Modelling and Constraint Hardness Characterisation of the Unique-Path OSPF Weight Setting Problem

Changyong Zhang and Robert Rodosek

IC-Parc, Imperial College London, London SW7 2AZ, United Kingdom
{cz, r.rodosek}@icparc.imperial.ac.uk

Abstract. Link weight is the primary parameter of OSPF, the most commonly used IP routing protocol. The problem of setting link weights optimally for unique-path OSPF routing is addressed. A complete formulation with a polynomial number of constraints is introduced and is mathematically proved to model the problem correctly. An exact algorithm is thereby proposed to solve the problem based on the analysis of the hardness of problem constraints.

1 Introduction

Open Shortest Path First (OSPF) [13] is the most widely deployed protocol for IP networks. As with most other conventional IP routing protocols [6], OSPF is a shortest path routing protocol, where traffic flows between origin and destination nodes are routed along the shortest paths, based on a shortest path first (SPF) algorithm [5]. Given a network topology, the SPF algorithm uses link weights to compute shortest paths. The link weights are hence the principal parameters of OSPF.

A simple way of setting link weights is the hop-count method, assigning the weight of each link to one. The length of a path is thereby equal to the number of hops. Another default way recommended by Cisco is the inv-cap method, setting the weight of a link inversely proportional to its capacity, without taking traffic into consideration. More generally, the weight of a link may depend on its transmission capacity and its projected traffic load. Accordingly, a task is to find an optimal weight set for OSPF routing, given a network topology, a projected traffic matrix [8], and an objective function. This is known as the OSPF weight setting problem.

The problem has two instances, depending on whether multiple shortest paths or only a unique one from an origin to a destination is allowed. For the first instance, a number of heuristic methods have been developed, based on genetic algorithm [7] and local search method [9]. For the second instance, Lagrangian relaxation method [12], local search method [15], and sequential method [2] have been proposed to solve the problem. With these heuristic methods, the problem is not formulated completely or explicitly and so generally is not solved optimally.

From a management point of view, unique-path routing requires much simpler routing mechanisms to deploy and allows for easier monitoring of traffic flows [3]. Therefore, this paper focuses on the unique-path instance. The problem is referred as the unique-path OSPF weight setting (1-WS) problem. It is a reduction of the NP-complete integer multicommodity flow problem [16].

V.S. Sunderam et al. (Eds.): ICCS 2005, LNCS 3514, pp. 804 – 811, 2005.

With the aim of developing a scalable approach to solve the 1-WS problem optimally, a complete formulation with a polynomial number of constraints is introduced and is mathematically proved to model the problem correctly in Section 2. The hardness of problem constraints is studied in Section 3. Based on the analysis of constraint hardness, an exact algorithm is proposed in Section 4. Conclusions and further work are presented in Section 5.

2 A Complete Formulation

2.1 Problem Definition

The unique-path OSPF weight setting problem is defined as follows. Given

- A network topology, which is a directed graph structure $G=(V, E)$ where V is a finite set of nodes and E is a set of directed links. For each $(i, j) \in E$, i is the starting node, j is the end node, and $c_{ij} \geq 0$ is the capacity of the link.
- A traffic matrix, which is a set of demands D. For each demand $k \in D$, $s_k \in V$ is the origin node, $t_k \in V$ is the destination node, and $d_k \geq 0$ is the demand bandwidth. Accordingly, S is the set of all origin nodes.
- Lower and upper bounds of link weights, which are positive real numbers $w_{\min}$ and $w_{\max}$, respectively.
- A pre-specified objective function, e.g., to maximise the residual capacities.

