Solution Strategies for the Multi-Hour Network Design Problem

S.E. Terblanche[1], R. Wessäly[2], J.M. Hattingh[1]

[1] School of Computer, Statistical and Mathematical Sciences
North-West University (Potchefstroom), South Africa
{fanie.terblanche,giel.hattingh}@nwu.ac.za

[2] atesio GmbH, Zuse Institute Berlin, Germany
wessaely@{atesio,zib}.de

Abstract

The problem of finding the most cost efficient network design by considering two or more demand matrices that represent varying network traffic over several time periods, is referred to as multi-hour network design. Capacity installations need to be determined such that the network is robust enough to allow routing of the demand matrices non-simultaneously. In the paper the multi-hour network design problem is formulated as a mixed-integer programming model and the solution approach is based on a branch-and-cut framework that incorporates several classes of valid inequalities. The aim of the paper is to investigate the effect of problem reduction on computing times. Results from domination theory are applied to reduce the number of non-simultaneous traffic demand matrices to a so-called dominant (sub-)set. That is, matrices from the original data set are removed which do not change the solution space of a feasible network. In addition, strategies for strengthening the right-hand side of metric inequalities are presented and computational results are provided based on measurements from an operational network.

Keywords: Network design, multi-hour, integer programming

1. Introduction

The aim of solving the multi-hour network design problem is to find the most cost efficient network design that specifies equipment to be installed, capacities for all communication links, and routing information by considering multiple non-simultaneous traffic demand matrices. The basic philosophy is that traffic variations over several time periods can be represented by a finite range of multiple traffic demand matrices.

From an algorithmic point of view it is expected that the number of demand matrices that will be considered as part of the problem domain will contribute to the complexity of the problem. The network design problem with one traffic demand matrix is already considered to be a hard problem (see [1]). It is, therefore, the aim of this paper to investigate what the effect is on computing times by reducing the number of non-simultaneous demand matrices. The algorithm developed in [19] is applied to identify traffic demand matrices which can be removed from the problem description without changing the cost of an optimal network design. The remaining sub-set of traffic demand matrices is referred to as a dominant set. The second objective of the paper is to present an approach for strengthening the right-hand side of metric inequalities and to produce computational results based on measurements from an operational network in Germany called G-WIN.

Several papers have been published that address the problem of network dimensioning with multi-hour traffic conditions, but the underlying model assumptions and application areas distinguish them. Some of the earlier models for multi-hour network design have simplifying assumption such as continuous capacities (see e.g. [7], [12] and [16]). These type of models are computationally very attractive but are not practical since communication equipment cannot be installed in fractional quantities. It is for this reason then that we are only interested in models having integrality restrictions on at least the capacity variables.

Several solution approaches for solving the multi-hour network design problem with integrality restrictions have been proposed. A Lagrangian decomposition approach with subgradient optimization has been suggested by e.g. [11], [6], and [2]. In [3] a Benders decomposition followed by a so-called 3-phase heuristic is applied, and in [10] details on bounding procedures and heuristics are provided. In both [17] and [18] a Benders decomposition approach is followed where metric inequalities are separated for purposes of generating feasibility cuts.

The content of this paper is organized as follows: In Section 2. an overview of the model is presented which comprises two sub-problems, the hardware configuration and the routing problem. In Section 3. details of an algorithmic approach is presented and attention is drawn to problem reduction and the strengthening of metric inequalities. Some implementation details are presented in Section 4. and computational results are presented in Section 5. based on real world data that was provided by DFN-Verein, the German IP network provider for universities and research institutes. Finally a summary and some concluding remarks are given in Section 6..

2. The multiple demand set network design model

The proposed model is a mixed-integer programming model that is an extension of the component-resource model developed by [9]. It consists out of two parts, the hardware model and the routing model. The objective of the hardware model is to allow a detailed representation of potential hardware configurations found in typical SDH or opaque WDM networks. For instance, at each node of the network various options for installing ADMs (add-drop-multiplexers) or a DXCs/OXCs (digital/optical cross connects) are available. These technologies differ w.r.t. switching capacities, the number of available slots for interface cards, the type of interfaces supported, and costs etc. The potential technologies are abstracted into so called node designs and the optimization process is responsible for selecting the most cost effective node design for each node that will satisfy capacity requirements. Similarly, link designs enable the modeling of different transmission technologies that may have different capacities, cost structures and port requirements.

