SHORTEST PATHS: APPLICATIONS, OPTIMIZATION, VARIATIONS, AND SOLVING THE CONSTRAINED SHORTEST PATH PROBLEM

EXERCISES

Prepared by Natashia Boland[1] and Irina Dumitrescu[2]

1 Applications and Modelling

1.1 Questions from "Network Flows"

Exercises from Ahuja et al.'s book "Network Flows" marked with an asterisk are more highly recommended, and those marked with two asterisks are very strongly recommended.

1. Exercise 4.2 from Ahuja et al.'s book "Network Flows".

   Solution: See photocopies from the book's Solution Manual

2. *Exercise 4.3 from Ahuja et al.'s book "Network Flows". Note that in answering this question, you may assume that all books with same height are stored on same height shelf, i.e. that no splitting of height classes is allowed.

   Solution: See photocopies from the book's Solution Manual

3. *Exercise 4.5 from Ahuja et al.'s book "Network Flows".

   Solution: See photocopies from the book's Solution Manual

4. *Exercise 4.6 from Ahuja et al.'s book "Network Flows".

   Solution: See photocopies from the book's Solution Manual

[1]Associate Professor, Department of Mathematics and Statistics, University of Melbourne

[2]HEC Montreal, Canada Research Chair in Distribution Management

5. **Exercise 4.7 from Ahuja et al.'s book "Network Flows". Note that in answering this question, you may assume that a concentrator can only be located at one of the nodes. Use c_q to denote the cost of establishing a concentrator at node q, and c_{kq} to denote the cost of homing node k onto a concentrator located at node q.

   Solution: See photocopies from the book's Solution Manual

6. Exercise 4.8 from Ahuja et al.'s book "Network Flows".

   Solution: See photocopies from the book's Solution Manual

7. **Exercise 4.9 from Ahuja et al.'s book "Network Flows". What is the complexity of the solution approach suggested in this question?

   Solution: See photocopies from the book's Solution Manual. The complexity of the approach is pseudopolynomial. Since it requires solution of a shortest path problem, it is polynomial in the number of nodes (and arcs) in the network; here the number of nodes is polynomial in items of input data, in particular the number of personnel required. This is pseudopolynomial in the length of the input data, which is given by the log of the data items.

1.2 Systems of Difference Constraints

8. Find a feasible solution or determine that no feasible solution exists for the following system of difference constraints.

$$
\begin{aligned}
x_1 - x_2 &\leq 1, & x_1 - x_4 &\leq -4, \\
x_2 - x_3 &\leq 2, & x_2 - x_5 &\leq 7, \\
x_2 - x_6 &\leq 5, & x_3 - x_6 &\leq 10, \\
x_4 - x_2 &\leq 2, & x_5 - x_1 &\leq -1, \\
x_5 - x_4 &\leq 3, & x_6 - x_3 &\leq -8
\end{aligned}
$$

   Solution: After constructing the augmented constraint graph and finding the shortest path from node 0 to all other nodes, one finds there are no negative length cycles, and gets the following labels on nodes.

Node	0	1	2	3	4	5	6
Label	0	-5	-3	0	-1	-6	-8

   Setting x_i to be the value of the label on node i for each $i = 1, \ldots, 6$ gives a feasible solution to the system of difference constraints.

9. Find a feasible solution or determine that no feasible solution exists for the following system of difference constraints.

$$\begin{aligned} x_1 - x_2 &\leq 4, & x_1 - x_5 &\leq 5, \\ x_2 - x_4 &\leq -6, & x_3 - x_2 &\leq 1, \\ x_4 - x_1 &\leq 3, & x_4 - x_3 &\leq 5, \\ x_4 - x_5 &\leq 10, & x_5 - x_3 &\leq -7, \\ x_5 - x_4 &\leq -8 \end{aligned}$$

Solution: After forming the augmented constraint graph and seeking the shortest path from node 0 to all other nodes, using an algorithm with negative length cycle detection, one finds there is a negative length cycle $(2, 3, 5, 4, 2)$ with length $1 - 7 + 10 - 6 = -2$. Thus the system is infeasible.

10. Can any shortest-path length from the new node, node 0, in the augmented constraint graph, be positive? Explain.

Solution: Since there is an arc of length 0 from node 0 to every other node, the label on every node (representing the length of the shortest path found so far from node 0 to that node) is set to 0 in the first step. Since it is only modified if a shorter path is found, of necessity such a path must have length *less than* 0, and so cannot be positive; the answer is "no".

