Solving a Multi-Constrained Network Design Problem by Lagrangean Decomposition and Column Generation[1]

Andreas M. Chwatal* Nina Musil* Günther R. Raidl*

*Institute of Computer Graphics and Algorithms, Vienna University of Technology
Favoritenstraße 9-11/E186, 1040 Wien, Austria
Email: {chwatal|raidl}@ads.tuwien.ac.at, nina.musil@yahoo.com

Abstract

In this paper we describe two approaches to solve a real-world multi-constrained network-design problem. The objective is to select the cheapest subset of links in a given network which enables to feasibly route messages from respective source to target nodes regarding various constraints. These constraints include particular capacity and delay constraints for each message, as well as a global delay constraint. Furthermore some messages may only be routed on connections supporting a secure protocol. The problem is strongly NP-hard and larger instances cannot be solved to provable optimality in practice. Hence, we present two heuristic approaches based on Lagrangean Decomposition and Column Generation, which turned out to be well suited. From these methods we obtain lower bounds as well as feasible solutions.

Keywords: *Network Design, Lagrangean Decomposition, Column Generation*

1 Introduction

In this work we consider a complex large-scale network-design problem. The particular problem emerged from an industry cooperation in the context of air traffic management. Various devices should be connected by a uniform network in order to facilitate efficient information exchange. As these communications ("messages") are highly safety critical, several restrictions need to be taken into account. Most important, it must be ensured that all messages can be routed in a given amount of time. Some communications need to take place faster than others and the total time to route all messages must not exceed a given time limit. In addition, some communications need to take place on secure network links, which usually implies higher costs. The optimization goal is to derive a minimum cost network that is sufficient to cover a simultaneous feasible routing of the given set of messages, which correspond to a maximum-load scenario.

The rest of the paper is organized as follows: In Section 2 we give a formal problem definition by presenting an Integer Linear Programming (ILP) formulation; then, in Section 3 we review underlying literature; Sections 4 and 5 describe our approaches to solve the given problem by either Lagrangean Decomposition (LD) or Column Generation (CG); after presenting some computational results in Section 6 we finally give our conclusions in Section 7.

[1]This work is supported by the Austrian Science Fund (FWF) under contract number P20342-N13.

2 Problem Formulation

We are given an undirected graph $G = (V, E)$ with a set of edges E defined on a set of nodes $V = S \cup T \cup M$ that is partitioned into source, target, and intermediate (Steiner) nodes, respectively. The edge set $E = E^+ \cup E^- \cup E^*$ is partitioned into a set E^+ of edges only supporting a *secure protocol*, a set E^- only supporting an *insecure protocol* and, a set E^* of edges supporting both kinds of protocols.

Further we are given a set of m *transports* T which model all communications and messages that should be sent over the network. Transport k is defined by the tuple $(s_k, t_k, v_k, \sigma_k, \delta_k)$, where $s_k, t_k \in V$ denote the source and target nodes respectively, $v_k \in \mathbb{R}^+$ the size of the transport and $\sigma_k \in \{0, 1\}$ a Boolean variable indicating if the transport is classified to take place over secure protocols only. Time-critical transports need to be routed within specified time-limits $\delta_k > 0$; otherwise we assume $\delta_k = \infty$.

For each edge we are further given nonnegative costs $c_{ij} \geq 0$, delays $d_{ij} \geq 0$ and capacities $u_{ij} \geq 0$. In addition to the edge costs, there are costs for the use of a certain protocol

$$(1) \qquad p_{ij}^k = \begin{cases} \min(p_{\text{sec}}, p_{\text{insec}}), & \text{if } \sigma_k = 0 \wedge (i,j) \in E^* \\ p_{\text{insec}}, & \text{if } (i,j) \in E^- \\ p_{\text{sec}}, & \text{otherwise,} \end{cases}$$

where $p_{\text{sec}} \geq 0$ denotes the costs for the secure protocol and $p_{\text{insec}} \geq 0$ the costs for the insecure protocol. Delays for the use of a certain protocol are defined analogously by

