Solutions for Multi-Hour Survivable Network Design Problems

S.E. Terblanche*, R. Wessäly†‡ and J.M. Hattingh*
*School of Computer, Statistical and Mathematical Sciences
North-West University (Potchefstroom)
Email: rkwset@puk.ac.za
†Konrad-Zuse-Institute für Informationstechnik Berlin, Germany
‡atesio GmbH Berlin, Germany
Email: wessaely@{zib,atesio}.de

***Abstract*— Solving the multi-hour survivable network design problem entails finding the most cost efficient network design given two or more demand matrices that represent network traffic for different (busy-)hours. In addition to capacity installations, feasible routings that satisfy the traffic requirements for all of the demand matrices need to be determined. The 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. Strategies for separating metric inequalities are presented and computational results are provided based on measurements from an operational network.**

I. Introduction

The aim of designing survivable networks is to find the most cost efficient network design that specifies equipment to be installed, capacities for all communication links, and routing information. In addition to typical inputs like equipment cost, allowable capacity types and survivability parameters, a traffic demand matrix specifying point-to-point demands is required.

A single traffic demand matrix may, however, not be sufficient in describing varying traffic conditions. During the course of a day several periods of peak network traffic may be observed. These periods are coined busy-hours. Apart from overall busy-hours, there is also the phenomenon that different communicating node pairs could have noncoincident busy-hours. For example, cities located in different time zones have different working hours and, consequently, will have noncoincident busy-hours. Another example is the noncoincident nature of business related traffic vs. residential traffic. In the evenings a certain amount of traffic might be relocated from some part of the network to another as residential traffic increases and work related traffic decreases.

Dimensioning a network following a conservative approach, that is considering the busy-hour traffic estimates over all communicating node pairs as coincident, could result in a very expensive network design. If on the other hand, average estimates are used for capacity planing, the bandwidth might be insufficient to meet QoS requirements. The ideal solution is to have a network design that is cost efficient but at the same time is capable of carrying noncoincident peak hour traffic by dynamically routing demands through parts of the network that are under utilised.

The objective of this paper is to present a model and solution approach where we consider multiple demand matrices for the problem of dimensioning survivable telecommunication networks at minimum cost. The basic philosophy is that traffic variations over several time periods can be represented by a finite range of multiple demand matrices. The solution obtained is expected to be a survivable network design that efficiently utilises capacity by allowing alternative routings for each of the different demand matrices.

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 apart. Some of the earlier models for multi-hour design have simplifying assumption such as continuous capacities (see e.g. [1] and [2]). These type of models are computationally very attractive, but are not practical since communication equipment cannot be installed in fractional quantities. For the remainder of this overview we will therefore only consider literature where models have integrality restrictions on at least the capacity variables.

In [3] the multi-hour network design problem for reconfigurable ATM networks has been considered. The model assumes unsplittable flows, i.e. a single virtual path (VP) between demands. The concept of different traffic classes is introduced that allows more than one VP per demand pair. Path selection variables are binary and are indexed according to a traffic class, a commodity, and a time period. Integer variables are used to define the number of capacity units that can be installed on an edge. A Lagrangian decomposition approach is followed by applying subgradient optimisation. Computational results are provided for data sets with up to 23 nodes, 33 edges, 105 commodities, 2 traffic classes and 3 time periods. In [4] variations of the multi-hour dimensioning problem are presented that includes e.g., link blocking for circuit-switched networks, reconfigurable splittable flows, and non-reconfigurable unsplittable flows.

The mixed-integer programming model presented in [5] uses binary variables for modelling different capacity types that can be installed on an edge. Integer variables are used for specifying the number of circuits carried on one or more paths in order to satisfy demand requirements. Lagrangian relaxation is used as a solution approach and computational results are provided for problem instances of up to 20 nodes and 3 busy-hours. In [6] a similar model is presented but with the slight modification that the capacity variables are treated as pure integers and, consequently, the number of units installable for one type of capacity is modeled instead of different capacity types. A Benders decomposition approach is followed whereby both the master and subproblem are solved heuristically. Computational results are provided for up to 8 nodes and 3 busy-hours.

