On Minimum Reload Cost Paths, Tours and Flows

Edoardo AMALDI
Politecnico of Milano

Giulia GALBIATI
University of Pavia

Francesco MAFFIOLI
Politecnico of Milano

CTW 2008 - Gargnano, 13-15 May 2008

The Model

The Model

- a directed graph $G = (V, A)$

The Model

- a directed graph $G = (V, A)$
- a non-negative cost $w(a)$ for each arc $a \in A$

The Model

- a directed graph $G = (V, A)$
- a non-negative cost $w(a)$ for each arc $a \in A$
- a color $l(a)$ for each arc $a \in A$ out of a set L of colors

The Model

- a directed graph $G = (V, A)$
- a non-negative cost $w(a)$ for each arc $a \in A$
- a color $l(a)$ for each arc $a \in A$ out of a set L of colors
- a non-negative integer reload cost matrix $R = \{r_{l\,l'}\}_{l,l' \in L}$ where $r_{l\,l'}$ represents the cost of moving from an arc of color l to another of color $l^{'}$

The Model

- a directed graph $G = (V, A)$
- a non-negative cost $w(a)$ for each arc $a \in A$
- a color $l(a)$ for each arc $a \in A$ out of a set L of colors
- a non-negative integer reload cost matrix $R = \{r_{l\,l'}\}_{l,l' \in L}$ where $r_{l\,l'}$ represents the cost of moving from an arc of color l to another of color $l^{'}$

For any path $P = (a_1, ..., a_k)$, with arcs colored by $(l_1, ..., l_k)$:

The Model

- a directed graph $G = (V, A)$
- a non-negative cost $w(a)$ for each arc $a \in A$
- a color $l(a)$ for each arc $a \in A$ out of a set L of colors
- a non-negative integer reload cost matrix $R = \{r_{l\,l'}\}_{l,l' \in L}$ where $r_{l\,l'}$ represents the cost of moving from an arc of color l to another of color $l^{'}$

For any path $P = (a_1, ..., a_k)$, with arcs colored by $(l_1, ..., l_k)$:

$w(P)$ is the arc cost of P

The Model

- a directed graph $G = (V, A)$
- a non-negative cost $w(a)$ for each arc $a \in A$
- a color $l(a)$ for each arc $a \in A$ out of a set L of colors
- a non-negative integer reload cost matrix $R = \{r_{l\,l'}\}_{l,l' \in L}$ where $r_{l\,l'}$ represents the cost of moving from an arc of color l to another of color $l^{'}$

For any path $P = (a_1, \ldots, a_k)$, with arcs colored by $(l_1, \ldots, l_k)$:

$w(P)$ is the arc cost of P
$r(P) = \sum_{j=1}^{k-1} r_{l_j l_{j+1}}$ is the reload cost of P

The Model

- a directed graph $G = (V, A)$
- a non-negative cost $w(a)$ for each arc $a \in A$
- a color $l(a)$ for each arc $a \in A$ out of a set L of colors
- a non-negative integer reload cost matrix $R = \{r_{l\,l'}\}_{l,l' \in L}$
  where $r_{l\,l'}$ represents the cost of moving from an arc of color l to another of color l'

For any path $P = (a_1, ..., a_k)$, with arcs colored by $(l_1, ..., l_k)$:

$w(P)$ is the arc cost of P
$r(P) = \sum_{j=1}^{k-1} r_{l_j l_{j+1}}$ is the reload cost of P
$c(P) = w(P) + r(P)$ is the transportation cost of P

The Model

- a directed graph $G = (V, A)$
- a non-negative cost $w(a)$ for each arc $a \in A$
- a color $l(a)$ for each arc $a \in A$ out of a set L of colors
- a non-negative integer reload cost matrix $R = \{r_{l\,l'}\}_{l,l' \in L}$ where $r_{l\,l'}$ represents the cost of moving from an arc of color l to another of color l'

For any path $P = (a_1, \ldots, a_k)$, with arcs colored by $(l_1, \ldots, l_k)$:

$w(P)$ is the arc cost of P
$r(P) = \sum_{j=1}^{k-1} r_{l_j l_{j+1}}$ is the reload cost of P
$c(P) = w(P) + r(P)$ is the transportation cost of P

Note - A similar model can be defined for undirected graphs.

