# Applications of WinQSB

László Losonczi

University of Debreceni, Faculty of Economics and Business Administration

Debrecen, 2012/13, 2nd semester

# Software

WinQSB= **Win**dows based (**Q**uantitative **S**ystem for **B**usiness
`http://www.econ.unideb.hu/sites/download/WinQSB.zip`
WinQSB has **19 application modules**. Here is an overview of each application. The user who is familiar with these applications will be able to go directly to the application module and will find the use of the software self-explanatory. Data input can be made in spreadsheet format.

# 1. Linear programming (LP) and integer linear programming (ILP)=Lineáris és egész (értékű) lineáris programozás

This programming module solves linear programming (LP) and integer linear programming (ILP) problems. Both LP and ILP problems have one linear objective function and set of linear constraints. Decision variables for LP problems are continuous (that is not an integer) and usually non-negative. Decision variables for ILP problems are either integer or binary (0 or 1), or one can have mixed integer linear programming problems where the some decision variables could be continuous, some could be binary and others could be integer. Decision variables may be bounded with limited values. The LP module uses simplex and graphical methods for solving problems. If the LP problem consists of two decision variables, one can use either of the methods to solve it. A LP problem with more than two decision variables is solved using the simplex method. The ILP module uses a branch-and-bound method for solving problems.

# 2. Linear goal programming (GP) and integer linear goal programming (IGP)=Lineáris és egész (értékű) lineáris célprogramozás

This program solves Goal programming and Integer Goal Programming problems where you have more than one linear objective to be satisfied and have a limited number of linear constraints. The goals are either prioritized or ordered. The decision variable may have any continuous value or an integer value or may have binary (0 or 1) values. One can get a graphic solution (for two decision variables only), a final solution or a step-by-step solution, as well as sensitivity analysis.

# 3. Quadratic programming (QP) and integer quadratic programming (IQP)=Kvadratikus és egész (értékű) kvadratikus programozás

This module solves Quadratic programming and Integer Quadratic Programming problems, as it has a quadratic objective function and a limited number of linear constraints. Decision variables may be bounded by certain values. One can get a graphic solution (for two decision variables), a final solution or step-by-step solution, as well as a sensitivity analysis.

# 4. Network modeling (NET)=Hálózat modellezés

This module solves network problems such as capacitated network flow (transshipment), transportation, assignment, maximal flow, minimal spanning tree, shortest path and traveling salesperson problems. A network model includes nodes and arcs/links (connections).

# 5. Nonlinear programming (NLP)=Nemlineáris programozás

This module solves a nonlinear objective function with linear and/or nonlinear constraints. The decision variables may be constrained or unconstrained. The NLP can be classified as an unconstrained single variable problem, an unconstrained multiple variables problem and a constrained problem, employing different techniques to solve them.

# 6. Dynamic programming (DP)=Dinamikus programozás

Dynamic programming is a mathematical technique for making a sequence of interrelated decisions. Each problem tends to be unique. This module handles three typical dynamic programming problems: knapsack, stagecoach, and production and inventory scheduling.

# 7. PERT/CPM

This program is for project management. One can use either Program Evaluation and Review Technique (PERT) or Critical Path Method (CAM) or both. A project consists of activities and precedence relations. It will identify critical activities, slacks available for other activities and the completion time of project. It also develops bar (Gantt) charts for activities for visual monitoring.

# 8. Queuing analysis=Sorbaállási analízis

This program evaluates a single stage queuing/waiting line system. It allows the user to select from 15 different probability distributions, including Monte Carlo simulation, for inter-arrival service time and arrival batch size. Output constitutes performance measurements of the queuing system, as well as a cost/benefit analysis.

# 9. Queuing system simulation (QSS)=Sorbaállási rendszer szimulálás

This module performs discrete event simulation of single queuing and multiple queuing systems. The input requirements are customer arrival population, number of servers, queues, and/or garbage collectors (customer leaving the system before completing the service). The output constitutes performance measurements of the queuing system in both tabular form and graphic form.

# 10. Inventory theory and systems (ITS)=Leltár elmélet és rendszerek

This program solves and evaluates inventory control systems, including the conventional EOQ model, quantity discount model, stochastic inventory model, Monte Carlo simulation of inventory control system and the single period model.

# 11. Forecasting (FC)=Előrejelzési modellek

This module provides eleven different forecasting models. Output includes forecast, tracking signal and error measurements. The data and forecast are also displayed in graphical form.

# 12. Decision analysis (DA)=Döntésanalízis

This module solves four decision problems: Bayesian, decision tree, payoff tables and zero-sum game theory (game play and Monte Carlo simulation).

# 13. Markov process (MKP)=Markov folyamat

A system exists in different conditions (states). Over time, the system will move from one state to another state. The Markov process will give a probability of going from one state to another state. A typical example could be brand switching by a consumer. Here the module will solve for steady state probabilities and analyze total cost or return.

# 14. Quality control charts (QCC)=Minőségellenőrzési diagrammok

This module performs statistical analysis and constructs quality control charts. It will construct 21 different control charts, including X-bar, R chart, p chart, and C chart. It also performs process capability analysis. Output is displayed in both table form and graph form.

# 15. Acceptance sampling analysis (ASA)=Átvételi mintavétel analízis

This module develops and analyzes acceptance-sampling plans for attribute and variable quality characteristics, such as single sampling, multiple sampling etc. It will construct OC, AOQ, ATI; ASN cost curves and can perform what-if analyses.

# 16. Job scheduling (JOB)=Munka és folyamat tervezés

This module solves scheduling problems for a job shop and a flow shop. There are 15 priority rules available for job shop scheduling, including the best solution based on selected criterion. Seven popular heuristics are available for flow shop scheduling, including the best solution based on selected criterion. Output can be viewed both in tabular form and on a Gantt chart.

# 17. Aggregate planning (AP)=Összesített (aggregált) tervezés

Aggregate planning deals with capacity planning and production schedule to meet demand requirements for the intermediate planning horizon. Typical decisions are aggregate production, manpower requirements and scheduling, inventory levels, subcontracting, backorder and lor lost sales.

# 18. Facility location and layout (FLL)=Létesítmény helyének és elrendezésének optimalizálása

This module evaluates the facility location for a two or three-dimensional pattern (plant and/or warehouse), facility design for functional Gob shop) layout and production line (flow shop). The facility location finds the location, which minimizes the weighted distance. Functional facility design is based on modified CRAFT algorithm. For flow shop design (line balancing), three different algorithms are available.

