Graphs and Integer Programming

Case Study

Finding Shortest Path Itineraries for Security Runs

Marcello Sanguineti

marcello.sanguineti@unige.it

DIBRIS - University of Genova

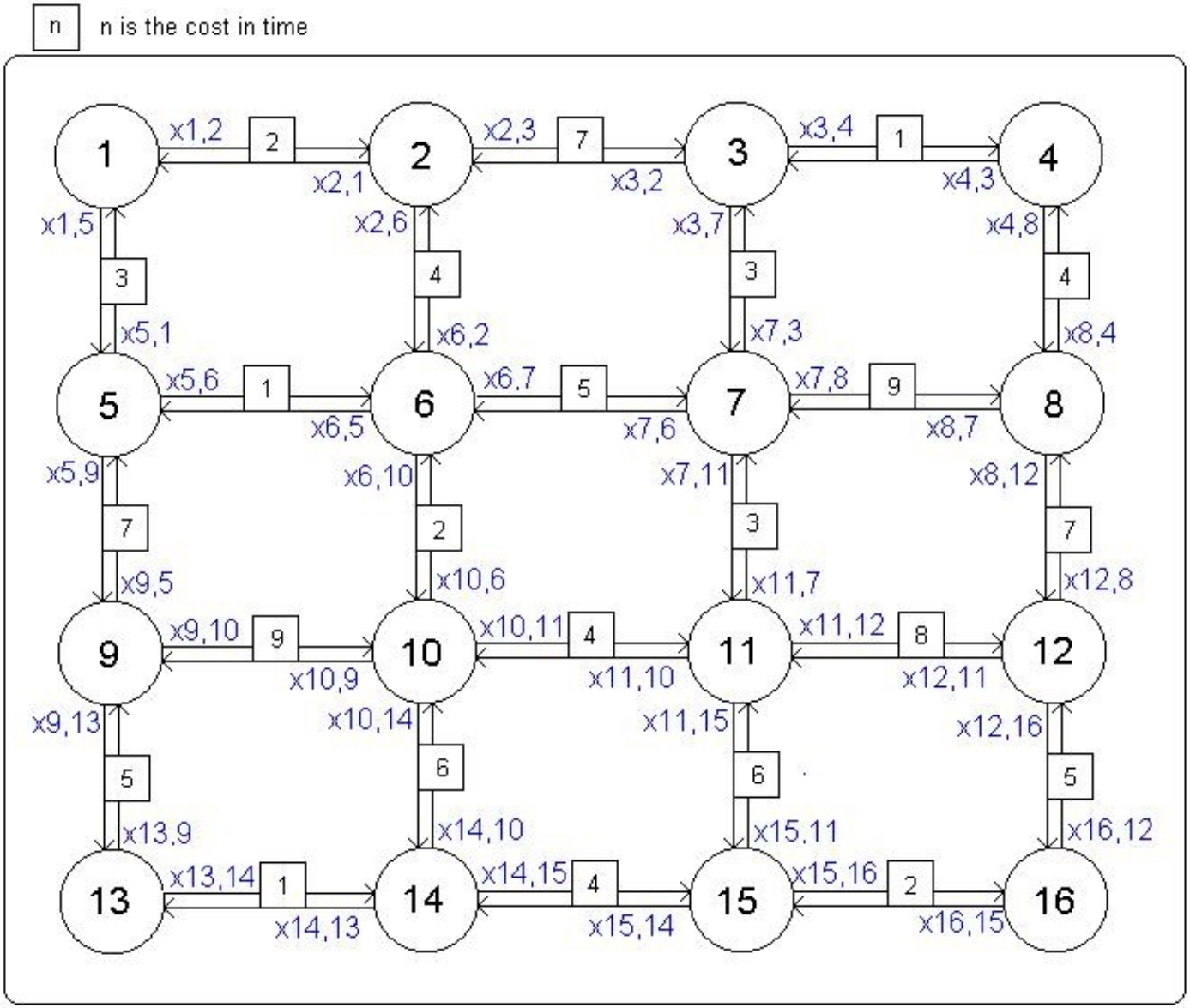
Security Run Problem I

- In Autonomous Mobile Robotics (AMR), robots are confronted regularly with the problem of finding the best path between a source location and a destination location.