11. Suppose that in addition to a system of difference constraints, we want to handle equality constraints of the form $x_i = x_j + b_{ij}$. Explain how the augmented constraint graph and shortest path approach can be adapted to solve this variety of constraint system.

Solution: Such an equality constraint can be written as two difference constraints: $x_i - x_j \leq b_{ij}$ and $x_i - x_j \geq b_{ij}$, or equivalently, $x_j - x_i \leq -b_{ij}$.

12. Suppose a system of constraints in variables $x_1, x_2, \ldots, x_n$ contains difference constraints, and single variable bounds of the form $x_i \leq u_i$ or $x_i \geq l_i$. Explain how the augmented constraint graph and shortest path approach can be adapted to solve this variety of constraint system.

Solution: Recall that we can add a constant to every variable in a feasible solution to a system of difference constraints, and still have a solution. We will add a new variable x_{n+1}, and whatever its value is, we will take that and add its negative to all other variables in the solution found by the augmented constraint graph and shortest path approach, to generate our final solution. Thus if $x_1, \ldots, x_n, x_{n+1}$ is the shortest path solution, our final solution will be $x_i' = x_i - x_{n+1}$. To satisfy the upper bound constraint we need $x_i' \leq u_i$, which is equivalent to asking that $x_i - x_{n+1} \leq u_i$. Note that this is a difference constraint! To satisfy the lower bound constraint we need $x'_i \geq l_i$, which is equivalent to asking that $x_i - x_{n+1} \geq l_i$, or equivalently that $x_{n+1} - x_i \leq -l_i$. Note that this, too, is a difference constraint! Thus we can include a new node $n+1$ in the augmented constraint graph, with an arc from $n+1$ to node i of length u_i for every variable x_i with an upper bound constraint, and an arc from node i to node $n+1$ of length $-l_i$ for every variable x_i with a lower bound constraint. By adding the value of $-x_{n+1}$ to the value of every other variable found by the shortest path approach, we obtain a solution feasible to the original system.

2 Optimization and Duality

13. Write down the LP dual of the LP formulation of the shortest path problem, over digraph $G = (V, A)$, with start node s, end node t, and lengths c_{ij} for all $(i, j) \in A$, using dual variables u_i for each node $i \in V$.

Solution: The LP dual is as follows.

$$\begin{array}{lll} \max & u_t - u_s & \\ \text{s.t.} & u_j - u_i \leq c_{ij}, & \forall (i,j) \in A \end{array}$$

In what follows, you may assume that the shortest path problem is feasible, i.e. that there exists path from s to t in the digraph. Furthermore, for simplicity, you may assume that all nodes in the graph are reachable from node s.

(a) Explain why it is that we can, without loss of generality, add a constraint $u_s = 0$ to the LP dual.

Solution: The feasible set for the LP dual is a system of difference constraints, thus for any solution u, the solution u' formed by taking $u'_i = u_i + \Delta$ for some constant Δ, for all $i \in V$, is also feasible. Furthermore, u and u' have the same objective value: $u'_t - u'_s = (u_t + \Delta) - (u_s + \Delta) = u_t - u_s$. So for any optimal solution u to the LP dual, we may take $\Delta = -u_s$, and form u' also optimal, satisfying $u'_s = 0$.

(b) Prove that if there is a negative length cycle in the network reachable from s, then the (primal) shortest path LP is unbounded below.

Solution: Let Δ be the indicator vector for arcs in the negative length cycle. Then obviously Δ satisfies $\sum_{j\,:\,(j,i)\in A} \Delta_{ji} - \sum_{j\,:\,(i,j)\in A} \Delta_{ij} = 0$ for all $i \in V$. Since by assumption the problem is feasible, there exists a path P from s to t in the digraph. Let x be the indicator vector for the arcs in this path. Then obviously x is feasible for the primal shortest path LP, and it must be that $x + \alpha\Delta$ is feasible for the shortest path LP for any $\alpha \in \mathbb{R}$, as

$$\begin{aligned} & \sum_{j\,:\,(j,i)\in A} (x_{ji} + \alpha\Delta_{ji}) - \sum_{j\,:\,(i,j)\in A} (x_{ij} - \alpha\Delta_{ij}) \\ = & \sum_{j\,:\,(j,i)\in A} x_{ji} - \sum_{j\,:\,(i,j)\in A} x_{ij} + \alpha(\sum_{j\,:\,(j,i)\in A} \Delta_{ji} - \sum_{j\,:\,(i,j)\in A} \Delta_{ij}) \\ = & \sum_{j\,:\,(j,i)\in A} x_{ji} - \sum_{j\,:\,(i,j)\in A} x_{ij}. \end{aligned}$$

