

Constraint Based Resilience Analysis

Helmut Simonis¹

CrossCore Optimization Ltd, London, UK
helmut.simonis@crosscoreop.com,
WWW home page: <http://www.crosscoreop.com>

Abstract. In this paper we give an overview of applications of Constraint Programming for IP (Internet Protocol) data networks, and discuss the problem of *Resilience Analysis* in more detail. In this problem we try to predict the loading of a network in different failure scenarios, without knowing end-to-end flow values throughout the network; the inference is based only on observed link traffic values. The related problem of *Traffic Flow Analysis* aims to derive a traffic matrix from the observed link traffic data. This is a severely under-constrained problem, we can show that the obtained flow values vary widely in different, feasible solutions. Experimental results indicate that using the same data much more accurate, bounded results can be obtained for *Resilience Analysis*.

1 Introduction

In this paper we discuss the use of Constraint Programming (CP) for IP (Internet Protocol) data network applications. The bulk of constraint applications for networks [40] are in the context of data networks, covering either traditional, connection oriented networks or packet-switched, routed networks like the Internet. The survey [40] classifies them into a number of different groups:

- The first real-world constraint application in this domain was a problem of *application placement* for the Italian Inter banking network [6], a problem very closely related to the warehouse location problem [45, 32].
- In many networks, the task of *path placement* is to define the route on which a demand will be sent through the network. This is a fundamental networking problem, for which many competing CP methods have been proposed. The models fall into three main sub-classes, link-based models [33–35, 27, 11, 12, 22], path-based models [5, 18, 25, 26] and node-based models [37].
- One possible extension is the use of *multiple paths* for demands, where the secondary path is only active when the primary connection has failed [50].
- Another possible extension is to add a time dimension, where traffic demands have given start and end times, and demands compete for network bandwidth if they overlap in time. This application is called *Bandwidth on Demand* [23, 28, 43, 7, 37].
- In the previous problems, the network structure and capacity was fixed. The problem of *Network Design* deals with defining connectivity and finding the right link capacity to satisfy a projected set of demands [24, 5, 9, 10, 41].

- IP (Internet Protocol) networks usually do not use explicit routes for traffic demands. Instead, packets are routed based on a distributed shortest path algorithm. *Metric optimization* deals with choosing metric weights to influence the routing in the network and to optimize the network utilization [1, 2, 16].
- Secondary paths and routing algorithms provide some methods to maintain network communications in case of element failure. The idea of *Bandwidth Protection* offers an alternative, purely local mechanism for improving network resilience [49, 48].

The spread of wireless networks has led to a whole new class of problems, two examples of the use of CP are shown in [19, 38].

2 Flow Analysis and Resilience Analysis

Most of the problems described above assume that there is a well-defined set of demands, the *Traffic Matrix*. We know who wants to use the network for connections between specific points and how much bandwidth they require. For IP based networks this assumption is, surprisingly, not valid. In an operational, routed network there is no (simple) way of collecting data about end-to-end traffic flows, we don't know who is talking to whom and how much bandwidth the customers use. The only information we can collect is the overall traffic on each link e of the network $\mathbf{traf}(e)$ and the external traffic entering $\mathbf{ext}^{in}(i)$ and leaving $\mathbf{ext}^{out}(j)$ at nodes i and j of the network. We can try to reconstruct a traffic matrix from these measurements, this is an active research area called *traffic flow analysis*.

2.1 Related Work

Most of the related work is concerned with the identification of the traffic matrix for all PE (provider edge router) to PE or PoP (Point of presence) to PoP flows. [21] discuss use cases for this flow analysis and compare different means of collecting this data. We will concentrate on methods which deduce the traffic matrix indirectly, without collecting flow data throughout the network. There are two main directions this work is taking, the *tomography method* and the *gravity approach*.