Applications

Applications

Telecommunication: data conversion at interchange points

Overlay: change of technology

Transportation: unloading and reloading goods at junctions

Energy distribution: different kypes of carriers

Applications

Telecommunication: data conversion at interchange points

Overlay: change of technology

Transportation: unloading and reloading goods at junctions

Energy distribution: different kypes of carriers

Previous work

This natural concept has received very little attention.

Applications

Telecommunication: data conversion at interchange points

Overlay: change of technology

Transportation: unloading and reloading goods at junctions

Energy distribution: different kypes of carriers

Previous work

This natural concept has received very little attention.

- H.-C. Wirth, J. Steffan, Discrete Appl. Math. 113 (2001).
- S. Raghavan, I. Gamvros, L. Gouveia, Proc. International Network Optimization Conference (INOC 2007), Spa, 2007.
- Giulia Galbiati, to appear in Discrete Appl. Math.(2008).

The path problems

The path problems

- *Problem* P1: find a minimum transportation cost path between two given nodes s and t of G.
- *Problem* P2: find a set of paths from a given node s to the other nodes of G minimizing the sum of their transportation costs.
- *Problem* P3: find a minimum transportation cost path-tree from s to the other nodes of G.
- *Problem* P4: find a path-tree from s to the other nodes of G minimizing the maximum among the transportation costs of its paths.

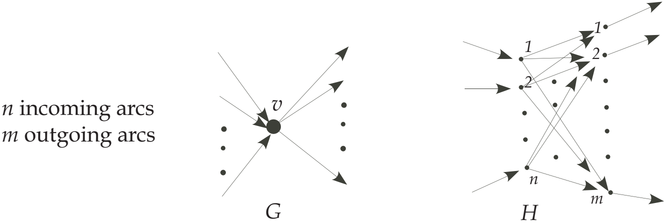
Problem P1

Problem $\mathrm{P1}$ *is polynomially solvable*.

Problem P1

Problem P1 *is polynomially solvable*.

Apply to all nodes v of G except s and t the splitting procedure:



Original arcs of G maintain their costs in H. Arcs of the complete bipartite graphs receive appropriate costs: an arc from node x to node y receives as cost the reload cost $r_{l\,l'}$, where l and l' are the colors of the arc entering x and of the arc leaving y in G .

A minimum cost $s-t$ path in H corresponds to a minimum transportation cost $s-t$ path in G (eventually visiting a node more than once).

Remarks.

When G is an undirected graph, the splitting procedure has to be modified as follows. Each node v must be substituted by a clique of order equal to the degree of v in G, so that each edge incident in G to v is attached to one and only one node of the clique; each edge of the clique, say from node x to node y, receives an arc cost equal to the reload cost of moving from the color of the edge of G incident to x to that of the edge of G incident to y.

Notice that the results for problem $\mathrm{P1}$ can also be obtained using the line-graph of G, instead of resorting to the splitting procedure.

Problem P2

Problem $\mathrm{P2}$ *is polynomially solvable*.

Problem P2

Problem $\mathrm{P2}$ *is polynomially solvable*.

Apply to G the same splitting procedure of Problem $\mathrm{P1}$ and let H be the resulting graph.

Compute a minimum cost path-tree T in H with origin s using, say, Dijkstra's algorithm.

The paths in T from s to all vertices of H allow to identify a set S of paths from s to all vertices of G, such that the sum of their transportation costs is minimum:

for each node v of G select, in the left shore of the bipartite graph replacing v in H, the node closest to s in H; the path in T from s to this node identifies a path in G from s to v. The resulting set S of paths is a set of minimum transportation cost paths from s that solves $\mathrm{P2}$.

Note - These paths do not usually form a tree of G.