Now for any $M \in \mathbb{R}$, there exists α such that $c(x+\alpha\Delta) < M$. For example, take $\alpha = \frac{M-cx}{c\Delta} + 1$, which is well defined since Δ indicates a negative length cycle means $c\Delta < 0$. Then

$$\alpha > \frac{M - cx}{c\Delta} \quad \Leftrightarrow \quad \alpha c\Delta < M - cx \quad \Leftrightarrow \quad cx + \alpha c\Delta < M \quad \Leftrightarrow \quad c(x + \alpha\Delta) < M.$$

This shows that there exist feasible solutions for the LP with arbitrarily small objective values; the LP is unbounded below.

Alternatively, observe that the network is actually the constraint graph for the dual problem. Let $(i_0, i_1, \ldots, i_{k-1})$ denote the negative length cycle. Suppose that the LP dual is feasible, with feasible solution u. Then u must satisfy the dual constraints for this cycle:

$$u_{i_{(j+1)\bmod k}} - u_{i_j} \leq c_{i_j i_{(j+1)\bmod k}}$$

for all $j = 0, \ldots, k-1$. Adding up these constraints over all $j = 0, \ldots, k-1$ yields zero on the left-hand side, since every variable appears as many times subtracted as it is added, and yields the length of the cycle (the c values summed over all arcs in the cycle), on the right-hand side. Since the cycle has negative length, we can deduce that zero is less than or equal to a negative number, which is obviously impossible. Thus it must be that the LP dual is infeasible, and hence by LP duality theory the primal shortest path LP is unbounded below.

(c) Explain why it is that if the (primal) shortest path LP is unbounded below, then there must exist a negative length cycle in the network.

Solution: Suppose the (primal) shortest path LP is unbounded below, and let r be a ray of the LP feasible polyhedron, with $cr < 0$. Now r a ray of the LP feasible polyhedron means that $r \geq 0$ and

$$\sum_{j\,:\,(j,i)\in A} r_{ji} - \sum_{j\,:\,(i,j)\in A} r_{ij} = 0,$$

for all $i \in V$, i.e. r induces a non-negative circulation in G. Now any such circulation can be decomposed into a sum of positive flows on cycles. Suppose that there are K such cycles, let

Δ_k be the indicator vector for the arcs in the kth cycle, and let $\alpha_k > 0$ denote the flow on the kth cycle, so we have that

$$r = \sum_{k=1}^{K} \alpha_k \Delta_k.$$

Now it must be that for some $k \in \{1, \ldots, K\}$, $c\Delta_k < 0$, since $cr < 0$ and $\alpha_j > 0$ for all $j = 1, \ldots, K$. Thus the kth cycle is a negative length cycle; such a cycle must exist.

(d) Suppose that there are no negative length cycles in the network. Show that if u_i is taken to be the length of the shortest path from node s to node i for each $i \in V$, then u must be feasible for the LP dual.

Solution: If there are no negative length cycles in the network, then the primal shortest path LPs for paths from s to i for any node $i \in V$ cannot be unbounded below, (by part (13c)), and so must have an optimal solution. Thus the shortest path from node s to node i for each $i \in V$ exists. Let u_i be the length of this path. The Bellman-Ford optimality condition for shortest paths ensures that for each $(i, j) \in A$,

$$u_j \leq u_i + c_{ij} \quad \Leftrightarrow \quad u_j - u_i \leq c_{ij}.$$

These are precisely the dual LP constraints, hence u is feasible for the LP dual.

(e) Prove that if the network has no negative length cycle, and the shortest path tree is unique, then the solution to the (primal) shortest path LP is unique, and is the indicator vector for the shortest path from s to t.