The *tomography approach*, pioneered by Vardi [47], is based on a model where the traffic matrix is deduced from the link traffic. As this problem is under-constrained, a series of observations are used, assuming that the measurements are independent and reflect the same traffic matrix. A stochastic algorithm is used to find the most likely traffic matrix which fits the available data. Slightly different models and assumptions are used in [46, 44, 3, 8], a survey is given in [4]. The work in [20] that uses a linear model to calculate a traffic matrix from the link data observations is the one most closely related to the model presented here. The use of lower and upper bounds for the flow analysis was suggested in

[36]. An important requirement for the tomography approach is the need for a routing model, which understands how traffic is flowing through the network.

An alternative approach is the *gravity model* [31], which is based on the traffic data of the external links only. Based on the assumption that the flow between two customers is proportional to the product of their input and output traffic values, we find a traffic matrix which is consistent with all external measurements, but not necessarily with the link loads inside the network. The underlying assumption that customers talk to each other with equal likelihood can be justified for large ISPs (Internet service provider) with a consistent user base, but are harder to maintain for enterprise networks or ISPs offering mainly VPN (virtual private network) connections, where we already know that certain customers only talk to a subset of all customers. The gravity model originates in social sciences, where it is used for example to predict traffic flows in public transport systems. The data requirements for a gravity model are much more restricted than for the tomography method, it neither needs a routing model nor traffic data from the interior of the network. But for the same reason it is inherently less accurate than the tomography approach.

[54] propose a combination of the gravity based approach and the tomography approach, which they validate on parts of the AT&T network. For a large tier-1 service provider the assumptions of the gravity model seem valid, if you distinguish between end-user connections and peering and up-link lines.

As described in [30], the traffic matrix can be defined at different levels of network abstractions. Typical variants are customer to customer flows, edge to edge flows, flows between core routers or aggregated flows, for example between PoPs. Different levels of abstraction are useful for different use cases, e.g. edge-to-edge flows for traffic engineering, PoP to PoP flows for capacity planning. In the context of resilience analysis, we will use edge to edge flows.

[17] deals with the problem of directly collecting flow data from the network and propose two methods which concentrate on large flows only. It shows that the current implementation of netflow has significant scaling problems in a large network. [14, 15] deal with the problem of sampling of traffic flows and the resulting inaccuracies.

2.2 Flow Analysis Model

A model for the traffic flow analysis is shown below. We describe the network as a directed graph $G = (\mathbf{N}, \mathbf{E})$ with nodes \mathbf{N} and directed edges \mathbf{E} . We use non-negative flow variables F_{ij} to denote the traffic flow from node i to node j in the network. The $[0, 1]$ constants r_{ij}^e define the routing in the network, they indicate what fraction of the flow between nodes i and j is routed over edge e .

$$\forall i, j \in \mathbf{N} : \quad \min_{\{F_{ij}\}} / \max_{\{F_{ij}\}} F_{ij} \quad (1)$$

st.

$$\forall e \in \mathbf{E} : \sum_{i,j \in \mathbf{N}} r_{ij}^e F_{ij} = \mathbf{traf}(e) \quad (2)$$

$$\forall i \in \mathbf{N} : \sum_{j \in \mathbf{N}} F_{ij} = \mathbf{ext}^{in}(i) \quad (3)$$

$$\forall j \in \mathbf{N} : \sum_{i \in \mathbf{N}} F_{ij} = \mathbf{ext}^{out}(j) \quad (4)$$

$$F_{ij} \geq 0$$

For every flow, we try to find a lower and an upper bound as the result of an optimization run with the objective (1). We know that the sum of all flows routed over an edge is equal to the observed traffic on the edge (2), and that the sum of all flows starting (3) or ending (4) in a node must be equal to the observed external traffic.