Problem P3

Problem P3 *is* NP-HARD.

Reduction from MIN SET COVER:

Instance: Collection C of subsets of a set Q having q elements, positive integer k.

Question: Does C contain a cover for Q of size k or less?

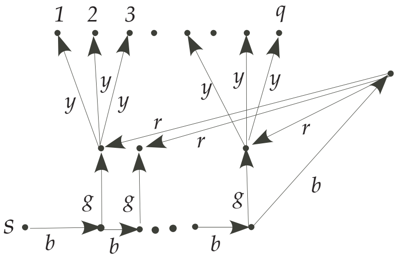
Problem P3

Problem P3 *is* NP-HARD.

Reduction from MIN SET COVER:

Instance: Collection C of subsets of a set Q having q elements, positive integer k.

Question: Does C contain a cover for Q of size k or less?



r_{bg}=1

r_{br}=0

r_{gy}=0

r_{ry}=M

There exists a cover for Q of size $\leq k$ iff the graph has a path-tree from s of reload (transportation) cost $\leq k + q$.

Problem P4

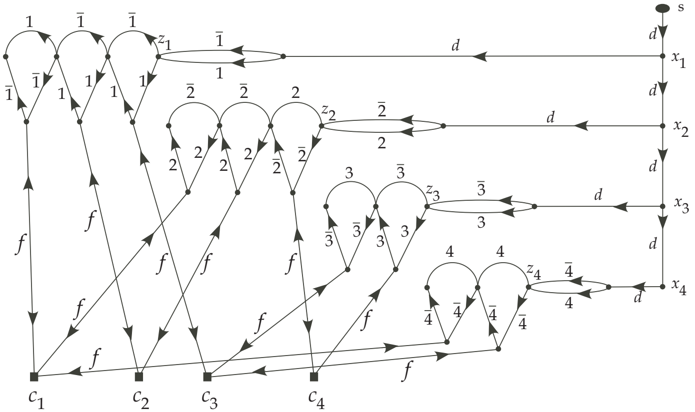
Problem P4 *is* NP-HARD.

Reduction from 3-SAT-3

Instance: set $X = \{x_1, ..., x_n\}$ of boolean variables, collection $C = \{C_1, ..., C_m\}$ of clauses, with $|C_h| \leq 3$, and with at most 3 clauses in C that contain either x_j or $\overline{x_j}$.

Question: does there exist a satisfying truth assignment for C?

(from Giulia Galbiati, to appear in Discrete Appl. Math.(2008))



all reload costs are equal to K except $r_{d,*}$=1

i. I is satisfiable $\implies opt(G) \leq K + 1$

ii. I is not satisfiable $\implies opt(G) \geq 2K + 1$.

The tour problems

The tour problems

Traversing all arcs (edges) of a directed (undirected) graph G with a tour of minimum cost so that each arc (edge) is used at least once is the famous Chinese Postman Problem CPP, which is solvable in polynomial time. We look for a similar tour of minimum transportation cost.

The tour problems

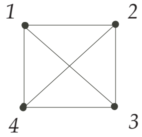
Traversing all arcs (edges) of a directed (undirected) graph G with a tour of minimum cost so that each arc (edge) is used at least once is the famous Chinese Postman Problem CPP, which is solvable in polynomial time. We look for a similar tour of minimum transportation cost.

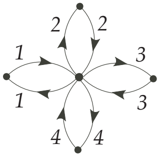
- *Problem* $\mathrm{T1}$: Find a Eulerian tour of minimum transportation cost in Eulerian graph G.
- *Problem* $\mathrm{T2}$: Find a Hamiltonian tour of minimum transportation cost in Hamiltonian graph G.

Both problems are NP-hard even if all arc costs are zero.
The same results hold for undirected graphs.

Problem T1

Reduction from the TRAVELING SALESMAN problem:





$r_{ij} = w_{ij} \quad \textit{for } i,j \in \{1,2,3,4\}$

Problem T2

Reduction from the VERTEX COVER problem (an adaptation of that in Garey and Johnson '79)

The flow problems

Let digraph G have arc capacities, unitary costs, and unitary reload costs. We are given also: origin s, destination t and demand d.