Solution: If there are no negative length cycles in the network, then the shortest path from node s to node i for each $i \in V$ exists. Let u_i be the length of this path. If arc $(i, j) \in A$ is on the shortest path tree, then

$$u_j = u_i + c_{ij},$$

so for the shortest path tree to be unique, it must be that

$$T = \{(i, j) \in A \ : \ u_j = u_i + c_{ij}\}$$

induces a (spanning) tree in G rooted at s. (Since by assumption every node in the network is reachable from s, the tree must span the graph.) Now by LP complementary slackness conditions,

$$\text{if} \quad u_j - u_i < c_{ij} \quad \text{then} \quad x_{ij} = 0.$$

Thus the support graph of x, (the set of arcs on which x can take on non-zero, i.e. positive, values), must be a subgraph of (V, T). Now there is only one, unique, path in (V, T) from s to

t, since T induces a tree, so the unique solution to the equations

$$\sum_{j\ :\ (j,i)\in T} x_{ji} - \sum_{j\ :\ (i,j)\in T} x_{ij} = \begin{cases} -1, & i = s \\ 0, & i \neq s,t \\ 1, & i = t \end{cases} \qquad \forall i \in V$$

is the binary vector indicating the arcs in the unique path from s to t in (V, T). This can be proved more formally by considering the above equations in order of decreasing depth in the tree. The equations for all leaf nodes j take the form $x_{i_T(j),j} = d$, where $i_T(j)$ is the unique predecessor of node j in the tree T, and $d = 1$ if $j = t$, $d = 0$ otherwise. This determines the values of the x variables on all arcs entering leaf nodes. Inductive arguments yield the desired result.

14. Write down a (Mixed) Integer Linear Programming formulation of the shortest path problem with renewable node resources. Use the notation $G = (V, A)$ for the digraph, s and t for the path start and end nodes, c for the arc lengths, D for the node resources, Q for the resource limit and $R \subseteq V$ for the nodes at which resources may be renewed. You may assume that any node in R supplies itself, or equivalently that it has zero demand.

Solution:

$$\begin{aligned} \min \quad & \sum_{(i,j)\in A} c_{ij}x_{ij} \\ \text{s.t.} \quad & \sum_{j:(j,i)\in A} x_{ji} - \sum_{j:(i,j)\in A} x_{ij} = \begin{cases} -1, & i = s \\ 0, & i \neq s,t \\ 1, & i = t \end{cases} & \forall i \in V \\ & d_j \geq d_i + D_i - (Q + D_i)(1 - x_{ij}), & \forall (i,j) \in A, i \notin R \\ & d_i \leq Q, & \forall i \in V \\ & d \geq 0 \\ & x \in \{0,1\}^{|A|} \end{aligned}$$

3 Variations and Modelling

15. In the case of minefield path planning, the Euclidean length of the path may need to be constrained to be no more than some limit L. Explain how to model this as an additive resource constrained shortest path problem.

Solution: For each arc $(i, j) \in A$ in the space discretization graph, let δ_{ij} be the Euclidean distance between the point in space represented by node i and that represented by node j. We then require the path to satisfy an additive arc resource constraint with arc resources given by δ and capacity limit L.

16. Suppose that in the case of piecewise linear function approximation, one seeks to minimize the number of points used, subject to constraining the squared error in the approximation to be no worse than some limit L. Show how to model this as an additive resource constrained shortest path problem.

    Solution: We define the digraph as discussed in the lecture, but define arc lengths c_{ij} for each $(i, j) \in A$ by: $c_{ij} = 1$ if $j \neq n$ and $c_{ij} = 0$ otherwise. Thus the total length of any path is the number of breakpoints (other than the start and end point) used in the function approximation. We also define an additive arc resource δ to be the function error embodied by each arc:

$$\delta_{ij} = \sum_{k=i+1}^{j-1} (f(x_k) - g(x_k))^2, \quad \forall (i, j) \in A.$$

    Thus the total resource consumed on any path is the squared error of the approximating function corresponding to that path; the total (additive) resource used on the path should be no more than L.

17. Suppose that in the case of piecewise linear function approximation, one seeks to minimize the error in the approximation, subject to constraining the number of points used to be no more than some limit M. Show how to model this as an additive resource constrained shortest path problem.

    Solution: This is identical to the previous question, but the definitions of c and δ are reversed, and M replaces L.