2.3 Data

For an evaluation of the model, we use 6 networks from the Rocketfuel project [29] and one other network topology (dexa) of a global enterprise network. The networks range from 51 to 315 routers, and also have quite different connectivity. Table 1 compares the major parameters of the network. Lines are bi-directional connections between routers, PoPs (Points of presence) indicate places where all routers for an area are co-located. Connections inside a PoP often are LAN (local area network) type, whereas connections between PoPs typically are WAN (wide area network) type and are more expensive. All networks are nearly real-life, the

Table 1. Test networks

Network	Routers	PoPs	Lines	Lines/Router
dexa	51	24	59	1.15
as1221	108	57	153	1.41
as1239	315	44	972	3.08
as1755	87	23	161	1.85
as3257	161	49	328	2.03
as3967	79	22	147	1.86
as6461	141	22	374	2.65

topology of the Rocketfuel networks is deduced from data collected remotely off the actual ISP networks, the dexa network is an operational network. For the dexa network, we also have actual link speeds and IGP (interior gateway protocol) metric values, while the metric values for the Rocketfuel networks are derived from traceroute information and no link speed is available (we assume

the same speed for all link for simplicity). While we can clearly distinguish P (core) and PE (edge) routers in the dexa network, we can only do so heuristically in the other networks.

For the dexa network, we use the actual customer VPN structure, for all other networks we generate VPNs of different sizes randomly. We then generate for all networks random traffic flows between the VPN end points, and calculate from these simulated flows s_{ij} the expected traffic load at each interface. These traffic loads are consistent with each other and are generated using the same routing model that is used in the analysis part.

2.4 Traffic Flow Analysis Results

The basic problem with the model above is that it is very under-constrained. We have $|\mathbf{N}|^2$ flow variables F_{ij} , but only $|\mathbf{E}| + 2|\mathbf{N}|$ constraints. Results from [39] shown below indicate that the values for the flows can vary in a very wide interval, with no clear preference for any particular value. It is therefore unclear how to use the results for answering further questions about the network, for example how the traffic will change in case of an element failure.

In table 2 we present the sum of all lower bounds as a percentage of the sum of the simulated flow values ($100 * \frac{\sum_{i,j} \min F_{ij}}{\sum_{i,j} s_{ij}}$), as well as the sum of all upper bounds as a percentage of the sum of the simulated flows ($100 * \frac{\sum_{i,j} \max F_{ij}}{\sum_{i,j} s_{ij}}$). A value of 100% would be the optimal result. We also show the number of objective functions and the total time (in seconds) to run the test.

Table 2. TFA results

Network	Low/Simul	High/Simul	Obj	Time
dexa	0	2310.65	1190	11
as1221	0.09	8398.64	11556	1318
as1239	n/a	n/a	n/a	n/a
as1755	0.15	6255.31	7482	699
as3257	0.04	12260.03	25760	12389
as3967	0.1	5387.10	6162	500
as6461	0.28	8688.39	19740	8676

We could not obtain a result for network AS1239, but we estimate, based on a partial result, that a complete analysis would take more than 5 days.

All results are obtained on Linux PCs running ECLiPSe 5.6, using CPLEX 6.5 as the linear solver.

The lower bound tells us how much of the traffic in the network we can associate with specific flows between routers, i.e. we know where the traffic originates and where it ends. The upper bound indicates how uncertain we are about the exact source and destination. If the number is very high, then many flows may

be the cause of the traffic, and we lack the ability to differentiate between them. In this particular setting, the results are unimpressive. The lower bounds are very close to zero, and the upper bounds over-estimate the simulated flows by a factor from 23 to 122! This means that we can not pin any traffic on particular flows and most of the traffic may have been caused by lots of flows between quite different routers.

Aggregating the flows for a PoP to PoP analysis improves the results, but not significantly. Table 3 considers flows between pairs of PoPs, which is the sum of all flows between all routers in either PoP. We can note two points:

Table 3. PoP TFA results

Network	Low/Simul	High/Simul	Obj	Time
dexa	0	1068.37	557	5
as1221	0.24	2964.93	3205	424
as1239	0.63	1401.72	1931	101359
as1755	0.66	1263.28	526	103
as3257	0.30	2028.73	2378	2052
as3967	0.1	1209.37	483	90
as6461	1.47	951.41	481	768

- The results are significantly better than for the router to router flow analysis, but not nearly good enough to identify the flows. The lower bounds are still very nearly zero, and the upper bounds overestimate the flows in total between 10 and 30 times.
- The run time is much reduced, since there are far fewer objectives to calculate, but per objective the runtime did slightly increase.

