Lagrangian Cardinality Cuts and Variable Fixing for Capacitated Network Design [⋆]

Meinolf Sellmann, Georg Kliewer, Achim Koberstein

University of Paderborn
Department of Mathematics and Computer Science
Fürstenallee 11, D-33102 Paderborn
{sello,geokl,akober}@upb.de

Abstract. We present a branch-and-bound approach for the Capacitated Network Design Problem. We focus on tightening strategies such as variable fixing and local cuts that can be applied in every search node. Different variable fixing algorithms based on Lagrangian relaxations are evaluated solitarily and in combined versions. Moreover, we develop cardinality cuts for the problem and evaluate their usefulness empirically by numerous tests.

1 Introduction

When solving discrete optimization problems to optimality, really two tasks have to be considered. First, an optimal solution must be constructed, and second, the algorithm must prove its optimality. Optimal or at least near optimal solutions can often be found quickly by heuristics or by approximation algorithms, both specially tailored for the given problem. In contrast to the construction of a high quality solution, the algorithmic optimality proof requires the investigation of the entire search space, which in general is much harder than to partly explore the most promising regions only. By eliminating parts of the search space that do not contain improving solutions, tightening strategies can help with respect to both aspects of discrete optimization.

In this paper, we focus on local tightening strategies that can be applied in every search node of a branch-and-bound tree and that may only be valid in the current subtree. We review bound computation algorithms based on Lagrangian relaxation that have been proposed for the CNDP and evaluate their performance in practice.

It is important to note that the algorithms used for bound computations within a branch-and-bound algorithm should not only be measured in terms of quality and computation time. In many successful approaches, they are also used for the selection of the branching cut that should favorably be introduced in the next branching step, and sometimes they can also be used to tighten the problem formulation within a search node by variable fixing. Or, more generally, by generating local cuts that may only be valid for the subtree rooted by the current node. We embed our algorithms for the computation

⋆ This work was partly supported by the German Science Foundation (DFG) project SFB-376, the project "Optimierung in Netzwerken" under grant MO 285/15-1, and by the UP-TV project, partially funded by the IST program of the Commission of the European Union as project number 1999-20 751, and by the IST Programme of the EU under contract number IST-1999-14186 (ALCOM-FT).

of the linear continuous relaxation bound of the CNDP in a branch-and-bound framework. We investigate independent variable fixing algorithms and a coupling technique for variable fixing algorithms based on Lagrangian relaxations to tighten the problem formulation within a search node. Additionally, we derive local Lagrangian Cardinality Cuts and evaluate their usefulness in practice.

The paper is structured as follows: In Section 2, we introduce the Capacitated Network Design Problem (CNDP). To solve the problem, we use bounds, variable fixing algorithms and local cardinality cuts based on Lagrangian relaxation as described in Section 3. The entire branch-and-bound-approach is described in Section 4. Finally, in Section 5, we give numerical results. Generally, because of space restrictions we omit all proofs. A full version of the paper can be found in [15].

2 The Capacitated Network Design Problem

The Capacitated Network Design Problem consists of finding an optimal subset of edges in a network $G = (V, E)$ such that we can transport a given demand of goods (so called *commodities*) at optimal total cost. The latter consists of two components: the flow costs and the design costs. The flow cost is the sum of costs for the routing of each commodity, whereby for each arc (i, j) and commodity k a scalar c^k_{ij} determines the cost of routing one unit of commodity k via (i, j). The design costs are determined by the costs of installing the chosen arcs, whereby for each arc (i, j) we are given a fixed edge installation cost f_{ij}. Additionally, there is a capacity u_{ij} on each arc that limits the total amount of flow that can be routed via (i, j).

For all edges $(i, j) \in E$ and commodities $1 \leq l \leq K$, let $b^l_{ij} = \min\{|d^l|, u_{ij}\}$. Using variables $x \in \mathcal{R}^{|E|}_+$ for the flows and $y \in \{0, 1\}^{|E|}$ for the design decisions, the mixed-integer linear optimization problem for the capacitated network design is defined as follows:

$$
\begin{array}{llll}
\text{Minimize} & L_{CNDP} = \sum_l (c^l)^T x^l + f^T y & & \\
\text{subject to} & N x^l = d^l & & (1) \\
& \sum_l x^l_{ij} \leq u_{ij} y_{ij} & \forall\, (i, j) \in E & (2) \\
& x^l_{ij} \leq b^l_{ij} y_{ij} & \forall\, (i, j) \in E,\ 1 \leq l \leq K & (3) \\
& x \geq 0 & & (4) \\
& y \in \{0, 1\}^{|E|} & & (5)
\end{array}
$$