The model suggested by [7] is a non-linear mixed integer programming model, where the non-linearity is a result of the requirement to model the demand as independent M/M/1 queues in which links are treated as servers with service rates proportional to the link capacities. Binary variables are used to model discrete capacity types on the links, and also for assigning paths to edges for each time period. By applying Lagrangian relaxation the problem is decomposed into a capacity allocation and routing subproblem. Computational results are provided for problem instances of up to 32 nodes, 60 edges and 4 busy-hours.

In [8], the authors investigated the network synthesis problem with non-simultaneous multicommodity flow requirements. Details on bounding procedures and heuristics are provided for both the multi-hour and the restoration problems. Integrality restrictions are imposed on the capacities as well as on the flow variables. Computational results are provided for up to 45 nodes, 63 edges, 297 commodities and 4 time periods.

In [9] a stochastic integer programming model is considered for the problem of network design with uncertain demands. Although not explicitly shown in the paper, the problem is analogous to the multi-hour network design problem since the discretisation of the traffic distributions results in several noncoincident demand matrices. For this model, capacities can be installed in integer multiples of a low bandwidth type and a high bandwidth type, where as the flow variables are treated as continuous. The solution approach applied is a modified L-shape method that combines ordinary Benders decomposition with a branch-and-cut scheme. Proofs are provided to generalise well known valid inequalities (e.g. metric and partition inequalities) to the case where multiple scenarios for modelling random demands are considered. Computational results are provided for two sets of problems, the Atlanta data set (15 nodes and 22 edges) and the New York data set (16 nodes and 49 edges). Optimal solutions are obtained for the first data set. But for the latter, quality gaps of up to 4.88% are reported with the algorithm being terminated after 3 hours.

To the best of our knowledge no other references in literature addresses the problem of multi-hour network design by taking into account survivability.

The content of this paper is organised as follows: In Section II, an overview of the model is presented which comprises of two sub-problems, the hardware configuration and the routing problem. In Section III, details of an algorithmic approach is presented and attention is drawn to the application of valid inequalities that are used as part of a branch-and-cut approach. Some implementation details are presented in Section IV. Computational results are presented in Section V for the solution approaches that we applied using real world data that was provided by the DFN-Verein, the German IP network provider for universities and research institutes. Finally a summary and some concluding remarks are given in Section VI.

II. The multi demand network design model

The proposed model is a mixed-integer programming model that is an extension of the component-resource model developed by [10]. 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 optimisation process is responsible for selecting the most cost effective node design for each node that will satisfy capacity requirements. Similarly, link designs enable the modelling 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 are, therefore, not dependent on any demand information. The hardware model will however be presented for the sake of completeness.

A. The Hardware (HW) model

Let $G = (V, E)$ be the supply graph where V is the set of potential node locations and E is the set of potential communication edges.

At each node $v \in V$ a set of admissible node designs $\mathcal{N}(v)$ are defined, and at most one node design $n \in \mathcal{N}(v)$ can be selected for installation. Likewise, for each edge $e \in E$ at most one link design $l \in \mathcal{L}(e)$ can be selected for installation from the predefined set of admissible link designs $\mathcal{L}(e)$.

The attributes of a node design $n \in \mathcal{N}(v)$ are, a set of installable module types $\mathcal{M}(n)$, a limit $M^{m,n} \in \mathbb{Z}_+$ on the number of modules of type $m \in \mathcal{M}(n)$ that may be installed, the available number of slots $S^n \in \mathbb{Z}_+$ that can be occupied by one or more modules, a switching capacity $C^n \in \mathbb{Z}_+$, and a cost $c^n \in \mathbb{R}_+$. Furthermore, each type of module $m \in \mathcal{M}(n)$ are designed to occupy a number of slots $S^m \in \mathbb{Z}_+$ and can accommodate several types of interfaces $\mathcal{I}(m)$. Each module also has a cost of $c^m \in \mathbb{R}_+$.

To enable the matching of link technologies with node technologies, each link design $l \in \mathcal{L}(e)$ for an edge $e \in E$ has a set of interface types $\mathcal{I}(l)$. Matching the interface types of a module m on a node n with the interface types of an edge e allows us to connect the edge e with the node n. There is a limit $I_i^m \in \mathbb{Z}_+$ on the number of interfaces of type $i \in \mathcal{I}(m)$ for a module m and a limit $I_i^l \in \mathbb{Z}_+$ on the number of interfaces of type $i \in \mathcal{I}(l)$ for a link design l.

