CIRRELT

Centre interuniversitaire de recherche sur les réseaux d'entreprise, la logistique et le transport

Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation

Computing Shortest Paths with Logistic Constraints

Teodor Gabriel Crainic
Michael Florian
Yolanda Noriega

April 2007

CIRRELT-2007-06

Bureaux de Montréal :
Université de Montréal
C.P. 6128, succ. Centre-ville
Montréal (Québec)
Canada H3C 3J7
Téléphone : 514 343-7575
Télécopie : 514 343-7121

Bureaux de Québec :
Université Laval
Pavillon Palasis-Prince, local 2642
Québec (Québec)
Canada G1K 7P4
Téléphone : 418 656-2073
Télécopie : 418 656-2624

www.cirrelt.ca

Computing Shortest Paths with Logistic Constraints

Teodor Gabriel Crainic[1,2*], Michael Florian[2] , Yolanda Noriega[2]

[1] NSERC Industrial Research Chair in Logistics Management, ESG, Université du Québec à Montréal, C.P. 8888, succursale Centre-ville, Montréal, Canada H3C 3P8

[2] Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT), Université de Montréal, C.P. 6128, succursale Centre-ville, Montréal, Canada H3C 3J7

Abstract. The issue considered in this contribution is the computation of shortest paths on networks that represent freight movements. The features of a particular distribution system can be abstracted with a series of conditions which are referred to as *logistic constraints*. We propose a polynomial shortest path algorithm that takes into account such logistic constraints.

Keywords. Constrained shortest paths, multicommodity, multimodal networks, logistics constraints.

Acknowledgements. Partial support for this project has been provided by the Natural Sciences and Engineering Research Council of Canada (NSERC) through its Discovery Grants program.

---

* Corresponding author: Teodor-Gabriel.Crainic@cirrelt.ca

Dépôt légal – Bibliothèque nationale du Québec,
Bibliothèque nationale du Canada, 2007

INTRODUCTION

Long-distance freight transportation and distribution activities make up a complex system involving a large number of rather diverse stakeholders. Demand for transportation arises from the needs of producers and consumers of goods and services and the significant distances that often separate them. Raw materials and intermediate products and parts must be moved to provide the means of producing the finite products which are then distributed via various channels to the final consumers. Carriers answer this demand by supplying transportation services. Railroads, ocean, coastal, and river shipping lines, and trucking companies are examples of carriers. Considering the type of services they provide, ports, intermodal platforms, and other such facilities may be described as carriers as well. Finally, governments contribute the infrastructure, roads and highways, as well as significant portions of ports, internal navigation, and rail facilities, while also regulating and taxing the industry.

The distribution of product flows in such networks is the result of the combined decisions concerned stakeholders take individually or in collaboration. Thus, the producers or consumers of the goods select a transportation mode or an intermodal transportation chain to satisfy their requirements in terms of cost, delivery time, quality of service, and so on. Alternatively, they may call upon an intermediate firm (e.g., a carrier, a broker, a third-party logistics company, a service integrator, etc.) to perform the service within specified performance levels. Then, according to the volumes and values of the respective goods, these intermediate firms may ship them individually or consolidate them with other shipments for part or the entire length of the trip. Detailed consolidation and routing decisions are then taken by the respective carriers according to their own network configurations, operating policies, and economic and regulatory environment. All such decisions are product specific in most cases, even when the same stakeholders are involved. The globalization of the economy and international exchanges, together with the emergence of integrated networks of production and distribution enterprises and the advanced planning of world-wide logistic (value) chains is further defining transportation and distribution channels and strategies for particular products.

This evolution challenges the methods used to represent, analyze, and predict the behaviour of freight transportation systems. Part of the answer to this challenge resides in a more detailed network representation of the multimodal transportation system and the development of more refined demand, mode choice, and assignment models (see Crainic and Florian, 2006 and Crainic and Kim, 2007, for reviews of these methodologies). This paper aims to contribute toward this goal by focussing on a key component of these methodologies, the representation and computation of shortest paths. Indeed, the basic building block of system optimal, user optimal, and stochastic assignment is the shortest path routine. Once this sub-problem is solved by an efficient algorithm it may serve for the construction of any assignment method based on these principles.

The issue thus considered in this contribution is the computation of shortest paths on networks that represent freight movements through multimodal systems. The modeling of paths used in freight transportation must take into account not only the modes that make up the transportation chain for a given product (and set of origins and destinations, eventually), but also the way the corresponding carrier networks and the particular logistic distribution systems are operated. As a consequence, the modeling and computation of such paths are considerably more complex than those for the transportation of persons (e.g., Florian and Hearn, 1995). Other than classical mode choice specifications, the features of a particular distribution system can be abstracted with a series of conditions which are referred to as *logistic constraints*. The objective is to propose and develop a shortest path algorithm that takes into account such logistic constraints.

There are no previous contributions to this particular shortest path problem we are aware of (see Gallo and Pallottino, 1986, 1988; Ahuja, 1997; Pallottino and Scutellà, 1998). It is worth mentioning that the proposed algorithm uses multiple link labels which are reminiscent of the shortest path algorithms with resource constraints (see, for instance, Desrochers, 1986; Desrochers and Soumis, 1988; Desrochers, Desrosiers, and Solomon, 1992 and Feillet *et al.*, 2004) and of algorithms for bi-criterion and weight-constrained shortest paths (e.g., Dumitrescu and Boland, 2003).

The paper is organized as follows. The notation used is given in Section 1 and the following section considers the concept of a logistic chain and its characterization with logistic constraints in order to compute logistic paths. Section 3 considers the way that logistic constraints may be satisfied in a shortest path computation. Section 4 provides a statement of the algorithm and a demonstration of its complexity. Data structure issues are considered in Section 5. Numerical results obtained with an implementation of this algorithm are given in Section 6 before conclusions are given in the last section.

1. NOTATION FOR FREIGHT NETWORKS

The problems considered involve the movement of, potentially different, product flows between given origin and destination points, over multimodal transportation networks. The representation of the transportation supply takes thus the form of a multimodal network. Here, a mode is a means of transportation with particular characteristics, such as vehicle type and capacity, as well as specific cost measures. Depending on the scope and level of detail of the study, a mode may represent a carrier or part of its network representing a particular transportation service, an aggregation of several carrier networks, or specific transportation infrastructures such as ports. To further capture the modal characteristics of transportation and to simplify the notation, we consider modal links.

The multimodal network is denoted (M,N,A,T), where M is the set of modes, $m \in M$, N is the set of nodes $i, j \in N$, A is the set of modal links $(i, j, m) \in A$, and T is the set of transfers.
It is important to note that parallel links are allowed between two nodes if there is **more** than one modal link between two nodes, to represent situations where more than one mode is available for transporting goods between two adjacent nodes; i.e. $(i, j, m) \in A$ and $(i, j, \hat{m}) \in A$ if $(m \neq \hat{m})$.

A intermodal transfer $t \in T$ is defined as the sequence of two modal links at a node as the quintuple "from-node, from-mode, at-node, to-node, to-mode" or $\left[(i,m)\, j\,(k,\hat{m})\right]$. In order to simplify the notation, a transfer will be denoted $(a,\hat{a})$ where $a = (i, j, m)$ and $\hat{a} = (j, k, \hat{m})$ are modal links. A path from an origin to a destination may use several modes, but, a mode change can occur ***only*** at a transfer mode. Some examples are the transfer of goods from ship to rail or truck in ports, the transfer of goods from truck to rail in a train yard and so on.