For ease of notation, we refer to the above LP with L_{CNDP}, which is also used to denote the optimal objective value. The network flow constraints (also called *mass balance constraints*) (1) are defined by the node-arc-incidence matrix $N = (n_{ia})_{i \in V, a \in E}$ and a demand vector $d^k \in \mathcal{R}^{|\mathcal{V}|}$ for all commodities k, whereby $n_{ia} = 1$ iff $a = (h, i)$, $n_{ia} = -1$ iff $a = (i, h)$, and $n_{ia} = 0$ otherwise, and $d^k_i > 0$ iff node $i \in V$ is a demand node and $d^k_i < 0$ iff node i is a supply node for commodity k. Without loss of generality, we may assume that there is exactly one demand node and one supply node for each commodity [11].

The total flow on an arc (i, j) is constrained by the capacity u_{ij} (so called *capacity* or *bundle constraints* (2)). The set of *upper bound constraints* (3) is redundant to the problem formulation and provides a tighter LP relaxation of the MIP.

2.1 State of the Art

In several research papers, Crainic, Frangioni, and Gendron develop lower bounding procedures for the CNDP [4]. The main insights are the following: Tight approximations of the so called *strong* LP-relaxation (see L_{CNDP} including the redundant constraints (3)) can be found much faster by Lagrangian relaxation than by optimizing the LP using standard LP-solvers. The authors investigate so called *shortest path* and *knapsack relaxations* (see Section 3). When solving the Lagrangian dual, bundle methods converge faster than ordinary subgradient methods and are more robust. Motivated by this successful work, we evaluate several Lagrangian relaxations in the context of branch-and-bound.

In [11], Holmberg and Yuan present a method to compute exact or heuristic solutions for the CNDP. They use the Lagrangian *knapsack relaxation* in each node of the branch-and-bound tree to efficiently compute lower bounds. Special penalty tests were developed which correspond to variable fixing strategies presented in the paper at hand. An evaluation of the following components is given: subgradient search procedure for solving the Lagrangian dual, primal heuristic for finding feasible solutions, interplay between branch-and-bound and the subgradient search. On top of that work, a heuristic is developed that is embedded in the tree search procedure. That heuristic is able to provide near-optimal solutions on CNDP instances which are far beyond the range of exact methods like Lagrangian relaxation based branch-and-bound or branch-and-cut (represented e.g. by the Cplex implementation).

In [2], Bienstock et al. describe two cutting-plane algorithms for a variant of the CNDP with multi-edges (i.e., an edge can be inserted multiple times). One of them is based on the multicommodity formulation of CNDP and uses cutset and three-partition inequalities. The other one adds the following cutting planes: total capacity, partition and rounded metric inequalities. In a branch-and-cut framework, both variants provide sound results on a benchmark of realistic data. A substantial improvement to this procedure is proposed by Bienstock in [3]. The branch-and-cut algorithm based on ϵ-approximations of linear programs performs better on the same benchmark data.

3 Lagrangian Relaxation Bounds

The CNDP can be viewed as a mixture of a continuous and a discrete optimization problem. The latter is obviously constituted by the design variables, whereas the first is a min cost multi-commodity flow problem (MMCF) that evolves when the design variables are fixed. For the MMCF, besides linear programming solvers, especially cost decomposition approaches based on Lagrangian relaxation have been applied successfully [6]. The bounds we will use for the CNDP will be based on those cost decomposition approaches for the MMCF.

Used for more than 30 years now, Lagrangian relaxation can well be referred to as a standard technique for the bound computation of combinatorial optimization problems. The pioneering work was done by Held and Karp [9, 10] who introduced the new idea when tackling the traveling salesman problem. By omitting some hard constraints and incorporating them in the objective function via a penalty term, upper bounds on the performance (that is, for the CNDP, lower bounds on the costs) can be computed.

Regarding the MMCF and also for the CNDP, we are left with two promising choices of which hard constraints should be softened:

– the bundle constraints ("shortest path relaxation"), or
– the mass balance constraints ("knapsack relaxation").

In the following, we discuss the knapsack relaxation in more detail. For an in depth presentation of the shortest path relaxation, we refer to [15].