For ease of notation let $\mathcal{M}(v) := \bigcup_{n\in\mathcal{N}(v)} \mathcal{M}(n, v)$ denote the set of modules installable at node $v \in V$ and $\mathcal{I}(v) := \bigcup_{m\in\mathcal{M}(v)} \mathcal{I}(m)$ the set of potential interfaces at node $v \in V$.

The following decision variables are required to model the installation of the node designs, the modules, and the link designs.

- x_v^n ∈{0,1} indicates whether node design $n \in \mathcal{N}$ is installed on node $v \in V$ or not.
- x_e^l ∈{0,1} indicates whether edge design $l \in \mathcal{L}$ is installed on edge $e \in E$ or not.
- $x_v^m \geq 0$ integer, the number of modules of type $m \in \mathcal{M}$ installed at node $v \in V$.

Subsequently, the objective function of the hardware model is to minimise the installation cost of the node designs, the modules, and the link designs.

$$\min \sum_{v\in V} \left(\sum_{n\in\mathcal{N}(v)} c^n x_v^n + \sum_{m\in\mathcal{M}(v)} c^m x_v^m \right) + \sum_{e\in E} \sum_{l\in\mathcal{L}(e)} c^l x_e^l \tag{1}$$

The following set of constraints defines the rules for installing the necessary hardware that will satisfy a capacity vector y_e.

$$\sum_{n\in\mathcal{N}(v)} x_v^n \leq 1 \quad \forall v \in V \tag{2}$$

$$\sum_{l\in\mathcal{L}(e)} x_e^l \leq 1 \quad \forall e \in E \tag{3}$$

$$\sum_{l\in\mathcal{L}(e)} C^l x_e^l = y_e \ \forall e \in E \tag{4}$$

$$\sum_{e\in\delta(v)} \sum_{l\in\mathcal{L}(e)} C^l x_e^l - \sum_{n\in\mathcal{N}(v)} C^n x_v^n \leq 0 \quad \forall v \in V \tag{5}$$

$$\sum_{m\in\mathcal{M}(v)} S^m x_v^m - \sum_{n\in\mathcal{N}(v)} S^n x_v^n \leq 0 \quad \forall v \in V \tag{6}$$

$$\sum_{e\in\delta(v)} \sum_{l\in\mathcal{L}(e)} I_i^l x_e^l - \sum_{m\in\mathcal{M}(v)} \sum_{i\in\mathcal{I}(m)} I_i^m x_v^m \leq 0 \quad \begin{array}{l}\forall v \in V,\\ \forall i \in \mathcal{I}(v)\end{array} \tag{7}$$

$$x_v^m - \sum_{n\in\mathcal{N}(v)} M^{m,n} x_v^n \leq 0 \quad \begin{array}{l}\forall v \in V,\\ \forall m \in \mathcal{M}(v)\end{array} \tag{8}$$

Constraints (2) and (3) enforces the rule that only one node design and one edge design be assigned to a each node and each edge respectively. Constraint sets (4) and (5) ensure that the capacity of a node is sufficient to switch all the capacity of its incident edges. Note that y_e is an auxiliary variable that gives the edge capacity resulting from a valid inequality generated from the solution of the MDR model (see subsequent section). To ensure that the slot requirements of installed modules do not exceed the available slots of a node design, constraint set (6) is applied. Constraint set (7) will ensure that the number of interfaces of type $i \in \mathcal{I}(v)$ that will be installed at node v is sufficient to accommodate the number of interfaces of type $i \in \mathcal{I}(l)$ required by the incident edges. An upper bound on the maximum number of modules of type m at a node v is defined by the constraint set (8).