- This is an optimization concern, since the robot wants to minimize its cost in time or in energy while achieving its goal.

- Security run problem. A robot is located in a (source) room of a building. We want the robot to find the shortest path (in terms of time) to reach a destination room, visiting a set of rooms, to verify the presence of burglars.

Security Problem II



Security Problem II

- Consider the map in the figure (the reader can imagine is a building floor's viewed from above):
  - the nodes symbolize the possible starting and arriving locations allowed (source and destination);
  - the edges represent the possible paths that the robot can take to travel from node to node (location to location);
  - the values in the boxes attached to each pair of edges indicate the cost to travel along them; in this case, we assume the units are minutes.

- Instance of the problem: among all the possible paths between node 1 (source) and node 7 (destination), passing through nodes 13, 11 and 8, find the one with associated minimum cost.
  The robot must find the shortest path among these five nodes.

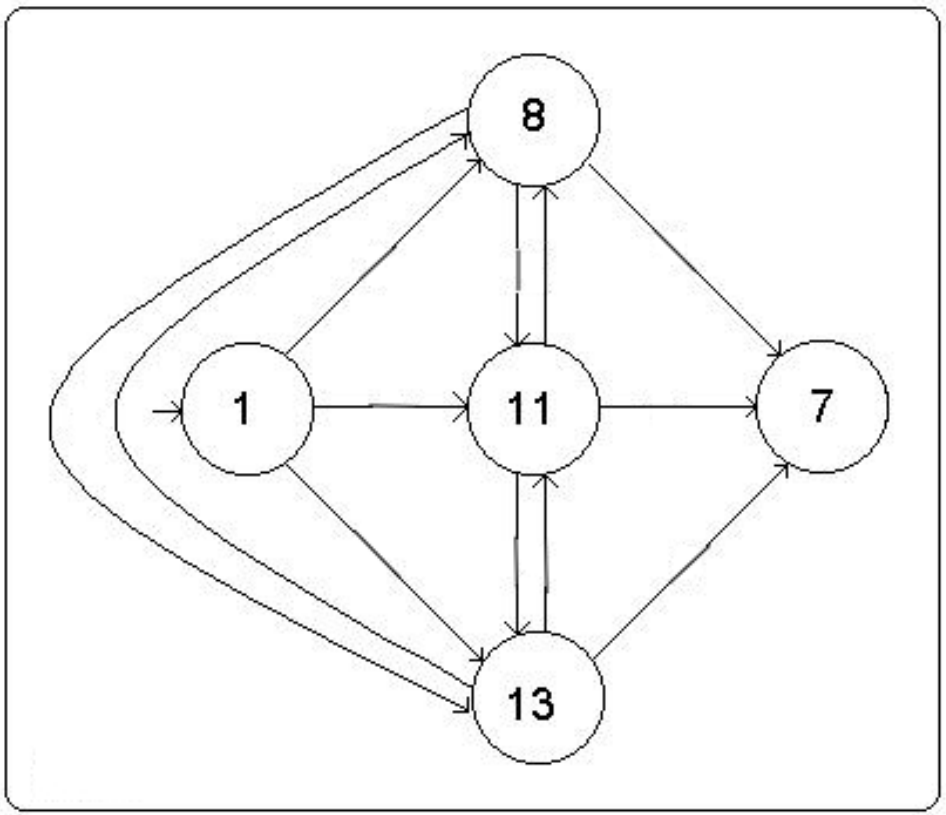
Security Problem III

- Remarks:
  1. The order in which nodes 13, 11 and 8 are visited is object of the optimisation.
  2. This is not a trivial shortest path problem, where you have only to find the optimal route between a source and a destination.
  3. The problem is similar to Travelling Salesman Problem (TSP), but in this case the robot must visit only 3 "rooms". In the TSP all the 16 nodes in the graph must be visited avoiding subtours.
  4. In the present work as TSP we mean the problem of finding an optimal Hamiltonian path between two edges, while in the literature the path must be a cyle.

- This security run scenario shows that Dijkstra's Shortest Path (SP) algorithm alone is not enough to solve the problem under consideration.