3.1 Knapsack Relaxation

For the mass balance constraints to be relaxed, we introduce Lagrangian multipliers μ_i^l for all $1 \leq l \leq K$ and $i \in V$. We get the following linear program:

$$\begin{array}{lll} \text{Minimize} & L_{KP}(\mu) = \sum_l \sum_{ij} (c_{ij}^l + \mu_i^l - \mu_j^l)^T x_{ij}^l + f^T y + \mu^T d & \\ \text{subject to} & \sum_l x_{ij}^l \leq u_{ij} y_{ij} & \forall\, (i,j) \in E \\ & x_{ij}^l \leq b_{ij}^l y_{ij} & \forall\, (i,j) \in E \\ & x \geq 0 & \\ & y \in \{0,1\}^{|E|} & \end{array}$$

Whereas the shortest path relaxation decomposes the Lagrangian subproblem by the different commodities, here we achieve an edge-wise decomposition. To solve the above LP, for each $(i,j) \in E$ we consider the following linear program, that is similar to the linear continuous relaxation of a knapsack problem:

$$\begin{array}{lll} \text{Minimize} & L_{KP}^{(i,j)}(\mu) = \sum_l \overline{c}_{ij}^l \overline{x}_{ij}^l & \\ \text{subject to} & \sum_l \overline{x}_{ij}^l \leq u_{ij} & \\ & \overline{x}_{ij}^l \leq b_{ij}^l & \forall\, 1 \leq l \leq K \\ & \overline{x} \geq 0 & \end{array}$$

where $\overline{c}_{ij}^l = c_{ij}^l + \mu_i^l - \mu_j^l$. For each $(i,j) \in E$, we set $x_{ij}^l = \overline{x}_{ij}^l$ for all $1 \leq l \leq K$, and $y_{ij} = 1$, iff $f_{ij} + L_{KP}^{(i,j)}(\mu) < 0$. Otherwise, we set $x_{ij}^l = 0$ for all $1 \leq l \leq K$, and $y_{ij} = 0$. Obviously, this setting provides us with an optimal solution for $L_{KP}^{(i,j)}(\mu)$. Thus, the main effort is to solve the problems $L_{KP}^{(i,j)}(\mu)$. But this is an easy task (compare with [12, 13]): first, we can eliminate all variables with positive cost coefficients, i.e., we set $\overline{x}_{ij}^l = 0$ for all $1 \leq l \leq K$ with $\overline{c}_{ij}^l \geq 0$. Next, we sort the $\overline{x}_{ij}^l$ according to increasing cost coefficients $\overline{c}_{ij}^l$, that is, from now on we may assume that $\overline{c}_{ij}^l < \overline{c}_{ij}^{l+1} < 0$ for all $1 \leq l < s \leq K$, where s is the number of negative objective coefficients. Let $k \in \mathbb{N}$ denote the *critical item* with $k = \min\{l \leq s \mid \sum_{h \leq l} b_{ij}^h > u_{ij}\} \cup \{s+1\}$. We obtain $L_{KP}^{(i,j)}(\mu)$ by setting $\overline{x}_{ij}^h = b_{ij}^h$ for all $h < k$, $\overline{x}_{ij}^h = 0$ for all $h > \min\{k, s\}$, and, in case of $k < s+1$, $\overline{x}_{ij}^k = u_{ij} - \sum_{h<k} b_{ij}^h$. Thus, the knapsack subproblem can be solved in time $O(|E|(K \log K))$.

Note, that both relaxations have the integrality property. Thus, the bound we achieve in both settings equals the linear continuous relaxation bound of the CNDP [4].

3.2 Lagrangian Cardinality Cuts

In the presence of a near optimal solution to the CNDP with associated objective value B, in each Lagrangian subproblem we can infer restrictions on the number of edges that need to be installed in any improving solution.

Before we state the idea idea more formally, to ease the notation we introduce some identifiers. For the knapsack relaxation we set $\hat{f}_{ij} = f_{ij} + L_{KP}^{(i,j)}(\mu)$ for all $(i,j) \in E$ and $\hat{B} = B + \mu^T d$ for Lagrangian multipliers μ. Further, denote with L_R the current Lagrangian subproblem $L_{KP}(\mu)$.

Theorem 1. *Denote with $e_1, \ldots, e_{|E|}$ an ordering of the edges in E such that $i < j$ implies $\hat{f}_{e_i} \leq \hat{f}_{e_j}$. It holds, if $L_R < B$, then*

a) there exist values

$$F = \text{argmin}_{0 \leq u \leq |E|} \{ \sum_{h \leq u} \hat{f}_{e_h} < \hat{B} \} \text{ and} \tag{1}$$

$$U = \text{argmax}_{0 \leq u \leq |E|} \{ \sum_{h \leq u} \hat{f}_{e_h} < \hat{B} \}. \tag{2}$$

b) And, in any improving solution (x, y), it holds that $F \leq \sum_{(i,j) \in E} y_{ij} \leq U$.