  Find an optimal weight w_{ij} for each link $(i, j) \in E$, subject to

- Flow conservation constraints. For each demand, at each node, the sum of all incoming flows (including demand bandwidth at origin) is equal to the sum of all outgoing flows (including demand bandwidth at destination).
- Link capacity constraints. For each link, the traffic load over the link does not exceed the capacity of the link.
- Path uniqueness constraints. Each demand has only one routing path.
- Path length constraints. For each demand, the length of each path assigned to route the demand is less than that of any other possible and unassigned path to route the demand.
- Link weight constraints. For each link $(i, j) \in E$, the weight w_{ij} is within the weight bounds, i.e., $w_{\min} \leq w_{ij} \leq w_{\max}$.

2.2 Mathematical Modelling

According to the requirements of the 1-WS problem, the routing path of a demand is the shortest one among all possible paths. For each link, the routing path of a demand either traverses it or not. Based on this observation and the relationship between the length of a shortest path and the weights of links that it traverses, the problem can be formulated by defining one routing decision variable for each link and each demand, which results in the following model.

Routing decision variables:

$$x_{ij}^k \in \{0,1\}, \forall k \in D, \forall (i,j) \in E \tag{1}$$

is equal to 1 if and only if the path assigned to route demand *k* traverses link (*i*, *j*).

Link weight variables:

$$w_{ij} \in [w_{\min}, w_{\max}], \forall (i,j) \in E \tag{2}$$

represents routing cost of link (*i*, *j*).

Path length variables:

$$l_i^s \begin{cases} = 0, i = s \\ \in [0, +\infty), i \neq s \end{cases}, \forall s \in S, \forall i \in V \tag{3}$$

represents the length of the shortest path from origin node *s* to node *i*.

Flow conservation constraints:

$$\sum_{h:(h,i)\in E} x_{hi}^k - \sum_{j:(i,j)\in E} x_{ij}^k = b_i^k, \forall k \in D, \forall i \in V \tag{4}$$

where $b_i^k = -1$ if $i = s_k$, $b_i^k = 1$ if $i = t_k$, and $b_i^k = 0$ otherwise.

Link capacity constraints:

$$\sum_{k\in D} d_k x_{ij}^k \le c_{ij}, \forall (i,j) \in E \tag{5}$$

Path length constraints:

$$\left.\begin{array}{r} x_{ij}^k = 0 \wedge \sum_{h:(h,j)\in E} x_{hj}^k = 0 \Rightarrow l_j^{s_k} \le l_i^{s_k} + w_{ij} \\ x_{ij}^k = 0 \wedge \sum_{h:(h,j)\in E} x_{hj}^k = 1 \Rightarrow l_j^{s_k} < l_i^{s_k} + w_{ij} \\ x_{ij}^k = 1 \Rightarrow l_j^{s_k} = l_i^{s_k} + w_{ij} \end{array}\right\}, \forall k \in D, \forall (i,j) \in E \tag{6'}$$

The above logic constraints can be linearised by introducing appropriate constants ε and M with $0 < \varepsilon << M$.

$$\begin{cases} l_j^{s_k} \le l_i^{s_k} + w_{ij} - \varepsilon(\sum_{h:(h,j)\in E} x_{hj}^k - x_{ij}^k) \\ l_j^{s_k} \ge l_i^{s_k} + w_{ij} - M(1 - x_{ij}^k) \end{cases}, \forall k \in D, \forall (i,j) \in E \tag{6}$$

Objective function: to maximise the residual capacities, alternatively, to minimise the throughput:

$$\min \sum_{(i,j)\in E} \sum_{k\in D} d_k x_{ij}^k \tag{7}$$

Accordingly, the complete model is presented as follows:

1-WS 0: Optimise (7) Subject to (4), (5), (6), (1), (2), (3)

2.3 Proof of Correctness

A relaxation of the 1-WS problem is the integer multicommodity flow problem [1], a recognised correct model of which is presented as follows:

1-WS I: Optimise (7) Subject to (4), (5), (1)

Apparently, the difference between 1-WS 0 and 1-WS I are path length constraints (6) and the resulting additional link weight variables (2) as well as path length variables (3). In order to ensure that 1-WS 0 formulates the 1-WS problem correctly, constraints (6) are proved to represent correctly the additional path length as well as path uniqueness constraints in the following. As the initial logic constraints are identical to the linearised constraints (6), the following proof is based on the initial constraints (6').

