Path Search Algorithm for Connections with Pumps in Crude Oil Pipe Networks

Jorge Luis Rojas-d'Onofrio, Jack Márquez, Emmanuel Boutleux, Eric Niel

Path Search Algorithm for Connections with Pumps in Crude Oil Pipe Networks

Jorge L. Rojas-D'Onofrio*. Jack Márquez**
Emmanuel Boutleux***, Eric Niel*

*Institut National des Sciences Appliquées de Lyon, Laboratoire AMPERE, UMR CNRS 5005, 69621, Villeurbanne Cedex,
France (e-mail: jorge_rojasve@yahoo.com)
**Distrito Socialista Tecnológico, AIT-PDVSA, Mérida, Venezuela
*** Ecole Centrale de Lyon, Laboratoire AMPERE, UMR CNRS 5005, 69134, Ecully Cedex, France.

Abstract: The work presented addresses the case of connections in pipe networks for crude oil transportation, using pumps to overcome negative differences of pressure. The studied systems are operated by controlling valves and pumps, and can be modelled as discrete event systems. We use models based on undirected graphs. We present a path-search algorithm with obligatory pump nodes. This algorithm is meant to provide assistance to the operator of the pipe network SCADA. We tested the algorithms using data from real pipe networks located in Venezuela.

Keywords: Crude Oil Transportation, Operative Capacity, Pipe Network, Discrete Event Systems, Graphs, Routing Algorithms.

1. INTRODUCTION

The studied systems are two pipe networks for crude oil transportation: one located at the Oil Seaport of Guaraguao, Puerto La Cruz, and the other at the Crude Oil Tanks Yard of Punta de Palmas, both in Venezuela, belonging to PDVSA, the Venezuelan state-owned company in charge of hydrocarbons exploration, transport, and treatment. These two separate systems must assure, each one with particular conditions and specifications, the transport of crude oil between different elements such as pipelines, tanks, and quays. Connections are established using remote controlled valves: some valves are opened along a path linking the elements, whereas some other valves are closed around the path, isolating the connection from the rest of the pipe network. In this work we consider only connections between two elements which are the more frequent. Some connections also need a pump to overcome the negative pressure difference between the delivering element and the receiving element. We use a model based on graphs to represent these connections as paths. The choice of connections which use pumps is the main problem addressed in this paper.

Valves and pumps are the only elements that can be commanded, thus they are the basis of our model. Valves are, most of the time, in one of two different states: open or closed, whereas pumps can be on or off. The transitions between these states can be considered instantaneous when compared with the time spent in any of the states.

Many different types of oil are managed in the studied systems. A connection is meant to assure the transport of only one type of oil. Frequently many possible paths exist between the elements that must be connected. On the other hand, a connection path may interfere with other connections (transporting other types of oil). Choosing among many possible paths must be made in a manner that allows other operations (other connections) to be performed at the same time. In (Rojas et al., 2009) we have presented, in the same context, a path search algorithm minimizing interferences for the case of connections not using pumps (which are the more frequent type). In this paper the graph model is taken up, and we propose a path search algorithm for connections with pumps.

The main problem addressed in this paper is to find a path between two nodes passing through one pump in the correct direction. Even if many works concerning path search (Cherkassky and Goldberg, 1996) (Eppstein, 1999), and multi-constraint path routing (Ziegelmannm, 2007) can be found, the specific problem of paths with obligatory intermediate nodes is hardly addressed. The related problem of finding the Shortest Path Problem with Obligatory Intermediate Nodes is addressed by Wei and Quiqi (2006), using an hierarchical approach. In our case, we consider the particular case of only one obligatory intermediate node. The problem is relaxed by not considering the shortest path but any path connecting a given pair of nodes. On the other hand, the problem is constrained by the fact that the path must enter the intermediate node in a specific direction, since the pump only works in a specific sense and the connection must allow oil transport form a specific delivering element to a specific receiving element. This problem is solved by applying to this particular case the path-search algorithm minimizing interferences with other paths (PSAMI) proposed in (Rojas et al., 2009).

2. PIPE NETWORK MODELLING

The pipe network is modelled as an undirected graph that allows translating any path computed by the algorithms into an understandable set of instructions for the operators, whichare basica valves that must be closed in order to establish the connection.