Theorem 1 allows us to add cardinality cuts on the number of edges to be used without loosing improving solutions. We will evaluate the effect of local cardinality cuts on the solution process in Section 5.

3.3 Variable Fixing

A big advantage of Lagrangian relaxation based bound computations is that they can be used for variable fixing in a very efficient way. In the presence of an optimal or at least high quality upper bound $B \in \mathbb{R}$ for the CNDP, it is an easy task to check whether a variable y_{ij} can still be set to either of its bounds without worsening the lower bound too much. More formally, given the Lagrangian multipliers μ in the current knapsack subproblem, a value $l \in \{0, 1\}$ and any edge $(i,j) \in E$, we can set

$$y_{ij} = l \quad \text{if } (2l - 1)(f_{ij} + L_{KP}^{(i,j)}(\mu)) > B - L_{KP}(\mu) \tag{3}$$

A similar statement holds for the shortest path relaxation. Using these implications, we can derive two variable fixing algorithms for the two different Lagrangian subproblems. Of course, we could just choose one of the two alternatives (for example the one for which the Lagrangian dual can be solved more quickly) and apply the corresponding variable fixing algorithm. But when using a coupling method for variable fixing algorithms that was published in [14], we can do even more: with the help of dual values gained in the solution process of the Lagrangian subproblem, in every Lagrange iteration we can apply both variable fixing algorithms.

When using the knapsack relaxation, the idea of the coupling method consists in using dual values as Lagrangian multipliers for the shortest path subproblem next. Thus, we need to provide dual values for the bundle as well as the upper bound constraints. Given the current Lagrangian multipliers μ, we solve $|E|$ knapsack subproblems as described in Section 3.1. Again, when given any edge $(i,j) \in E$, we assume that $s \in \mathbb{N}$ denotes the number of negative cost coefficients in $L_{KP}^{(i,j)}(\mu)$, that the remaining variables $\overline{x}_{ij}^l$ are ordered with respect to increasing cost coefficients, and that $k \leq s+1$ is the critical item.

In case of $k < s+1$, we set $\overline{\lambda}_{ij} = \overline{c}_{ij}^k$, $\overline{\nu}_{ij}^l = \overline{c}_{ij}^l - \overline{c}_{ij}^k$ for all $l < k$ and $\overline{\nu}_{ij}^l = 0$ for all $l \geq k$. And for $k = s+1$, we set $\overline{\lambda}_{ij} = 0$, $\overline{\nu}_{ij}^l = \overline{c}_{ij}^l$ for all $l < k$ and $\overline{\nu}_{ij}^l = 0$ for all $l \geq k$.

Theorem 2. *The vectors $\overline{\lambda}$ and $\overline{\nu}$ define optimal dual values for $L_{KP}(\mu)$.*

Now, if we choose to use the knapsack relaxation, in every Lagrangian subproblem we solve $|E|$ linear continuous knapsack problems and achieve a lower bound for the CNDP. If that bound is worse than B, we can prune the current choice point. Otherwise, we fix variables according to Equation 3. Next, we set up the Lagrangian shortest path subproblems that evolves when using the optimal dual values of the knapsack subproblem in Theorem 2 as Lagrangian multipliers. Then, we fix variables with respect to this substructure.

As our experiments show, it is favorable to use the knapsack relaxation to solve the Lagrangian dual quickly. As one would expect, solving K shortest path problems in every Lagrangian subproblem in addition to the $|E|$ knapsack subproblems is rather costly and slows down the solution process considerably. The following Theorem helps to cope with this situation more efficiently.

Theorem 3. *Given Lagrangian multipliers μ in the knapsack relaxation, denote with $\overline{\lambda} \leq 0$ and $\overline{\nu} \leq 0$ optimal dual values for $L_{KP}(\mu)$. Then,*

$$L_{SP}(\overline{\lambda}, \overline{\nu}) \geq L_{KP}(\mu), \tag{4}$$

where $L_{SP}(\overline{\lambda}, \overline{\nu})$ denotes the value of the next shortest path subproblem.