# 19. Material requirement planning (MRP)=Anyagszükséglet tervezés

This module addresses the issues related to material requirement planning (MRP) for production planning. Based on final demand requirements, both in terms of how many and when products will be delivered to customers, the MRP method will determine net requirements, planned orders, and projected inventory for materials and components items. This module will perform capacity analysis, cost analysis, and inventory analysis.

# Using WinQSB

This section describes the **common features** of all 19 modules of WinQSB.
After starting WinQSB, **select the application** you want to execute. You will see the screen with a tool bar consisting of file and help . In the file pull down menu, you will see the following options:

New problem

Load problem (*previously saved problems*)

Exit

Once the problem is entered, the tools bar will appear (More discussion of this later on).

# File menu

The **file menu** includes the following commands:

New Problem: to start a new problem

Load Problem: to open and load a previously saved problem

Close Problem: to close the current problem

Save Problem: to save the current problem with the current file name

Save Problem As: to save the current problem under a new name

Print Problem: to print the problem

Print Font: to select the print font

Print Setup: to setup the print page

Exit: to exit the program

# Edit menu

The next item on the tool bar is the **Edit menu**. It has the following commands:

Cut: to copy the selected areas in the spreadsheet to the clipboard and clear the selected area

Copy: to copy the selected areas in the spreadsheet to the clipboard

Paste: to past the content of the clipboard to selected areas in the spreadsheet

Clear: to clear the selected areas in the spreadsheet

Undo: to undo the above action

Problem Name: to change the problem name

Other commands, which appear here, are applicable to the particular application you are using.

# Format menu

**Format menu:** this pull-down menu includes following commands:

Number: to change the number format for the current spreadsheet or grid

Font: to change the font for the current spreadsheet or grid

Alignment: to change the alignment for selected columns or rows of the current spreadsheet or grid

Row Height: to change the height for the selected rows of the current spreadsheet or grid

Column Width: to change the width for the selected columns of the current spreadsheet or grid.

Other commands, which appear here, are applicable to the particular application you are using (more later on).

# Solve and analyze menu

**Solve and analyze, result menu**: this pull-down menu typically includes the following commands:

Solve the Problem: to solve the problem and display the results

Solve and Display Steps: to solve and display the solution iterations step by step

**Result menu**: this includes the options to display the solution results and analyses.

# Utility and window menu

**Utility menu**: this pull-down menu includes following commands:

Calculator: pops open the Windows system calculator

Clock: display the Windows system clock

Graph/Chart: to call a general graph and chart designer

**Window menu**: this pull-down menu includes following commands:

Cascade: to cascade windows for the current problem

Tile: to tile all windows for the current problem

Arrange Icons: to arrange all windows if they are minimized to icons

# WinQSB and help menu

**WinQSB menu**: this pull-down menu includes the option to switch to another application module without shutting down the WinQSB program .

**Help menu:** this pull-down menu includes following commands:

Contents: to display the main help categories in the help file

Search for Help on: to start the search for a keyword in the help file

How to Use Help: to start the standard Windows help instruction

Help on Current window: to display the help for the current window. You can click any area of the window to display more information

About the Program: to display the short information about the program

# LP example

CMP is a cherry furniture manufacturer. Their primary products are chairs and tables. CMP has a limited supply of wood available for making chairs and tables. Each week their suppler provides them with 100 board feet of cherry wood. Each chair requires 4 board feet of wood and a table needs 8 board feet. CMP can sell all chairs it can make but only 10 tables per week. CMP has at their disposal only 120 man-hours available per week for making the furniture. It takes 6 man-hours to make a chair and 4 man-hours to make a table. CMP makes $ 20 profit for every chair it sells and makes $ 30 profit for every table sold. Here CMP wants to decide how many chairs and tables it should make to maximize total profits.

# LP example, continued

**Solution**: suppose that the number of chairs and tables produced are $x_1$ and $x_2$ respectively.

$$\begin{aligned} \text{The cherry wood constraint:} \quad & 4x_1 + 8x_2 \leq 100, \\ \text{selling constraint:} \quad & x_2 \leq 10, \\ \text{labor constraint:} \quad & 6x_1 + 4x_2 \leq 120, \\ \text{nonnegativity constraint:} \quad & x_1 \geq 0,\ x_2 \geq 0. \end{aligned}$$

The weekly profit: $20x_1 + 30x_2$ dollars should be maximal.
Summarizing:

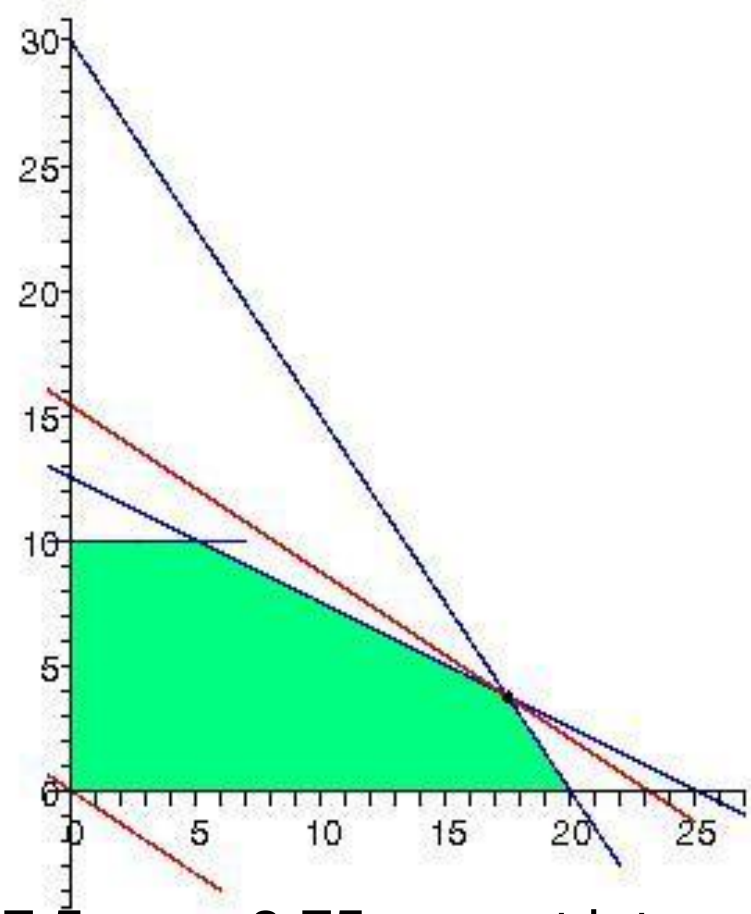
$$z = 20x_1 + \quad 30x_2 \to \text{maximum, subject to}$$

$$\begin{aligned} 4x_1 + 8x_2 &\leq 100 \\ x_2 &\leq 10 \\ 6x_1 + 4x_2 &\leq 120 \end{aligned} \qquad x_1 \geq 0, x_2 \geq 0.$$