$$(2) \qquad a_{ij}^k = \begin{cases} \min(a_{\text{sec}}, a_{\text{insec}}), & \text{if } \sigma_k = 0 \wedge (i,j) \in E^* \\ a_{\text{insec}}, & \text{if } (i,j) \in E^- \\ a_{\text{sec}}, & \text{otherwise,} \end{cases}$$

where $a_{\text{sec}} \geq 0$ denotes the delay for the secure protocol and $a_{\text{insec}} \geq 0$ the delay of the insecure protocol. Here we assume that $p_{\text{insec}} < p_{\text{sec}}$ implies that $a_{\text{insec}} < a_{\text{sec}}$ and vice versa.

A solution to our problem consists of a feasible routing for each message k, i.e. a path from s_k to t_k, satisfying the delay constraints, and we are interested in a minimum cost solution. Of course one arc can be used by more than one transport, but as all transports are routed simultaneously, they must share its capacity. In addition to the delay-restrictions for each transport, there exists a global delay-constraint, which enforces the sum of all routing times to be less than a constant D.

We now formulate the problem as an ILP using a multi-commodity flow (MCF) approach, see [1]. For this purpose we define a directed graph $G' = (V, A)$, where $(i,j) \in A \wedge (j,i) \in A$ exactly when $(i,j) \in E$. Binary variables x_{ij}, $\forall (i,j) \in E$, indicate if an edge $(i,j) \in E$ is used by any transport. The routing-paths for each transport k are described by flow variables $f_{ij}^k \in \{0,1\}$, $\forall (i,j) \in A$, $1 \leq k \leq m$, indicating if arc (i,j) is used by this transport. We further define $A^k = \{(i,j) \in A \mid \sigma_k = 0 \vee (i,j) \notin E^-\}$, and E^k analogously for the undirected case. These sets of arcs are appropriate for routing transport k in correspondence to its security classification.

$$(3a) \qquad \min \sum_{(i,j) \in E} \Big(\underbrace{c_{ij} \cdot x_{ij}}_{\text{edge costs}} + \sum_{k=1}^{m} \underbrace{f_{ij}^k \cdot p_{ij}^k + f_{ji}^k \cdot p_{ji}^k}_{\text{protocol costs}} \Big)$$

$$(3b) \qquad \text{s.t.} \sum_{(i',i) \in A^k | i' \neq t_k} f_{i'i}^k - \sum_{(i,i'') \in A^k | i'' \neq s_k} f_{ii''}^k = 0 \qquad \forall i \in V \backslash \{s_k, t_k\},\ k = 1, \ldots, m$$

$$(3c) \qquad \sum_{(s_k,j) \in A^k} f_{s_k j}^k = 1 \qquad \forall k = 1, \ldots, m$$

$$(3d) \qquad \sum_{(i,t_k) \in A^k} f_{i t_k}^k = 1 \qquad \forall k = 1, \ldots, m$$

$$\sum_{k=1}^{m}(f_{ij}^k \cdot v_k + f_{ji}^k \cdot v_k) \leq u_{ij} \qquad \forall (i,j) \in E \tag{3e}$$

$$\sum_{(i,j)\in A^k} (d_{ij} + a_{ij}^k) \cdot f_{ij}^k \leq \delta_k \qquad \forall k = 1,\ldots,m \tag{3f}$$

$$\sum_{k=1}^{m} \sum_{(i,j)\in A} (d_{ij} + a_{ij}^k) \cdot f_{ij}^k \leq D \tag{3g}$$

$$x_{ij} \geq f_{ij}^k + f_{ji}^k \qquad \forall k = 1,\ldots,m,\ (i,j) \in A^k \tag{3h}$$

$$x_{ij} \in \{0,1\} \qquad \forall (i,j) \in E \tag{3i}$$

$$f_{ij}^k \in \{0,1\} \qquad \forall k = 1,\ldots,m,\ (i,j) \in A \tag{3j}$$

The objective function (3a) minimizes total edge and protocol costs. The latter ones are proportional to the number of transports routed over this arc. Each transport is required to be routed on an elementary path, which is enforced by flow conservation equalities (3b, 3c, 3d). Inequalities (3e) ensure that the capacity limit of each edge is not exceeded. Inequalities (3f) state that the delay constraint of each transport must be fulfilled, and finally inequality (3g) restricts the sum over all delays to D.