This Theorem allows to fix variables with respect to the shortest path relaxation without having to solve the corresponding shortest path subproblems, which improves on the running time of our variable fixing algorithm, but makes it also less effective. Unfortunately, as we shall see in Section 5, even in its strong version the shortest path variable fixing algorithm is already almost ineffective, and therefore this idea cannot be used to improve on the running time of our CNDP solver.

4 A branch-and-bound Algorithm

After having described the bound computation and possible tightening strategies based on Lagrangian relaxation, now we sketch the decisions taken in the tree search.
Dominance Cut-Off Rule Apart from the lower bound exceeding the upper bound, the search in the current node can be pruned if the min-cost routing of all commodities only uses edges that have already been decided to be installed. Thus, in every choice point we use a column generation approach to solve the min-cost multicommodity flow

problem on the subset of edges with associated y_{ij} that have a current upper bound of 1. And if that routing only uses edges (i, j) with y_{ij} with lower bound 1, we can prune the search and backtrack.

Branching Variable Selection The previous discussion also induces a rule for the selection of the branching variable: it is clearly favorable to choose a variable for branching that is being used by the current optimal min-cost multicommodity flow. Of course, there may be more than just one such variable. Then, we can choose the one with minimal or maximal reduced costs $|\hat{f}_{ij}|$ in the Lagrangian subproblem with the best associated multipliers. The different choices will be evaluated in Section 5.

Tree Traversal A simple depth first search procedure is used to choose the next search node. This allows to find feasible solutions quickly and eases the reuse of Lagrangian multipliers.

Primal Heuristic To find reasonably good and near optimal solutions quickly, in every search node we apply a Lagrangian heuristic that was suggested by Holmberg and Yuan. It works by computing multicommodity flow solutions on a subset of the edges and de-assigning all arcs that carry no flow. For further details, we refer to [11].

Variable Fixing Heuristic Because the Network Design Problem is very hard to be solved exactly, we may decide to search for relatively good solutions quickly. The exact approach can be transformed in a heuristic for the problem by fixing variables more optimistically. Holmberg and Yuan [11] developed the so called α-heuristic for this purpose: While solving the Lagrangian dual, we protocol how often a variable is set to one or to zero. And if one of the values is dominant with respect to a given parameter, the variable is simply set to this value.

5 Numerical Results

We report on our computational experience with the algorithms developed in this paper. The section is structured as follows: first, we introduce the benchmark data used in the experiments. Then, we define the possible parameter settings that activate and deactivate different algorithmic components. And finally, we compare the variants when solving the CNDP from scratch, in the optimality proof, and when using the approach as a heuristic.

All tests were carried out on systems with AMD Athlon, 600MHz processors, and 256 MByte main memory. The code was compiled with the GNU g++ 2.95 compiler using optimization level O3.

5.1 Benchmark Data

Surprisingly, in spite of the theoretical interest the CNDP has drawn and the large number of research groups that have dealt with the problem, apparently there has been no benchmark set established on which researchers can compare algorithms that solve the CNDP exactly. Much work has been done with respect to the computation of lower bounds and the heuristic solution of the problem. Benchmarks used for this purpose (to be found in [4, 11], for example) are still too large to allow the computation of optimal solutions. For variations of the problem (such as the multi-edge CNDP, Network Loading, etc.) benchmark data exists, but it is not straight forward to see how it could be converted into meaningful instances for the pure CNDP as we consider it here.

Thus, we decided to base our comparison on a benchmark of 48 instances generated by a CNDP generator developed by Crainic et al. and described in [4]. It appears as a generator that is used by different research groups, and it was enhanced with a stable random number generator by A. Frangioni. We generated graphs with 12, 18, and 24 nodes with 50 to 440 arcs and 50 to 160 commodities. For the heuristic comparison we use the benchmark set from [4, 5]. The exact details about the benchmark sets we use is given in [15]. There, the exact data regarding our experiments can be found as well, that is left out in the paper because of space restrictions.

5.2 Algorithm Variants Considered in the Experiments

The optimization system developed consists of several parts. The ones compared and evaluated in the experiments are: different Lagrangian relaxation algorithms based on the shortest path (SP) or the knapsack relaxation (KP), respectively; a branch-and-bound algorithm using bounds based on those relaxations, where the branching variable is chosen according to minimal (BR0) or maximal (BR1) absolute reduced cost values $\hat{f}_{ij}$; two different variable fixing algorithms based on the shortest path relaxation (SF) and the knapsack relaxation (KF); and finally, the cardinality interval tightening algorithm that adds Lagrangian cardinality cuts to the problem (CIT).