# LP example, solution

Graphical solution: sketch on the plane the $(x_1, x_2)$ domain satisfying the constraints. The objective function with constant value of $z$ are parallel lines which have to have common points with the above domain. Hence we shift these lines parallel until we get the maximum on $z$. In our case this is $x_1 = 17.5, x_2 = 3.75, z_{max} = 462.5$. The next figure shows the domain satisfying the constraints colored by green. The parallel lines $z = 20x_1 + 30x_2$ or $x_2 = -\frac{x_1}{3} + \frac{z}{30}$ are shown by red color for $z = 0$ and $z = z_{max} = 462.5$.

# LP example, graphical solution




The solutions $x_1 = 17.5, x_2 = 3.75$ are not integers. Thus we have to find the point(s) in the green area **which have integer coordinates and for which the value of** $z = 20x_1 + 30x_2$ **is the largest**. This point is $x_1 = 17, x_2 = 4$, and $z_{max} = 460$.

# LP example:ELOAD1B.LPP, with fives variables

$$
\begin{aligned}
x_1, x_2, x_3, x_4, x_5 &\geq 0, \\
\\
x_1 + 2x_3 - 2x_4 + 3x_5 &\leq 60 \\
x_1 + 3x_2 + x_3 + x_5 &\leq 12 \\
x_2 + x_3 + x_4 &\leq 10 \\
2x_1 + 2x_3 &\leq 20 \\
\\
3x_1 + 4x_2 + 5x_3 + 3x_4 - 2x_5 &= z \rightarrow \max \quad \text{or} \quad \min
\end{aligned}
$$

# LP example: data entry

Data entry into WinQSB in matrix form:

| Variable --> | X1 | X2 | X3 | X4 | X5 | Direction | R. H. S. |
|---|---|---|---|---|---|---|---|
| Maximize | 3 | 4 | 5 | 3 | -2 | | |
| C1 | 1 | | 2 | -2 | 3 | <= | 60 |
| C2 | 1 | 3 | 1 | | 1 | <= | 12 |
| C3 | | 1 | 1 | 1 | | <= | 10 |
| C4 | 2 | | 2 | | | <= | 20 |
| LowerBound | 0 | 0 | 0 | 0 | 0 | | |
| UpperBound | M | M | M | M | M | | |
| VariableType | Continuous | Continuous | Continuous | Continuous | Continuous | | |

# LP example:solution

The table of solution:

**Combined Report for eload1b**

| | 13:07:30 | | Sunday | February | 28 | 2010 | | |
|---|---|---|---|---|---|---|---|---|
| | Decision Variable | Solution Value | Unit Cost or Profit c(j) | Total Contribution | Reduced Cost | Basis Status | Allowable Min. c(j) | Allowable Max. c(j) |
| 1 | X1 | 10,00 | 3,00 | 30,00 | 0 | basic | 2,00 | M |
| 2 | X2 | 0,67 | 4,00 | 2,67 | 0 | basic | 3,00 | 12,00 |
| 3 | X3 | 0 | 5,00 | 0 | -1,00 | at bound | -M | 6,00 |
| 4 | X4 | 9,33 | 3,00 | 28,00 | 0 | basic | 2,00 | 4,00 |
| 5 | X5 | 0 | -2,00 | 0 | -2,33 | at bound | -M | 0,33 |
| | Objective | Function | (Max.) = | 60,67 | | | | |
| | | | | | | | | |
| | Constraint | Left Hand Side | Direction | Right Hand Side | Slack or Surplus | Shadow Price | Allowable Min. RHS | Allowable Max. RHS |
| 1 | C1 | -8,67 | <= | 60,00 | 68,67 | 0 | -8,67 | M |
| 2 | C2 | 12,00 | <= | 12,00 | 0 | 0,33 | 10,00 | 40,00 |
| 3 | C3 | 10,00 | <= | 10,00 | 0 | 3,00 | 0,67 | M |
| 4 | C4 | 20,00 | <= | 20,00 | 0 | 1,33 | 0 | 24,00 |

# LP example:explanation of the solution

The **reduced cost** appears at decision variables which have (the optimal value) zero, and it shows the change of the objective function when we require positive value for that decision variable (instead requiring it to be nonnegative only). **For example at $x_3 = 0$ the reduced cost is $-1$, which means that requiring $x_3 \geq a_3 (> 0)$ instead of $x_3 \geq 0$ the objective function will (approximately) change by $(-1) \cdot a_3$.**
The **shadow price** appearing at a constraint shows that the change of the constant at the right hand side of the constraint how influences the value of the objective function. **For example at the constraint $C_3$ the shadow price is 3, this means that increasing the right hand side of $C_3$ by $b_3$ (in our case to $10 + b_3$) the value of the objective function will (approximately) change by $3b_3$ (positive sign means increase, negative decrease).**

# LP example:explanation of the solution

The **shadow price** appearing at a constraint shows that the change of the constant at the right hand side of the constraint how influences the value of the objective function. **For example at the constraint $C_3$ the shadow price is** 3**, this means that increasing the right hand side of $C_3$ by $b_3$ (in our case to** $10 + b_3$**) the value of the objective function will (approximately) change by** $3b_3$ **(positive sign means increase, negative decrease).**

# LP example:explanation of the solution

The **slack or surplus** appearing at a constraint is the difference between the two sides of that constraint (at the optimal values of the decision variables).
The upper 1-5 lines of the last two columns show the lower and upper bounds of the coefficients in that decision variable in the objective function by which there is still optimal solution of the problem.
The last 4 lines of the last two columns show the maximum and minimum values of the right hand sides of the constraints by which there is still optimal solution.

# Example for Linear Goal Programming (GP)

A company is considering three form of advertising:

1. by TV ads, cost is 3000\$, reaches 1000 new potential customers,
2. through radio, cost is 800\$, reaches 500 new potential customers,
3. in newspaper, cost is 250\$, reaches 200 new potential customers.