Proposition 1. *The path length constraints in 1-WS 0 restrict that each routing path is a shortest path.*

Proof. Assume for demand $k, P_j=(j_1,j_2)\rightarrow(j_2,j_3)\rightarrow\ldots\rightarrow(j_{n-1},j_n)$, $j_1=s_k, j_n=t_k$ is the assigned routing path and $P_i=(i_1,i_2)\rightarrow(i_2,i_3)\rightarrow\ldots\rightarrow(i_{m-1},i_m)$, $i_1=s_k, i_m=t_k$ is one of any other possible and non-assigned paths. Then, according to the definition of routing decision variables, $x^k_{j_l j_{l+1}}=1, l=1,\ldots,n-1$ and $\exists(i_q,i_{q+1})\in P_i$, $x^k_{i_q i_{q+1}}=0$, $q\in\{1,2,\ldots,m-1\}$.

As a result, according to (6'), on one hand, since $x^k_{ij}=1\Rightarrow l^{s_k}_j=l^{s_k}_i+w_{ij}$,

$$l^{s_k}_{t_k}=l^{s_k}_{j_n}=l^{s_k}_{j_{n-1}}+w_{j_{n-1}j_n}=l^{s_k}_{j_{n-2}}+w_{j_{n-2}j_{n-1}}+w_{j_{n-1}j_n}=\ldots=l^{s_k}_{j_1}+w_{j_1j_2}+\ldots+w_{j_{n-1}j_n}=l_{P_j}$$

On the other hand, since $x^k_{ij}=0\Rightarrow l^{s_k}_j\le l^{s_k}_i+w_{ij}$,

$$l^{s_k}_{t_k}=l^{s_k}_{i_m}\le l^{s_k}_{i_{m-1}}+w_{i_{m-1}i_m}\le l^{s_k}_{i_{m-2}}+w_{i_{m-2}i_{m-1}}+w_{i_{m-1}i_m}\le\ldots\le l^{s_k}_{i_1}+w_{i_1i_2}+\ldots+w_{i_{m-1}i_m}=l_{P_i}$$

Therefore, $l_{P_j}\le l_{P_i}$. It is proved that path P_j is a shortest path. □

Lemma 1. *The path uniqueness constraints are satisfied by 1-WS I.*

Proposition 2. *The path length constraints in 1-WS 0 restrict that the resulting routing path of each demand is a unique shortest path.*

Proof. As 1-WS 0 is a reduction of 1-WS I, the solution to routing decision variables x^k_{ij} of 1-WS 0 is a solution to 1-WS I.

According to Lemma 1, there is only one path to route each demand. Suppose for demand k, $P_j=(j_1,j_2)\rightarrow(j_2,j_3)\rightarrow\ldots\rightarrow(j_{n-1},j_n), j_1=s_k, j_n=t_k$ is the assigned routing path, and $P_i=(i_1,i_2)\rightarrow(i_2,i_3)\rightarrow\ldots\rightarrow(i_{m-1},i_m), i_1=s_k, i_m=t_k$ is one of any other possible and non-assigned paths to route demand k. Then, according to the definition of routing decision variables, $x^k_{j_l j_{l+1}}=1, l=1,\ldots,n-1$.

As a result, according to (6'), since $x^k_{ij}=1\Rightarrow l^{s_k}_j=l^{s_k}_i+w_{ij}$,

$$l^{s_k}_{t_k}=l^{s_k}_{j_n}=l^{s_k}_{j_{n-1}}+w_{j_{n-1}j_n}=l^{s_k}_{j_{n-2}}+w_{j_{n-2}j_{n-1}}+w_{j_{n-1}j_n}=\ldots=l^{s_k}_{j_1}+w_{j_1j_2}+\ldots+w_{j_{n-1}j_n}=l_{P_j}$$