The cost of a link is denoted $c(a)$ and the cost of a mode to mode transfer is denoted $c(a,\hat{a})$. The particular meaning of "cost" is application specific but has no relevance on the shortest path algorithm we propose. The paths are defined from origins $o, o \in O$ contained in N to destinations $d, d \in D$ contained in N. An origin-destination pair will be denoted O-D.

A freight network assignment method that minimizes total costs will distribute the product flows over the paths of the network for each set of origin-destination pairs, according to a given mode choice for each product given the current value of the link costs. However, not all the conditions determining the possible itineraries of given product flows can be captured through these costs. For example, interchanges between two carriers, e.g., railroads may take place at pre-specified facilities which do not correspond to a cost minimizing flow; consolidation carriers, e.g., railroads, less-than-truckload motor carriers, container shipping lines, etc., move freight through major consolidation terminals such as rail yards, breakbulks, and container-port terminals, respectively; particular contracts may govern the flow of products out of specific production zones through particular facilities and transportation channels, and so on. We introduce in the following the concept of *logistics constrains* to represent such particular conditions and force them on the computation of the path used to transport the corresponding commodities.

2. LOGISTIC CHAINS, LOGISTIC CONSTRAINTS AND LOGISTIC PATHS

A *logistic chain* specifies the sequence of transportation modes, transfer terminals and (eventually) storage facilities used to transport a product (or a set of products) from given origin(s) to given destination(s). Here are a few examples of logistic chains:

- Ship newsprint by truck to a port → by specialized vessel to particular destination port → by truck to a warehouse → by truck to a printing location;
- Ship oranges by truck to airport → by one of a set air freighters to the destination airport → by truck to a warehouse → by truck to store;
- Ship furniture by truck to a specific port → load in container and use one of a set of container liner services to one of a set of destination ports → ship to a warehouse by train → unload the containers at the warehouse and store → deliver by truck to the final destination;
- Ship a train full of automobiles on a given service to a given port → load on the ship of a give line → move to a destination port → unload and move using a given unit train service to the main distribution center;
- Ship by train from the local station (plant is linked to the rail network) to one of the marshalling yards of the railway → then on a train to the final local station.

A *logistic path* connects an origin O to a destination D and satisfies certain *logistic constraints*, so called because they aim to represent logistic chains. The specification of the logistic path that models a specific logistic chain of a distribution system, for a particular product, is defined as follows (elements within [] are optional since they depend on the number of such constraints):

[origin(s), destination(s)], and a sequence of *k, k=1, 2, …, K,* logistic constraints which are defined as sets of modes and intermediate nodes:

[modes (1), intermediate nodes (1),
modes (2), intermediate nodes (2),
modes (3), intermediate nodes (3),
. . . .
modes (*K*), intermediate nodes (K),]
modes

Each set of **modes** indicates that the chain must use at least one of the modes included in that set. Each set **of intermediate nodes** indicates that the chain must pass through at least one of the nodes of the set, which may contain regular nodes or nodes with transfers where a mode change may occur. In particular, a path must pass through at least one of the nodes of intermediate node set (1) before it can access the modes and services of set (2) and so on until constraint (*K*) is satisfied. In general, the sets of intermediate modes and nodes of the different constraints are not necessarily disjoint. Finally, the logistic constraints must include at least one set of **modes** specified in order to reach the destination (if a path exists). The sets of origins and destinations are optional since the logistic constraints may apply to all the O-D pairs of the network. Furthermore, all the destinations are always considered for selected origins.

The number of logistic constraints may be very large. An example of a logistic path that requires only *four* such constraints are:

- Modes for the links from the origin centroid (representing the zone) to the exit ponts (nodes) from the zone; $k = 1$;
- Modes for local transportation until the “origin” transfer or consolidation terminal; $k = 2$;
- Modes for long-haul transport until the “destination" transfer or consolidation terminal; $k = 3$;
- Modes and services for local transportation until the entry points into destination zone; $k = 4$;
- Modes to the destination centroid (zone).

3. SHORTEST PATHS WITH LOGISTIC CONSTRAINTS - CONCEPT

In this section the shortest path algorithm that takes into account logistic constraints is developed. There are trivial instances of this problem where a shortest path is required to visit a particular set of intermediate nodes, in sequence, such as the itinerary of a particular vehicle. The shortest path may be found by simply applying sequentially a conventional shortest path algorithm from the origin to each intermediate node and then between intermediate nodes until the destination. The problem that we consider requires the path to use one node of a set of successive node sets, which are not necessarily contiguous, and permits the use of any one of a set of modes between each such successive set of intermediate nodes.

It is worthwhile to note that in most cases the intermediate node sets ***do not*** form a cut in the network. This implies that one cannot decompose the problem along intermediate node sets. Hence one has to explore sub-paths until a path that satisfies all constraints is found. This may be determined in some cases only at the destination, if an allowed mode in the first constraint set provides a complete path which does not satisfy the constraints (e.g., the road mode in most cases).

Consider now a label setting algorithm that labels links with tentative shortest path distances, predecessors, and the state of the sub-path. The state of the sub-path constructed accounts for the number of logistic constraints that are satisfied. Since the freight network contains transfers, it is natural to use link labels (intuitively, the beginning of the link) and node labels for the origins and the destinations. This is similar to the shortest path algorithm used in Guelat, Florian, and Crainic (1990).

Let the link labels be denoted: $\left[u_{ak}, p_{ak}\right], k = 0,...,K$, where u_{ak} denotes the cost to link "a" associated with k satisfied constraints, p_{ak} is the pointer to the predecessor (origin or link) network element which was used for computing u_{ak} . Hence the algorithm would generate labels:

$$
\begin{array}{l}
\left[u_{a0}, p_{a0}\right] \\
\left[u_{a1}, p_{a1}\right] \\
\vdots \\
\left[u_{aK}, p_{aK}\right]
\end{array}
$$

where $p_{ak} = (\tilde{a}, \tilde{k}) = \left(\left(\tilde{i}, \tilde{j}, \tilde{m}\right), \tilde{k}\right)$ is the pointer to the predecessor link $\tilde{a}$ with $\tilde{k}$ satisfied constraints.

The shortest path labelling of all links reachable from the head node *j*, node of the current link, must be done with multiple labels, each one corresponding to a number of satisfied logistic constraints, if they are feasible. A link may be reached by more than one sub-path; each of these sub-paths may have a different number of logistic constraints satisfied. Since the intermediate nodes do not necessarily form a cut in the permitted sub-network between O and D, sub-paths that reach nodes that do not belong to the intermediate set must be continued until a better sub-path eliminates it. Forward labelling is done from all present link labels of the current link.

The labels of all states are kept in one common binary heap. Hence one constructs simultaneously the shortest paths that satisfy no constraints, some constraints and all constraints. The data structure to be used for implementing the computations is discussed in detail in Section 5 after the statement of the algorithm.

Denote by $m\left(k+1\right)$ and $n\left(k+1\right)$ the modes and intermediate nodes permitted in the constraint set corresponding to state k $\left(k = 0,1,...,K\right); m(K+1)$ is the last set of modes that permit access to the destination(s).

Starting from the origin node, the links are examined by following the forward star of the j-node of a link that has the current lowest cost label. The label of this link becomes permanent at each step of the algorithm. The successors receive updated labels in the basic recursion step.

4. THE ALGORITHM AND ITS COMPLEXITY

The algorithm proceeds origin by origin.

$\left[u_{ak}, p_{ak}\right]$, for $k = 0,1,...K$ and $a \in A$ are the link labels.
$\widetilde{A}$ is the set of links to label.