The objective of the routing model is to provide the hardware model with the capacity requirements on the communication links by finding feasible routings that will satisfy restrictions like hop count limits, survivability requirements, integral routing requirements etc. The overall model is, therefore, responsible for selecting an optimal hardware configuration and, simultaneously, find feasible routings such that all point-to-point traffic requirements are satisfied.

Extending the overall model to facilitate multiple demand matrices requires only modification of the routing model. The hardware model only receives as input from the routing model the capacity requirements and is, therefore, only indirectly dependent on demand information.

2..1 The Multiple Demand set Routing (MDR) model

Modeling the demand requirements for a communication network requires the concept of a demand matrix that specifies point to point demands. For ease of notation we will refer to a demand matrix as a *demand vector*. The notion of a commodity $k \in \mathcal{K}$ with $\mathcal{K} = \{1, 2, \ldots, K\}$ is adopted to differentiate between communicating node pairs. Thus, the demand vector $d \in \mathbb{R}_+^K$ represents the demand requirements for all communicating node pairs in the network. If no demand requirements exist for a commodity k, then $d_k = 0$. To facilitate the modeling of multiple demand vectors, a set of time periods $\mathcal{T} = \{1, 2, \ldots, T\}$ is used to index the set of demand vectors $D = \{d^1, d^2, \ldots, d^T\}$

Let $p \in \mathcal{P}$ be a set of edges that define a non-cyclic undirected path between a communicating node pair. The set $\mathcal{P}$ contains all possible paths for all possible node pairs and the subset $\mathcal{P}(k) \subseteq \mathcal{P}$ contains all paths that can route traffic for a commodity $k \in \mathcal{K}$. For each edge $e \in E$ the subsets $\mathcal{P}(k, e) \subseteq \mathcal{P}(k)$ contain all paths for a commodity k that traverse the edge e. The variables $f_p^t \in \mathbb{R}_+$ are introduced to define the flow of traffic on path $p \in \mathcal{P}$ for a time period $t \in \mathcal{T}$. The following set of constraints is found in the MDR model:

$$\sum_{p \in \mathcal{P}(k)} f_p^t = d_k^t \quad \forall\, k \in \mathcal{K},\, \forall\, t \in \mathcal{T} \tag{1}$$

$$\sum_{k \in \mathcal{K}} \sum_{p \in \mathcal{P}(k,e)} f_p^t \leq y_e \quad \forall\, e \in E,\, \forall\, t \in \mathcal{T} \tag{2}$$

with $y \in \mathbb{R}_+^{|E|}$ the capacity vector for which feasible routings should be found. The constraints (1) state that the sum of all the flows for a commodity should satisfy the demand requirements for each of the demand vectors. The capacity constraints (2) require that the capacity vector should be sufficiently large to accommodate the flow requirements.

3. Solution approach

A branch-and-cut scheme [15] is adopted to solve the joint hardware and routing problem. Several classes of valid inequalities are incorporated such as GUB cover and k-graph-partition inequalities [4], band inequalities [21], and strengthened metric inequalities [20].

At each node of the branch-and-bound process, the relaxation of the hardware model model yields a solution y^* for the capacity vector. A separation procedure is employed to generate any inequalities that are being violated by the current solution y^* (see [4]). In our case this separation procedure involves solving the linear programming problem MDR with $y := y^*$ as the capacity vector.

The contribution of this paper is towards investigating problem reduction as part of the solution approach as well as applying a procedure for strengthening the metric inequality right-hand side. The problem reduction approach followed is based on the theory of domination between traffic demand matrices [14].