A pipe network with valves can be modelled as an undirected graph, based on the analogy between valves connecting portions of a pipe network, and edges connecting nodes in a graph. The edges are undirected as the oil can be transported through a valve in both directions. While edges represent the valves in the network, the objects representing the nodes must be defined. In this article these objects are called segments (Bentley WaterCAD, 2010). A segment is defined as a portion of the network that is bounded by valves, but not containing any valve within its bounds.
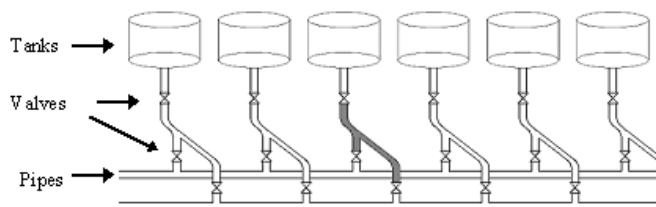
Fig. 1. shows (in grey) a segment in a pipe network.

They may contain a tank, a pump, a loading arm, or they can correspond to pipelines. Segments corresponding to tanks, loading arms or pipelines (which connect the system to other transportation systems out of the control of operators) will be represented as dead-ends nodes. Nodes corresponding to manifolds will be connected to several edges (valves). Pumps will be added to the graph only when connections that use pumps are considered, since connections not needing pumps must not pass through them.
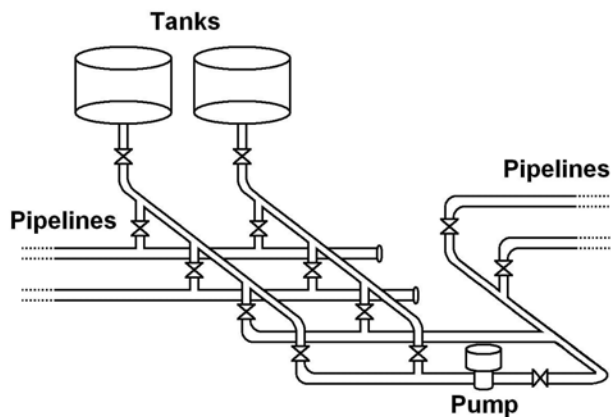
Fig. 2. A simple crude oil pipe network with two incoming pipelines (left), two tanks for treatment, a pump, and two exiting pipelines (right).

Fig. 2 shows a simple pipe network with valves. It is a simple model based on the referred Crude Oil Tanks Yard of Punta de Palmas. It has two incoming pipelines that deliver crude oil with sufficient pressure to fill any of the tanks without using a pump (this pressure is generated by pumps out of the control of this system). The crude oil is treated in the tanks, and then is transported out of the yard through any of the two exiting pipelines. For this it is necessary to use the pump represented in the figure. It is also possible to use the pump to transport the oil between two tanks.
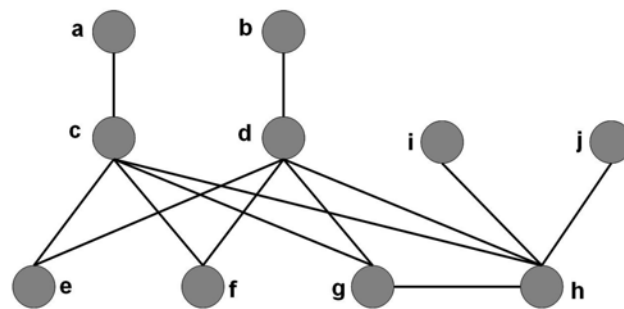
Fig. 3. Undirected graph model corresponding to the crude oil pipe network of Fig 2.

Fig. 3 shows the undirected graph model of the pipe network of Fig. 2 Tanks are segments connected to only one valve, thus their corresponding nodes (a, b) have only one edge. Each of the two arriving pipelines (e, f) is connected to each tank through the segments c and d. These segments also connect each tank to the downstream segment (h) and the upstream segment (g) of the pump. The two exit pipelines are connected to the downstream segment of the pump.