Shortest path algorithm with logistic constraints

Step 0 *Initialization of labels* $\left[u, p\right]$

For $k = 0$ to K do :
$u_{ok} = 0$; $u_{dk} = \infty, d \in D$; $u_{ak} = \infty, a \in A$
$p_{ok} = 0$; $p_{dk} = -1, d \in D$; $p_{ak} = 0, a \in A$
$\tilde{j} = o$; $\tilde{k} = 0$
Go to Step 3

Step 1 *Choice of arc to label*

If $\tilde{A} = \phi$ go to step 4
Choose $\left(\tilde{a}, \tilde{k}\right)$ where $\tilde{a} = (\tilde{i}, \tilde{j}, \tilde{m})$ such that $u_{\tilde{a}\tilde{k}} \leq u_{ak}, a \in \tilde{A}$
$\tilde{A} = \tilde{A} - \left\{(\tilde{a}, \tilde{k})\right\}$
If $\tilde{j}$ is a regular or a transfer node go to Step 3
If $\tilde{j}$ is a destination go to Step 2

Step 2 *Node* $\tilde{j}$ *is a destination node*

If $u_{\tilde{a}\tilde{k}} < u_{d\tilde{k}}$ then $u_{d\tilde{k}} = u_{\tilde{a}\tilde{k}}$; $p_{d\tilde{k}} = (\tilde{a}, \tilde{k})$
Return to Step 1

Step 3 *Regular or transfer node: scan of successors*
If $\tilde{j} \notin n(\tilde{k}+1)$ then consider links with modes belonging to $m(\tilde{k}+1)$ if they exist.
For each link $a = (i, j, m)$ such that $i = \tilde{j}$ do:
If there is a transfer $t = (\tilde{a}, a)$ then:

If $u_{\tilde{a}\tilde{k}} + c(\tilde{c}, a) + c_a < u_{a\tilde{k}}$ then
$u_{a\tilde{k}} = u_{\tilde{a}\tilde{k}} + c(\tilde{a}, a) + c_a$
$p_{a\tilde{k}} = (\tilde{a}, \tilde{k})$
$\tilde{A} = \tilde{A} \cup \left\{\left(a, \tilde{k}\right)\right\}$

Otherwise
If $u_{\tilde{a}\tilde{k}} + c_a < u_{a\tilde{k}}$ then $u_{a\tilde{k}} = u_{\tilde{a}\tilde{k}} + c_a$
$p_{a\tilde{k}} = (\tilde{a}, \tilde{k})$
$\tilde{A} = \tilde{A} \cup \left\{\left(a, \tilde{k}\right)\right\}$

If $\tilde{j} \in n(\tilde{k}+1)$ then consider links with modes belonging to $m(\tilde{k}+2)$ if they exist.
For each link $a = (i, j, m)$ such that $i = \tilde{j}$ do:
If there is a transfer $t = (\tilde{a}, a)$ then:
If $u_{\tilde{a}\tilde{k}} + c(\tilde{a}, a) + c_a < u_{a\tilde{k}+1}$ then
$u_{a\tilde{k}+1} = u_{\tilde{a}\tilde{k}} + c(\tilde{a}, a) + c_a$
$p_{a\tilde{k}+1} = (\tilde{a}, \tilde{k})$
$\tilde{A} = \tilde{A} \cup \left\{\left(a, \tilde{k}+1\right)\right\}$

Otherwise
If $u_{\tilde{a}\tilde{k}} + c_a < u_{a\tilde{k}+1}$ then $u_{a\tilde{k}+1} = u_{\tilde{a}\tilde{k}} + c_a$
$p_{a\tilde{k}+1} = (\tilde{a}, \tilde{k})$
$\tilde{A} = \tilde{A} \cup \left\{\left(a, \tilde{k}+1\right)\right\}$

Return to Step 1

Step 4 *Determine if an optimal path exists (for destination j)*

If $u_{jK} < \infty,\ j \in D$ then an optimal path was found for destination j.
Otherwise, no optimal path exists for destination j.

□

The algorithm is polynomial. To see this, consider that the network is composed of K disjoint paths:

- one that has 0 constraints satisfied;
- one that has 1 constraint satisfied;
- one that has 2 constraints satisfied;
- one that has 3 constraints satisfied;
- ……………………………………
- one that has K constraints satisfied.

The algorithm will actually compute K shortest paths and the one that satisfies the logistic constraints is found at the destination node. Each of these computations is of the order of $(|A|+|T|)log2(|A|+|T|)$, where $|A|$ is the cardinality of the set of links and $|T|$ is the cardinality of the set of transfers. Hence the worst case complexity is $K*(|A|+|T|)log2(n*|A|+|T|)$.

5. DATA STRUCTURE ISSUES

The implementation of the prototype code uses the following data structures. The aim of this implementation was to achieve a reasonably efficient code. The data structures may be further improved if one were to implement the algorithm for very large scale computations.

The modes are stored in an array. The nodes are stored in two different data structures. The first is an array which is sorted in increasing order of node ID's. The elements are retrieved by using a binary search. The second consists of a hash table. The elements are accessed by using a hash function.

Most of the remaining elements are stored in lists in order to use dynamic memory allocation and thus achieve relatively fast access.

Each origin node has two lists which contain the logistic constraints, one for the modes and the other one for the nodes. The algorithm computes the shortest path from an origin towards all the destinations and the constraints may differ among origins.

Each node has a list with its outgoing links, or a "forward star" representation. Each link has a linked list with the permitted transfers (if they exist) and another one with the link labels. These labels contain information about the costs and the preceding links.
Figure 1 shows the organisation of the data structures starting by the nodes. A binary heap (through a vector) is used to manage the set A containing the links to label.

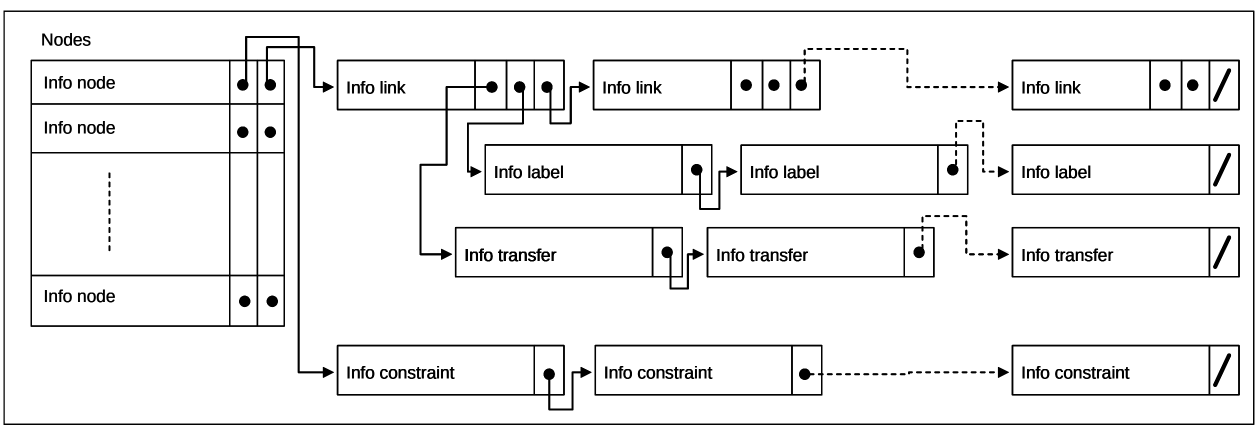


Figure 1 Node's related data structures

Other data structures may be used; however, the performance of the algorithm with and without the logistic constraints may be properly evaluated with this implementation.

6. SOME NUMERICAL EXAMPLES

The algorithm was applied to several simple networks to illustrate its operation. These examples are given in the Appendix so that an interested reader can follow the details of the computations. The link labels as well as the elements of the set $\widetilde{A}$ at each step of the algorithm are showed.