2.5 Resilience Analysis Model

The idea behind *resilience analysis* is to avoid the generation of the intermediate traffic matrix, and to pose questions about the network behavior directly in the initial model. For example, we may be interested in understanding the traffic in the network under an element failure and resulting re-routing. The routing in the normal network operation is denoted with r_{ij}^e , the routing after the element failure is given by \overline{r}_{ij}^e . The model for resilience analysis below uses the flow variables F_{ij} only internally, without trying to deduce particular values.

$$\forall e \in \mathbf{E} : \quad \min_{\{F_{ij}\}} / \max_{\{F_{ij}\}} \quad \sum_{i,j \in \mathbf{N}} \overline{r}_{ij}^e F_{ij} \quad (5)$$

st.

$$\forall e \in \mathbf{E}: \sum_{i,j \in \mathbf{N}} r_{ij}^e F_{ij} = \mathbf{traf}(e) \quad (6)$$

$$\forall i \in \mathbf{N}: \sum_{j \in \mathbf{N}} F_{ij} = \mathbf{ext}^{in}(i) \quad (7)$$

$$\forall j \in \mathbf{N}: \sum_{i \in \mathbf{N}} F_{ij} = \mathbf{ext}^{out}(j) \quad (8)$$

$$F_{ij} \geq 0$$

The objective function (5) now tries to find a value the predicted traffic on each edge in the network under the failure scenario, and finds bounds by running minimization and maximization optimization queries. The constraints (6, 7 and 8) are the same as for the traffic flow analysis.

2.6 Resilience Analysis Results

Results on the resilience analysis are a lot more encouraging as shown in table 4. The entry Low/Simul is calculated as

$$\left(100 * \frac{\sum_{e \in \mathbf{E}} \min \sum_{i,j \in \mathbf{N}} \overline{r_{ij}^e} * F_{ij}}{\sum_{e \in \mathbf{E}} \sum_{i,j \in \mathbf{N}} \overline{r_{ij}^e} * s_{ij}} \right)$$

the value High/Simul is

$$\left(100 * \frac{\sum_{kl \in \mathbf{E}} \max \sum_{i,j \in \mathbf{N}} \overline{r_{ij}^{kl}} * F_{ij}}{\sum_{kl \in \mathbf{E}} \sum_{i,j \in \mathbf{N}} \overline{r_{ij}^{kl}} * s_{ij}} \right)$$

We also report the number of objective functions (Obj), the total time (Time) and the number of failure cases (Cases) considered. We identify between 68 and 96% of the simulated traffic volume in the lower bound, and the sum of the upper bounds over-estimates the simulated traffic by a maximum of 9%.

Table 4. Resilience Analysis

Network	Low/Simul	High/Simul	Obj	Time	Cases
dexa	68.91	108.25	3503	57	59
as1221	85.75	102.60	14191	2869	153
as1239	92.53	102.64	4499	44205	10
as1755	92.82	105.39	8409	1815	161
as3257	93.69	103.15	31093	39934	328
as3967	91.60	108.79	9090	1635	141
as6461	96.51	103.44	24808	20840	374

Note that we did not run all failure cases on the largest network due to time limitations.

To check if the results are typical, we repeated the experiments with one hundred randomly generated data sets for the four smallest networks. In table 5 we show the average value and its standard deviation for both lower bounds and upper bounds. Results are quite consistent and confirm our initial values.