3. PATH-SEARCH ALGORITHM MINIMIZING INTERFERENCES WITH OTHER PATHS (PSAMI)

We address the problem of finding a path connecting two nodes in a graph and passing through an intermediary node in a fixed direction. The path must enter the intermediate node from the source node in the correct direction, exit the intermediate node and continue to the terminal node. This problem is solved using the PSAMI presented in a previous paper (Rojas 2009). The path found using this algorithm assures that the number of simultaneous connections between two predefined sets of elements, and not interfering with this path, will be maximum. Fig. 4 shows an undirected graph with a possible set of maximum node-disjoint paths between two sets of nodes determined using the OCA. These paths are: (a,h,j), (b,f,k), (c,,g,l), and (d,i,m).
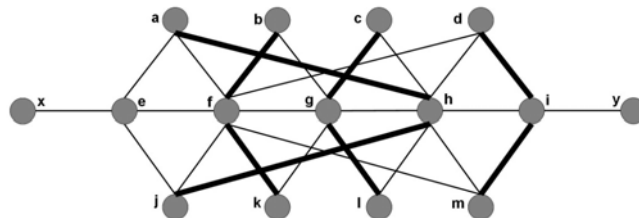
Fig. 4. Node-disjoint paths (thick lines) between the sets: *(a,b,c,d)* and *(j,k,l,m)* determined using OCA in a undirected graph. shows (in grey) a segment in a pipe network.

A path connecting the nodes *x* and *y* may interfere with the possible paths between the two sets. By instance, the path *(x,c,f,g,h,i,y)* will block all paths between the two sets.

In Fig. 5 the PSAMI is used to determined a path between the nodes x and y interfering as less as possible with the node disjoint paths between the same two sets of nodes. The path found is (x,c,a,h,i,y), with a resultant maximum flow of 2. This maximum flow value means that it is impossible to find more than two paths between the referred sets of nodes that do not interfere with a path between x and y. Paths (b,f,k) and (c,g,l) correspond to the resultant maximum flow.
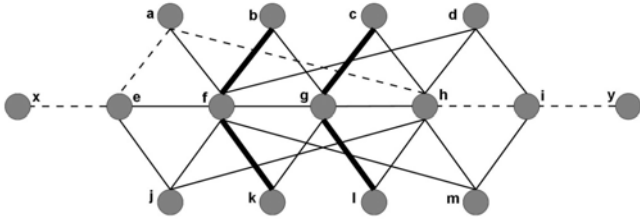

Fig. 5. Path between x and y (dot line) determined using PSAMI for the sets of nodes: (a,b,c,d) and (j,k,l,m). Thick lines correspond to the resultant possible node-disjoint paths between the sets.

PSAMI is based on Dijkstra's shortest path algorithm (1959) which is used in graphs with a cost assigned to each edge. Classical Dijkstra's algorithm considers edge costs that are known in advance. On the contrary, the PSAMI calculates costs dynamically during path search. At each relaxation step, the concerned edge cost is computed. This cost corresponds to the decrease in the maximum number of possible disjoint paths between two sets of nodes caused by the aimed node removal. The maximum number of possible disjoint paths is in turn calculated at each iteration using a maximum flow algorithm forbidding flow convergences/divergences. Once the shortest path to an intermediate node is found, the sub-graph induced by shortest path removal is saved. These sub-graphs will be used for costs calculation of edges exiting the corresponding node.
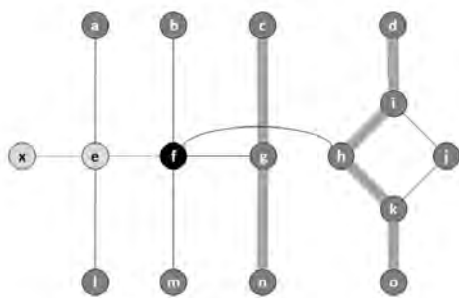

Fig. 6. Path search using PSAMI. In black, a node for which the shortest path has already been calculated. In light grey, the shortest path to the black node. Grey lines correspond to the maximum number of disjoint paths between the sets $A=(a,b,c,d,)$ and $B = (l,m,n,o)$.

Fig. 5 shows a graph where PSAMI is being executed. The black node f corresponds to a node for which the shortest path has been calculated. The nodes in light grey x,e correspond to the shortest path to f. The 2 grey lines correspond to the maximum number of disjoint paths between the sets

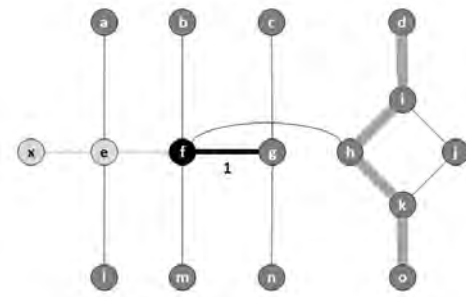$A=(a,b,c,d,)$ and $B = (l,m,n,o)$ for the sub-graph induced by x, e, f removal.