The goals are (in priority order)

1. Cost of advertisement $\leq$ 25000\$,
2. Reach at least 30000 new potential customers.
3. Run at least 10 TV spots.

The number TV spots, radio, and newspaper ads be $x_1, x_2$ and $x_3$ respectively. If the above goals were constraints rather than goals then:

$$\begin{aligned} \text{cost constraint:} \quad & 3000x_1 + 800x_2 + 250x_3 \leq 25000, \\ \text{new customer constraint:} \quad & 800x_1 + 500x_2 + 200x_3 \geq 30000, \\ \text{TV spots min. constraint:} \quad & x_1 \geq 10. \end{aligned}$$

If the third inequality holds then the left hand side of the first constraint $\geq 3000 \cdot 10 = 30000$ hence there is **no feasible solution** $x_1, x_2, x_3$ **satisfying the constraints.**

# Example for Linear Goal Programming (GP)

We introduce 6 new non negative variables $u_i, e_i, (i = 1, 2, 3)$ corresponding to the goals.
$u_i =$ the $i$th goal minus the value achieved (under achieving)
$e_i =$ the value achieved minus the $i$th goal (exceeds achieving)
The goals can now be written as

$$\begin{aligned}
\text{cost goal:} \quad & 3000x_1 + 800x_2 + 250x_3 + u_1 - e_1 = 25000, \\
\text{new customer goal:} \quad & 800x_1 + 500x_2 + 200x_3 + u_2 - e_2 = 30000, \\
\text{TV spots goal:} \quad & x_1 + u_3 - e_3 = 10,
\end{aligned}$$

Here, $e_1$ (the amount spent over 25000), $u_2$ (the number of potential new customers under 30,000 reached), and $u_3$ (the number of television ads below 10 aired) are called the **detrimental variables/deviations** for this problem. **The objective of a goal programming problem somehow involves minimizing the detrimental variables** $e_1, u_2, u_3$**.** Just how this is done differentiates two approaches to solving goal programming problems the non preemptive and preemptive (lexicographic) method.

# Solution of Linear Goal Programming (GP)

In the **non-preemptive approach of the solution** we give relative weights to each detrimental variable ($e_1, u_2, u_3$), e.g. if the weights are 10,5,2, then we seek the minimum of the weighted average

$$z = 10e_1 + 5u_2 + 2u_3 \to \text{minimum, subject to}$$

$$\begin{aligned} 3000x_1 + 800x_2 + 250x_3 + u_1 - e_1 &= 25000, \\ 800x_1 + 500x_2 + 200x_3 + u_2 - e_2 &= 30000, \\ x_1 + u_3 - e_3 &= 10, \\ x_1, x_2, x_3, u_1, e_1, u_2, e_2, u_3, e_3 &\geq 0. \end{aligned}$$

# Solution of Linear Goal Programming (GP)

In the **preemptive approach (or lexicographic goal)** we first try to satisfy the goal at priority level 1 (if there are more than one goal at priority level 1, then we apply the non-preemptive method for these goals). In our case we start with the LP problem

$$e_1 \to \text{minimum, subject to}$$

$$\begin{aligned} 3000x_1 + 800x_2 + 250x_3 + u_1 - e_1 &= 25000, \\ 800x_1 + 500x_2 + 200x_3 + u_2 - e_2 &= 30000, \\ x_1 + u_3 - e_3 &= 10, \\ x_1, x_2, x_3, u_1, e_1, u_2, e_2, u_3, e_3 &\geq 0. \end{aligned}$$

If the minimum value of the objective function $e_1$ is $v_1$, then we continue with the LP problem (for the priority 2 goal)

$$u_2 \to \text{minimum, subject to}$$

$$\begin{aligned} 3000x_1 + 800x_2 + 250x_3 + u_1 - v_1 &= 25000, \\ 800x_1 + 500x_2 + 200x_3 + u_2 - e_2 &= 30000, \\ x_1 + u_3 - e_3 &= 10, \\ x_1, x_2, x_3, u_1, u_2, e_2, u_3, e_3 &\geq 0. \end{aligned}$$

# Solution of Linear Goal Programming (GP)

Again, if the minimum value of the objective function $u_2$ is $v_2$, then for the goal of third priority we solve the LP problem

$$u_3 \to \text{minimum, subject to}$$

$$\begin{aligned} 3000x_1 + 800x_2 + 250x_3 + u_1 - v_1 &= 25000, \\ 800x_1 + 500x_2 + 200x_3 + v_2 - e_2 &= 30000, \\ x_1 + u_3 - e_3 &= 10, \\ x_1, x_2, x_3, u_1, e_2, u_3, e_3 &\geq 0. \end{aligned}$$

# Network Modeling

This programming module, Network Modeling, solves network problems including **transportation, assignment, shortest path, maximal flow, minimal spanning tree, traveling salesperson and capacitated network (transshipment) problems**. A network consists of nodes and connections (arcs/links). Each node may have a capacity (in case of the network flow and transportation problems). If there is a connection between two nodes, there may be a cost, profit, distance, or flow capacity associated with the connection. Based on the nature of the problem, NET solves the link or shipment to optimize the specific objective function.

# Transportation problem

The objective is to **ship goods from supply sources to demand locations at the lowest total cost**.
The transportation problem could be **balanced** (the supplies and demands are equal) or could be **unbalanced.**
CMP has production facilities in Nashville and Atlanta and markets its products in New York, Miami, and Dallas. Production capacities and demands and transport costs are given at the table below (supply demand in some units, cost in $).

| From/to | New York | Miami | Dallas | Supply |
|---|---|---|---|---|
| Nashville | 10 | 12 | 8 | 300 |
| Atlanta | 7 | 10 | 14 | 500 |
| Demand | 150 | 300 | 350 | |

How many unit need to be shipped from Nashville and Atlanta to New York, Miami, and Dallas **at the lowest total cost**.