3..1 Domination

The decision to include or exclude a demand vector in the computation of a network design depends on whether the demand vector is being dominated by some other demand vector or set of demand vectors. A demand vector d^1 *dominates* a demand vector d^2 if every capacity vector y that allows a feasible routing of d^1 also allows a feasible routing of d^2. The algorithm developed in [19] extends the notion of domination between two demand matrices to multiple demand matrices by utilizing convexity properties of multi-commodity flow requirements. The algorithm has therefore been used as a preprocessing step to reduce the problem instances for our computational study to a dominant subset of demand vectors.

3..2 Strengthening of the metric inequality right-hand side

Let μ_e for all $e \in E$ be arbitrary non-negative edge weights and $\pi_k = \min_{p \in \mathcal{P}(k)} \left\{ \sum_{e \in E(p)} \mu_e \right\}$ for all $k \in \mathcal{K}$ be the shortest path lengths for each of the commodities. Then a given capacity vector y^* will allow a feasible routing of the demand vector d if the following inequality holds:

$$\sum_{e \in E} y_e^* \mu_e \geq \sum_{k \in \mathcal{K}} d_k \pi_k \tag{3}$$

Inequalities of the form (3) are referred to as *metric inequalities* (see [13] and [8]) and are computed for the hardware model model by using duality information from the MDR model.

The following sub-problem (P) is obtained by introducing the vector $\alpha \in \mathbb{R}_+^T$ in the MDR model to represent the shortfall in capacity for a given capacity vector y^*:

$$\min \sum_{t \in \mathcal{T}} \alpha^t$$

$$\sum_{p \in \mathcal{P}(k)} f_p^t = d_k^t \quad \forall\, k \in \mathcal{K},\ \forall\, t \in \mathcal{T} \tag{4}$$

$$\sum_{k \in \mathcal{K}} \sum_{p \in \mathcal{P}(k,e)} f_p^t - \alpha^t \leq y_e^* \quad \forall\, e \in E,\ \forall\, t \in \mathcal{T} \tag{5}$$

To acknowledge feasibility of a capacity vector y^* the optimal solution to (P) should yield $\alpha^t = 0$ for all $t \in \mathcal{T}$. By taking the dual of (P) and associating the variables $\pi \in \mathbb{R}^{K \times T}$ and $\mu \in \mathbb{R}_+^{E \times T}$ with the constraints (4) and (5) respectively, the sub-problem (D) is obtained:

$$\max \sum_{t\in\mathcal{T}} \left(\sum_{k\in\mathcal{K}^t} d_k^t \pi_k^t - \sum_{e\in E} y_e^* \mu_e^t \right)$$

$$\sum_{e\in E} \mu_e^t = 1 \quad \forall\, t \,\in\, \mathcal{T} \tag{6}$$

$$\pi_k^t - \sum_{e\in E(p)} \mu_e^t \leq 0 \quad \forall\, p \in \mathcal{P}(k), \quad \begin{array}{l}\forall\, k \in \mathcal{K},\\ \forall\, t \in \mathcal{T}\end{array} \tag{7}$$

If the solution of (D) indicates an infeasible capacity vector, i.e. the optimal objective value to (D) is positive, then the following valid metric inequality can be added to the hardware model model to cut off this infeasible capacity vector :

$$\sum_{t\in\mathcal{T}} \left(\sum_{e\in E} y_e^* \mu_e^t \geq \sum_{k\in\mathcal{K}} d_k^t \pi_k^t \right) \tag{8}$$

Note that the single metric inequality (8) can be replaced by the following multi-cut equivalent since the problem (D) can be decomposed nicely into T sub-problems and each sub-problem can be solved independently:

$$\sum_{e\in E} y_e^* \mu_e^t \geq \sum_{k\in\mathcal{K}} d_k^t \pi_k^t \quad \forall\, t \,\in\, \mathcal{T} \tag{9}$$