18. In the case of minefield path planning, the number of turns made along the path may need to be constrained to be no more than some limit M. Show how to model this as an additive resource constrained shortest path problem, in the case that the spatial discretization used is a simple rectangular grid, with arcs only along the sides of each rectangular grid element. [Hint: you may need to introduce new nodes and arcs]. Generalize your model to the case that arcs are allowed along the diagonals of each rectangular grid element.

    Solution: Let the two directions in the rectangular grid be described as "horizontal" and "vertical". So every arc is either a horizontal arc, or a vertical arc. We construct a new network to model turning from the rectangular grid network as follows. Replace each node $i \in V$ in the rectangular grid network with two nodes i_v and i_h, to capture incoming vertical and horizontal arcs respectively. Let δ denote an arc resource in the new network used to model turning. For each vertical arc (i,j) in the old network, we create two arcs in the new network: (i_v, j_v) with $\delta_{i_v j_v} = 0$ and (i_h, j_v) with $\delta_{i_h j_v} = 1$; both new arcs have the same cost (risk) as the old arc. Similarly, for each horizontal arc (i,j) in the old network, we create two arcs in the new network: (i_h, j_h) with $\delta_{i_h j_h} = 0$ and (i_v, j_h) with $\delta_{i_v j_h} = 1$; both new arcs have the same cost (risk) as the old arc. Now a turn in the path in the rectangular grid network would be represented by an arc (i,j) followed by (j,k), where one of these is horizontal and the other vertical. Consider the case where the first is horizontal, the second vertical. In the new network, whichever of the new arcs is used in place of the first arc, (i,j), it must enter the node j_h, in which case it must be that the arc used in the new network in place of (j,k) is (j_h, k_v). This arc uses up one unit of resource, thus counting the turn from horizontal to vertical. We now ask for the minimum cost (risk) path in the new network subject to the additive arc resource constraint with resource δ and resource limit M.

The approach for diagonal arcs is similar, except that we need to distinguish between entering a node on the "left-right" diagonal, or on the "right-left" diagonal. For each node i in the rectangular grid network, we will create four nodes, i_h, i_v, i_L, and i_R, to represent the case of entering the node along the horizontal, vertical, left-right diagonal and right-left diagonal respectively. For arc (i,j) in the old network, we make create four arcs in the new network, one starting at each of the four copies of i, but all ending at the copy of j corresponding to the direction of the arc. All would use one unit of resource, except for that starting at the copy of node i corresponding to the direction of the arc. For example, if (i,j) was a left-right diagonal arc, we would have arcs (i_v, j_L), (i_h, j_L), (i_L, j_L), and (i_R, j_L); all would use one unit of resource except for (i_L, j_L), which would use none.

19. Suggest how you might, in general, formulate the problem of approximating an arbitrary given function $f(x)$ in space between the points $(a, f(a))$ and $(b, f(b))$ by a piecewise linear function, trading off approximation accuracy against the number of linear pieces used, as a shortest path problem in a network. You may assume f has properties that make numerical calculations using f tractable, for example, you may assume f is integrable on the interval $[a,b]$.

    Solution: We could create a spatial discretization of the $x-y$ plane, where $y = f(x)$, the function value axis. Construct the network to ensure that points $(a, f(a))$ and $(b, f(b))$ are represented by nodes in the network, labelled s and t respectively. For a network node i, write $x(i)$ to denote the x-coordinate of the point in space it represents, and $y(i)$ to denote the y-coordinate. All network arcs should have the form (i,j) where $x(i) < x(j)$. The equation of the piece of the piecewise linear function represented by (i, j) is

$$g_{ij}(x) = \frac{y(j) - y(i)}{x(j) - x(i)}(x - x(j)) + y(j).$$

If arc (i, j) is used in the path, then this linear piece is used in the piecewise linear approximating function. So the piecewise linear function induced by a path P in the network from s to t is given by

$$g(x) = g_{ij}(x)$$

for $x \in [a, b)$, where $(i, j) \in P$ is the (unique) arc in P having $x(i) \leq x < x(j)$, and $g(b) = f(b)$. One approach to measuring the accuracy of the approximation, would be to define the error e_{ij} for each linear piece represented by arc (i, j) as the area between the curves $f(x)$ and $g_{ij}(x)$ on the interval $[x(i), x(j)]$; this could be calculated by integration, for example. We then set the length of each arc (i, j) to be

$$c_{ij} = e_{ij} + \beta.$$

The length of a path would then represent the total area between $f(x)$ and $g(x)$ on the interval $[a, b]$, modelling the error in the approximation, plus β times the number of linear pieces used in the approximating function.