The flow problems

Let digraph G have arc capacities, unitary costs, and unitary reload costs. We are given also: origin s, destination t and demand d.

- *Problem* F1: find a flow of minimum transportation cost from s to t satisfying demand d.

The flow problems

Let digraph G have arc capacities, unitary costs, and unitary reload costs. We are given also: origin s, destination t and demand d.

- *Problem* F1: find a flow of minimum transportation cost from s to t satisfying demand d.

Let digraph G as above. We are given now a set of q origin-destination pairs (s_i, t_i), and corresponding demand d_i.

The flow problems

Let digraph G have arc capacities, unitary costs, and unitary reload costs. We are given also: origin s, destination t and demand d.

- *Problem* F1: find a flow of minimum transportation cost from s to t satisfying demand d.

Let digraph G as above. We are given now a set of q origin-destination pairs (s_i, t_i), and corresponding demand d_i.

- *Problem* F2: find a multicommodity flow of minimum transportation cost, satisfying all demands, with unsplittable commodity flows.
- *Problem* F3: find a multicommodity flow of minimum transportation cost, satisfying all demands, with unsplittable commodity flows, but unlimited arc capacities.
- *Problem* F4: find a multicommodity flow of minimum transportation cost, satisfying all demands, with splittable commodity flows.

Problem F1

Problem $\mathrm{F1}$ *is polynomially solvable*.

The splitting procedure, applied to all nodes except origin s and destination t, reduces $\mathrm{F1}$ to a minimum cost $s - t$-flow problem in network H.

Problem F1

Problem $\mathrm{F1}$ *is polynomially solvable*.

The splitting procedure, applied to all nodes except origin s and destination t, reduces $\mathrm{F1}$ to a minimum cost $s-t$-flow problem in network H.

Problem F2

Problem $\mathrm{F2}$ *is* NP*-hard*.

Reduction from the shortest arc-disjoint q paths problem proved NP-hard in:

U. Brandes, W. Schlickenrieder, G. Neyer, D. Wagner and K. Weihe A software package of algorithms and heuristics for disjoint paths in Planar Networks, DAM Vol. 92(1999) 91-110.

It is enough to set all reload cost equal to 1, costs equal to 0, capacities equal to 1 and demands equal to 1.

Problem F3

Problem $\mathrm{F3}$ *is polynomially solvable*.

An optimum solution is obtained by just superimposing the minimum transportation cost paths for each origin-destination pair.

Problem F3

Problem F3 *is polynomially solvable*.

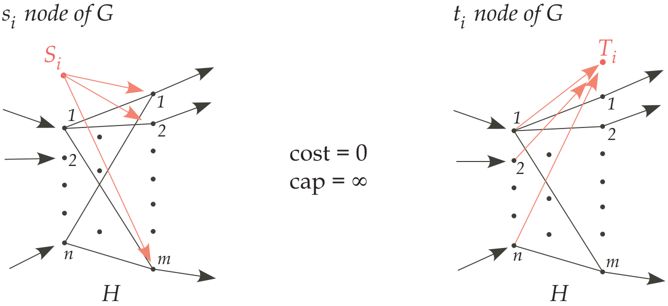
An optimum solution is obtained by just superimposing the minimum transportation cost paths for each origin-destination pair.

Problem F4

Problem F4 *is polynomially solvable*.

It can be reduced to that of finding a splittable multicommodity flow of minimum cost.

Apply the splitting procedure to all nodes. Add to each complete bipartite graph corresponding to an origin (destination) of G a new left shore node S_i (right shore node T_i) connected with arcs from S_i to all nodes of the right shore (from all nodes of the left shore to T_i). All these additional arcs have zero cost and unbounded capacity.



Depending on whether the node in the original graph G is an origin s_i or a destination t_i, the corresponding node S_i or T_i is considered as the origin or destination in the new network.

Thank you for your attention