5.3 Evaluation

With the first experiments we performed we wanted to find out which type of Lagrangian relaxation was preferable. In accordance to the results reported in [11], we found that the knapsack relaxation is clearly superior both with respect to the number of subgradient iterations needed to solve the Lagrangian dual as well as the time needed to solve the Lagrangian subproblems. Because of the space restrictions, we omit a detailed comparison here, and start right away with an evaluation of the impact of Lagrangian cardinality cuts when solving the CNDP using the knapsack relaxation. Table 1 shows a comparison of lower bound routines using the Lagrangian knapsack relaxation with and without cardinality cuts. And Table 2 shows a comparison of two different strategies for the selection of the branching variable. Comparing two variants, the tables give the average percentage of the second variant when compared to the first (that is always set to 100%) with respect to running times and the number of search nodes visited in the branch-and-bound trees. Moreover, we specify minima, maxima, and the variance of those percentages.

	BR0 → BR0-CIT	BR1 → BR1-CIT
time	**93.7%**	**25.3%**
min	4.72%	0.28%
max	353%	131.5%
variance	62.1%	11.5%
nodes	**38.6%**	**10.1%**
min	0.73%	0.02%
max	120.7%	78.4%
variance	14.5%	2.9%

Table 1. Impact of cardinality interval tightening using knapsack relaxation with fixation based on knapsack relaxation for solving CNDP. Mean, minimum, maximum values and variance of running time and nodes in the branch-and-bound tree are given.

Clearly, choosing a branching variable with minimal reduced costs is favorable, no matter if cardinality cuts are introduced or not. This result contradicts the recommendation given in [11]. Actually, this result is not very surprising. Intuitively, the variable with the minimal absolute reduced costs is least likely to be set to either of its bounds by variable fixing. It is the variable we have the least knowledge about, and therefore it is a good choice to base a case distinction on it. In contrast, the variable with the largest absolute reduced costs is most likely to be set by variable fixing, and therefore it is no good idea to double the effort by using this variable for branching.

Regarding the introduction of Lagrangian cardinality cuts, Table 1 shows that they have a great impact on the number of search nodes that have to be investigated. Cardinality cuts are also favorable with respect to the total running time, but the gains are not as large as with respect to the size of the search tree. The trade off is caused by the additional effort that is necessary to sort the edges with respect to the current reduced costs $\hat{f}_{ij}$.

	BR0 → BR1	BR0-CIT → BR1-CIT
time	**1817.7%**	**235.1%**
min	68.03%	30.91%
max	7445.5%	1221.7%
variance	37944.6%	604.7%
nodes	**2750.4%**	**311.1%**
min	89.832%	15.636%
max	19415.3%	1427%
variance	172454%	1163.3%

Table 2. Impact of branching variable selection using knapsack relaxation with fixation based on knapsack relaxation for solving CNDP.

When looking at the data more precisely, we find that the primal heuristic works much better in the presence of cardinality cuts. The result of this positive effect is clear: high quality upper bounds are found much earlier in the search, pruning and variable fixing work much better, and the number of search nodes is greatly reduced, which explains the numbers in Table 1.

We conjecture that the primal heuristic works so well in the presence of cardinality cuts because they provide a good estimate on the number of edges that need to be installed in order to improve the current solution. Thus, the right amount of edges is opened for the heuristic, and it is able to compute near optimal solutions at a higher rate.

Next, we evaluate the use of the coupling method for variable fixing algorithms for the CNDP. Table 3 shows a comparison of runs when using shortest path variable fixing in addition to the knapsack variable fixing algorithm. The results are very disappointing: not only is the coupled approach inferior with respect to the total running time. On top of that, the reduction of choice points is negligible, and therefore the additional effort taken is almost worthless.

	SOLVE: KF → KF-SF	OPT: KF → KF-SF
time	**148.6%**	**144.1%**
min	96.59%	51.87%
max	466%	271.3%
variance	46.1%	13.5%
nodes	**133.8%**	**94.9%**
min	71.42%	20%
max	677.1%	180.3%
variance	166.3%	7.5%

Table 3. Impact of additional shortest path fixing using knapsack relaxation with fixation based on knapsack relaxation. Branching strategy BR0 and cardinality interval tightening are used.

Note, that the number of search nodes when using the coupling method sometimes even exceeds the value when using knapsack variable fixing only. This is caused by differences when building up the search tree: the Lagrangian dual usually stops with different Lagrangian multipliers that have a severe impact on the variable selection. Moreover, the generation of primal solutions differs, which makes the comparison particularly difficult, because variable fixing is highly sensitive to the quality of upper bounds. Thus, to eliminate the last perturbation, we repeated the experiment on the algorithmic optimality proof. That is, in the experiments we present in the following, we provide the algorithm with an optimal solution and let it prove its optimality only.