20. Given a digraph $G = (V, A)$, path start and end nodes s and t respectively, and a set $T = \{S_1, S_2, \ldots, S_K\}$ of distinct subsets of the nodes, $S_k \subset V$ with $|S_k| \geq 2$ for each $k = 1, \ldots, K$, a *subset disjoint s-t path* P is a path in G from s to t that contains at most one node from any subset. In other words, if we write $V(P)$ to denote the set of nodes in the path P, then P is a subset disjoint path if $|V(P) \cap S_k| \leq 1$ for all $k = 1, \ldots, K$. Given arc lengths c_{ij} for each arc $(i, j) \in A$, the shortest subset disjoint path problem is the problem of finding a subset disjoint s-t path minimizing the sum of c values on the arcs in the path. Show how to model the subset disjoint path problem as a constrained shortest path problem with multiple additive resources.

    Solution: Define a node resource r^k for each set $k = 1, \ldots, K$, defined by $r^k_i = 1$ if $i \in S_k$ and $r^i_k = 0$ otherwise, for all $i \in V$. Set the resource limit for the kth resource to be $R^k = 1$. Then the problem is to find the shortest path from s to t satisfying additive node resource constraints for all K of these resources.

4 Constrained Shortest Path Problems

4.1 Complexity

In these questions, we are not really concerned with the precise complexity of an algorithm, but are only interested in whether or not a problem (or algorithm) is polynomially solvable, or if, as far as is known, it will require exponential time to solve; in the latter case, we are interested in whether or not it is pseudopolynomially solvable.

21. Prove that the constrained shortest path problems you derived in Questions 17 and 18 are solvable in polynomial time, in the former case in time polynomial in the number of points in the original function, and in the latter case in the number of nodes in the original space discretization network.

    Solution: We use the construction given in lectures to convert an RCSPP to an SPP.

    We can see that the number of nodes in the SPP graph G' in the function approximation case is $O(nM)$ where n is the number of points in the original function. However as M is the upper limit on the number of points used in the approximation, clearly $M < n$, otherwise the constraint can be ignored and the solution will obviously be the original function. Thus the number of nodes in the SPP graph is $\mathrm{O}(n^2)$. Since SPP can be solved in time polynomial in the number of nodes in the graph, we deduce the desired result for the function approximation problem.

    Similarly, the number of nodes in the SPP graph G' in the minefield case is $O(nM)$ where n is the number of points in the RCSPP graph. However as M is the upper limit on the number of turns used, and each arc in the RCSPP graph either uses one turn or none, clearly $M < m$, where m is the number of arcs in the RCSPP graph; otherwise the constraint can be ignored and the solution will just be the shortest path in the RCSPP graph. Thus the number of nodes in the SPP graph is $\mathrm{O}(nm)$. Now the number of nodes and arcs in the RCSPP graph is twice the number in the original space discretization graph, so the number of nodes in the SPP graph is $O(4nm)$ which is $O(n^3)$. Since SPP can be solved in time polynomial in the number of nodes in the graph, we deduce the desired result for the minefield problem.

22. Show how to solve an additive resource constrained shortest path problem with multiple resources as a shortest path problem in a related graph, i.e. construct this graph. You may assume that at least one of the resources has positive values on every arc, and that all resources are non-negative integers. How many nodes does the graph you constructed have? What does this tell you about the complexity of the additive resource constrained shortest path problem with multiple resources?