# Transportation problem, LP model

Total supply is 800 units, total demand is 800 units hence the supply covers the demand (balanced problem). Denote the number of units shipped from Nashville to the 3 destinations by $x_{11}, x_{12}, x_{13}$ respectively, and the number of units shipped from Atlanta to the 3 destinations by $x_{21}, x_{22}, x_{23}$ respectively. Then we have to minimize the total cost being

$$z = 10x_{11} + 12x_{12} + 8x_{13} + 7x_{21} + 10x_{22} + 14x_{23}$$

subject to the constraints:

| | | |
|---|---|---|
| $x_{11} + x_{12} + x_{13}$ | $\leq 300$ | supply constraint in Nashville |
| $x_{21} + x_{22} + x_{23}$ | $\leq 500$ | supply constraint in Atlanta |
| $x_{11} + x_{21}$ | $= 150$ | demand satisfaction constraint in NY |
| $x_{12} + x_{22}$ | $= 300$ | demand satisfaction constraint in Miami |
| $x_{13} + x_{23}$ | $= 350$ | demand satisfaction constraint in Dallas |

Solution by WinQSB, either LP problem as above or by Network Modeling.(TRANSPORT.NET)

# An assignment problem

Assignment problem is a special type of network or linear programming problem where objects or assignees are being allocated to assignments on a one-to-one basis. The object or assignee can be a resource, employee, machine, parking space, time slot, plant, or player, and the assignment can be an activity, task, site, event,asset, demand, or team. Our terminology: **we assign agents to do tasks.** The problem is ”linear” because the cost function to be optimized as well as all the constraints contain only linear terms. The assignment problem is also considered as a **special type of transportation problem with unity supplies and demands** and is solved by the network simplex method.

# Example: assignment problem

**Example.** A factory unit has 4 agents (workers) $A_1$ (Tim), $A_2$ (Peter), $A_3$ (John), $A_4$ (Rudy) and they have to make 4 different type of tasks (jobs): $J_1, J_2, J_3, J_4$. The next table shows how many hours each of them needs to do the same jobs.

| From/to | $J_1$ | $J_2$ | $J_3$ | $J_4$ |
|---|---|---|---|---|
| $A_1$ (Tim) | 3 | 6 | 7 | 10 |
| $A_2$ (Peter) | 5 | 6 | 3 | 8 |
| $A_3$ (John) | 2 | 8 | 4 | 16 |
| $A_4$ (Rudy) | 8 | 6 | 5 | 9 |

Which worker should be given the task (job) $J_1, J_2, J_3, J_4$ if **all the 4 jobs should be done in minimal time and each worker can be given only one type of job.**

# Assignment problem, example

Let the variable $x_{ij}$ represents the assignment of agent $A_i$ to the job $J_j$ taking value1 if the assignment is done and 0 otherwise. This formulation allows also fractional variable values, but there is always an optimal solution where the variables take integer values. The objective function (cost)

$$z = \begin{aligned} &3x_{11}+6x_{12}+7x_{13}+10x_{14}+5x_{21}+6x_{22}+3x_{23}+8x_{24}+ \\ &2x_{31}+8x_{32}+4x_{33}+16x_{34}+8x_{41}+6x_{42}+5x_{43}+9x_{44} = \min. \end{aligned}$$

subject to the constraints:

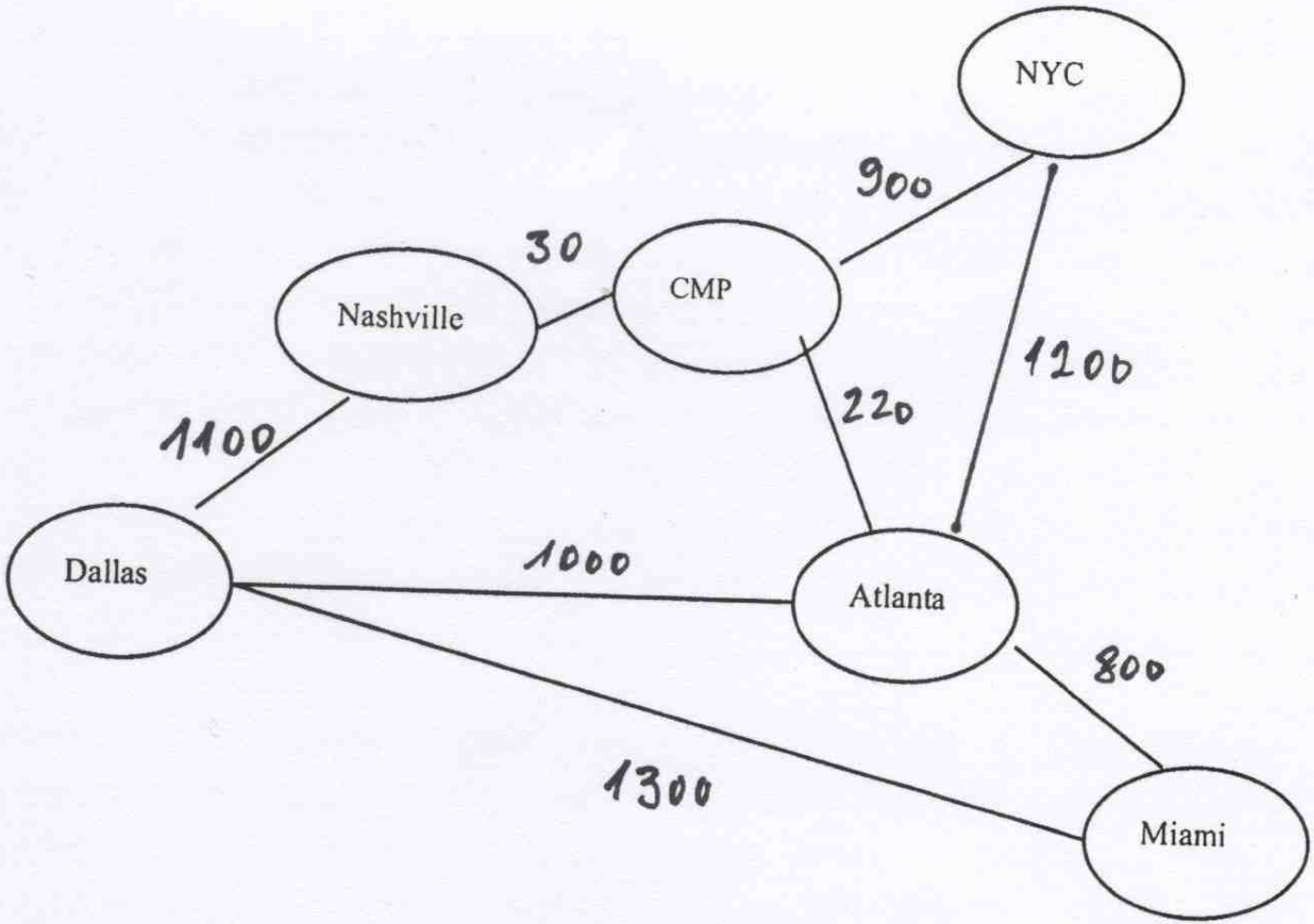
| | |
|---|---|
| $x_{11}+x_{12}+x_{13}+x_{14} = 1$ | agent $A_1$ performs one of the 4 jobs |
| $x_{21}+x_{22}+x_{23}+x_{24} = 1$ | agent $A_2$ performs one of the 4 jobs |
| $x_{31}+x_{32}+x_{33}+x_{34} = 1$ | agent $A_3$ performs one of the 4 jobs |
| $x_{41}+x_{42}+x_{43}+x_{44} = 1$ | agent $A_4$ performs one of the 4 jobs |
| $x_{11}+x_{21}+x_{31}+x_{41} = 1$ | the job $J_1$ is done by one of the 4 agents |
| $x_{12}+x_{22}+x_{32}+x_{42} = 1$ | the job $J_2$ is done by one of the 4 agents |
| $x_{13}+x_{23}+x_{33}+x_{43} = 1$ | the job $J_3$ is done by one of the 4 agents |
| $x_{14}+x_{24}+x_{34}+x_{44} = 1$ | the job $J_4$ is done by one of the 4 agents |