Table 3 shows the results, that still reveal the poor performance of the additional application of shortest path variable fixing. The reason for this is, that the shortest path variable fixing algorithm is much less effective than the one based on the knapsack relaxation. We tried to improve on the effectiveness of the algorithm by adding node-capacity constraints. If a node is a source for some commodities, its out-capacity must be large enough to push the corresponding supply into the network. Similarly, if a node is a sink node for some commodities, its in-capacity must be large enough to let the required demand in. In contrast to the knapsack relaxation, where the x- and y-variables are not independent, the shortest path relaxation allows to incorporate those constraints very easily. However, even this strengthening did not result in a filtering algorithm that was effective enough to be worth applying.

Next, in Table 4 we compare the performance of the algorithm we developed and the standard MIP-solver Cplex 7.5 when solving the CNDP and when proving the optimality of a given solution. Clearly, using LP-bounds improved by several kinds of cuts that Cplex adds to the problem results in a huge reduction of search nodes. However, Lagrangian relaxation allows to compute lower bounds much faster, so that the approach presented here is still competitive when solving the CNDP. And it achieves an improvement on the running time in the optimality proof.

Regarding the fact that we set up our system for the evaluation of variable fixing algorithms and local Lagrangian cardinality cuts, and taking into account that no sophisticated methods (like, e.g., Bundle methods) for the optimization of the Lagrangian dual are used, and, most importantly, that no global cuts are introduced yet to strengthen the lower bounds computed, we consider these results as very encouraging.

	OPT: CPLEX $\rightarrow$ KP-BR0-KF-CIT	SOLVE: CPLEX $\rightarrow$ KP-BR0-KF-CIT
time	**73.5%**	**229.2%**
min	9.63%	22.48%
max	259%	753.5%
variance	36.5%	356.5%
nodes	**1148.1%**	**3014.6%**
min	196.666%	100%
max	5250%	10279.5%
variance	10762.4%	73762.5%

Table 4. Comparison of CPLEX branch-and-cut algorithm against knapsack relaxation with fixation based on knapsack relaxation and cardinality interval tightening.

Finally, we compare the non-exact version of our approach (using the α-fixing heuristic) with other heuristic approaches that have been developed for the CNDP (see [5, 7, 8]). In Figure 1, we give the percentage of instances in a benchmark set (set C in [4, 5], containing 31 instances) that have been solved within a given solution quality (in percent, compared with the best known solution). Not only are the α-fixing with and without cardinality cuts clearly superior with respect to the achieved solution quality. On top of that, the heuristic variable fixing approach was stopped after at most 300 seconds cpu time. On this benchmark set, heuristic variable fixing is on average about 6 times faster than TABU-PATH and 23 times faster than PATH-RELINKING (using SPECint values to make different architectures comparable).

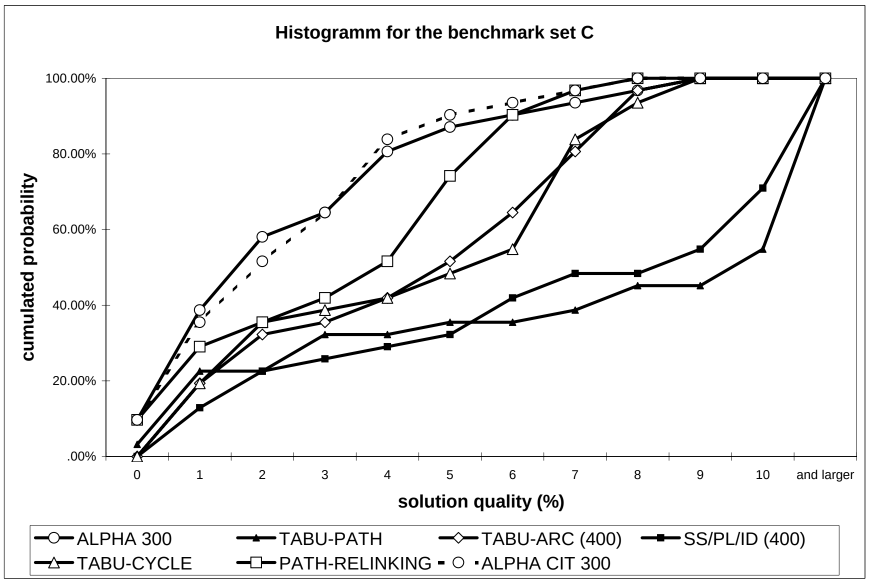

Fig. 1. Comparison of different heuristic solvers for the CNDP.