Solution Strategy

- We can split the problem into two subproblems:
  1. Find the optimal paths between every pair of nodes among 1, 13, 11, 8, 7 (remebering that 1 is the source and 7 is the destination);
  2. Find the optimal Hamiltonian path between 1 and 7 considering only the nodes 13, 11 and 8.



Shortest Path as ILP

$$\text{Minimise} \sum_{\text{all edges}} c_{ij} x_{ij}$$

$$\text{such that: } \sum_{j,k} x_{ij} - x_{ki} = b_i \text{ (for each node } i \text{ in the network)}$$

$$x_{ij} \in \{0, 1\} \text{ (for each edge in the network)}$$

- $x_{ij} = 1$ if the robot takes edge joining node i to node j, 0 otherwise.
- c_{ij} represents the cost (expressed in minutes) of travelling the edge from i to j.
- b_i represents the net supply. If node i is a source (starting node), it has value 1. If node i is a destination, it has value -1. If i is a transitory node, it has value 0 (because what enters must leave).

Example: shortest path between 1 and 8

$$
\begin{aligned}
\min z = \; & 2x_{1,2} + 2x_{2,1} + 3x_{1,5} + 3x_{5,1} + \\
& 7x_{2,3} + 7x_{3,2} + 4x_{2,6} + 4x_{6,2} + \\
& 1x_{3,4} + 1x_{4,3} + 3x_{3,7} + 3x_{7,3} + \\
& 4x_{4,8} + 4x_{8,4} + 1x_{5,6} + 1x_{6,5} + \\
& 7x_{5,9} + 7x_{9,5} + 5x_{6,7} + 5x_{7,6} + \\
& 2x_{6,10} + 2x_{10,6} + 9x_{7,8} + 9x_{8,7} + \\
& 3x_{7,11} + 3x_{11,7} + 7x_{8,12} + 7x_{12,8} + \\
& 9x_{9,10} + 9x_{10,9} + 5x_{9,13} + 5x_{13,9} + \\
& 4x_{10,11} + 4x_{11,10} + 6x_{10,14} + 6x_{14,10} + \\
& 8x_{11,12} + 8x_{12,11} + 6x_{11,15} + 6x_{15,11} + \\
& 5x_{12,16} + 5x_{16,12} + 1x_{13,14} + 1x_{14,13} + \\
& 4x_{14,15} + 4x_{15,14} + 2x_{15,16} + 2x_{16,15}
\end{aligned}
$$

$$
\begin{aligned}
1) \;& x_{1,2} - x_{2,1} + x_{1,5} - x_{5,1} = 1 \\
2) \;& -x_{1,2} + x_{2,1} + x_{2,3} - x_{3,2} + x_{2,6} - x_{6,2} = 0 \\
3) \;& -x_{2,3} + x_{3,2} + x_{3,4} - x_{4,3} + x_{3,7} - x_{7,3} = 0 \\
4) \;& -x_{3,4} + x_{4,3} + x_{4,8} - x_{8,4} = 0 \\
5) \;& -x_{1,5} + x_{5,1} + x_{5,6} - x_{6,5} + x_{5,9} - x_{9,5} = 0 \\
6) \;& -x_{2,6} + x_{6,2} - x_{5,6} + x_{6,5} + x_{6,7} - x_{7,6} + x_{6,10} - x_{10,6} = 0 \\
7) \;& -x_{3,7} + x_{7,3} - x_{6,7} + x_{7,6} + x_{7,8} - x_{8,7} + x_{7,11} - x_{11,7} = 0 \\
8) \;& -x_{4,8} + x_{8,4} - x_{7,8} + x_{8,7} + x_{8,12} - x_{12,8} = -1 \\
9) \;& -x_{5,9} + x_{9,5} + x_{9,10} - x_{10,9} + x_{9,13} - x_{13,9} = 0 \\
10) \;& -x_{6,10} + x_{10,6} - x_{9,10} + x_{10,9} + x_{10,11} - x_{11,10} + x_{10,14} - x_{14,10} = 0 \\
11) \;& -x_{7,11} + x_{11,7} - x_{10,11} + x_{11,10} + x_{11,12} - x_{12,11} + x_{11,15} - x_{15,11} = 0 \\
12) \;& -x_{8,12} + x_{8,12} - x_{11,12} + x_{12,11} + x_{12,16} - x_{16,12} = 0 \\
13) \;& -x_{9,13} + x_{13,9} + x_{13,14} - x_{14,13} = 0 \\
14) \;& -x_{10,14} + x_{14,10} - x_{13,14} + x_{14,13} + x_{14,15} - x_{15,14} = 0 \\
15) \;& -x_{11,15} + x_{15,11} - x_{14,15} + x_{15,14} + x_{15,16} - x_{16,15} = 0 \\
16) \;& -x_{12,16} + x_{16,12} - x_{15,16} + x_{16,15} = 0
\end{aligned}
$$