Table 5. Average results (100 runs) for resilience analysis

Network	Lower bound/Simul		Upper bound/Simul	
	Average	Stdev	Average	Stdev
dexa	91.50	0.14	108.28	0.16
as1755	88.65	0.11	106.08	0.056
as3967	94.08	0.073	106.88	0.091
as1221	87.34	0.10	102.05	0.025

2.7 Adding Information

One basic limit of the flow analysis is that we have too few constraints for too many variables. We now add a new data source to the problem, which will provide us with many additional constraints. In MPLS networks [13], we can not only use interface traffic counters, but also use counters on each LSP (label switched path) [42]. For each output interface, we get a counter for each LSP routed through the interface leading to a destination router. This counter will give us the sum of all flows to the destination which have been forwarded through the router, but it does not contain the flow that starts in the router. The MIB (management information base) also defines LSP counters on all input interfaces, but these counters are not meaningful on Cisco routers in current IOS versions.

We can define the LSP counters formally with the following definition.

Definition 1. *The constant v_e^j is the (consistent) LSP counter volume on the directed link e from node k to node l for all flows with destination node j which are forwarded on the link. The counter does not include the flow that originates in node k .*

This naturally leads to the next constraint, which links a sum of flow variables to the counter value.

Constraint 1 *The constraint states that the sum of all flows through a link towards a destination is equal to the LSP counter for that destination on the link.*

$$\forall j \in \mathbf{N}, e = (kl) \in \mathbf{E}: \sum_{i \in \mathbf{N}, i \neq k} r_{ij}^e * F_{ij} = v_e^j$$

The exact number of non-trivial constraints of this form depends on the topology, but usually is $O(n^2)$. This means that the problem is much more tightly constrained, and the results of the flow analysis should improve dramatically.

Table 6 shows the results of the experiments for traffic flow analysis. The lower bounds now range from 10 to 30 % of the simulated flows, the upper bounds range between 2.5 and 10 times the simulated flows. This is much better than before (see table 2), but still quite disappointing. The run times decreased a lot as well, adding more constraints helped the problem solving.

Table 6. TFA results with LSP counters

Network	Low/Simul	High/Simul	Obj	Time
dexa	30.35	249.71	1190	7
as1221	9.94	685.37	11556	885
as1239	10.74	1151.03	98910	72461
as1755	25.29	269.30	7482	397
as3257	23.77	425.67	25760	5121
as3967	24.47	300.17	6162	275
as6461	19.43	477.44	19740	2683

We also repeated the experiments for the PoP flow analysis. Table 7 shows the result. For some networks (dexa and AS1755) the results are nearly usable, but in general they are still not good enough to identify the flow values.

Table 7. PoP TFA results with LSP counters

Network	Low/Simul	High/Simul	Obj	Time
dexa	60.62	145.85	557	3
as1221	28.49	499.16	3205	271
as1239	33.36	211.84	1931	2569
as1755	50.33	169.37	526	46
as3257	36.82	249.16	2378	640
as3967	40.72	182.97	483	36
as6461	34.05	210.93	481	136

If we add LSP counter constraints to the model, then the results for resilience analysis are even more impressive. The lower bounds in table 8 reach 97 to 99.44 % of the simulated values, and the sum of the upper bounds is 101.33 % of the simulated values in the worst case. Also note that again the execution times decrease when we add the LSP counter constraints to the model, by more than a factor of 10 for the largest network.

Again we check the results for the 4 smallest networks by generating one hundred data sets and recording average percentage and standard deviation for

Table 8. Resilience Analysis with LSP counters

Network	Low/Simul	High/Simul	Obj	Time	Cases
dexa	97.76	101.33	3503	36	59
as1221	98.15	100.69	14191	1840	153
as1239	99.37	100.38	4499	3974	10
as1755	99.28	100.66	8409	964	161
as3257	99.41	100.44	31093	13381	328
as3967	98.88	101.00	9090	819	147
as6461	99.44	100.52	24808	8006	374

the sums of the lower bounds and the sums of the upper bounds compared to the simulated link traffic values in all single-node failure cases as shown in table 9.