6 Conclusions and Future Work

We have presented an approach for the solution of the Capacitated Network Design Problem. It is based on a tree search where lower bounds based on Lagrangian relaxation are used for pruning. Two kinds of relaxation are considered, the shortest path and the knapsack relaxation. The latter is clearly favorable with respect to the convergence of the subgradient algorithm that optimizes the Lagrangian dual.

Two different variable fixing algorithms have been proposed in the literature that belong to the kind of relaxation that is chosen. When using the knapsack relaxation, we have shown how variables can also be fixed with respect to shortest path considerations by using dual values in the Lagrangian knapsack subproblem. However, even in combination with node-capacity constraints the shortest path variable fixing algorithm is too ineffective to justify the additional effort that is necessary for its application.

To tighten the problem formulation in a search node, we introduced the idea of local Lagrangian Cardinality Cuts. Experimental results show that their application improves on the overall running time, even though the time per search node increases considerably when they are applied.

Finally, we compared the heuristic variable fixing approach with other heuristic approaches developed for the CNDP. The results show, that the tree search approach we implemented clearly outperforms other heuristics both with respect to the cpu time needed and the quality achieved.

As a subject of further research, methods for the strengthening of the shortest path variable fixing that incorporate the routing costs may improve on the effectivity. Moreover, other tightening strategies such as global cuts can be incorporated to improve on the relaxation gap, which may improve on the overall performance of a Lagrangian relaxation based CNDP solver.

References

1. R.K. Ahuja, T.L. Magnati, J.B. Orlin. *Network Flows.* Prentice Hall, 1993.
2. D. Bienstock, O. Günlük, S. Chopra, C.Y. Tsai. Mininum cost capacity installation for multicommodity flows. *Mathematical Programming*, 81:177-199, 1998.
3. D. Bienstock. Experiments with a network design algorithm using epsilon-approximate linear programs. CORC Report 1999-4.
4. T.G. Crainic, A. Frangioni, B. Gendron. Bundle-based relaxation methods for multicommodity capacitated fixed charge network design. *Discrete Applied Mathematics*, 112: 73-99, 2001.
5. T.G. Crainic, M. Gendreau, and J.M. Farvolden. A simplex-based tabu search method for capacitated network design. *INFORMS Journal on Computing*, 12(3):223–236, 2000.
6. A. Frangioni. Dual Ascent Methods and Multicommodity Flow Problems. Ph.D. Dissertation TD 97-5, Dip. di Informatica, Univ. di Pisa, 1997.
7. I. Ghamlouche, T.G. Crainic, M. Gendreau. Cycle-based neighbourhoods for fixed-charge capacitated multicommodity network design. Technical report CRT-2001-01. Centre de recherche sur les transports, Université de Montréal.
8. I. Ghamlouche, T.G. Crainic, M. Gendreau. Path relinking, cycle-based neighbourhoods and capacitated multicommodity network design Technical report CRT-2002-01. Centre de recherche sur les transports, Université de Montréal.
9. M. Held and R.M. Karp. The travelling-salesman problem and minimum spanning trees. *Operations Research*, 18:1138–1162, 1970.
10. M. Held and R.M. Karp. The travelling-salesman problem and minimum spanning trees: Part II. *Mathematical Programming*, 1:6–25, 1971.
11. K. Holmberg and D. Yuan. A Lagrangean Heuristic Based Branch-and-Bound Approach for the Capacitated Network Design Problem. *Operations Research*, 48: 461-481, 2000.
12. S. Martello and P. Toth. An upper bound for the zero-one knapsack problem and a branch and bound algorithm. *European Journal of Operational Research*, 1:169–175, 1977.
13. S. Martello and P. Toth. Knapsack Problems – Algorithms and Computer Implementations. *Wiley Interscience*, 1990.
14. M. Sellmann and T. Fahle. Coupling Variable Fixing Algorithms for the Automatic Recording Problem. *9th Annual European Symposium on Algorithms (ESA 2001)*, Springer LNCS 2161: 134–145, 2001.
15. M. Sellmann, G. Kliewer, A. Koberstein. Lagrangian Cardinality Cuts and Variable Fixing for Capacitated Network Design. Technical report tr-ri-02-234. University of Paderborn.