Solution of the first subproblem

In order to solve the first subproblem, we only need to run an ILP solver 9 times and find the trivial shortest paths for the paths $1 \to 8$, $1 \to 11$, $1 \to 13$, $8 \to 7$, $8 \leftrightarrow 11$, $8 \leftrightarrow 13$, $11 \leftrightarrow 13$, $11 \to 7$ and $13 \to 7$.

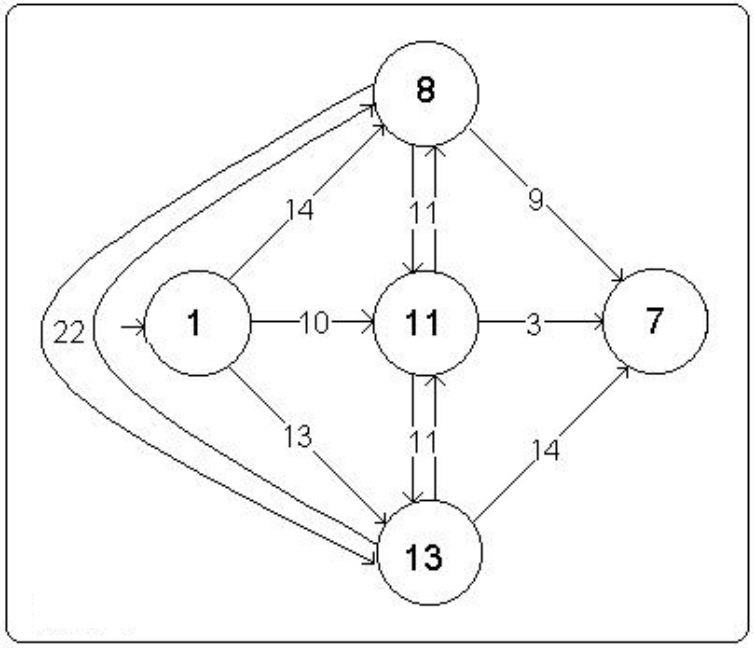


Tableau 2	node 8	node 11	node 13	node 7
node 1	$x_{1,2} \to x_{2,3} \to x_{4,8}$	$x_{1,5} \to x_{5,6} \to x_{6,9} \to x_{10,11}$	$x_{1,5} \to x_{5,6} \to x_{6,10} \to x_{10,14} \to x_{14,13}$	–
node 8	–	$x_{8,4} \to x_{4,3} \to x_{3,7} \to x_{7,11}$	$x_{8,4} \to x_{4,3} \to x_{3,7} \to x_{7,11} \to x_{11,15} \to x_{15,14} \to x_{14,13}$	$x_{8,4} \to x_{4,3} \to x_{3,7}$
node 11	$x_{11,7} \to x_{7,3} \to x_{3,4} \to x_{4,8}$	–	$x_{11,15} \to x_{15,14} \to x_{14,13}$	$x_{11,7}$
node 13	$x_{13,14} \to x_{14,15} \to x_{15,11} \to x_{11,7} \to x_{7,3} \to x_{3,4} \to x_{4,8}$	$x_{13,14} \to x_{14,15} \to x_{15,11}$	–	$x_{13,14} \to x_{14,15} \to x_{15,11} \to x_{11,7}$