B. The Multiple Demand Routing (MDR) model

Modelling the demand requirements for a communication network requires the concept of a demand matrix that specifies point to point demands. For ease of notation and mathematical correctness, we will refer to a demand matrix as a *demand vector*. The notion of a commodity $k \in \mathcal{K}$ with $\mathcal{K} = 1, 2, \dots, K$ 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. To facilitate the modelling of multiple demand vectors, a set of time periods $\mathcal{T} = 1, 2, \dots, T$ is introduced and for each time period $t \in \mathcal{T}$ a set of commodities $\mathcal{K}^t$ exist. The notation for a demand $d_k \in \mathbb{R}_+$ is now interpreted as the demand for commodity $k \in \mathcal{K}^t$, for a time period $t \in \mathcal{T}$. The absence of a superscript "t" will as rule indicate that a single time period and a single demand vector is applicable.

The concept of failure states is used to facilitate the modelling of survivability. Different failure scenarios can be modelled by letting each failure state $s \in S$ contain all the network components (nodes and edges) that fail simultaneously. A survivable network design therefore comprises feasible routings over all failure states.

Survivability could be achieved by considering two alternatives [11]. The first is to introduce a *diversification* parameter $\delta_k \subseteq (0, 1] \in \mathbb{R}_+$ that defines a limit on the fraction of the demand d_k that can be routed through any node or edge that is effected by a failure state. The result is that traffic is distributed more evenly across the network and in the case of component failure, traffic can be redirected through surviving paths. The second alternative is to impose a path length restriction. The parameter $\gamma_k \in \mathbb{N}_+$ is an upper bound on the number of edges being traversed by any path that routes (part of) the demand d_k. The idea is to avoid long paths that have a higher probability of being effected in the case of component failures.

The formulation of the MDR model is a path flow multicommodity problem that is extended to include survivability and multiple time periods. The variable $f_p \in \mathbb{R}_+$ is introduced to define the flow of traffic on path $p \in \mathcal{P}(k)$ for the commodity $k \in \mathcal{K}^t$. The set $\mathcal{P}(k)$ therefore contains all (non-cyclic) paths that can rout traffic for a commodity k in time period

t. The set $\mathcal{P}(k,s) \subseteq \mathcal{P}(k)$ is used to index all paths for a commodity k that is effected by a failure state $s \in S$, and the set $\mathcal{P}(k,e) \subseteq \mathcal{P}(k)$ is required to index all paths for a commodity k that traverses an edge e.

Recall that the HW model is independent of any demand information but depends on the MDR model to provide it with information on the capacity requirements. To achieve this the auxiliary variables $y_e \geq 0$ found in the HW model is also defined for the MDR model.

The following is a formulation of all the constraints found in the MDR model.

$$\sum_{p \in \mathcal{P}(k)} f_p = d_k \quad \forall\, k \in \mathcal{K}^t,\ \forall\, t \in \mathcal{T} \tag{9}$$

$$\sum_{p \in \mathcal{P}(k,s)} f_p \leq \delta_k d_k \quad \forall\, k \in \mathcal{K}^t,\ \begin{matrix}\forall\, t \in \mathcal{T}, \\ \forall\, s \in \mathcal{S}\end{matrix} \tag{10}$$

$$\sum_{k \in \mathcal{K}^t} \sum_{p \in \mathcal{P}(k,e)} f_p \leq y_e \quad \forall\, t \in \mathcal{T},\ \forall\, e \in E \tag{11}$$

The set of constraints (9) state that the sum of all the flows for a commodity should satisfy the demand requirements for each of the demand vectors. The diversification constraints (10) restricts the total amount of flow for a commodity to no more than a fraction δ_k of the total demand d_k. The capacity constraints (11) requires that the capacity vector should be sufficiently large to accommodate the flow requirements.

III. Algorithmic approach

Recognising that the mixed integer programming problem HW+MDR (presented in Section II) can be intractable, a branch-and-cut scheme [12] is adopted that incorporates several classes of valid inequalities. These inequalities include GUB cover and k-graph-partition inequalities [13], band inequalities [14], and strengthened metric inequalities [15].

At each node of the branch-and-bound process, the relaxation of the HW 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 [13]). In our case this separation procedure involves solving the linear programming problem MDR with $y := y^*$ as the capacity vector.

The inequalities directly effected by our extension of the routing model to cater for multiple demand vectors, are the metric and partition inequalities. These are the inequalities that depend on demand information during the separation process.