For each of these multi cuts a strengthening procedure is applied before being added to the hardware model model. Consider a metric inequality for a time period $t \in \mathcal{T}$ and for an arbitrary capacity vector y^*. Note that although we can obtain values for μ_e^t and π_k^t from duality, the metric inequality should hold irrespective of how the edge weights have been calculated. More specifically, the edge weights could be calculated independently of the demand vector d^t. This in turn implies that for an arbitrary set of edge weights μ_e^t for all $e \in E$ and shortest path lengths π_k^t for all $k \in \mathcal{K}$, the capacity vector y^* will allow a feasible routing of the demand vectors $d^1, d^2, \ldots, d^T$ if the following inequality holds:

$$\sum_{e\in E} y_e^* \mu_e^t \geq \sum_{k\in\mathcal{K}} d_k^s \pi_k^t \quad \forall\, s \,\in\, \mathcal{T}$$

which is equivalent to saying

$$\sum_{e\in E} y_e^* \mu_e^t \geq \max_{s\in\mathcal{T}} \left\{ \sum_{k\in\mathcal{K}} d_k^s \pi_k^t \right\} \tag{10}$$

4. Implementation details

The implementation of the network design problem with multiple demand vectors has been done using C++ and is based on the network design tool DISCNET (see [20] and [5]). The separation procedures for valid inequalities have been implemented as part of call-back functions in the commercial mathematical programming software CPLEX.

Two approaches for implementing separation of metric inequalities for multiple demand vectors have been investigated. The first approach is to solve a linear programming problem (LP) for each of the time periods at each node of the branch-and-cut process. Or secondly, to have only one LP and to replace, prior to solving the LP for a specific time period, the right hand side of the LP with the corresponding demand vector. From our experience using only one LP has proven to perform better in most cases, especially with larger problem instances.

In order to manage the enormous number of path variables in the problem a column generation approach is followed. Consider the sub-problem (D) in Section 3..2 for a single time period $t \in \mathcal{T}$. The constraint set (7) states that for a commodity $k \in \mathcal{K}$ variable π_k^t is less than or equal to the minimum of the expression $\sum_{e\in E(p)} \mu_e^t$ for all paths $p \in \mathcal{P}(k)$. This is equivalent to saying that each π_k is the shortest path length for

Matrices	*%Gap*	*Nodes Expl.*	*LB*	*UB*
Original Data Sets				
12	8.14	18580	254707734	275448511
28	9.97	6589	311695117	342783859
288	17.9	570	277402845	327098006
Dominant Data Sets				
10	8.23	20150	255416923	276453911
21	11.76	7439	312046568	348762771
109	15.4	1240	269239807	310649877
Strengthening Procedure Applied				
10	10.1	19510	255660104	281591951
21	10.6	7420	312215254	345363761
109	14.7	1130	270424802	310200341

Table 1: Computational Results for G-WIN data

Matrices	*%Gap*	*Nodes Expl.*	*LB*	*UB*
Original Data Sets				
100	13.23	2828	255678463	289516052
200	16.10	837	252554215	293226696
400	11.08	385	255640657	283968377
800	-	-	-	-
1600	-	-	-	-
Dominant Data Sets				
15	10.16	17276	259693296	286082448
20	11.03	13700	259337982	287949236
40	12.65	9537	260451539	293419345
80	12.37	4693	258729596	290734932
154	13.77	1808	260626802	296522532
Strengthening Procedure Applied				
15	11.06	16448	259148685	287812188
20	10.69	14861	259394937	287148133
40	13.42	9402	260619779	295602231
80	11.33	5732	259290148	288678634
154	17.11	2132	260373234	304943047

Table 2: Computational Results for random data

a commodity $k \in \mathcal{K}$ calculated with edge weights μ_e^t, for all edges on the shortest path. It is, therefore, not necessary to include all possible paths in the formulation. The pricing operation required is simply to calculate shortest paths using the values for μ_e^t for each of the commodities. Whenever it is found that a shortest path has been calculated that is less than any of the π_k's, this new shortest path is added to the formulation.