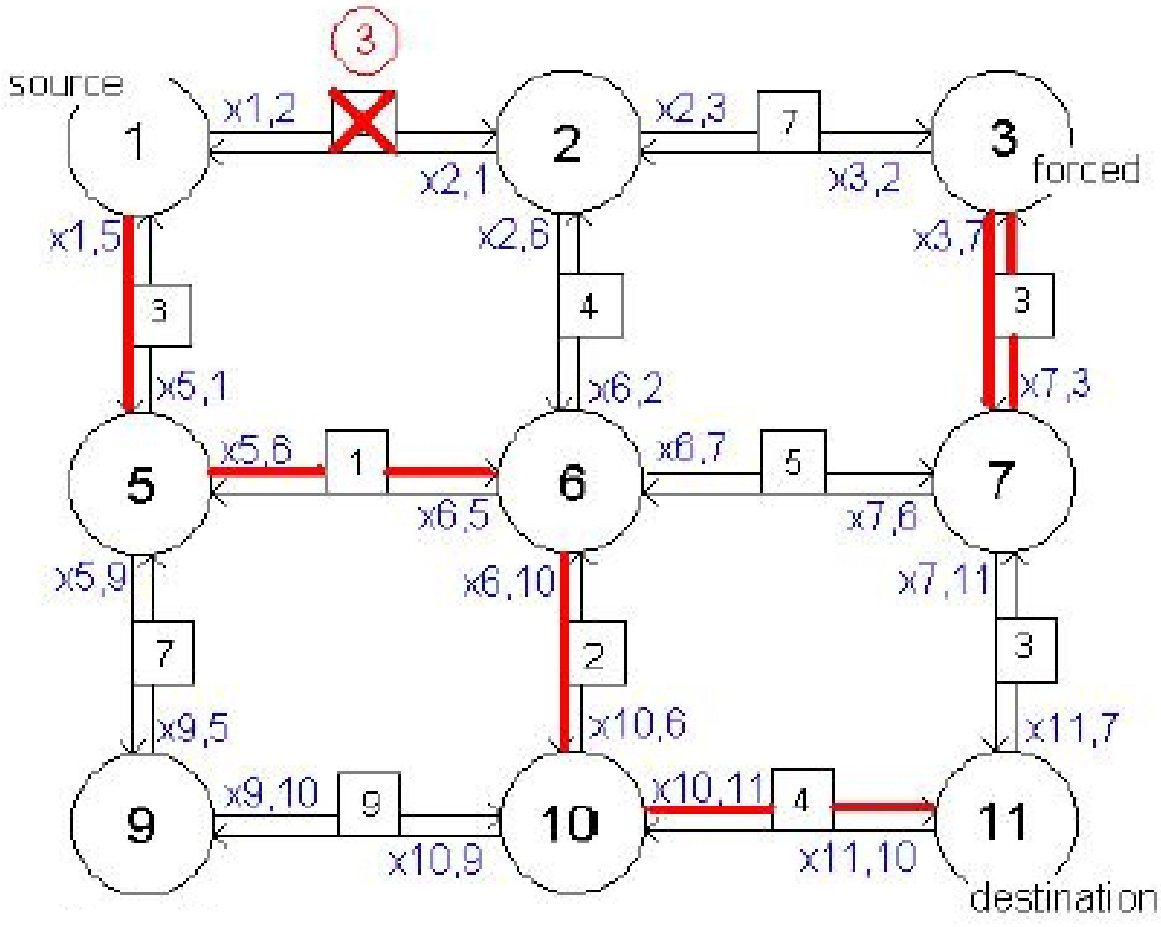
Avoiding an edge or a node

- In order to avoid the edge (i, j), we can either add the constraint $x_{ij} = 0$ or setting $c_{ij} = M$, where M is a very large number.
- Avoiding a node is equivalent to avoiding all the edges that converge on that node. Therefore, if we do not want to pass through node j we must avoid all the edges (i, j).
- Remark: Keep in mind that adding constraints is more costly in computational power than changing the costs.

Forcing a Node I

- If we want to visit a node i at least once, we must ensure that the robot goes "in" i, then "out" at least once.
- This should force a visit, and it will. However, it will not necessarily guarantee a fully connected solution; subtours may appear.
- Consider the scenario where node 1 is the source, node 11 is the destination, and node 3 is a node that the robot must visit.
  (a) $x_{2,3} + x_{7,3} \geqslant 1$: the robot must absolutely "get in" node 3 by one of these two edges;
  (b) $x_{3,2} + x_{3,7} \geqslant 1$: the robot must absolutely "get out" node 3 by one of these two edges.
- Adding the two above constaints to the ILP formulation leads to the following not fully connected solution.