A. Metric Inequalities

For purposes of simplifying the overview on metric inequalities, the survivability constraints in the MDR model is ignored. This can be done without loss of generality. The following sub-problem (P) is obtained by introducing the variable $\alpha \in \mathbb{R}^T$ in the MDR model to represent the shortcomming in capacity for a given capacity vector y^*:

$$\min \sum_{t \in \mathcal{T}} \alpha^t \tag{12}$$

$$\sum_{p \in \mathcal{P}(k)} f_p = d_k \quad \forall\, k \in \mathcal{K}^t,\ \forall\, t\ \in\ \mathcal{T} \tag{13}$$

$$\sum_{k \in \mathcal{K}^t} \sum_{p \in \mathcal{P}(k,e)} f_p - \alpha^t \leq y_e^* \quad \forall\, e \in E,\ \forall\, t\ \in\ \mathcal{T} \tag{14}$$

$$\tag{15}$$

To acknowledge feasibility of a capacity vector y_e^* the optimal solution to (P) should yield $(\alpha^t)^* \leq 0$ for all $t \in \mathcal{T}$. Taking the dual of (P) the sub-problem (D) is obtained:

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

$$\sum_{e \in E} \mu_e^t = 1 \quad \forall\, t\ \in\ \mathcal{T} \tag{16}$$

$$\pi_k - \sum_{e \in E(p)} \mu_e^t \leq 0 \quad \forall\, p \in \mathcal{P}(k),\ \begin{matrix}\forall\, k \in \mathcal{K}^t, \\ \forall\, t \in \mathcal{T}\end{matrix} \tag{17}$$

In the case of a single time period, the feasibility test performed by solving (D) is considered successfull if the dual objective value is non-positive. In the case of multiple time periods, however, this might not be sufficient. The sum of feasibilities (i.e. sum of non-positive α's) might outweigh the sum of infeasibilities (i.e. sum of positive α's), falsely indicating that the feasibility test over all periods has been successfull. It is therefore necessary to evaluate the feasibility w.r.t. each time period separately. That is, the following collection of inequalities should hold to acknowledge feasibility of a capacity vector:

$$\sum_{e \in E} y_e (\mu_e^t)^* \geq \sum_{k \in \mathcal{K}} d_k \pi_k^* \ \forall\, t\ \in\ \mathcal{T} \tag{18}$$

If any of these inequalities are violated, they are added to the HW model to cut off the infeasible capacity vector y^*. The sub-problem (D) is referred to as the *feasibility test problem* and the inequalities (18) that are constructed if the feasibility test fails, are referred to as *metric inequalities* (see [16] and [17]).

The problem (D) decomposes nicely into T sub-problems and each sub-problem can be solved independently. An advantage anticipated by doing this is that path variables generated during the separation of one inequality may be reused in the next. Furthermore, optimal solutions for (D) found in a specific time period could be carried over as starting solutions in the subsequent period.

B. Partition Inequalities

Let $\tau = \{V_1, V_2, \ldots, V_m\}$ with $V_i \cap V_j =$Ø, $i \neq j$ be a partition of the node set V into $m \geq 2$ subsets. If $\phi(e) = \{u, v\} \in V$ is defined as the set of end nodes for an edge $e \in E$, then the set $E(\tau) = \{e \in E | \exists i \in \{1, \ldots, m\} : |\phi(e) \cap V_i| = 1\}$ contains all the edges for which the end nodes are in distinctive subsets. Similarly, by defining the set $\theta(k) = \{u, v\} \in V$ as the set of demand nodes for a commodity $k \in \mathcal{K}$, the set $\mathcal{K}(\tau) = \{k \in \mathcal{K} | \exists i \in \{1, \ldots, m\} : |\theta(k) \cap V_i| = 1\}$ contains all the commodities for which the demand nodes are in distinctive subsets.

The partition inequality for a partition τ is the following:

$$\sum_{e \in E(\tau)} y_e \geq \sum_{k \in \mathcal{K}(\tau)} d_k \tag{19}$$

To extend partition inequalities for multiple time periods the right hand side of (19) is replaced by the maximum demand over all the time periods since this will yield a stronger inequality:

$$\sum_{e \in E(\tau)} y_e \geq \max_{t \in \mathcal{T}} \left\{ \sum_{k \in \mathcal{K}^t(\tau)} d_k \right\} \tag{20}$$

IV. Implementation details

A. Strategies for separating Metric Inequalities

Several strategies for separating metric inequalities exist for the multi-hour problem since each time period is treated independently. Recall that the only data differentiating the time periods apart are the different demand vectors that are associated with them. The different strategies will consequently be defined in terms of functions related to the demand vectors.