5. Computational results

The data for our empirical work has been collected through measurements from an operational network called G-WIN (German Research Network). The network has 32 demand nodes and 64 potential links and three sets of demand data where measured. The first data set consisting out of 12 demand matrices was measured over a time period of a year, one demand matrix per month. The second data set consisting out of 28 demand matrices, one demand matrix per day, was measured in the month of February. The third data set was measured over a period of one day by taking samples every 5 minutes resulting in a total of 288 demand matrices.

Some preprocessing was done on the raw data that resulted in a reduced number of commodities. A minimum cut-off value of 1Mbps was applied and demands in opposite directions for the same communicating node pairs were aggregated to comply with an undirected commodity model. The average number of commodities per data set is 218 for the 12 monthly demand matrices, 242 for the 28 daily demand matrices and 211 for the 288 demand matrices measured from 5 minute intervals.

In addition to the data sets collected through measurements, random data was generated based on a worst case demand vector w that was created from the 12 monthly matrices. This was done by taking only the maximum demand values for each component over the 12 demand matrices. Demand vectors were then created randomly by using the following:

$$d_k^i = \rho^i \gamma^k w_k$$

where d_k^i is the k-th demand for a demand vector d^i generated for a scenario i. The parameter ρ^i is for introducing variation across all the scenarios and the parameter γ^k is to randomize demands within a demand vector. The two parameters were sampled out of the uniform distributions $U(0.5, 1.5)$ and $U(0.2, 1, 2)$ respectively. The ranges for the two distributions were selected arbitrarily. The number of scenarios that we considered was 100, 200, 400, 800, and 1600.

In terms of the hardware model, only one potential node design was specified for each of the nodes in the network. This node design can host up to five different modules, with each module compatible to a different interface. To enable modeling different edge capacities a total of 18 different link designs was considered. Each link design has in addition to different cost structures different interface requirements. The resulting hardware model formulation has 1984 integer variables and 1568 constraints.

The domination checking algorithm in [19] was applied to both the G-WIN data as well as the randomly generated data. For the G-WIN data the 288 matrices measured in 5 minute intervals were reduced to 109, the 28 daily matrices were reduced to 21, and the 12 monthly matrices were reduced to 10. For the randomly generated data, the original data sets consisting out of 100, 200, 400, 800, and 1600 matrices, were reduced to 15, 20, 40, 80 and 154 respectively.

Running times for all of the data sets were limited to 3 hours. Table 1 shows a summary of the computational results for the G-WIN data and Table 2 shows the results for the randomly generated data. The column labeled *Nodes Expl.* gives the number of branch-and-bound nodes explored and the columns *LB* and *UB* gives the lower and upper bounds respectively.

From the computational results it is interesting to note that not all problem reductions resulted in an improvement in the percentage gap. For instance, the only G-WIN data set for which an improvement can be observed is for the data set with 109 demand matrices. However, for all three dominant set of demand matrices 10, 21 and 109, the total number of branch-and-bound nodes explored are higher than for the original data sets. For instance, for the 288 demand matrices that were reduced to 109 (i.e. 62% reduction), the number of nodes explored increased from 570 to 1240.

For the randomly generated data sets the results are more promising. It is only for the random data set of 400 (reduced to 40) for which no improvement in percentage gap can be observed. In fact, for the original random data sets with 800 and 1600 matrices no feasible solutions could be found after 3 hours of running time. But after reducing the two data sets to 80 and 154 respectively, feasible solution could be found.

There has been an improvement in percentage gap due to the application of the strengthening procedure for some of the problem instances. In most of the problem instances there has been at least an improvement in the lower bound.

6. Summary and conclusion

In this paper we considered the multi-hour network design problem where variations in network traffic are being represented by a collection of non-simultaneous demand matrices. The model considered takes cognizance of detailed hardware requirements and computational results are based on demand matrices measured from an operational network.

An existing dominance checking algorithm was applied to reduce the problem space without discarding feasible solutions to the network design problem. The most successful reduction was for the data set of 288 demand matrices that was reduced to 109, resulting in an improvement in quality gap from 17.9% to 15.4%. A further improvement, resulting in a gap of 14.7%, was achieved for the same data set by applying the strengthening procedure. The overall gain, for applying the reduction algorithm based on domination theory, is that there was an increase in the number of branch-and-bound nodes explored. Furthermore, solveability was gained for larger instances such as the data sets with 800 and 1600 demand matrices.