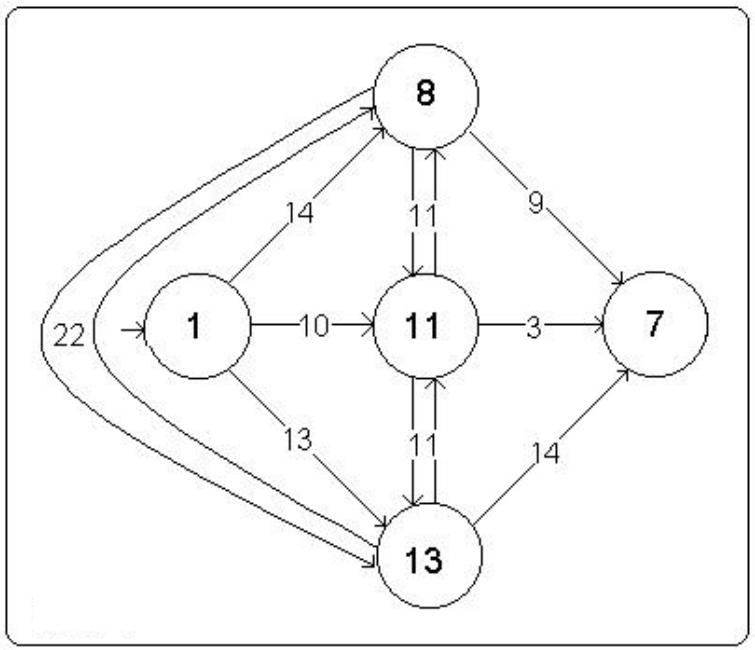
Forcing a Node II



- However, the two above mentioned constaints, in the case of the second subproblem where you must solve an instance of TSP, will help us to solve the last round of our security run problem.

Solution of the second subproblem I

- We are now in a position to solve the second subproblem:
  Find the optimal Hamiltonian path between 1 and 7 considering only the nodes 13, 11 and 8.



Solution of the second subproblem II

- Brute Force solution: since we have only 6 possible paths from node 1 to node 7, visiting the other 3 nodes, we can evaluate the best solution by inspection:
  (a) $1 \to 13 \to 11 \to 8 \to 7$: associated cost $Z = 44$;
  (b) $1 \to 13 \to 8 \to 11 \to 7$: associated cost $Z = 49$;
  (c) $1 \to 11 \to 13 \to 8 \to 7$: associated cost $Z = 52$;
  (d) $1 \to 11 \to 8 \to 13 \to 7$: associated cost $Z = 57$;
  (e) $1 \to 8 \to 11 \to 13 \to 7$: associated cost $Z = 50$;
  (f) $1 \to 8 \to 13 \to 11 \to 7$: associated cost $Z = 50$;

Solution of the second subproblem III

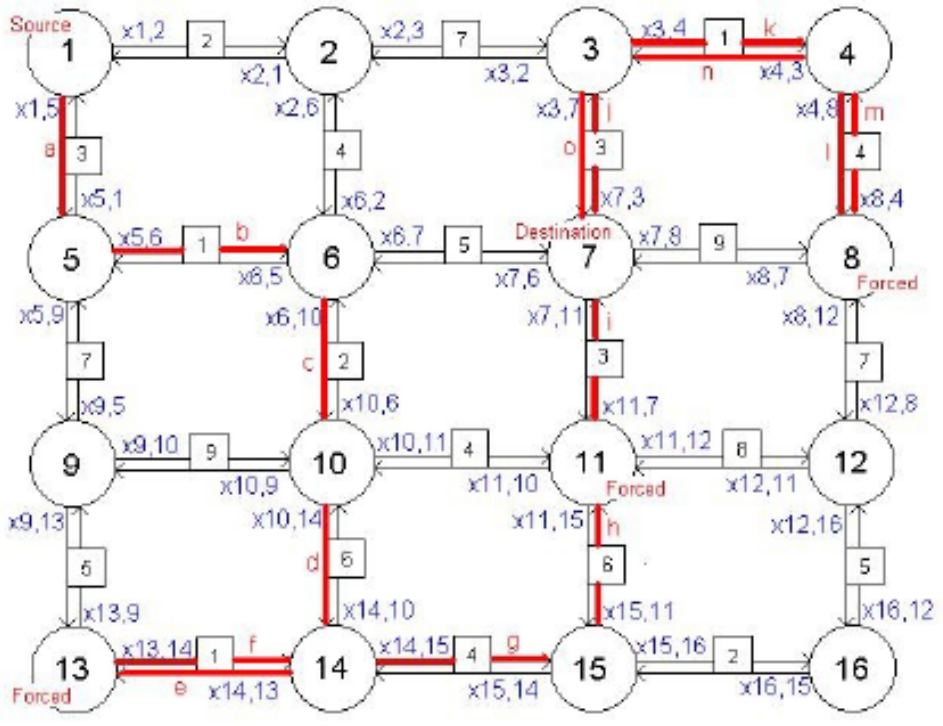
- The problem can also be modelled as ILP in the following way:

$$\begin{aligned} \min z = {} & 14x_{1,8} + 10x_{1,11} + 13x_{1,13} + 11x_{8,11} + \\ & 22x_{8,13} + 9x_{8,7} + 11x_{11,8} + 11x_{11,13} + \\ & 3x_{11,7} + 22x_{13,8} + 11x_{13,11} + 14x_{13,7} \end{aligned}$$

$$\begin{array}{rlll}
1) & x_{1,13} + x_{1,11} + x_{1,8} = 1 & \text{start at node 1} \\
8) & x_{8,13} - x_{13,8} + x_{8,11} - x_{11,8} - x_{1,8} + x_{8,7} = 0 & \text{pass thru node 8} \\
11) & x_{11,8} - x_{8,11} + x_{11,13} - x_{13,11} - x_{1,11} + x_{11,7} = 0 & \text{pass thru node 11} \\
13) & x_{13,11} - x_{11,13} + x_{13,8} - x_{8,13} - x_{1,13} + x_{13,7} = 0 & \text{pass thru node 13} \\
7) & -x_{8,7} - x_{11,7} - x_{13,7} = -1 & \text{end at node 7} \\
20) & x_{11,8} + x_{11,13} + x_{11,7} = 1 & \text{forcing node 11} \\
21) & x_{1,11} + x_{8,11} + x_{13,11} = 1 & \text{forcing node 11} \\
22) & x_{13,8} + x_{13,7} + x_{13,11} = 1 & \text{forcing node 13} \\
23) & x_{8,13} + x_{11,13} + x_{1,13} = 1 & \text{forcing node 13} \\
24) & x_{8,11} + x_{8,7} + x_{8,13} = 1 & \text{forcing node 8} \\
25) & x_{1,8} + x_{11,8} + x_{13,8} = 1 & \text{forcing node 8} \\
26) & x_{1,8} + x_{1,11} + x_{1,13} + x_{8,11} + x_{8,13} + x_{8,7} + & \text{forcing a visit to...} \\
26) & x_{11,8} + x_{11,13} + x_{11,7} + x_{13,8} + x_{13,11} + x_{13,7} = 4 & \text{... exactly four nodes} \\
17) & \forall x_{ij} = (0,1) & \text{upper bound 1, lower bound 0} \\
18) & \forall x_{ij} \leq 1 &
\end{array}$$

Solution of the security run problem

High Level	Figure 1 Level
$x_{1,13}$	$x_{1,5} \rightarrow x_{5,6} \rightarrow x_{6,10} \rightarrow x_{10,14} \rightarrow x_{14,13}$
$x_{13,11}$	$x_{13,14} \rightarrow x_{14,15} \rightarrow x_{15,11}$
$x_{11,8}$	$x_{11,7} \rightarrow x_{7,3} \rightarrow x_{3,4} \rightarrow x_{4,8}$
$x_{8,7}$	$x_{8,4} \rightarrow x_{4,3} \rightarrow x_{3,7} \rightarrow x_{7,11}$



References

- M. Talbot, "A Dynamical Programming Solution for Shortest Path Itineraries in Robotics,", *Electronic Journal of Undergraduate mathematics*, Vol. 21, pp. 21–35, 2004.