Solution: Create a copy of every node in the network for every possible resource vector. Suppose there are K resources with limits $R_1, R_2, \ldots, R_K$. There are clearly $(R_1 + 1)(R_2 + 1) \ldots (R_K + 1)$ copies of each node. Write (i, r) to denote the copy of node i for resource vector $r \in \{0, 1, \ldots, R_1\} \times \{0, 1, \ldots, R_2\} \times \cdots \times \{0, 1, \ldots, R_K\}$. For each arc (i, j) in the original network, with resource useage vector r, construct a copy from node (i, q) to node $(j, q + r)$, having the same cost, for all $q \in \{0, 1, \ldots, R_1\} \times \{0, 1, \ldots, R_2\} \times \cdots \times \{0, 1, \ldots, R_K\}$ such that $q_k + r_k \leq R_k$ for all $k = 1, \ldots, K$. Include arcs to the "last" copy of node t, $(t, (R_1, \ldots, R_K))$, from all other copies of node t, of cost 0. Since at least one element of each resource usesage vector r is positive, and since $(R_1, \ldots, R_K)$ is the resource vector highest in the lexicographic order, all arcs are directed from a node having a resource vector lower in the lexicographic order to one that is higher. Thus the new graph is acyclic, and solving the SPP from node $(s, 0)$ to node $(t, (R_1, \ldots, R_K))$ solves the multiple resource constrained shortest path problem. Thus this problem can be solved in time polynomial in $|V|(R_1 + 1)(R_2 + 1) \ldots (R_K + 1)$.

23. Use your answer in Question 22 to deduce something about the complexity of the elementary resource constrained shortest path problem, in which the resource used on every arc is assumed to be non-negative.

    Solution: Every arc has at least one element of its resource vector set to one, to reflect the resource for one of its terminal nodes, hence has a positive element. The elements for node resources are either zero or one, so since the original resource is non-negative, all resource vectors are non-negative, and the conditions of Question 22 are met. Since there is a resource for each node in the graph, with resource limit 2, the complexity would be polynomial in $|V|R2^|V|$.

24. Use your answer in Question 22to discuss the complexity of the constrained shortest path problem you derived in Question 20, in terms of the number of subsets, K.

    Solution: We can remodel this as an additive arc resource for each node as described in the first lecture. Every arc then has at least one element of its resource vector set to one, to reflect the subset containing one of its terminal nodes, thus is positive. The other elements are zero, hence the vector is non-negative, and the conditions of Question 22 are met. Since there is a resource for each subset, with resource limit 2, the complexity would be polynomial in $|V|2^K$.

4.2 Preprocessing

25. Let $G = (V, A)$, where $V = \{1, 2, 3, 4, 5, 6\}$ and A is the set of arcs:

Arc	Resource	Cost
(1,2)	1	1
(1,4)	2	2
(1,3)	1	4
(2,3)	2	2
(2,4)	1	3
(3,4)	3	17
(3,5)	1	13
(3,6)	9	8
(4,3)	2	2
(4,6)	18	4
(5,6)	1	5

The origin is $s = 1$, the destination is $t = 6$, and the resource limit is $R = 19$. Apply the preprocessing procedure.

Solution: We first calculate the shortest paths from $s = 1$ to every node and from every node to $t = 6$, with arc lengths given by the costs.

Arc lengths given by costs				
	Shortest path from $s = 1$ to i		Shortest path from i to $t = 6$	
i	Path Length	Path	Path Length	Path
1	0	-	8	(1,2,4,6)
2	1	(1,2)	7	(2,4,6)
3	3	(1,2,3)	8	(3,6)
4	2	(1,4)	4	(4,6)
5	16	(1,2,3,5)	5	(5,6)
6	6	(1,4,6)	0	-

We now calculate the shortest paths from $s = 1$ to i and from i to $t = 6$, with arc lengths given by the resource consumption.

Arc lengths given by the resource consumption				
	Shortest path from $s = 1$ to i		Shortest path from i to $t = 6$	
i	Path Length	Path	Path Length	Path
1	0	-	3	(1,3,5,6)
2	1	(1,2)	4	(2,3,5,6)
3	1	(1,3)	2	(3,5,6)
4	2	(1,2,4)	4	(4,3,5,6)
5	2	(1,3,5)	1	(5,6)
6	3	(1,3,5,6)	0	-

The initial upper bound is the cost of the path $(1, 3, 5, 6)$: $U = 22$.

Node elimination:

- Cost: The cost of the least cost path from s to i plus the cost of the least cost path from i to t is less than $U = 22$, therefore no node is eliminated.
- Resource: The resource consumed along the least consumption path from s to i plus the resource consumed along the least consumption path from i to t is less than $R = 20$, therefore no node is eliminated.