In addition, tests were carried out on freight networks that have been used in the application of the STAN (INRO, 2002) software for Canada and Sweden. Table 1 shows the dimensions of these networks.

	Canada	Sweden
Modes	8	10
Centroids	67	287
Reg. nodes	668	5,450
Links	3,580	15,232
Transfers	2,028	1,893

Table 1 Size of the test networks

The algorithm was programmed in C++ and executed on two computers: an Intel Pentium 4, 2.4 GHz PC (under Win XP, 512 Mb RAM, Microsoft Visual C++), and a SUN workstation (sun4u Sun Fire 880, 3 Mb RAM 1200 MHz [4 CPUs], under Ultra-Sun-Solaris 2.10, GNU C++). The programs were compiled using the O2 optimization option.

We computed shortest paths for five different sets of constraints for the freight network of Canada. For this application, all the destinations were considered for each origin. Please refer to Figure 2 to identify the cities included in the constraints and see the division between the Eastern and Western Canada areas. The logistic constraints were defined as follows:

- Set 1:
  All the origins to all the destinations. No logistic constraints at all.
- Set 2:
  For origins Moncton, Rimouski and Quebec: road until Montreal, CN railway to Calgary, road to destination. No constraints for the remaining origins.
- Set 3:
  For origins Moncton, Rimouski and Quebec: road until Montreal, CP railway to Calgary, road to destination. No constraints for the remaining origins.
- Set 4:
  For east origins: road until Sudbury, CN railway to Winnipeg, road to destination.
  For west origins: road until Winnipeg, CN railway to Sudbury, road to destination.
- Set 5:
  For east origins: road until Sudbury, CP railway to Winnipeg, road to destination.

For west origins: road until Winnipeg, CP railway to Sudbury, road to destination.

Table 2 shows the computing times for the Canada network examples.

		Intel WinXP		SUN Solaris	
		Binary search	Hash table	Binary search	Hash table
Set 1	Printing results	4.04	4.24	1.40	1.29
	Without printing	1.34	1.54	0.67	0.70
Set 2	Printing results	4.15	4.36	1.47	1.36
	Without printing	1.30	1.60	0.70	0.72
Set 3	Printing results	4.17	4.36	1.47	1.36
	Without printing	1.30	1.59	0.69	0.71
Set 4	Printing results	4.98	5.19	1.81	1.68
	Without printing	1.44	1.72	0.86	0.87
Set 5	Printing results	5.14	5.37	1.87	1.74
	Without printing	1.42	1.71	0.86	0.87

Table 2 The freight network of Canada: average computing times (in sec)

The next example concerns the freight network of Sweden. Seven constraint sets were defined for this network. In all these computational experiments only one origin is considered for the shortest path calculations to all destinations. The freight network of the Sweden application is presented in the Figure 3. It highlights the modes of transportation and the cities used in the logistic constraints. The logistic constraint sets are as follows:

- Set 1:
  Origin Alvsbyn; no constraints.
- Set 2:
  Origin Alvsbyn (in the north): any mode to the Gulf at Lulea or Ojebyn, then ship to Stockholm, then train (mode « j ») to Vara, then any mode to destination.
- Set 3:
  Origin near Mora / Orsa / Rattvik; no constraints.

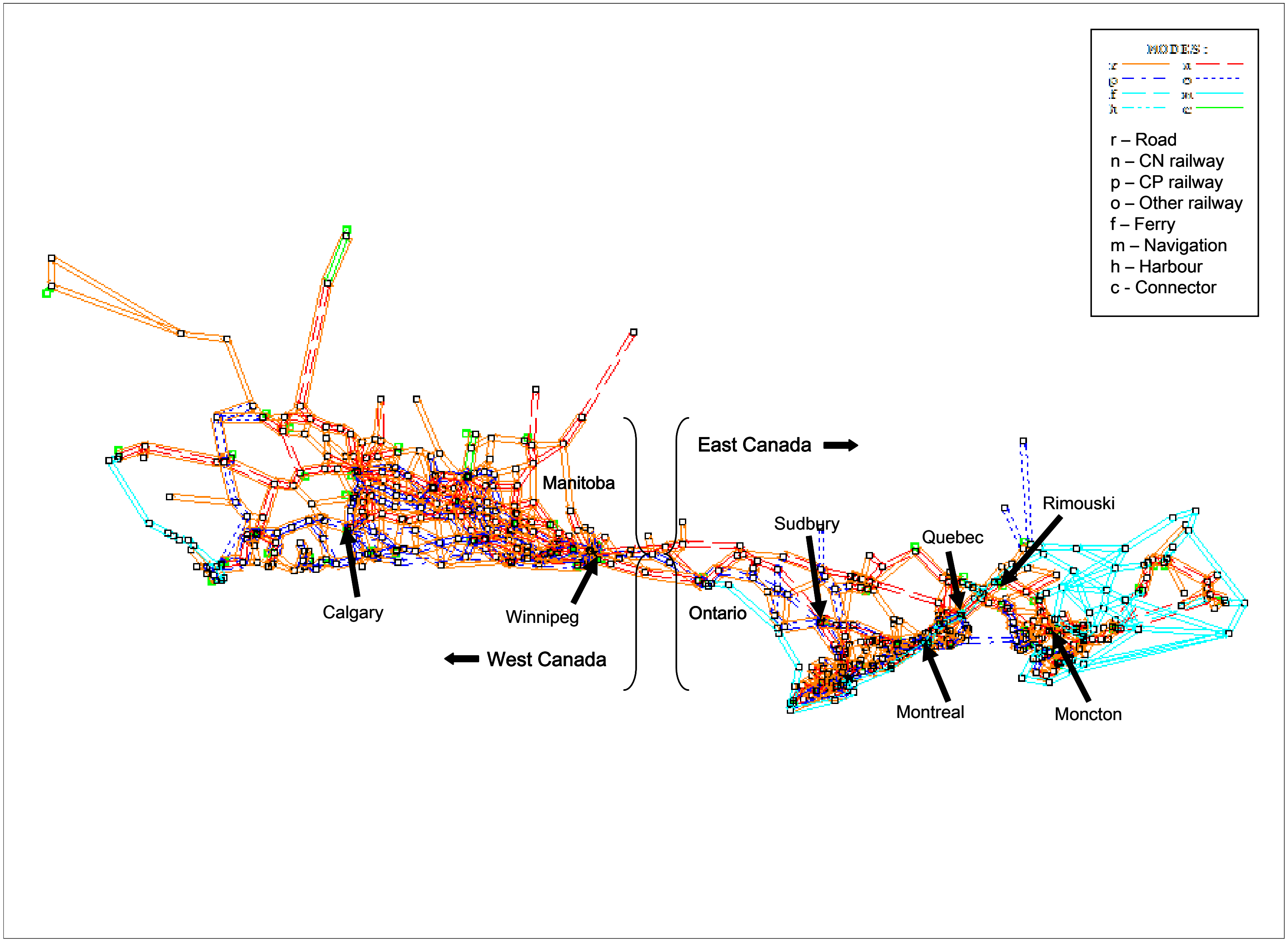


Figure 2 The freight network of Canada

- Set 4:
  Origin near Mora / Orsa / Rattvik: any mode to Mora, then train (mode « j ») to Hassleholm, then any mode to destination.
- Set 5:
  Origin Karlstad; no constraints.
- Set 6:
  Origin Karlstad: train (mode « q ») near to Hallsberg, then any mode to destination.
- Set 7:
  Origin Karlstad: any mode to Kil, then train (mode « j ») to Borlang, then any mode to destination.

Table 3 shows the computing times for the Sweden network examples.