3 Previous Work

A similar problem, related to the same industry project, has been presented and solved by *POEMS* in [5]. POEMS stands for **P**rototype **O**ptimization with **E**volved **IM**provement **S**teps and is a metaheuristic approach which does not provide quality guarantees. However, differences are that the total delay is computed in another way and feasibility can be achieved by adding further delay in the case some capacity constraints are exceeded. So far, no approaches based on mathematical programming techniques have been published for our particular problem, and also no other methods to derive upper and lower bounds exist. Nevertheless, similar multi-commodity flow problems with additional constraints have been studied in literature, and methods like Lagrangean relaxation and column generation turned out to often be very successful. Our particular problem can be classified as a generalization of the *multi-commodity capacitated network design problem* [4], which is NP-hard. Differences are, however, that we additionally have to consider delay constraints and distinguish between secure and insecure messages.

The extensively studied problem of searching a single shortest path not exceeding a certain resource limit is referred to as *constrained shortest path* or *constrained least cost problem* [3]. This problem is NP-hard as well. Anyway, many instances can viably be solved by preprocessing techniques with a subsequent labeling algorithm based on dynamic programming. This algorithm is used by the Lagrangean decomposition (LD) and column generation (CG) methods we propose in the following.

4 Lagrangean Decomposition

Directly solving the ILP as stated in Section 2 is only possible for relatively small problem instances. By Lagrangean decomposition it is possible to compute lower bounds in relatively short time, and to derive good heuristic solutions from these results. We can also expect the lower bounds to be better than the ones obtained from an LP-relaxation as the integrality property holds. The main idea is to relax the linking constraints (3h) as well as constraints (3e) and (3g) and iteratively adapt their Lagrangean coefficients by the *volume algorithm* [2], which is a special variant of a subgradient algorithm. By basically replacing $f_{ij}^k + f_{ji}^k$ by variables $b_{ij}^k \in \{0,1\},\ \forall (i,j) \in E$, and performing the relaxation we obtain:

$$\min \sum_{(i,j)\in E} \left(c_{ij} \cdot x_{ij} \ + \sum_{k=1}^{m} b_{ij}^k \cdot p_{ij}^k \right) \tag{4a}$$

$$+ \sum_{(i,j)\in E} \lambda_{ij} \cdot \left(\sum_{k=1}^{m} \left(b_{ij}^k \cdot \upsilon_k \right) - u_{ij} \right) \tag{4b}$$

$$+ \lambda' \left(\sum_{k=1}^{m} \sum_{(i,j)\in E^k} \left((d_{ij} + a_{ij}^k) \cdot b_{ij}^k \right) - D \right) \tag{4c}$$

$$+ \sum_{k=1}^{m} \sum_{(i,j)\in E} \lambda''_{k,ij} \left(b_{ij}^k - x_{ij} \right) \tag{4d}$$

$$\text{s.t.} \quad b_{ij}^k \in C^k \tag{4e}$$

$$\sum_{(i,j)\in E^k} (d_{ij} + a_{ij}^k) \cdot b_{ij}^k \leq \min(\delta_k, D) \qquad \forall k = 1, \ldots, m \tag{4f}$$

$$b_{ij}^k \cdot \upsilon_k \leq u_{ij} \qquad \forall k = 1, \ldots, m, \ (i,j) \in E^k \tag{4g}$$

$$x_{ij} \in \{0, 1\} \qquad \forall (i,j) \in E \tag{4h}$$

$$b_{ij}^k \in \{0, 1\} \qquad \forall (i,j) \in E \tag{4i}$$

Here, C^k is the set of incidence vectors representing all feasible paths connecting s_k with t_k. To tighten the model we keep the delay constraints (4f) in the ILP model. Constraints (4g) are only included for completeness of the formulation; in our implementation they are already met by a preprocessing step that just includes valid arcs for each transport k in the resulting subproblem. The objective function (4a) to (4d) can be rewritten as follows