1) Iteration strategies: As a first approach metric inequalities could be separated by simply iterating through the demand vectors associated with each of the time periods, while adding all violated metric inequalities to the LP relaxation. After solving the linear programming (LP) relaxation to the HW problem, the separation process is repeated if necessary by re-iterating through the list of demand vectors. A variation of this strategy is to specify a parameter that would stop after k successfull separation attempts and to pass the k generated cuts to the LP relaxation. If further separation is required in the next iteration of the cutting plane algorithm, the separation process restarts at the beginning of the list of demand vectors and the success counter is reset to zero. The motivation for exploring such a strategy is that the first k metric inequalities may be successfull in separating the infeasible capacity vector without considering the remaining violated inequalities. The strategy will be referred to as the k-Reset strategy reflecting the fact that the index to the set of demand vectors is being reset to the beginning of the list each time k successfull separations have taken place.

The alternative to the k-Reset strategy is the k-Continue strategy. As the name suggests, instead of resetting the index to the demand vector list after k successfull separations, the next round of separations continue from the previous index position. The result anticipated is that the danger of repeating the separation procedure for the same set of demand vectors, while making little progress in separating the infeasible capacity vector, is avoided.

Diversification	*%Gap*	*time(sec)*
1	0.078	8
$\frac{1}{2}$	0.075	68

TABLE I

INTEGER CAPACITY TYPES ($\lambda = 1$)

2) Ordering strategies: For both the k-Reset and the k-Continue strategies, it is expected that the ordering of demand vectors according to some ordering criteria, might improve the separation process. It is argued that if a metric inequality is separated using a "large" demand vector, then it is likely that a large portion of the infeasible region is separated and the capacity vector could accommodate the "smaller" demand vectors.

V. Computational results

The implementation of the survivable network design problem with multiple demand matrices is based on the network design tool DISCNET (see [15] and [18]). The code has been implemented using C++ and the commercial mathematical programming software CPLEX is used for solving the linear programming relaxations.

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 47 potential links. For our purpose 12 demand matrices have been identified from the set of measurements for representing different busy-hours.

As part of our study, the impact of two different capacity models on the quality of solutions, has been investigated. For the first capacity model under consideration, integer variables $y \in \mathbb{Z}_+^{|E|}$ are used for modelling the installation of multiple units of a single type of capacity (in [9] e.g., multiples of two capacity types were considered). To have a more realistic model it is assumed that the capacities are installed in multiples of λ batch sizes. Thus, the effective capacity is given by λy. The lower bound z_L^* was provided by the solution of the linear programming relaxation of the HW+MDR model. Let $y_L^* \in \mathbb{R}_+^{|E|}$ be the lower bound solution for the capacity vector y. The heuristic we applied for obtaining the upper bound z_U^* entailed calculating the upper bound solution for the capacity vector y as $y_U^* := \lceil y_L^* \rceil$. Table I shows the results for the cases where diversification parameters $\delta = 1$ (i.e. no survivability) and $\delta = \frac{1}{2}$ were used with $\lambda = 1$. The quality of the solutions is given by *%Gap* = $\frac{z_U^* - z_L^*}{z_U^*} \times 100$. From the results it is clear that very good solutions can be obtained for the multi-hour problem within a couple of seconds using a simple heuristic. Note, however, that there is a direct relationship between the quality of the solutions obtained and the value specified for λ. Larger values of λ will result in bigger quality gaps when applying the heuristic. For example, if $\lambda = 155$ (the base capacity STM-1), gaps of 12.313% and 11.362% are obtained for respectively solving the multi-hour problem with $\delta = 1$ and $\delta = \frac{1}{2}$.