		Intel Win XP		SUN Solaris	
		Binary search	Hash table	Binary search	Hash table
Set 1	Printing results	0.68	0.65	0.22	0.20
	Without printing	0.09	0.10	0.06	0.07
Set 2	Printing results	0.94	0.89	0.35	0.33
	Without printing	0.25	0.24	0.17	0.18
Set 3	Printing results	0.71	0.66	0.23	0.21
	Without printing	0.09	0.10	0.06	0.07
Set 4	Printing results	0.73	0.68	0.26	0.24
	Without printing	0.16	0.16	0.11	0.12
Set 5	Printing results	0.61	0.56	0.20	0.18
	Without printing	0.09	0.10	0.06	0.07
Set 6	Printing results	0.48	0.46	0.17	0.16
	Without printing	0.09	0.09	0.07	0.07
Set 7	Printing results	0.79	0.78	0.29	0.26
	Without printing	0.18	0.18	0.12	0.13

Table 3 The Sweden freight network: average computing times (in sec)

It is straightforward to see that the addition of the logistic constraints does not increase the computing times significantly. In some cases, the computing times with logistic constraints (Set 6 - Sweden) are smaller than those without (Set 5 - Sweden).

Figures 4 and 5 show the shortest path found by the algorithm in the cases without constraints and with logistic constraints. In the case with constraints the path is restricted to use the Ship mode between the Golf (in the north) and the city of Stockholm.

CONCLUSIONS

A new shortest path algorithm that computes shortest paths with logistic constraints has been developed and tested. In spite of the fact that the algorithm requires the verification of the logistic constraints and handles multiple labels as required by the method, the computing times are only slightly longer, in the worst cases seen in our experiments, than those required for computing a simple shortest path. Hence it has the potential to be integrated in a freight assignment method that considers logistic aspects of the freight network studied.