As both P_i and P_j are paths between s_k and t_k, they finally merge at one node. Assume it is node r and $r = j_p = i_q, p \in \{2,3,...,n-1,n\}, q \in \{2,3,...,m-1,m\}$. Then, according to the definition of routing decision variables, $x^k_{j_{p-1}r} = 1$ and $x^k_{i_{q-1}r} = 0$, and hence $\sum_{h:(h,r)\in E} x^k_{hr} = 1$. As a result, according to (6'),

$$\begin{aligned} l^{s_k}_{t_k} &= l^{s_k}_{i_m} = l^{s_k}_{i_{m-1}} + w_{i_{m-1}i_m} = ... = l^{s_k}_r + w_{ri_{q+1}} + ... + w_{i_{m-1}i_m} \\ &< l^{s_k}_{i_{q-1}} + w_{i_{q-1}r} + w_{ri_{q+1}} + ... + w_{i_{m-1}i_m} \le ... \le l^{s_k}_{i_1} + w_{i_1 i_2} + ... + w_{i_{m-1}i_m} = l_{P_i} \end{aligned}$$

Therefore, $l_{P_j} < l_{P_i}$. It is proved that path P_j is the unique shortest path to route demand *k*. □

3 Constraint Hardness Characterisation

There are three types of constraints in 1-WS 0, flow conservation constraints, link capacity constraints, and path length constraints. Among them, flow conservation constraints are the basic and core constraints of the problem. In order to compare the hardness of the other two types of constraints, two relaxed problems are studied.

First, path length constraints are relaxed from the 1-WS problem, which results in the integer multicommodity flow problem 1-WS I, as introduced in Section 2.3.

Second, link capacity constraints are relaxed from the 1-WS problem. This results in the un-capacitated unique-path OSPF weight setting problem:

1-WS II: Optimise (7) Subject to (4), (6), (1), (2), (3)

Forty-eight data sets with combinations of different parameter scenarios were generated for empirical study. In Table 1, Nds, Lnks, and Dmnds denote the numbers of nodes, links, and demands, respectively. All the three problems 1-WS 0, 1-WS I, and 1-WS II were implemented in ECLiPSe [11] and solved using CPLEX 6.5 [10] on all data sets generated. The timeout was set to be *3600 seconds* for each data instance. The following analyses are thereby based on the performance of using CPLEX.

Table 1. Details of data sets tested

ID	Nds	Lnks	Dmnds	ID	Nds	Lnks	Dmnds	ID	Nds	Lnks	Dmnds
1	10	22	3	17	30	78	60	33	50	130	49
2	10	26	5	18	30	78	375	34	50	128	50
3	10	24	9	19	30	136	3	35	50	130	100
4	10	24	10	20	30	144	15	36	50	136	788
5	10	26	20	21	30	142	29	37	50	238	3
6	10	24	50	22	30	144	30	38	50	238	25
7	10	46	3	23	30	142	60	39	50	238	49
8	10	46	5	24	30	142	450	40	50	242	50
9	10	46	9	25	30	236	3	41	50	240	100
10	10	46	10	26	30	234	15	42	50	238	1000
11	10	48	20	27	30	236	29	43	50	644	3
12	10	44	50	28	30	234	30	44	50	648	25
13	30	80	3	29	30	236	60	45	50	642	49
14	30	78	15	30	30	234	450	46	50	642	50

15	30	82	29	31	50	128	3	47	50	646	100
16	30	76	30	32	50	132	25	48	50	642	1000

Consider the OSPF weight setting problem, it is shown that routing performances resulting from the proposed complete formulation are much better than those from using the default methods. The resulting average maximum utilisation is *28.79%* of that from using the hop-count method and *40.68%* of that from using the inv-cap method, which demonstrates the significant gains achieved by formulating the problem completely and solving it optimally.