Table 9. Average results (100 runs) for resilience analysis with LSP counters

Network	Lower bound/Simul		Upper bound/Simul	
	Average	Stdev	Average	Stdev
dexa	99.60	0.029	100.33	0.025
as1755	99.31	0.016	100.63	0.015
as3967	99.41	0.014	100.61	0.014
as1221	98.10	0.025	100.57	0.010

The results indicate that the resilience analysis with LSP counters is able to predict the link load in the network very accurately. In a similar way, other information can be added, for example partial netflow results for selected routers, bounds obtained from specific applications or from the VPN structure. In each case, the added information adds constraints to the problem, tightening the bounds obtained from the model. Since we obtain both lower and upper bounds, we can also easily decide how much additional data is required. Once the bounds are close enough, we can stop adding more information, thereby reducing the data collection overhead.

2.8 Discussion

In the presentation above, we have oversimplified the use of the actual traffic measurements. The models as shown only work if a consistent snapshot of all values can be collected. In practice, this poses significant problems. If the data are not collected for exactly the same time periods, then inconsistencies may occur. There are further problems caused by queues in the routers and bugs in implementing data collection facilities in devices of multiple vendors. The data collection process itself uses unreliable communications (UDP) so that some measurements may be lost due to dropped packets. One approach to overcoming

these issues is the use of a separate error correction model, which tries to correct values before feeding them into the models above. Another, shown in [52, 51, 53] deals with the problem by integrating incomplete and inconsistent data into the constraint solving process.

References

1. F. Ajili, R. Rodosek, and A. Eremin. A branch-price-and-propagate approach for optimising IGP weight setting subject to unique shortest paths. In *Proceedings of the 20th Annual ACM Symposium on Applied Computing (ACM SAC '05)*, Santa Fe, New Mexico, March 2005.
2. F. Ajili, R. Rodosek, and A. Eremin. A scalable tabu search algorithm for optimising IGP routing. In *2nd International Network Optimization Conference (INOC '05)*, pages 348–354, March 2005.
3. J. Cao, D. Davis, S. Vander Weil, and B. Yu. Time-varying network tomography. *Journal of the American Statistical Association*, 2000.
4. R. Castro, M. Coates, G. Liang, R. Nowak, and B. Yu. Network tomography: Recent developments, 2003.
5. A. Chabrier. Heuristic branch-and-price-and-cut to solve a network design problem. In *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems CP-AI-OR 03*, Montreal, Canada, May 2003.
6. C. Chiopris and M. Fabris. Optimal management of a large computer network with CHIP. In *2nd Conf Practical Applications of Prolog*, London, UK, April 1994.
7. Y. Chu and Q. Xia. Bandwidth-on-demand problem and temporal decomposition. In *2nd International Network Optimization Conference (INOC '05)*, pages 542–550, Lisbon, Portugal, March 2005.
8. M. Coates, A. Hero, R. Nowak, and B. Yu. Internet tomography. *IEEE Signal Processing Magazine*, May 2002.
9. W. Cronholm and F. Ajili. Strong cost-based filtering for Lagrange decomposition applied to network design. In M. Wallace, editor, *10th International Conference on Principles and Practice of Constraint Programming (CP 2004)*, pages 726–730, Toronto, Canada, 2004. Springer-Verlag.
10. W. Cronholm and F. Ajili. Hybrid branch-and-price for multicast network design. In *2nd International Network Optimization Conference (INOC '05)*, pages 796–802, Lisbon, Portugal, March 2005.
11. W. Cronholm, W. Ouaja, and F. Ajili. Strengthening optimality reasoning for a network routing application. In *4th International Workshop on Cooperative Solvers in Constraint Programming (CoSolv '04)*, Toronto, Canada, September 2004.
12. W. Cronholm, W. Ouaja, and F. Ajili. Strong reduced cost fixing in network routing. In *2nd International Network Optimization Conference (INOC '05)*, pages 688–694, Lisbon, Portugal, March 2005.
13. B. Davie and Y. Rekhter. *MPLS: Technology and Applications*. Morgan Kaufmann Publishers, 2000.
14. N. Duffield, C. Lund, and M. Thorup. Charging from sampled network usage. In *SIGCOMM Internet Measurement workshop*, November 2001.
15. N. G. Duffield and M. Grossglauser. Trajectory sampling for direct traffic observation. *IEEE/ACM Transactions on Networking*, pages 226–237, June 2001.
16. A. Eremin, F. Ajili, and R. Rodosek. A set-based approach to the optimal IGP weight setting problem. In *2nd International Network Optimization Conference (INOC '05)*, pages 386–392, Lisbon, Portugal, March 2005.