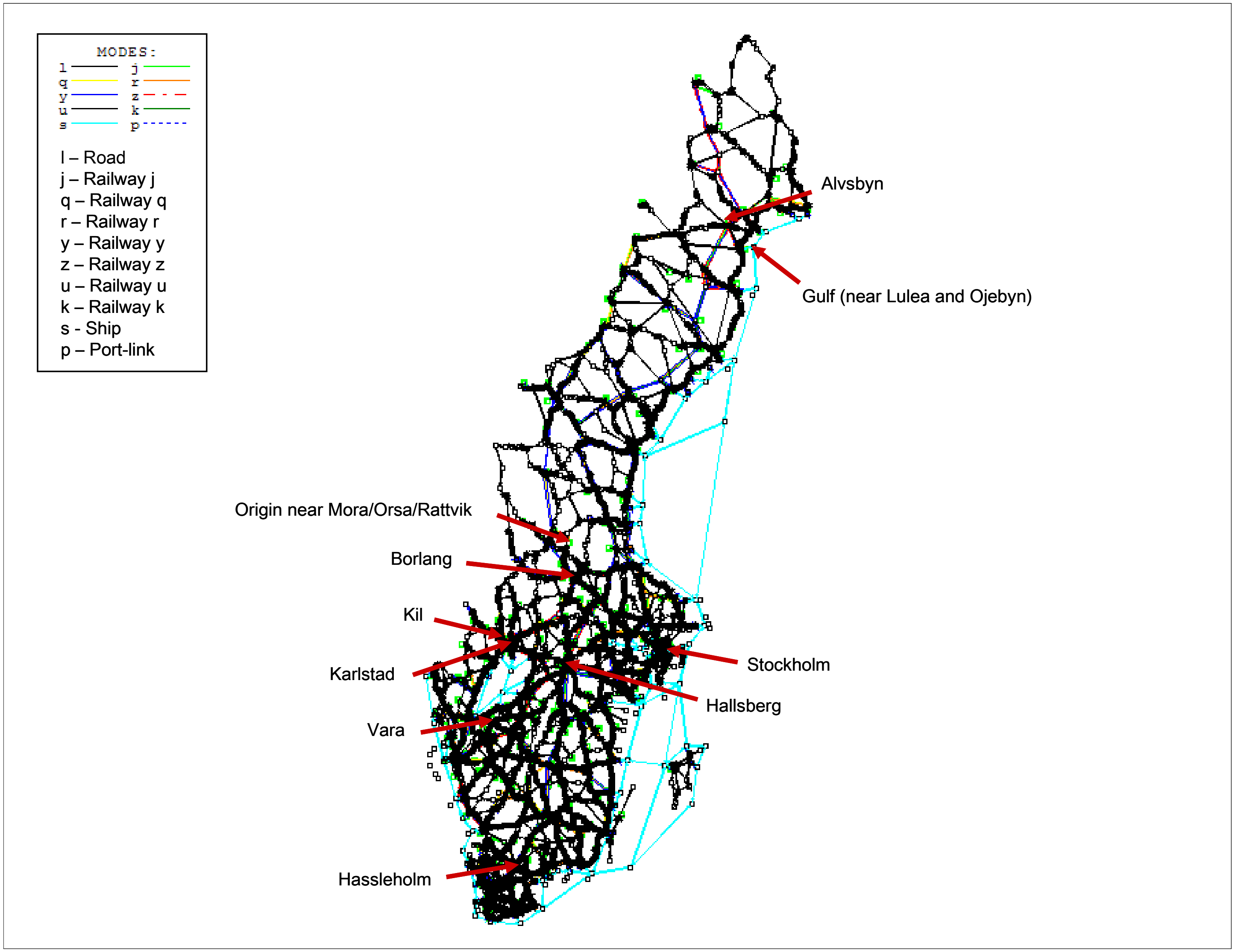


Figure 3 The freight network of Sweden

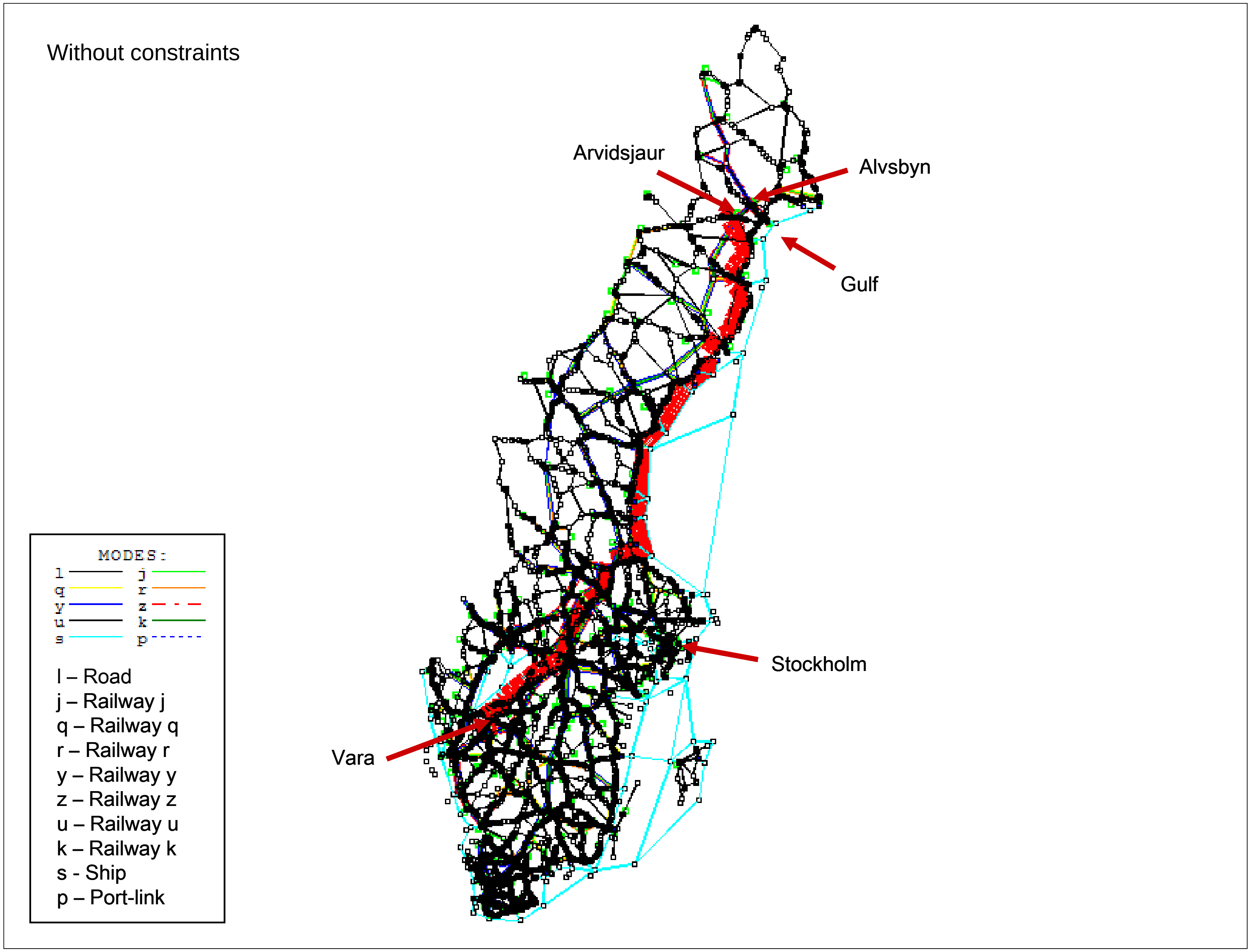


Figure 4 Shortest path between Arvidsjaur and Vara

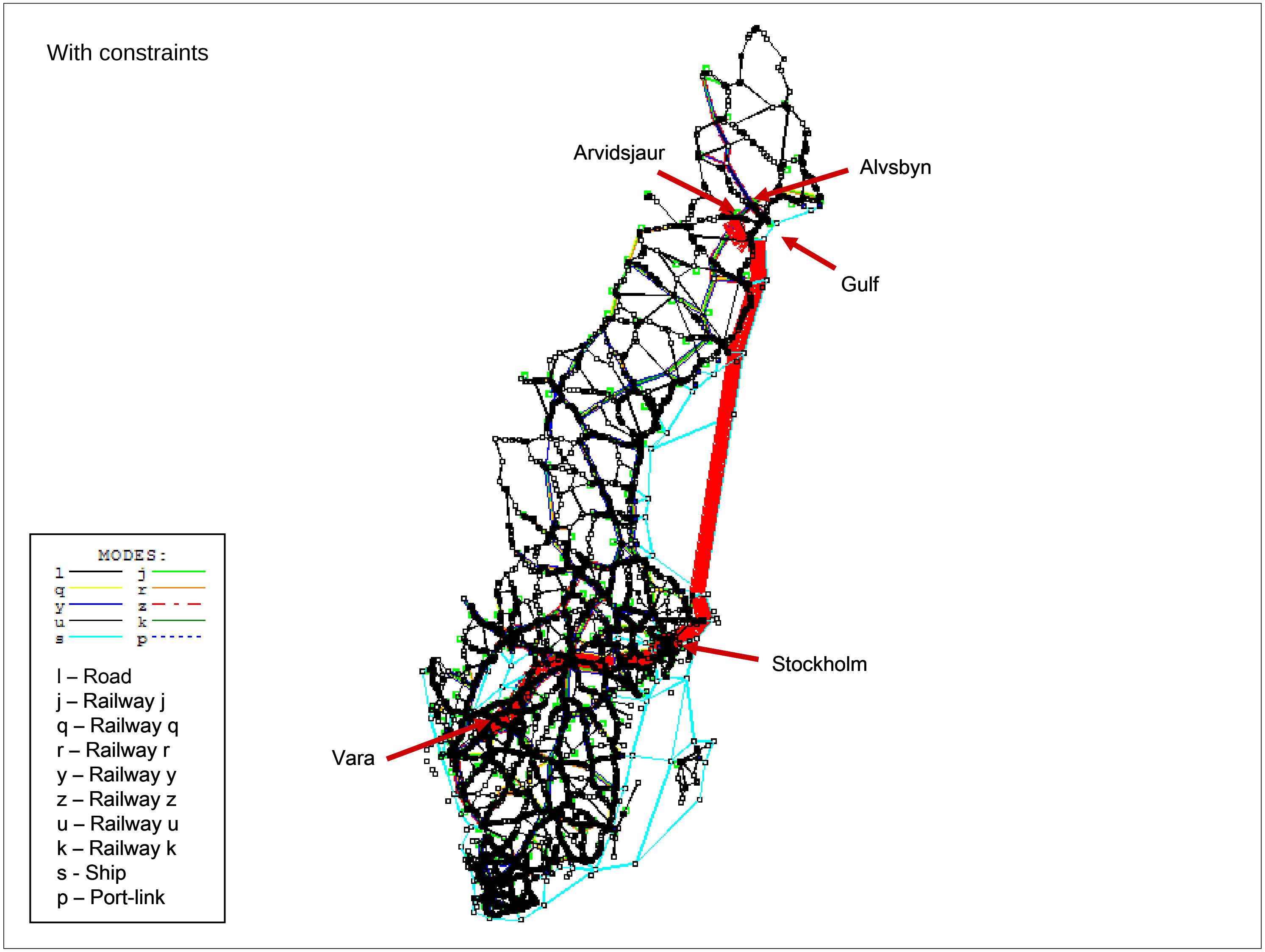


Figure 5 Shortest path with constraints between Arvidsjaur and Vara

ACKNOWLEDGMENTS

Partial support for this project has been provided by the Natural Sciences and Engineering Research Council of Canada (NSERC) through its Discovery Grants program.

REFERENCES

AHUJA, R.K., Flows and Paths, in *Annotated Bibliographies in Combinatorial Optimization*, Dell'Amico, M., Maffioli, F., Martello, S. (Eds.), John Wiley & Sons, New York, NY, 283-309, 1997.

CRAINIC, T.G., FLORIAN, M., Modèles d'optimisation pour la planification des systèmes de transport, in *Optimisation combinatoire : applications*. V. Th. Paschos (ed.). Hermès, Paris. 117-152, 2006.

CRAINIC, T.G., KIM, K.H., Intermodal transportation, in *Transportation*, *Handbooks in Operations Research and Management Science*, C. Barnhart et G. Laporte (Eds.), North-Holland, Amsterdam, 2007

DESROCHERS, M., An algorithm for the shortest path problem with resource constraints. Publication 421A, Centre de recherche sur les transports, 1986.

DESROCHERS, M., DESROSIERS, J., SOLOMON, M., A new optimization algorithm for the vehicle routing problem with time windows. *Operations Research*, 40(2):342-354, 1992.

DESROCHERS, M., SOUMIS, f., A Generalized Permanent Labelling Algorithm for the Shortest Path Problem with Time Windows, *INFOR* 26(3): 191-212, 1988.

DUMITRESCU, I., BOLAND, N., Improved preprocessing labelling and scaling algorithms for the weight-constrained shortest path problem. *Networks*. 42:136-153, 2003.

FEILLET, D., DEJAX, P., GENDREAU, M., GUEGUEN, C., An exact algorithm for the elementary shortest path problem with resource constraints : application to some vehicle routing problems. *Networks*, 43:216-229, 2004.