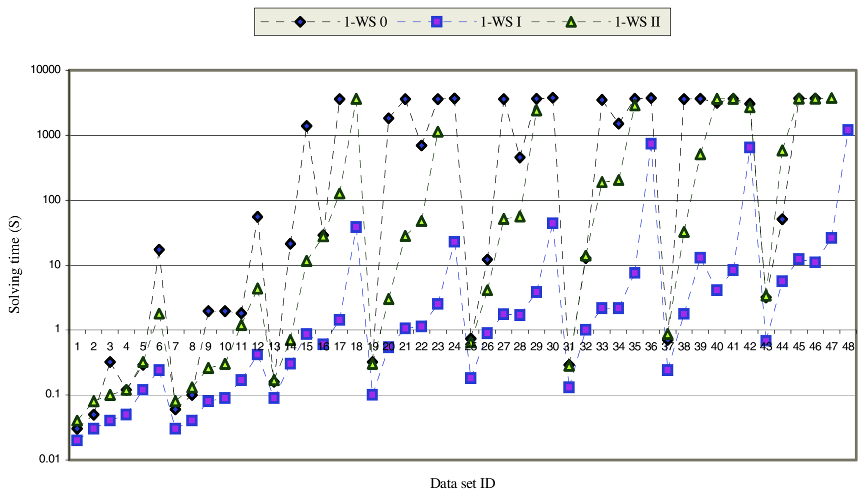

Fig. 1. Solving times of 1-WS 0, 1-WS I, and 1-WS II

Figure 1 compares the solving times of 1-WS 0 with those of 1-WS I and 1-WS II. It can be noted that 1-WS I is solved optimally within *1000 seconds* for all instances, except the last one, which is detected infeasible in *1191 seconds*. Meanwhile, it takes more time to solve 1-WS 0 than to solve 1-WS I on all instances. For most large-scale instances, it even cannot be solved when timeout. It is thus shown that path length constraints are very hard constraints for the 1-WS problem. It can be further seen that, although it takes less time to solve 1-WS II than the initial problem on most data instances, the difference is not so significant. The relaxed problem still cannot be solved when timeout on a few data instances. It is therefore indicated that the link capacity constraints are not the hardest constraints.

In addition, it can be observed that between the two relaxed problems, 1-WS I is much easier to solve than 1-WS II. Therefore, path length constraints, which are relaxed in 1-WS I, are the hardest constraints for the 1-WS problem.

In order to investigate further the reason behind the above observations, the constraint structure of the 1-WS problem is shown in Figure 2. The first row represents link capacity constraints (5), the next four rows correspond to flow conservation constraints (4), and the last four rows represent path length constraints (6). As it can be seen, among the three types of constraints, flow conservation constraints and link capacity constraints contain only routing decision variables, while path length constraints couple routing decision variables with link weight variables and path length variables, which makes the problem more complicated than the integer multicommodity flow problem. This observation can also be used to explain why path length constraints are the hardest constraints for the 1-WS problem, instead of link capacity constraints, which are the hardest for the integer multicommodity flow problem [14].

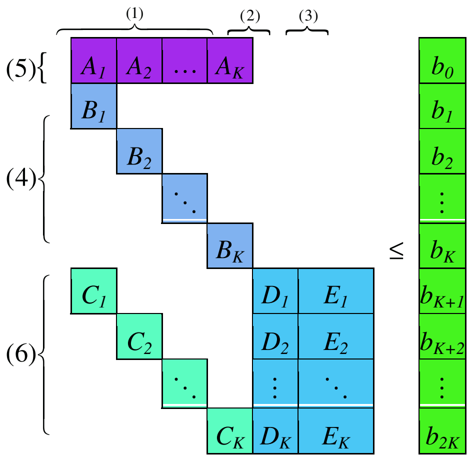

Fig. 2. Constraint structure of 1-WS 0

4 Proposed Algorithm

Based on the above study of constraint hardness, a proposed algorithm to solve the 1-WS problem is Benders decomposition method [4], which decomposes the problem into an integer multicommodity flow master problem and a linear programming (LP) subproblem. The master problem deals with flow conservation constraints and link capacity constraints, and so contains routing decision variables only. Accordingly, the LP subproblem deals with the hardest constraints, path length constraints. Compared with the initial mixed integer programming (MIP) problem, the resulting master problem has a much smaller model size. Therefore, instead of solving a larger and more complicated MIP problem in one step, the proposed algorithm solves the problem by dealing with a smaller and simpler master problem and an LP subproblem iteratively. For the integer master problem, Lagrangian relaxation method has been demonstrated to be an appealing algorithm [14].