An interesting observation is that the quality gaps obtained from our computational study for the network design problem with multiple traffic demand matrices, are within the same range as what we would expect for the network design problem with one traffic demand matrix. It therefore appears as if the computational complexity for the multi-hour case is not much more difficult than for the single traffic demand matrix case.

References

[1] SNDlib 1.0 – Survivable network design data library. http://sndlib.zib.de, 2005.

[2] A. Amiri and H. Pirkul. Routing and capacity assignment in backbone communication networks under time varying traffic conditions. *European Journal of Operational Research*, 117:15–29, 1999.

[3] K. Chari and A. Dutta. Design of private backbone networks - I: time varying traffic. *European Journal of Operational Research*, 67:428–442, 1993.

[4] G. Dahl and M. Stoer. A polyhedral approach to multicommodity survivable network design. *Numerische Mathematik*, 68:149–167, 1994.

[5] DISCNET, www.atesio.de, atesio GmbH, Berlin, Germany.

[6] A. Dutta. Capacity planning of private networks using DCS under multibusy-hour traffic. *IEEE Transactions on Communications*, 42(7):2371–2374, 1994.

[7] R.E. Gomory and T.C. Hu. Synthesis of a communication network. *Journal of the Society for Industrial and Applied Mathematics*, 12/2:348–369, 1964.

[8] O. Kakusho and K. Onaga. On feasibility conditions of multicommodity flows in networks. *Transactions on Circuit Theory*, 18:425–429, 1971.

[9] A. Kröller. Network optimization: Integration of hardware configuration and capacity dimensioning. Diploma thesis, Technische Universität Berlin, June 2003.

[10] M. Labbé, R. Séquin, P. Soriano, and C. Wynants. Network synthesis with non-simultaneous multicommodity flow requirements: Bounds and heuristics. Technical report, Centre for Research on Transportation, Université de Montréal, 1999.

[11] D. Medhi. Multi-hour, multi-traffic class network design for virtual path-based dynamically reconfigurable wide-area atm networks. *IEEE/ACM Transactions on Networking*, 3:809–818, 1995.

[12] M. Minoux. Optimum synthesis of a network with non-simultaneous multicommodity flow requirements. *Annals of Discrete Mathematics*, 11:269–277, 1981.

[13] M.Iri. On an extension of the maximum-flow minimum-cut theorem to multicommodity flows. *Journal of the Operations Research Society of Japan*, 13:129–135, 1971.

[14] G. Oriolo. Domination between traffic matrices. http://www.optimization-online.org, 2004.

[15] M. Padberg and G. Rinaldi. A branch and cut algorithm for the resolution of large-scale symmetric traveling salesman problems. *SIAM review*, 33:60–100, 1991.

[16] G.G. Polak and B.T. Smith. Multi-hour dimensioning in non-hierarchical telecommunications networks. In *Telecommunications Network Planning*. 1999.

[17] M. Riis and K.A. Andersen. Capacitated network design with uncertain demand. *INFORMS Journal on Computing*, 14(3):247–260, 2002.

[18] S.E. Terblanche, R. Wessäly, and J.M. Hattingh. Solutions for the survivable multi-hour network design problem. In *Proceedings of the South African Telecommuniations and Networks Applications Conference - SATNAC*, 2005.

[19] S.E. Terblanche, R. Wessäly, and J.M. Hattingh. Finding a dominant set of traffic demand matrices. In *Proceedings of the South African Telecommuniations and Networks Applications Conference - SATNAC*, 2006.

[20] R. Wessäly. *DImensioning Survivable Capacitated NETworks*. Phd thesis, Technische Universität Berlin, 2000.

[21] L.A. Wolsey. Valid inequalities for 0/1 knapsacks and mips with generalised upper bound constraints. *Discrete Applied Mathematics*, 29:251–261, 1990.