FLORIAN, M., HEARN, D., Network Equilibrium Models and Algorithms, in *Network Routing*, Volume 8 of *Handbooks in Operations Research and Management Science,* Ball M., Magnanti, T.L, Monma, C.L., and Nemhauser, G.L. (Eds.), North-Holland, Amsterdam,485-550, 1995.

GALLO, G., PALLOTTINO, S., Shortest path methods: a unifying approach. *Mathematical Programming Study*, 26:38-64, 1986.

GALLO, G., PALLOTTINO, S., Shortest path algorithms, *Annals of Operations Research*, 7:3-79, 1988

GUELAT, J., FLORIAN, M., CRAINIC, T.G., A Multimode Multiproduct Network Assignment Model for Strategic Planning of Freight Flows, *Transportation Science* 24(1): 25-39. 1990.

INRO CONSULTANTS INC., STAN On-line Documentation, Release 6.2, 2002.

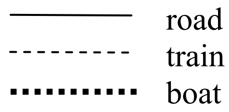
PALLOTTINO, S., SCUTELLÀ, M.G., Shortest Path Algorithms in Transportation Models: Classical and Innovative Aspects, in *Equilibrium and Advanced Transportation Modelling*, Marcotte, P. and Nguyen, S. (Eds.), Kluwer Academic Publishers, Norwell, MA, 245-281, 1998.

APPENDIX

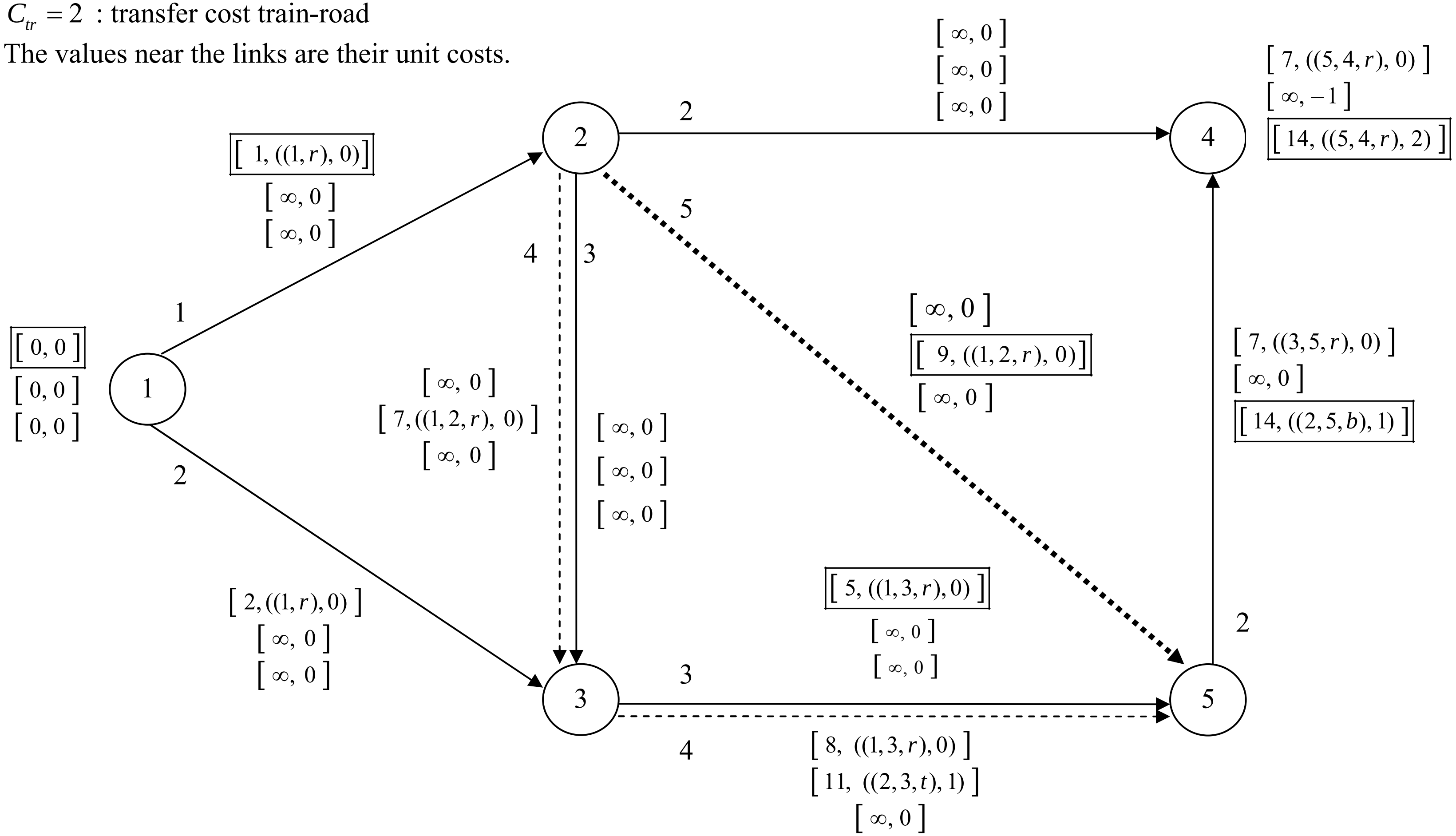
Figure 1.

$C_{rb} = 3$: transfer cost road-boat
$C_{br} = 3$: transfer cost boat-road
$C_{rt} = 2$: transfer cost road-train
$C_{tr} = 2$: transfer cost train-road

The values near the links are their unit costs.



Constraints : $\{(*,2),(t/b,5),*\}$



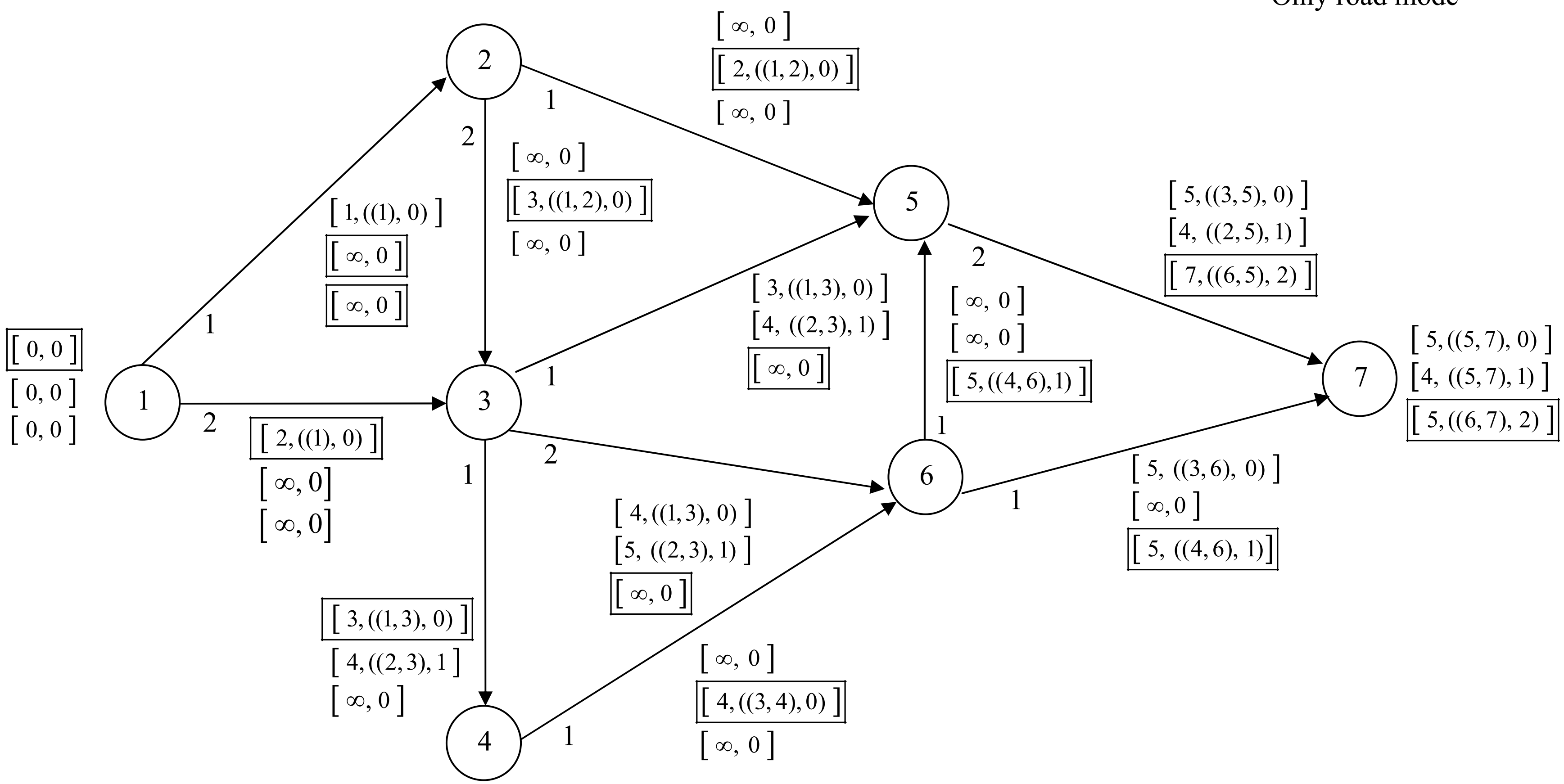
$$\tilde{A} = \{ ((1,2,r),0), ((1,3,r),0), ((2,5,b),1), ((2,3,t),1), ((3,5,r),0), ((3,5,t),0), ((5,4,r),0), ((3,5,t),1), (5,4,r),2) \}$$