Solution by WinQSB, select the Network Modeling. (ASSIGN.NET)

# Shortest path problem

The shortest path problem includes a set of connected nodes where only one node is considered as origin node and only one node is considered as destination node. The objective is to determine a path of connections that minimizes the total distance from the origin to the destination. The shortest path problem is solved by a Labeling Algorithm. Let say CMP's trucks can only travel between CMP headquarters and its manufacturing plants in Nashville and Atlanta and its markets in Dallas, Miami and New York as shown in the next figure.

# Shortest path problem

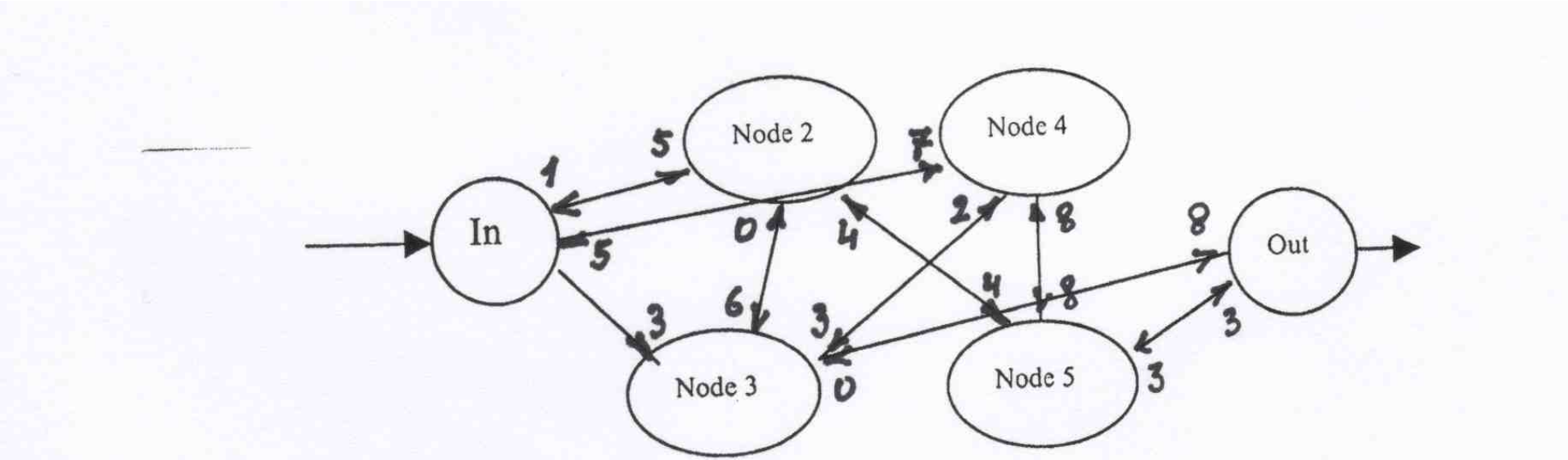


# Shortest path problem

The owner wants to know the **shortest distance from headquarters to Miami**. One would use the shortest path model to find the answer. Select Network Modeling from the WinQSB menu, and then select Shortest Path Problem from the Problem Type option. Objective Criterion is set to Minimization. Numbering the nodes are Dallas (1), Nashville (2), CMP (3), NYC (4), Atlanta (5), Miami (6). (SHORTEST.NET)