Fig. 7. Dynamic cost calculation of the edge (f,g) (black line). Node g removal reduces the maximum number of disjoint paths from 2 to 1. Therefore $cost(f,g) = 1$.

In Fig. 7 the cost of the edge (f,g) is calculated considering the removal of the aimed node g, and recalculating the maximum number of disjoint paths between the sets A and B. In this case, the maximum number of disjoint paths is reduced from 2 to 1. Therefore the cost of the edge (f,g) equals 1.
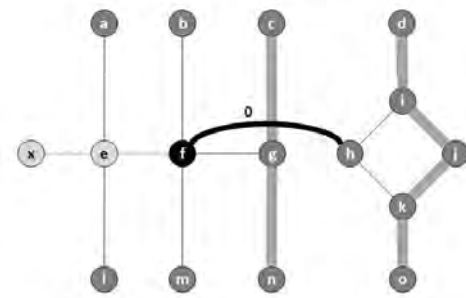

Fig. 8. Dynamic cost calculation of the edge (f,h) (black line). Node h removal does not reduce the maximum number of disjoint paths. Therefore $cost(f,h) = 0$.

In Fig. 8 the cost of the edge (f,h) is calculated considering the removal of the aimed node g, and recalculating the maximum number of disjoint paths between the sets A and B. In this case, the maximum number of disjoint paths does not change. Therefore the cost of the edge (f,h) equals 0.

Rojas (2009) gives a detailed explanation, with examples and the pseudo-code of the PSAMI. PSAMI implementation using a Ford Fulkerson algorithm (1956) for maximum number of disjoint path calculation has an asymptotic running time of $O(|V|^2 + |E|^2)$.

4. PATH-SEARCH ALGORITHM WITH AN OBLIGATORY PUMP NODE (PSAOP)

In our problem we consider the particular case of two sets of elements containing each one only one node. Hence, at a given time, only one connection can be established between the two sets of elements, although many different valid connections may exist. The path found using the algorithm will interfere as less as possible with the connection between the two sets. Let 'a' and 'b' be the pair of nodes to be

connected and let 'S' and 'T' be the pair of sets for which the number of node-disjoint paths must be maximized. The problem of finding a path between two elements and passing through a specific intermediate node (pump) in the correct direction is solved as follows. First, the intermediate node is split into two nodes. One node is connected to the upstream nodes of the intermediate node, and the other will be connected to the downstream nodes of the intermediate node. Then the PSAMI is applied, with 'a' corresponding to the crude oil delivering element, 'b' to the upstream node of the pump, 'S' to the downstream node of the pump, and 'T' to the receiving element. If the resultant flow between 'S' and 'T' is not null, then the problem has a solution and there is a path connecting the element through the pump in the correct direction. In this case, the path is completed searching for a path between 'S' and 'T', and not interfering with the path between 'a' and 'b'. Let PSAMI(G,a,b,s,t,F) be the procedure returning a path from a to b calculated with the PSAMI in the flow network G, maximizing the flow between s and t, and without passing through the nodes of the set F.

Let PATH(G,a,b,F) be a procedure searching for a path between a and b in the directed graph G without passing though the nodes oh the set F. The following procedure correspond to the PSAOP for a connection between the nodes x and y, and passing through the pump p. U is the set of upstream nodes of p, and D is the set of downstream nodes of p. The set F will be useful to avoid the use of many pumps in one connection. F will correspond usually to the pumps that are not considered. The pseudo-code notation of Cormen et al. (2001) is used:

```
PSAOP(G, x, y,p, U, D, F)
1   V[G] ← V[G] ∪ {pre(p)} ∪ {post(p)}
2   E[G] ← E[G] ∪ {(pre(p), post(p))}
3   for each node n ∈ U
4       do E[G] ← E[G] ∪ {(n, pre(p))}
5   for each node n ∈ U
6       do E[G] ← E[G] ∪ {(post(p), n)}
7   pre-path ← PSAMI(G,x,pre,post,y,F)
8   post-path ← PATH(G, post, y,{pre-path} ∪ F)
9   return (pre-path,post-path)
```