Arc elimination:

- **Arc (1,2)**: r_{12} + r(2 to 6)= 1 + 4<R (for path (1,2,3,5,6)) and c_{12}+c(2 to 6)=1+7<U (for path (1,2,4,6)), therefore no arc elimination.
  Cost of the path (1,2,3,5,6)=21 (the path is feasible!), therefore we update the upper bound: $U = 21$.
  Resource of the path (1,2,4,6)=20>R (path not feasible).
- **Arc (1,3)**: r_{13} + r(3 to 6)= 1+1<R (for path (1,3,5,6)) and c_{13}+c(3 to 6)=4+8=12<U (for path (1,3,6), therefore no arc elimination.
  Cost of the path (1,3,5,6,)=22>U.
  Resource of the path (1,3,6)=1+9=10<R (path feasible), therefore its cost is a new upper bound: U=12.
- **Arc (1,4)**: r_{14}+r(4 to 6)=2+4<R (for path (1,4,3,5,6)) and c_{14}+c(4 to 6)=2+4=6<U (for path (1,4,6)), therefore no arc elimination.
  Cost of the path (1,4,3,5,6)=22>U.
  Resource of the path (1,4,6)=20>R.

- **Arc (2,3)**: r(1 to 2)+r_{23}+r(3 to 6)=1+2+2<R (for path (1,2,3,5,6)) and c(1 to 2)+c_{23}+c(3 to 6)=1+2+8=11<U (for path (1,2,3,6).

  Cost of the path (1,2,3,5,6)=21 >U.

  Resource of the path (1,2,3,6)<R, therefore its cost is our new upper bound: U=11.

- **Arc (2,4)**: r(1 to 2)+r_{24}+r(4 to 6)=1+1+4<R (for path (1,2,4,3,5,6)) and c(1 to 2)+c_{24}+c(4 to 6)=1+3+4<U (for path (1,2,4,6)).

  Cost of the path (1,2,4,3,5,6) > U.

  Resource of the path (1,2,4,6)=20>R.

- **Arc (3,4)**: r(1 to 3)+r_{34}+r(4 to 6)=1+3+4<R (for path (1,3,4,3,5,6)) and c(1 to 3)+c_{34}+c(4 to 6)=3+17+4>U.

  Eliminate arc (3,4)!!

- **Arc (3,5)**: r(1 to 3)+r_{35}+r(5 to 6)=1+1+1<R (for path (1,3,5,6)) and c(1 to 3)+c_{35}+c(5 to 6)=3+13+5=21>U.

  Cost of the path (1,3,5,6)= 22.

- **Arc (3,6)**: r(1 to 3)+r_{36}=1+1<R (for path (1,3,6)) and c(1 to 3)+c_{36}=3+8=U.

  Eliminate arc (3,6)!!

- **Arc (4,6)**: r(1 to 4)+r_{46}=2+18>R.

  Eliminate arc (4,6)!!

- **Arc (5,6)**: r(1 to 5)+r_{56}=2+1<R (for path (1,3,5,6)) and c(1 to 5)+c_{56}=16+5>U.

  Cost of the path (1,3,5,6)>U.

Reduced network:

Arc	Resource	Cost
(1,2)	1	1
(1,4)	2	2
(1,3)	1	4
(2,3)	2	2
(2,4)	1	3
(3,5)	1	13
(4,3)	2	2
(5,6)	1	5

Since the network and the upper bound were modified, we re-start the preprocessing procedure.

We first calculate the shortest paths from $s = 1$ to every node and from every node to $t = 6$, with arc lengths given by the costs.

Arc lengths given by costs				
	Shortest path from $s = 1$ to i		Shortest path from i to $t = 6$	
i	Path Length	Path	Path Length	Path
1	0	-	22	(1,4,3,5,6)

We can STOP. The cheapest path from s to t has cost greater than the upper bound, which means the upper bound identified corresponds to the optimal solution. Therefore the optimal solution is the path (1,2,3,6) of cost 11 and resource consumption 12.

26. Solve the resource constrained shortest path problem defined in the previous exercise using the Label Setting Algorithm.

Solution: The efficient labels at the end of the algorithm are:

- Node 1: (0,0).
- Node 2: (1,1).
- Node 3: (1,4), (3,3).
- Node 4: (2,2).
- Node 5: (2,17).
- Node 6: (3,22), (6,21), (10,12), (12,11).

The optimal path is the one corresponding to the label (12,11), i.e. cost 11, resource consumed 12 (consistent with the previous exercise!).