It was shown be preliminary results that, for small data instances, MIP solver solves the problem slightly faster than Benders decomposition method. However, when data instances get larger, the latter takes the advantage.

5 Conclusion

In order to develop a complete solution approach to the unique-path OSPF weight setting problem, the problem has been explicitly formulated as a complete model, the correctness of which is mathematically proved. The model has three types of constraints, flow conservation constraints, link capacity constraints, and path length constraints. Among them, path length constraints have been identified to be the hardest constraints for the problem. Based on the study of constraint structure of the formulation, Benders decomposition method, embedded with Lagrangian relaxation method for the integer master problem, has been proposed to solve the problem.

Our future work includes developing the proposed algorithm completely and investigating possible improvements to both model formulation and solution algorithm to accelerate the convergence rate of the solution approach.

References

1. Ahuja, R. K., Magnanti, T. L., and Orlin, J. B.: Network Flows: Theory, Algorithms, and Applications. Prentice-Hall (1993)
2. Ameur, W. B., Bourquia, N., Gourdin, E., and Tolla, P.: Optimal Routing for Efficient Internet Networks. In Proc. of ECUMN (2002) 10-17
3. Ameur, W. B., and Gourdin, E.: Internet Routing and Related Topology Issues. SIAM J. on Discrete Mathematics, 17(1) (2003) 18-49
4. Benders, J.: Partitioning Procedures for Solving Mixed-Variables Programming Problems. Numerische Mathematik, 4 (1962) 238-252
5. Bertsekas, D., and Gallager, R.: Data Networks. Second Edition, Prentice Hall (1992)
6. Black, U.: IP Routing Protocols: RIP, OSPF, BGP, PNNI and Cisco Routing Protocols. Prentice Hall, (2000)
7. Ericsson, M., Resende, M. G. C., and Pardalos, P. M.: A Genetic Algorithm for the Weight Setting Problem in OSPF Routing. J. of Combinatorial Optimization, 6 (2002) 299-333
8. Feldmann, A., Greenberg, A., Lund, C., Reingold, N., Rexford, J., and True, F.: Deriving Traffic Demands for Operational IP Networks: Methodology and Experience. IEEE/ACM Transactions on Networking, 9(3) (2001) 265-279
9. Fortz, B., and Thorup, M.: Internet Traffic Engineering by Optimizing OSPF Weights. In Proc. of INFOCOM (2000) 519-528
10. ILOG Inc.: ILOG CPLEX 6.5 User's Manual. (1999)
11. Imperial College London: ECLiPSe 5.7 User's Manual. (2003)
12. Lin, F. Y. S., and Wang, J. L.: Minimax Open Shortest Path First Routing Algorithms in Networks Supporting the SMDS Services. In Proc. of ICC, 2 (1993) 666-670
13. Moy, J.: OSPF Anatomy of an Internet Routing Protocol. Addison-Wesley (1998)
14. Ouaja, W., and Richards, B.: A Hybrid Multicommodity Routing Algorithm for Traffic Engineering. Networks, 43(3) (2004) 125-140
15. Ramakrishnan, K. G., and Rodrigues, M. A.: Optimal Routing in Shortest-Path Data Network. Bell Labs Technical Journal, January-June (2001) 117-138
16. Wang, Y., and Wang, Z.: Explicit Routing Algorithms for Internet Traffic Engineering. In Proc. of ICCCN (1999) 582-588