Lines 1 to 6 add the nodes and edges corresponding to the pump. Line 7 use the PSAMI to calculate the first part of the path, and line 8 use a simple path-search algorithm to compute the rest of the path. Line 9 return the complete path.
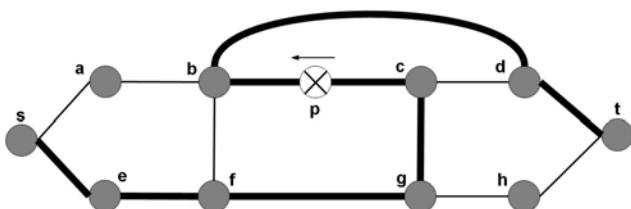

Fig. 9. Undirected graph model of a network with a pump. Many paths can connect the nodes 's' and 't'. The path in thick lines was calculated using the PSAOP.

Fig. 9 shows a network where many possible paths exist between a delivering element 's' and a receiving element 't'. Some of these paths connect the two elements without passing through the pump, for example (s,e,f,g,h,t) and (s,a,b,f,g,c,d,t), whereas other paths do pass through the pump but in the opposite direction (s,a,b,p,c,d,t). Only a path connecting this elements and passing through the pump in the correct direction is a valid connection.
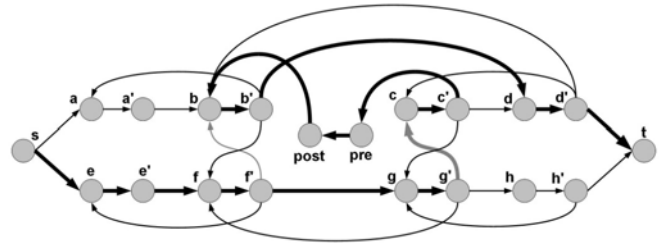

Fig. 10. Flow network constructed from the undirected graph model of Fig. 9. Nodes pre and post are created with PSAOP procedure. Thick lines show the path found with the PSAOP.

In Fig.10 the flow network corresponding to the network of Fig. 9 is shown. The nodes pre and post are created during the execution of the PSAOP (line 1). The nodes s' and t' were not represented for simplification, since these are the source and terminal nodes of the path.
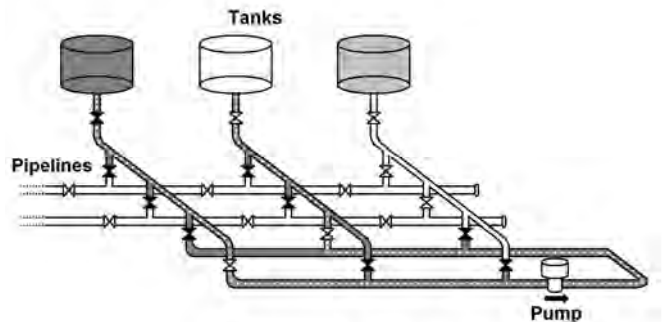

Fig. 11. Path between the first tank and the second tank calculated using PSAOP .
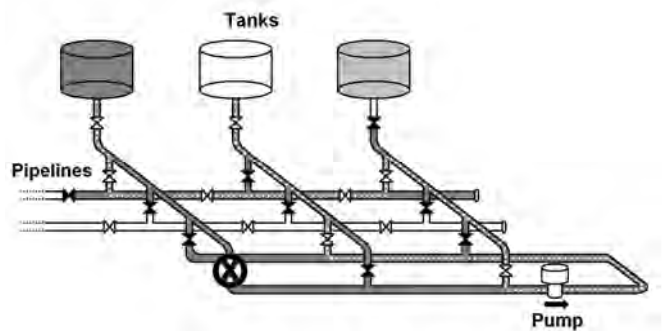

Fig. 12. Path between the first tank and the second tank calculated using PSAOP . A failure in a valve (X) impedes its opening.

In Fig. 11 the PSAOP calculates a path between the first and the second tank using the pump in the correct direction. In Fig. 12 a more complex case is considered. A failure in a valve is modelled removing the corresponding edge from the graph. In this case a alternative path is found.

5. APPLICATIONS

The program was applied to the cases of Punta de Palmas and Guaraguao pipe networks, being confidential the detailed information about these pipe networks. We have entered the pipe network data traducing the existing network displays used by the operators to the undirected graph representation presented previously. We have tested the program searching for connections between different elements. We used the programming language Python for the algorithms implementations.