# Maximal flow problem

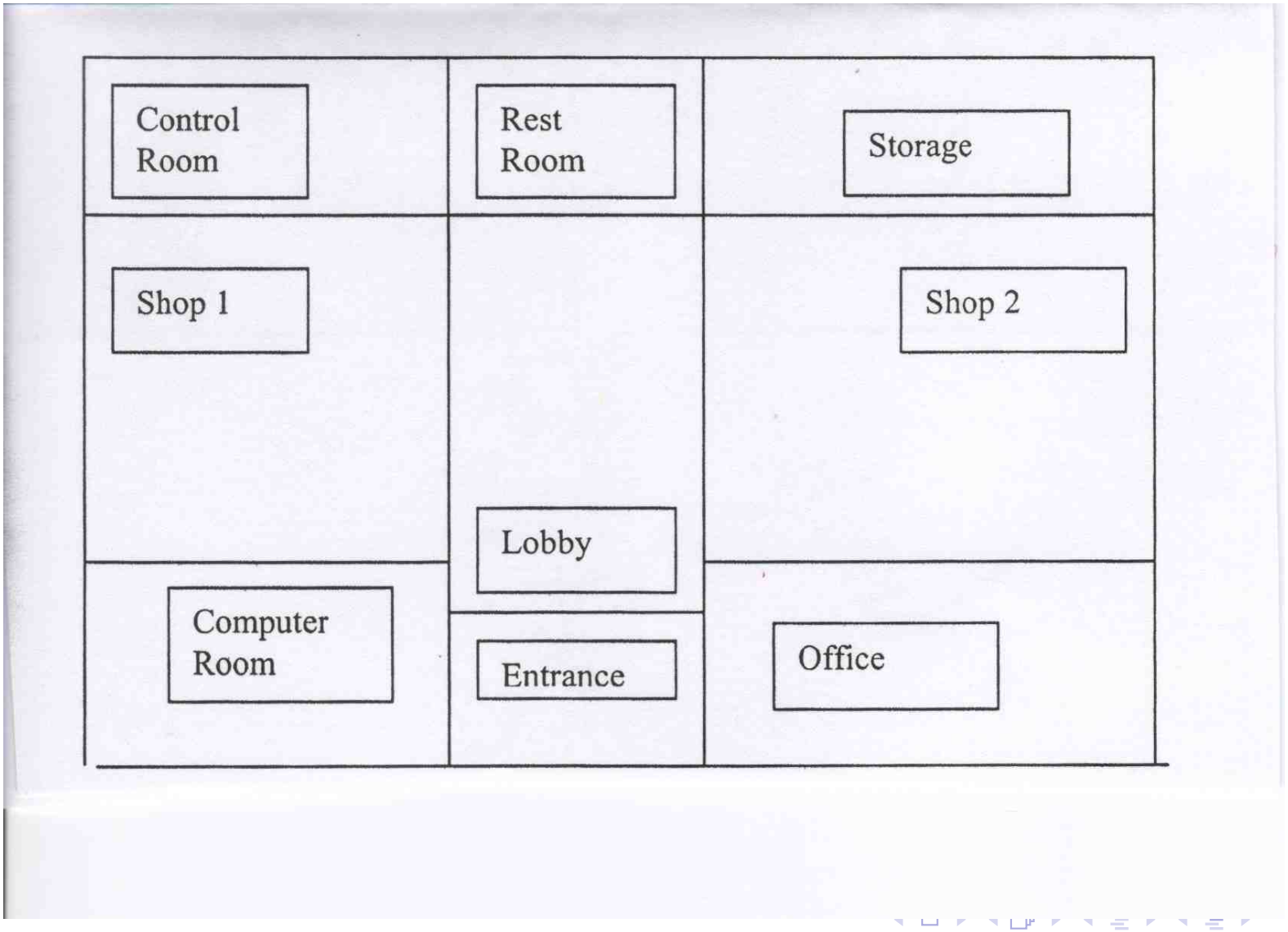
You are in charge of civilian defense of a city. You need to evacuate the city as quickly as possible. The road map for removing the citizens is shown in the next figure.




Capacities of roads in terms of the number of vehicles per minute. In WinQSB, select Network Modeling. In Problem Type, select Maximal Flow Problem. The Objective Criterion is by default Maximization. (EVACUATE.NET) Graphic solution

# Minimal spanning tree problem

CMP needs to install a sprinkler system in their office. The layout of the office shown in the next figure.

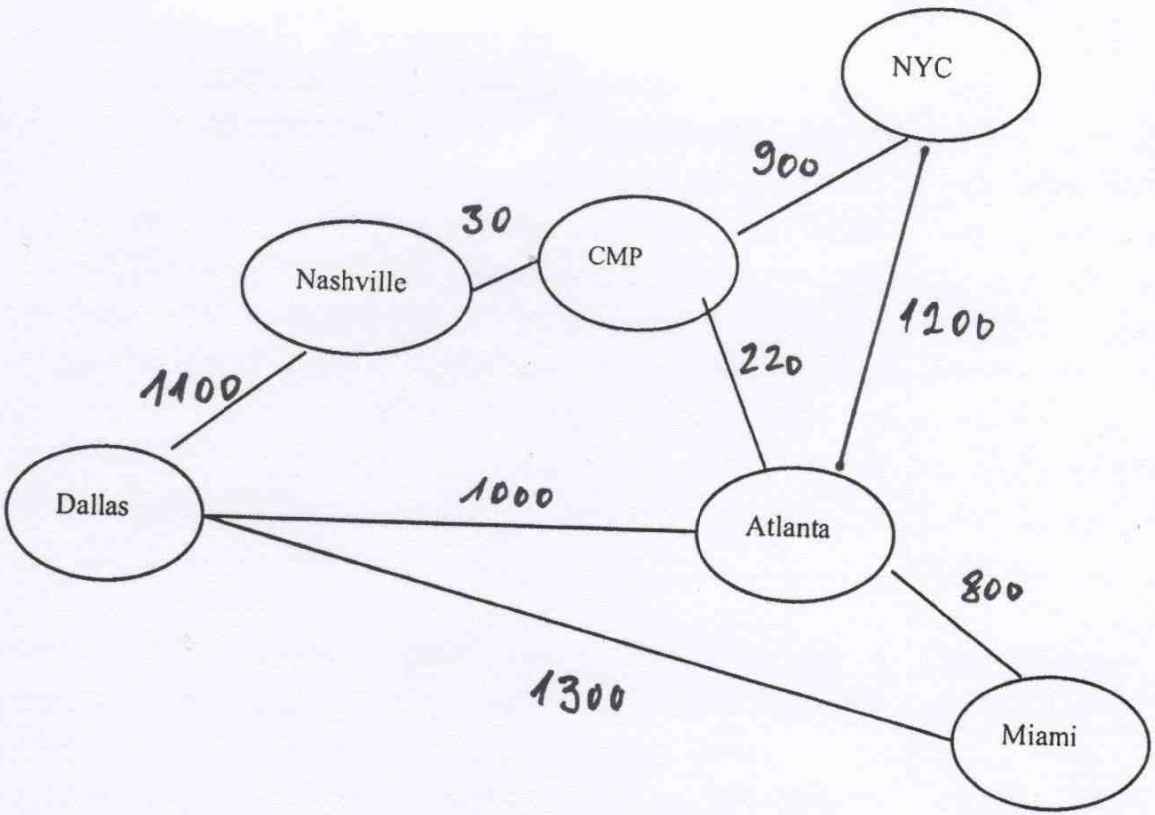


# Minimal spanning tree problem

CMP wants to minimize the total length of pipes to be installed. The minimal spanning tree procedure is to be used here. From the WinQSB menu se Network Modeling. In NET Problem Specification select Minimal Spanning Tree. The Objective Criterion by default is Minimization. Select Spreadsheet Matrix Form. Problem name is CMP Safety. There are a total of nine nodes: Entrance, Lobby, Office, Control Room, Computer Room, Shop 1, Shop 2, Restroom, and Storage. Click OK. Now you will see the spreadsheet matrix input form. The node names are edited from default to actual names. Enter the appropriate distances in the from/to cells. After entering all data, click on the Solve and Analyze button on the toolbar. The next screen will display the solution in terms of sprinkler piping connection between the locations. The total length of pipe is 507 feet. The graphical solution option gives you the graphical layout of different nodes and the piping connections between them.(SPRINKLER.NET) (symmetric arcs)

# Traveling salesman problem

CMP has sales representatives visiting between headquarters, the manufacturing facilities, and their customers. A sales representative starts the sales call from headquarters and must visit all the locations without revisiting them and then return to headquarters. This is the classical traveling salesman problem. The next figure displays the location of all the facilities and their distances.

