tổng các chi phí của các vòng cung trong mạng. Nếu vòng cung có chi phí tích cực, thì một sự phản ánh ít nên thuyết phục bạn mạng lưới chi phí tối thiểu không chứa các vòng lặp (ví dụ, đối với bất kỳ hai nút (hoặc nhà) trên mạng, có chính xác một con đường kết nối chúng). Một mạng như vậy được gọi là một cây bao trùm.

Một thuật toán đơn giản để tìm kiếm một chi phí tối thiểu của cây bao trùm, xem Kruskal (1956):

1) Thiết lập Y = {2, 3, 4 ... n} (tức là thiết lập các nút vẫn chưa được kết nối). A = {1} (tức là thiết lập các nút đã được kết nối). Chúng ta tự ý có thể định nghĩa nút 1 như là gốc của cây.

2) Nếu Y là trống rỗng, thì chúng ta đang thực hiện,

3) Nếu Y không rỗng, thì ta tìm thấy vòng cung ngắn nhất (i, j) như vậy là i có trong A và j là trong Y.

4) Bổ sung vòng cung (i, j) vào mạng và đặt A = A + j, Y = Y - j.

5) Lại quay lại bước (2).

Bởi vì các thuật toán ở trên là đơn giản và hiệu quả, không cần thiết phải sử dụng bài toán QHTT để giải quyết vấn đề chi phí tối thiểu của cây bao trùm. Trong thực tế, mô hình bài toán cây bao trùm với chi phí tối thiểu nếu xem như một bài toán QHTT thì rất tẻ nhạt.

Sau đây minh họa một mô hình bài toán cho một cây bao trùm trong LINGO. Mô hình này không giải quyết nó một cách rõ ràng như trên, nhưng chỉ giải quyết nó như là một chương trình số nguyên đơn giản:

```
MODEL: ! (MNSPTREE);
! Cho một tập các nút và khoảng cách giữa mỗi cặp, tìm tổng các khoảng
  cách ngắn nhất liên kết trên mạng để kết nối tất cả các nút. Đây là bài
  toán cây cổ điển kéo dài tối thiểu (MST – Minimal Spanning Tree);
SETS:
      CITY: LVL;
      ! LVL( I) = level of city I in tree. LVL( 1) = 0;
      LINK( CITY, CITY):
      DIST, ! The distance matrix;
      X; ! X( I,J) = 1 if we use link I, J;
ENDSETS
!  This model finds the minimum cost network connecting Atlanta,
   Chicago, Cincinnati, Houston, LA, and Montreal so that messages can
   be sent from Atlanta (base) to all other cities;
DATA:
      CITY= ATL  CHI  CIN  HOU  LAX  MON;
      ! Distance matrix need not be symmetric. City 1 is base;
      DIST = 0    702  454   842  2396 1196 ! from Atlanta;
             702    0  324  1093  2136  764 ! from Chicago;
             454  324    0  1137  2180  798 ! from Cinci;
             842 1093 1137     0  1616 1857 ! from Houston;
            2396 2136 2180  1616     0 2900 ! from LA;
            1196  764  798  1857  2900    0; ! from Montreal;
ENDDATA
!-------------------------------------------;
!The model size: Warning, may be slow for N > 8;
N = @SIZE( CITY);
!The objective is to minimize total dist. of links;
MIN = @SUM( LINK: DIST * X);
!For city K, except the base, ... ;
@FOR( CITY( K)| K #GT# 1: ! It must be entered;
```

```
@SUM( CITY( I)| I #NE# K: X( I, K)) = 1;
 !If there is a link from J-K, then
    LVL(K)=LVL(J)+1.
   Note:These are not very powerful for
    large problems;
  @FOR( CITY( J)| J #NE# K:
  LVL( K) >= LVL( J) + X( J, K)
             -( N - 2) * ( 1 - X( J, K))
            + ( N - 3) * X( K, J)));
  LVL( 1) = 0; ! City 1 has level 0;
!There must be an arc out of city 1;
@SUM( CITY( J)| J #GT# 1: X( 1, J)) >= 1;
!Make the X's 0/1;
@FOR( LINK: @BIN( X));
!The level of a city except the base is at least 1 but no more than N-1,
and is 1 if link to the base;
@FOR( CITY( K)| K #GT# 1:
@BND( 1, LVL( K), 999999);
LVL( K) <= N - 1 - ( N - 2) * X( 1,
K));
END
```

Lời giải tối ưu của mô hình bài toán này là:

Optimal solution found at step: 16

Objective value: 4000.000

Variable	Value	Reduced Cost
N	6.000000	0.0000000
LVL(CHI)	2.000000	0.0000000
LVL(CIN)	1.000000	0.0000000
LVL(HOU)	1.000000	0.0000000
LVL(LAX)	2.000000	0.0000000
LVL(MON)	3.000000	0.0000000
X(ATL, CIN)	1.000000	454.0000
X(ATL, HOU)	1.000000	842.0000
X(CHI, MON)	1.000000	764.0000
X(CIN, CHI)	1.000000	324.0000
X(HOU, LAX)	1.000000	1616.000

Giải pháp này cho thấy Atlanta nên kết nối với Cincinnati và Houston. Houston, lần lượt kết nối với LA. Cincinnati kết nối với Chicago, và Chicago kết nối với Montreal.

```
SETS:
       ALLNODE : U;
       ! U( I) = level of node I in the tree;
                ! U( 1) = 0;
        MUSTNOD( ALLNODE); ! The subset of nodes that must be
connected;
        LINK( ALLNODE, ALLNODE):
                  DIST, ! The distance matrix;
                  X;    ! X( I, J) = 1 if we use link I, J;
ENDSETS
DATA:
ALLNODE= ! Distance matrix need not be symmetric;
      A   B   C   D    E   F   G   H   I   J   K;
 DIST =0  14 999 999  999   4 999 999 999 999 999
      14   0 999 999  999 999 999   3 999 999 999
     999 999   0   9  999 999 999 999   2  99 999
     999 999   9   0  999 999 999 999 999   3   6
     999 999 999 999    0 999 999   5 999 999   3
       4 999 999 999  999   0 999 999   3 999 999
     999 999 999 999  999 999   0   2 999   3 999
     999   3 999 999    5 999   2   0 999 999 999
     999 999   2 999  999   3 999 999   0   8 999
     999 999 999   3  999 999   3 999   8   0 999
     999 999 999   6    3 999 999 999 999 999   0
! The subset of nodes that must be connected. The first node must be a
  must-do node;
MUSTNOD = A  B  C  D  E;
ENDDATA
!---------------------------------------------;
! The model size: Warning, may be slow for N > 8;
N = @SIZE( ALLNODE);
! Objective is minimize total distance of links;
MIN = @SUM( LINK: DIST * X);
! For each must-do node K, except the base, ... ;
@FOR( MUSTNOD( K)| K #GT# 1:
! It must be entered;
@SUM( ALLNODE( I)| I #NE# K: X( I, K)) = 1;
! Force U(J)=number arcs between node J and node 1. Note: This is not
  very strong for large problems;
@FOR( ALLNODE( J)| J #GT# 1 #AND# J #NE# K:
   U( J) >= U( K) + X ( K, J) -
   ( N - 2) * ( 1 - X( K, J)) +
```