(Cost)	(1)	(2)	(9)	(7)	(5)	(8)	(7)	(11)	(14)

Figure 2.

The values near the links are the link's costs.

Constraints : $\{(2/4), 6\}$
Only road mode

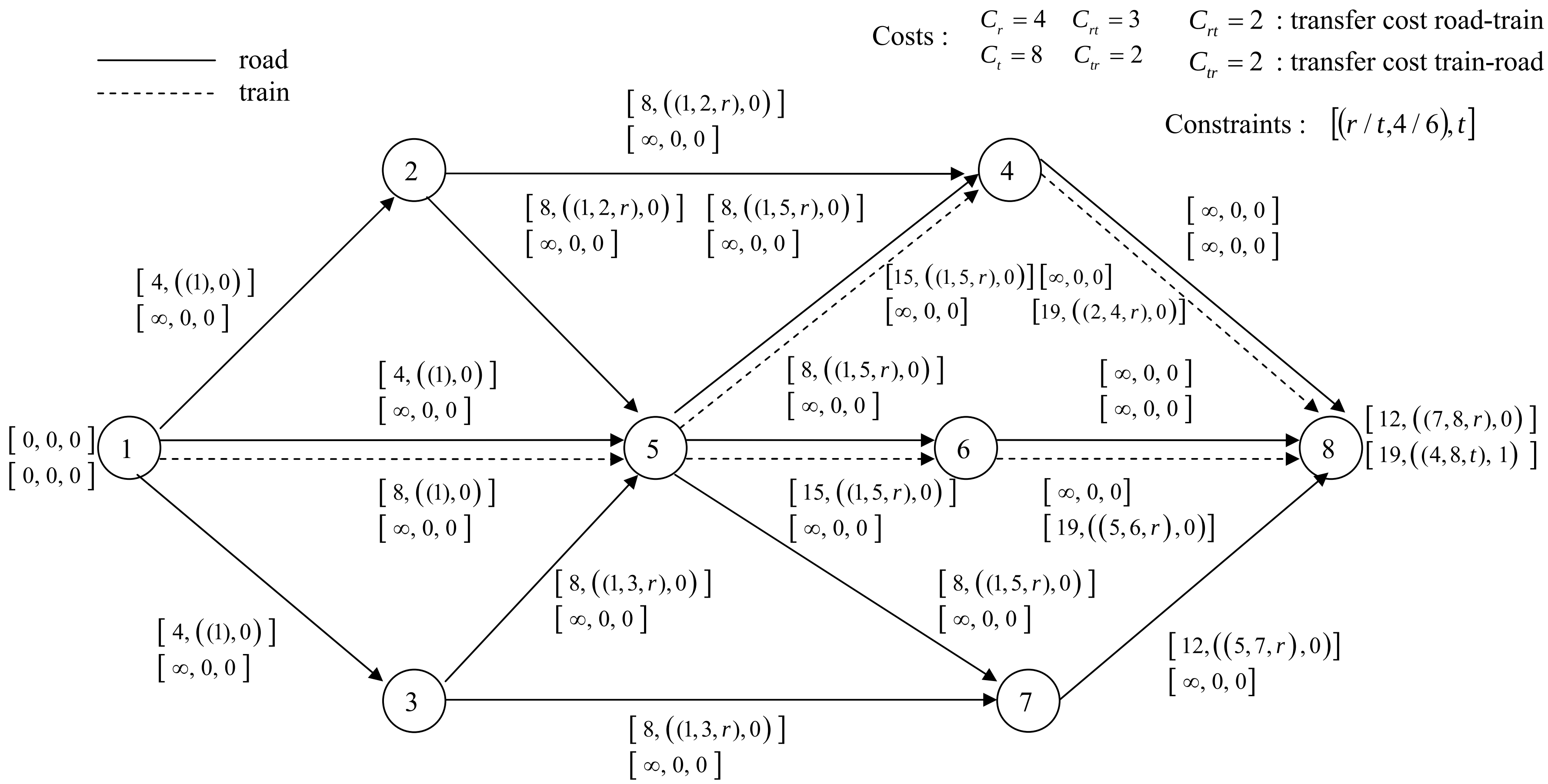


$$\tilde{A}=\{\underset{(1)}{((1,2),0)},\underset{(2)}{((1,3),0)},\underset{(2)}{((2,5),1)},\underset{(3)}{((2,3),1)},\underset{(3)}{((3,5),0)},\underset{(4)}{((3,6),0)},\underset{(3)}{((3,4),0)},\underset{(4)}{((5,7),1)},\underset{(4)}{((3,5),1)},$$

(Cost)

$$\underset{(5)}{((3,6),1)},\underset{(4)}{((3,4),1)},\underset{(5)}{((5,7),0)},\underset{(4)}{((4,6),1)},\underset{(5)}{((6,7),0)},\underset{(5)}{((6,7),2)},\underset{(5)}{((6,5),2)},\underset{(7)}{((5,7),2)}\}$$

Figure 3.



$$\begin{array}{lcccccccc}
\tilde{A} = \{ & ((1,2,r),0), & ((1,5,r),0), & ((1,5,t),0), & ((1,3,r),0), & ((2,4,r),0), & ((2,5,r),0), & ((5,4,r),0), & ((5,4,t),1), \\
(\text{Cost}) & (4) & (4) & (8) & (4) & (8) & (8) & (8) & (15)
\end{array}$$

$$\begin{array}{cccccccc}
((5,6,r),0), & ((5,6,t),0), & ((5,7,r),0), & ((3,5,r),0), & ((3,7,r),0), & ((4,8,t),1), & ((6,8,t),1), & ((7,8,r),0)\} \\
(8) & (15) & (8) & (8) & (8) & (19) & (19) & (12)
\end{array}$$