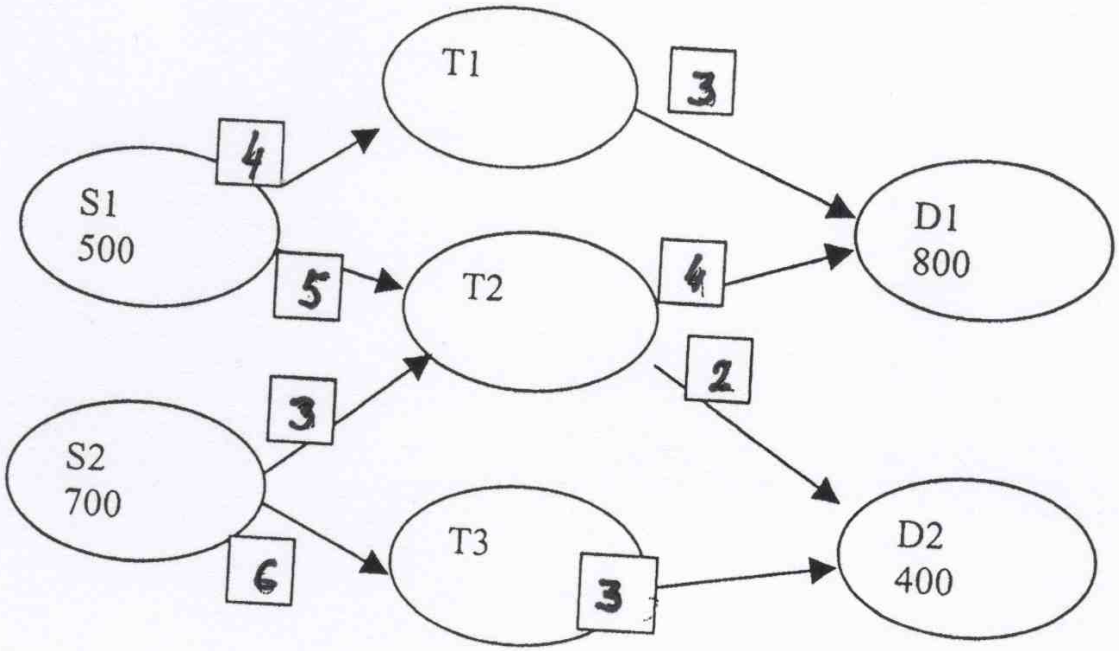


# Traveling salesman problem

In WinQSB select Network Modeling, then click on Traveling Salesman Problem. The Objective Criterion is to minimize the total distance a sales representative has to travel in visiting all of the places. We will use Spreadsheet Matrix form for data input. Enter the name of the problem in the Problem Title space. There are all together six places, hence enter Number of Nodes equal to six. Click OK. In input form edit the node variables from the Edit menu in the Toolbar and enter the names of nodes. Enter the distance data in appropriate cells. Click on Solve and Analyze. From popup menu, select appropriate solution method (here we have selected Branch and Bound Method) and click Solve. The computer will display the solution on the screen. The sales person should travel from CMP headquarters to New York and from New York to Atlanta, from Atlanta to to Miami, from Miami to Dallas, from Dallas to Nashville, from Nashville to CMP, with a total of 5330 miles traveled. There is also a graphical solution option output.

# A network flow (transshipment) problem

In the next transshipment problem there are two supply points $S_1$, $S_2$, three transshipment points $T_1$, $T_2$, $T_3$ and two demand points $D_1$, $D_2$. Supply capacities and demand requirements are shown in the circle and respective shipping costs are shown along the arrows in squares. We want to ship the goods through transshipment points to demand points at the least possible cost. The total supply = demand is 1200 u



# A network flow (transshipment) problem

Denote by $x_{11}, x_{12}, x_{13}$ the number of units shipped from S1 to T1, T2, T3 respectively, $x_{21}, x_{22}, x_{23}$ be the number of units shipped from S2 to T1, T2, T3 respectively. Further let $y_{11}, y_{12}$ be the number of units shipped from T1 to D1, D2 respectively, $y_{21}, y_{22}$ the units shipped from T2 to D1, D2 respectively, and $y_{31}, y_{32}$ the number of units shipped from T3 to D1, D2 respectively. The cost function

$$z = 4x_{11} + 5x_{12} + 3x_{22} + 6x_{23} + 3y_{11} + 4y_{21} + 2y_{22} + 3y_{32}$$

should be minimized subject to the constraints

| | | |
|---|---|---|
| $x_{11} + x_{12} + x_{13}$ | $\leq 500$ | supply constraint at S1 |
| $x_{21} + x_{22} + x_{23}$ | $\leq 700$ | supply constraint at S2 |
| $y_{11} + y_{21} + y_{31}$ | $= 800$ | demand at D1 must be satisfied |
| $y_{12} + y_{22} + y_{32}$ | $= 400$ | demand at D2 must be satisfied |
| $x_{11}$ | $= y_{11}$ | what goes in T1 also goes out |
| $x_{12} + x_{22}$ | $= y_{21} + y_{22}$ | what goes in T2 also goes out |
| $x_{23}$ | $= y_{32}$ | what goes in T3 also goes out |

and all variables $x_{ij}, y_{ki}$ should be nonnegative.

# A network flow (transshipment) problem

Click on WinQSB, and in the popup menu select Network Modeling. From there select Problem Type: Network Flow. The Objective Criterion is Minimization. Select Spreadsheet Matrix Form for data entries. Enter Problem Title: Transshipment Problem. In our problem we have seven nodes, hence enter Number of Nodes equals seven. Click OK. The screen will display a spreadsheet form for inputs with default node names. Click on EDIT on toolbar, select Node names and input appropriate node names, one to be replaced by S1 and so on and click OK. Input the data, note that blank cells represent no connections.Save it as TRSSHIPM.NET Now click on the Solve and Analyze button on the toolbar and select Solve Problem. Minimal cost is 7600 and in the solution the amount of units shipped (in various ways) will be displayed. If you want to see another form of the solution, click on result and then select graphical solution. The computer will display the graphical representation of the solution.