$$\min \quad -\lambda' \cdot D + \sum_{(i,j)\in E} \Big((c_{ij} \cdot x_{ij} - \lambda_{ij} \cdot u_{ij}) \tag{5}$$

$$+ \sum_{k=1}^{m} \left(b_{ij}^k \cdot \left(p_{ij}^k + \lambda_{ij} \cdot \upsilon_k + \lambda' \cdot (d_{ij} + a_{ij}^k) + \lambda''_{k,ij} \right) - \lambda''_{k,ij} \cdot x_{ij} \right) \Big) . \tag{6}$$

We can observe that in this relaxation we have now k independent delay-constrained shortest path problems to solve, where when considering transport k the new costs of an arc (i,j) are given by

$$\hat{c}_{ij}^k = p_{ij}^k + \lambda_{ij} \cdot \upsilon_k + \lambda'(d_{ij} + a_{ij}^k) + \lambda''_{k,ij} \, . \tag{7}$$

Each delay-constrained shortest path problem is solved by the algorithm from [3]. In a preprocessing step unnecessary edges are removed from the graph, which already yields the optimal solution in many cases. Otherwise the solution is obtained by a subsequent labeling algorithm, based on dynamic programming. The values x_{ij} are set by simple inspection according to

$$x_{ij} = \begin{cases} 0, & \text{if } c_{ij} - \sum_{k=1}^{m} \lambda''_{k,ij} > 0, \\ 1, & \text{otherwise.} \end{cases} \tag{8}$$

Based on this formulation we can solve the LD by the volume algorithm, which is a refined subgradient approach [2]. Afterwards, we try to derive a feasible solution by a Lagrangean heuristic. Paths obtained by LD are iteratively added to the solution. If the capacities of some arcs are exceeded by a transport k, the corresponding path is not added immediately. Instead our heuristic tries to find a new feasible path for this transport on the graph with residual capacities. For more details on this procedure see [7].

5 Column Generation

Alternatively, by using a path-based formulation rather than a flow-formulation, we can approach the problem by column generation; see [6] for a comprehensive review. Let P_k denote the set of all feasible, i.e. delay-constrained, paths for each transport k when no other transports are considered.

A particular path for a transport k from s_k to t_k is denoted by $\varphi_k \in P_k$. To indicate if an arc (i,j) belongs to path φ_k we use constants

$$\delta_{ij}^{\varphi_k} = \begin{cases} 1, & \text{if } (i,j) \in \varphi_k \\ 0, & \text{otherwise} \end{cases} \qquad \forall k = 1,\dots,m,\ (i,j) \in E. \tag{9}$$

For the total delay of a path $\delta_{ij}^{\varphi_k}$ we further introduce constants

$$a'_{\varphi_k} = \sum_{(i,j)\in E^k} \delta_{ij}^{\varphi_k} \cdot \left(a_{ij}^k + d_{ij}\right), \tag{10}$$

and analogously for the protocol costs

$$c'_{\varphi_k} = \sum_{(i,j)\in E^k} \delta_{ij}^{\varphi_k} \cdot p_{ij}^k. \tag{11}$$

To indicate if a path φ_k is selected we introduce the variable $\omega_{\varphi_k} \in \{0,1\}$. The problem can now be described by the following constrained set covering model, which is our *integer master problem* (IMP).

$$\min \sum_{k=1}^{m} \sum_{\varphi_k \in P_k} c'_{\varphi_k} \cdot w_{\varphi_k} + \sum_{(i,j)\in E} c_{ij} \cdot x_{ij} \tag{12a}$$

$$\text{s.t.} \sum_{\varphi_k \in P_k} w_{\varphi_k} \geq 1 \qquad \forall k = 1,\dots,m \tag{12b}$$

$$\sum_{\varphi_k \in P_k} \delta_{ij}^{\varphi_k} \cdot w_{\varphi_k} \leq x_{ij} \qquad \forall k = 1,\dots,m,\ (i,j) \in E \tag{12c}$$