17. C. Estan and G. Varghese. New directions in traffic measurement and accounting. In *SIGCOMM2002*, September 2002.
18. C. Frei and B. Faltings. Resource allocation in networks using abstraction and constraint satisfaction techniques. In *Principles and Practice of Constraint Programming - CP 1999*, Alexandria, Virginia, October 1999.
19. T. Fruehwirth and P. Brisset. Optimal placement of base stations in wireless indoor telecommunication. In *Principles and Practice of Constraint Programming - CP 1998*, Pisa, Italy, October 1998.
20. O. Goldschmidt. ISP backbone traffic inference methods to support traffic engineering. In *Internet Statistics and Metrics Analysis (ISMA) Workshop*, San Diego, CA, December 2000.
21. M. Grossglauser and J. Rexford. Passive traffic measurement for IP operations. Technical report, ATT, 2001.
22. O. Kamarainen and H. El Sakkout. Local probing applied to network routing. In *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems CP-AI-OR 04*, Nice, France, April 2004.
23. M. Lauvergne, P. David, and P. Bauzimaault. Connections reservation with rerouting for ATM networks: A hybrid approach with constraints. In P. Van Hentenryck, editor, *Principles and Practice of Constraint Programming - CP 2002*, Cornell University, Ithaca, N.Y., September 2002.
24. C. Le Pape, L. Perron, J. Regin, and P. Shaw. Robust and parallel solving of a network design problem. In P. Van Hentenryck, editor, *Principles and Practice of Constraint Programming - CP 2002*, Cornell University, Ithaca, N.Y., September 2002.
25. J. Lever. A local search/constraint propagation hybrid for a network routing problem. In *The 17th International FLAIRS Conference (FLAIRS-2004)*, Miami Beach, Florida, May 2004.
26. J. Lever. A local search/constraint propagation hybrid for a network routing problem. *International Journal on Artificial Intelligence Tools*, 14(1-2):43–60, 2005.
27. V. Liatsos, S. Novello, and H. El Sakkout. A probe backtrack search algorithm for network routing. In *Proceedings of the Third International Workshop on Cooperative Solvers in Constraint Programming, CoSol'03*, Kinsale, Ireland, September 2003.
28. S. Loudni, P. David, and P. Boizumault. On-line resource allocation for ATM networks with rerouting. In *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems CP-AI-OR 03*, Montreal, Canada, May 2003.
29. R. Mahajan, N. Spring, D. Wetherall, and T. Anderson. Inferring link weights using end-to-end measurements. In *IMW2002*, 2002.
30. A. Medina, C. Fraleigh, N. Taft, S. Bhattacharyya, and C. Diot. A taxonomy of IP traffic matrices. In *Workshop on Scalability and Traffic Control in IP Networks at the SPIE ITCOM+OPTICOMM 2002 Conference*, Boston, MA, June 2002.
31. A. Medina, N. Taft, K. Salamatian, S. Bhattacharyya, and C. Diot. Traffic matrices estimation: Existing techniques and new directions. In *ACM SIGCOMM2002*, Pittsburgh, PA, August 2002.
32. L. Michel and P. Van Hentenryck. A simple tabu search for warehouse location. *European Journal on Operations Research*, pages 576–591, 2004.
33. W. Ouaja and B. Richards. A hybrid solver for optimal routing of bandwidth-guaranteed traffic. In *INOC2003*, pages 441–447, 2003.
34. W. Ouaja and B. Richards. A hybrid multicommodity routing algorithm for traffic engineering. *Networks*, 43(3):125–140, 2004.