Diversification = 1		Diversification = $\frac{1}{2}$	
Strategy	*%Gap*	Strategy	*%Gap*
rest-order-max12	25.921	rest-rand-max8	29.675
cont-order-max12	26.03	rest-rand-max6	31.977
cont-rand-max2	27.151	rest-order-max10	34.275
rest-rand-max8	27.319	rest-rand-max2	35.595
rest-rand-max10	27.359	rest-rand-max4	35.795
cont-order-max2	27.409	cont-rand-max8	36.608
rest-rand-max6	27.469	rest-rand-max10	36.916
rest-rand-max4	27.628	rest-order-max12	41.141
rest-order-max4	27.768	cont-order-max12	41.266
cont-rand-max4	27.772	rest-rand-max12	41.729
cont-rand-max10	27.957	rest-order-max4	42.79
rest-rand-max12	27.958	rest-order-max8	43.453
cont-rand-max6	27.959	rest-order-max2	45.063
cont-order-max4	27.99	rest-order-max6	49.568
rest-order-max8	28.001	cont-order-max2	57.804

TABLE II
SEPARATION STRATEGIES (ORDERED BY *%Gap*)

The alternative to the integer capacity model is the use of binary variables. As described in Section II-A the set of binary capacity variables represents a set of potential capacity types from which only one will be selected for installation. Since the above heuristic only applies to the integer capacity model, solutions were computed by applying the separation strategies described in Section IV-A. The potential capacity types that were considered are STM-1, STM-4, STM-16 and an aggregated type STM-4 + STM-16. The multi-hour problem was solved for several combinations of the separation strategies with a time limit of 1 hour. Table II shows the results for the combination of separation strategies sorted in increasing order according to the quality of the solutions. Only the first 15 results are listed for both the cases where $\delta = 1$ and $\delta = \frac{1}{2}$.

The names of the different combinations listed in the table are constructed in the following way: The first term indicates what type of iteration strategy was applied, i.e. either continue (cont) or restart (rest). The next term indicates whether the demand vectors have been ordered (decreasingly) or selected randomly, and the last term shows the limit k on the number of separations that was considered during each iteration of the cutting plane algorithm.

It is evident from the results that there is no clear pattern that shows which combination of strategies are dominant. There is also no correlation between the results obtained for the case where $\delta = 1$ and for the case where $\delta = \frac{1}{2}$. The only significant difference noticeable is that there is more variation in the quality gap for $\delta = \frac{1}{2}$, indicating that the multi-hour network design problem with survivability requirements is more sensitive to the choice of a specific separation strategy.

By comparing the best quality gaps obtained for the combination of separation strategies (25.921% and 29.675%) with the gaps obtained in the first experiment with the rounding heuristic (0.075% and 0.078%), it appears that problems with integer capacity models are much easier to solve than problems with binary capacity models. Furthermore, in the case where integer capacity models are used, it is clear that the the quality of the solutions is dependent on the choice of problem data such as the capacity batch sizes (i.e. the λ values).

VI. SUMMARY AND CONCLUSION

In this paper we considered the problem of multi-hour survivable network design. We attempted to show the relevance of multi-hour network design and pointed out that busy-hour traffic for different communicating node pairs may be noncoincident. Furthermore, a model and solution approach were presented for solving the multi-hour network design problem, taking into account survivability parameters.

Computational results were provided using data from an operational network. The results obtained showed that model assumptions such as different capacity models, and data, could have a significant influence on the quality of the solutions.

REFERENCES

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

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

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

[4] M. Pioro and D. Medhi, *Routing, flow, and capacity design in communication and computer networks*, Morgan Kaufmann, 2004.

[5] A. Dutta, "Capacity planning of private networks using DCS under multibusy-hour traffic," *IEEE Transactions on Communications*, vol. 42, no. 7, pp. 2371–2374, 1994.

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

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

[8] M. Labbé, René Séquin, P. Soriano, and C. Wynants, "Network synthesis with non-simultaneous multicommodity flow requirements: Bounds and heuristics," Tech. Rep., Centre for Research on Transportation, Université de Montréal, 1999.

[9] M. Riis and K.A. Andersen, "Capacitated network design with uncertain demand," *INFORMS Journal on Computing*, vol. 14, no. 3, pp. 247–260, 2002.

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

[11] D Alveras, M. Grötschel, and R. Wessäly, "Cost-efficient network synthesis from leased lines," *Annals of Operations Research*, vol. 76, pp. 1–20, 1998.

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

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

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

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

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

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

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