Punta de Palmas network is the smaller one. It has several tanks, some pumps, some pipelines, and tens of valves. We have tested the algorithm for the cases of connections between tanks, and connections between tanks and exiting pipelines.

Guaraguao network is larger. Its complexity can be grasped if we take into account that it contains several quays, several arriving pipelines, tens of pumps, tens of tanks, and hundreds of valves. We have tested the algorithm for the cases of connections between tanks, and connection from quays to tanks. connection.

The results found are satisfactory, with calculation times of less than one minute.

```
>>> G.chemin_pompe('135x3','135x5','amont','aval')
[('135x3', '135x3*', 1), ('135x3*', 'MPC_135x3', 1), ('MPC_135x3', 'MPC_135x3*',
 1), ('MPC_135x3*', 'LT8', 1), ('LT8', 'LT8*', 1), ('LT8*', 'Merey_Multiple_4',
 1), ('Merey_Multiple_4', 'Merey_Multiple_4*', 1), ('Merey_Multiple_4*', 'tubo_mu
ltiple3', 1), ('tubo_multiple3', 'tubo_multiple3*', 1), ('tubo_multiple3*', 'Suc
cion1_97x1/7', 1), ('Succion1_97x1/7', 'Succion1_97x1/7*', 1), ('Succion1_97x1/7
*', 'Succion1_97x1/7_Bombas_1,6,7', 1), ('Succion1_97x1/7_Bombas_1,6,7', 'amont'
, 1), ('amont', 'aval', 1), [('aval', 'tubo_multiple5', 1), ('tubo_multiple5', '
tubo_multiple5*', 1), ('tubo_multiple5*', 'LT2', 1), ('LT2', 'LT2*', 1), ('LT2*'
, 'MPC_135x5', 1), ('MPC_135x5', 'MPC_135x5*', 1), ('MPC_135x5*', '135x5', 1)]]
>>> |
```

Fig. 12. Path found in Guaraguao network using PSAOP.

Fig.12. shows a list of edges corresponding to a path between two tanks of the Guaraguao network, using a chosen pump. The result is computed in 30 seconds using a regular PC.

6. CONCLUSIONS

The algorithm presented in this paper extends the works concerning the calculation of the operative capacity in crude oil pipe networks. It has direct practical application in the assistance programs currently being developed by PDVSA for the control of pipe networks.

Probably it is possible to use an extension of PSAOP to solve the problem of finding the shortest path with an obligatory pump node. For this, a classical Dijkstra's algorithm can be combined with PSAOP assigning very high costs in case of interferences with

The PSAOP can be utilized in the more general problem of finding a path with an unique obligatory intermediate node (without the restriction of an imposed direction). For this the sets of upstream and downstream nodes must be confounded. This extension of the PSAOP can have applications in many other areas (telecommunications, logistics, transportation systems, etc.).

REFERENCES

Bentley WaterCAD. (2010) Segmentation. *Bentley Systems Inc*; (Online). Available: http://docs.bentley.com/en/HMWaterCAD/

Cherkassky, B. V., A. V. Goldberg, T. R. (1996). Shortest paths algorithms: Theory and experimental evaluation, *Mathematical Programming*, 73, 129-174.

Cormen, T.H., Stein, C., Rivest, R.L., Leiserson, C.E. (2001). *Introduction to Algorithms*, chapter 24. 2nd ed. McGraw-Hill Higher Education.

Dijkstra, E. W. (1959). A note on two problems in connection with graphs, *Numerical Mathematics* 1, 395412. 25.

Eppstein, D. (1999). Finding the k Shortest Paths, *SIAM Journal on Computing,* 28, 652-673.

Ford, L.R. Jr., Fulkerson, D.R. (1956). Maximal Flow Through a Network. *Canadian Journal of Mathematics*, volume (8), 399-404.

Rojas, J.L., Gonzalez, J.G., Boutleux, E., Niel, E. (2009). Path Search Algorithm Minimizing Interferences with Envisaged Operations in a Pipe Network. *Proceedings of the European Control Conference 2009*, Budapest, Hungary.

Wei Wu, Qiuqi Ruan. (2006). A Hierarchical Approach for the Shortest Path Problem with Obligatory Intermediate Nodes. *Proceedings of the 8th International Conference on Signal Processing*, 2006, volume (4).

Ziegelmannm M. (2007). *Constrained Shortest Paths and Related Problems : Constrained Network Optimization.* VDM Verlag Dr. Müller.