35. W. Ouaja and E. B. Richards. Hybrid Lagrangian relaxation for bandwidth-constrained routing: Knapsack decomposition. In *20th Annual ACM Symposium on Applied Computing (ACM SAC '05)*, pages 383–387, Santa Fe, New Mexico, March 2005.
36. R. Rodosek and B. Richards. RiskWise constraint model. Internal Note, 2000.
37. L. Ros, T. Creemers, E. Tourouta, and J. Riera. A global constraint model for integrated routing and scheduling on a transmission network. In *7th International Conference on Information Networks, Systems and Technologies*, Minsk, October 2001.
38. Y. Shang, M. Fromherz, Y. Zhang, and L. S. Crawford. Constraint-based routing for ad-hoc networks. In *IEEE Int. Conf. on Information Technology: Research and Education (ITRE 2003)*, pages 306–310, Newark, NJ, USA, August 2003.
39. H. Simonis. Resilience analysis in MPLS networks. Technical report, Parc Technologies Ltd, 2003.
40. H. Simonis. Constraint applications in networks. In F. Rossi, P. van Beek, and T. Walsh, editors, *Handbook of Constraint Programming*, chapter 25. Elsevier, 2006.
41. B. M. Smith. Symmetry and search in a network design problem. In Roman Barták and Michela Milano, editors, *CPAIOR*, volume 3524 of *Lecture Notes in Computer Science*, pages 336–350. Springer, 2005.
42. C. Srinivasan, A. Viswanathan, and T. D. Nadeau. Multiprotocol label switching (MPLS) label switching router (LSR) management information base. Technical report, IETF, October 2003. draft-ietf-mpls-lsr-mib-13.txt.
43. J. Symes. Bandwidth-on-demand services using MPLS-TE. In *MPLS World Congress 2004*, Paris, France, February 2004.
44. Y. Tsang, M. Coates, and R. Nowak. Passive network tomography using EM algorithms. In *IEEE Conf Acoust. Speech and Signal Proc*, May 2001.
45. P. Van Hentenryck and J.P. Carillon. Generality versus specificity: An experience with AI and OR techniques. In *AAAI*, pages 660–664, 1988.
46. R. Vanderbei and J. Iannone. An EM approach to OD matrix estimation. Technical Report SOR 94-04, Princeton University, 1994.
47. Y. Vardi. Network tomography: Estimating source-destination traffic intensities from link data. *Journal of the American Statistical Association*, pages 365–377, 1996.
48. Q. Xia. Traffic diversion problem: Reformulation and new solutions. In *2nd International Network Optimization Conference (INOC '05)*, pages 235–241, Lisbon, Portugal, March 2005.
49. Q. Xia, A. Eremin, and M. Wallace. Problem decomposition for traffic diversions. In *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems CP-AI-OR 2004*, pages 348–363, Nice, France, April 2004.
50. Q. Xia and H. Simonis. Primary/secondary path generation problem: Reformulation, solutions and comparisons. In *4th International Conference on Networking*, Reunion Island, France, 2005. Springer Verlag.
51. N. Yorke-Smith. *Reliable Constraint Reasoning with Uncertain Data*. PhD thesis, IC-Parc, Imperial College London, University of London, June 2004.
52. N. Yorke-Smith and C. Gervet. On constraint problems with incomplete or erroneous data. In P. Van Hentenryck, editor, *Principles and Practice of Constraint Programming - CP 2002*, Cornell University, Ithaca, N.Y., September 2002.
53. N. Yorke-Smith and C. Gervet. Tight and tractable reformulations for uncertain CSPs. In *CP '04 Workshop on Modelling and Reformulating Constraint Satisfaction Problems*, Toronto, Canada, September 2004.

54. Y. Zhang, M. Roughan, N.G. Duffield, and A. Greenberg. Fast accurate computation of large-scale IP traffic matrices from link loads. In *ACM Sigmetrics*, 2003.