$$\sum_{k=1}^{m} \sum_{\varphi_k \in P_k} a_{\varphi_k} \cdot w_{\varphi_k} \leq D \tag{12d}$$

$$\sum_{k=1}^{m} \sum_{\varphi_k \in P_k} w_{\varphi_k} \cdot \delta_{ij}^{\varphi_k} \cdot \upsilon_k \leq u_{ij} \qquad \forall (i,j) \in E \tag{12e}$$

$$w_{\varphi_k} \in \{0,1\} \qquad \forall \varphi_k \in P_k \tag{12f}$$

The objective function (12a) minimizes the sum over all link and protocol costs. Inequalities (12b) ensure, that at least one path is selected for each transport. Inequalities (12c) enforce the selection of the edges used by the active paths. The global delay constraint is stated by inequality (12d), and the compliance with the capacity constraints is guaranteed by inequalities (12e).

We aim at solving the linear programming (LP) relaxation of the IMP, which is called *(linear) master problem* (MP). To do so, we start with a small, restricted set of considered paths $\tilde{P}_k \subset P_k$ for each transport. The corresponding LP is called *restricted master problem* (RMP). Then, we iteratively try to find new variables corresponding to paths whose inclusion in the RMP can further improve the objective value *(pricing problem)*.

Let μ_k be the dual variables for inequalities (12b), $\pi_{e,k}$ the dual variables for inequalities (12c); let further η be the dual variable for inequality (12d) and ρ_e the dual variables for inequalities (12e). Based on our MP and these dual variables, the reduced costs for a path $\varphi_k \in P_k$ are given by

$$\bar{c}_{\varphi_k} = c'_{\varphi_k} - \mu_k + \sum_{e \in \varphi_k} \pi_{e,k} + a_{\varphi_k} \cdot \eta + \sum_{e \in \varphi_k} \rho_e \upsilon_k \; . \tag{13}$$

A solution to the pricing problem is a path with negative reduced costs. By adding the corresponding variable to the model, the current solution of the RMP can be further improved. Such a path φ_k having $\bar{c}_{\varphi_k} < 0$ is thus a shortest path on a graph with arc costs

$$\hat{c}^k_{ij} = p^k_{ij} + \pi_{e,k} + \eta \cdot (a_{ij} + d_{ij}) + \rho_e \cdot \upsilon_k \; . \tag{14}$$

Note that this pricing problem corresponds to the subproblem in LD, and we therefore solve it again by the algorithm from [3]. In order to create the initial set of variables of the RPM, we use a heuristic that tries to find a feasible path for each transport, not considering mutual dependencies originating from the capacity constraints. Then, in an iterative process, these paths are modified in order to become compliant with the capacity constraints, if necessary. Again, the same heuristic as in LD is finally applied to derive a heuristic solution to IMP.

6 Computational Results

In order to evaluate our algorithms we created artificial test instances with 25 to 1000 nodes and 100 to 1000 transports. We further considered various network densities and ratios $\bar{c}_{ij}/\bar{p}_{ij}$, where $\bar{c}_{ij}$ and $\bar{p}_{ij}$ denote average link and protocol costs, respectively. Instances with $|V| \leq 30$, $|T| \leq 200$, $|A| \leq 10 \cdot |V|$ and $\bar{c}_{ij}/\bar{p}_{ij} \approx 1$ can typically be solved to proven optimality within a couple of minutes by directly solving the MCF model. For a subset of these instances we show the average optimality gaps of the upper- and lower bounds obtained by LD and CG in Table 1. Each group of instances (first column) contains nine instances with different network densities. Optimal solution values have been obtained by directly solving the ILP-model (3). For this purpose, as well as for the solution of the RMP of the CG, we used ILOG CPLEX in version 11.0. Runtimes of LD and CG have been limited to 1000 seconds and all tests have been performed on a Dual Core AMD Opteron 2214 machine with 4 GB RAM.

For larger instances feasible solutions can usually be obtained within a couple of seconds (mainly by the primal heuristic), but finding optimal solutions is no longer possible. By the LD and CG approaches high quality solutions can be obtained in particular for instances with $\bar{c}_{ij}/\bar{p}_{ij} \approx 1$. Table 2 shows the results for some larger instances. The first two columns list average gaps between upper and lower bounds of LD and CG, respectively, while the third column shows gaps between the best bounds of both methods. One can see that LR is superior to CG in most cases, however, using the best lower and upper bounds from both methods the best average gaps, see the last column of Table 2.

7 Conclusions and Future Work

In this work we presented new approaches to solve a real-world multi-constraint network design problem including a multi-commodity flow model and heuristic methods based on Lagrangean decomposition and column generation. Our results indicate that these methods are capable of finding high quality solutions of reasonably sized input instances in relatively short time. In particular for the column generation approach, however, further improvements like applying stabilization techniques [6] seem to be promising and will be part of future work.

Table 1: Average %-gaps (and standard deviations) of LD and CG w.r.t. optimal solution values.

	Lagrangean decomp.					column gen.				
inst. ($\|V\|$, $\|T\|$)	LB [%]		UB [%]		$t_{avg}[s]$	LB[%]		UB[%]		$t_{avg}[s]$
25,100	-0.95	(0.33)	5.17	(2.48)	87	-11.91	(33.03)	5.91	(8.57)	11
25,200	-0.64	(0.44)	4.18	(1.35)	334	-11.57	(33.16)	3.51	(2.85)	28
30,100	-1.30	(0.50)	6.05	(3.18)	225	-1.28	(0.50)	3.43	(1.57)	14
30,200	-0.74	(0.25)	4.98	(2.69)	528	-0.72	(0.24)	4.43	(3.22)	36

Table 2: Average gaps (and standard deviations) of LD and CG for larger problem instances between lower and upper bounds.

inst. ($\|V\|$, $\|T\|$)	LD-gap [%]		$t_{avg}[s]$	CG-gap [%]		$t_{avg}[s]$	total gap [%]	
50,100	9.47	(2.63)	361	6.80	(2.30)	84	6.80	(2.30)
50,200	12.08	(0.97)	581	27.93	(40.88)	515	8.30	(2.38)
100,100	11.22	(3.71)	665	17.14	(31.16)	487	6.68	(2.30)
100,200	16.31	(3.22)	926	90.26	(29.21)	946	15.66	(2.07)
500,100	14.23	(4.11)	1000	100.00	(0.00)	1000	14.23	(4.11)
500,200	21.95	(2.75)	1000	100.00	(0.00)	1000	21.83	(2.81)

References

[1] Ravindra K. Ahuja, Thomas L. Magnanti, James B. Orlin, and M.R. Reddy. Applications of network optimization. In M. Ball et al., editor, *Handbooks in Operations Research and Management Science, Volume 7: Network Models*, chapter 1, pages 1–84. Elsevier, 1995.

[2] Francisco Barahona and Ranga Anbil. The volume algorithm: producing primal solutions with a subgradient method. *Mathematical Programming*, 87(3):385–399, 2000.

[3] Irina Dumitrescu and Natashia Boland. Improved preprocessing, labeling and scaling algorithms for the Weight-Constrained Shortest Path Problem. In *Networks*, volume 42, pages 135–153, 2003.

[4] Bernard Gendron, Teodor Crainic, and Antonio Frangioni. Multicommodity capacitated network design. In *Telecommunications Network Planning*, pages 1–19. B. Sans'o and P. Soriano (eds.), 1998.

[5] Jiri Kubalik and Richard Mordinyi. Optimizing events traffic in event-based systems by means of evolutionary algorithms. In *ARES '07: Proceedings of the The Second International Conference on Availability, Reliability and Security*, pages 1101–1107. IEEE Press, 2007.

[6] Marco E. Lübbecke and Jacques Desrosiers. Selected topics in column generation. *Operations Research*, 53(6):1007–1023, 2005.

[7] Nina Musil. Design eines sicherheits-, zeit- und kostenkritischen Kommunikationsnetzwerkes mittels Lagrange Relaxierung und Spaltengenerierung. Master's thesis, Vienna University of Technology, Institute of Computer Graphics and Algorithms, Vienna, Austria, December 2008.