Capacitated Network Design

Cardinality Cuts
and Coupled Variable Fixing Algorithms
based on Lagrangian Relaxations *

Meinolf Sellmann, Georg Kliewer, Achim Koberstein

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

Student Paper

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 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 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

* 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).

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.

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_{ij}^k 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 $b_{ij}^l = \min |d^k|, u_{ij}$. 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:

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

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 $d_i^k > 0$ iff node $i \in V$ is a demand node and $d_i^k < 0$ 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 CDNP [6]. 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 [3], 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 [4]. The branch-and-cut algorithm based on epsilon-approximations of linear programs performs better on the same benchmark data.

Further branch-and-cut algorithms for the multi-edge CNDP are investigated in research papers by Günlük, and Atamtürk et al. [8, 2]). Several valid inequalities are considered, and it is found that branch-and-cut is substantially superior to branch-and-bound code on the investigated benchmark data. After adding the cuts, the integrality gap at the root node was reduced significantly and the total number of evaluated nodes diminishes. These results emphasize the importance of tight lower bounds for the CNDP.

Sridhar and Park report about an implementation of a Benders-and-cut algorithm for the CNDP [17]. The algorithm consists of three parts: a cutting planes algorithm for finding tight lower bounds, a heuristic to generate feasible solutions, and the Benders-and-cut algorithm itself. The computational results provided in the paper are based on a wide range of instances with varying traffic demand. The complexity of the problem instances depends heavily on the capacity provided in the network. CNDP problems with low transport demand are easy to solve, whereas on problems with higher demand the authors suggest to use the Benders-and-cut algorithm. In cases when the traffic demand becomes extremely high, the integrality gaps increase. To strengthen the LP-relaxation, flow inequalities are very effective.

A branch-and-price algorithm for the path-based formulation of the CNDP is compared with the traditional branch-and-bound for the link-based formulation by Clarke and Gong in [5]. The path-based formulation is computationally more efficient. Adding SOS-branching on each origin and destination node increases this advantage. The efficient solution of the pricing problem (which is a simple shortest-path problem) enables a faster solving of the LP-relaxations in the branch-and-bound tree. Computational results on instances with 6, 10, and 15 nodes are reported.

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 [7]. 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 travelling 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”).

3.1 Shortest Path Relaxation

Assume, in L_{CNDP}, for each edge $(i, j) \in E$ we introduce a Lagrangian multiplier $\lambda_{ij} \leq 0$ and transfer the corresponding bundle constraint into the objective function. And at the same time, we also relax the upper bound constraints $x_{ij}^l \leq b_{ij}^l$ using penalty costs $\nu_{ij}^l \leq 0$. Let $\otimes : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n$

denote the componentwise product of two vectors, i.e. $z = x \otimes y$ iff $z_s = x_s y_s \ \forall \ x, y \in \mathbb{R}^n$ and $1 \leq s \leq n$. Then, we get the following linear program:

$$\begin{array}{ll} \text{Minimize} & L_{SP}(\lambda,\nu) = \sum_l (c^l - \lambda - \nu^l)^T x^l + (f + \lambda \otimes u + \sum_l \nu^l \otimes b^l)^T y \\ \text{subject to} & \quad N x^l = d^l \qquad\qquad \forall\ 1 \leq l \leq K \\ & \qquad x \geq 0 \\ & \qquad y \in \{0,1\}^{|E|} \end{array}$$

The theory of Lagrangian relaxation shows that, for every choice of the Lagrangian multipliers $\lambda, \nu \leq 0$, $L_{SP}(\lambda,\nu)$ is a lower bound on the CNDP. Notice that this value can be obtained easily, because there is no crosstalking between y and x and between the x^l's anymore. Thus, we can compute $L_{SP}(\lambda,\nu)$ by solving K problems of the form

$$\begin{array}{ll} \text{Minimize} & L^l_{SP}(\lambda,\nu) = (c^l - \lambda - \nu^l)^T x^l \\ \text{subject to} & \quad N x^l = d^l \\ & \qquad x^l \geq 0 \end{array}$$

and by setting $y_{ij} = 1$ iff $f_{ij} + \lambda_{ij} u_{ij} + \sum_k \nu^l_{ij} b^l_{ij} < 0$, and $y_{ij} = 0$ otherwise. That is, we can solve the Lagrangian subproblem mainly by computing K single source shortest path problems with positive edge weights. Therefore, the shortest path subproblem can be solved in time $O(K(m + n \log n))$.

3.2 Knapsack Relaxation

The other promising alternative is to relax the mass balance constraints in L_{CNDP}. For each constraint to be relaxed, we introduce Lagrangian multipliers μ^l_i for all $1 \leq l \leq K$ and $i \in V$. We get the following linear program:

$$\begin{array}{lll} \text{Minimize} & L_{KP}(\mu) = \sum_l \sum_{ij} (c^l_{ij} + \mu^l_i - \mu^l_j)^T x^l_{ij} + f^T y + \mu^T d & \\ \text{subject to} & \sum_l x^l_{ij} \leq u_{ij} y_{ij} & \forall\ (i,j) \in E \\ & \quad x^l_{ij} \leq b^l_{ij} y_{ij} & \forall\ (i,j) \in E \\ & \quad x \geq 0 & \\ & \quad y \in \{0,1\}^{|E|} & \end{array}$$

Whereas in the shortest path relaxation the Lagrangian subproblem was decomposed by 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:

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

where $\overline{c}^l_{ij} = c^l_{ij} + \mu^l_i - \mu^l_j$. For each $(i,j) \in E$, we set $x^l_{ij} = \overline{x}^l_{ij}$ for all $1 \leq l \leq K$, and $y_{ij} = 1$, iff $f_{ij} + L^{(i,j)}_{KP}(\mu) < 0$. Otherwise, we set $x^l_{ij} = 0$ for all $1 \leq l \leq K$, and $y_{ij} = 0$. Obviously, this setting provides us with an optimal solution for L. Thus, the main effort is to solve the problems $L^{(i,j)}_{KP}(\mu)$. But this is an easy task (compare with [13, 14]): first, we can eliminate all variables with positive cost coefficients, i.e., we set $\overline{x}^l_{ij} = 0$ for all $1 \leq l \leq K$ with $\overline{c}^l_{ij} \geq 0$. Next, we sort the $\overline{x}^l_{ij}$ according to increasing cost coefficients $\overline{c}^l_{ij}$, that is, from now on we may assume that $\overline{c}^l_{ij} < \overline{c}^{l+1}_{ij} < 0$ 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 $k = \min\{l \leq s \mid \sum_{h \leq l} b^h_{ij} > u_{ij}\} \cup \{s+1\}$. We obtain $L^{(i,j)}_{KP}(\mu)$ by setting $\overline{x}^h_{ij} = b^h_{ij}$ for all $h < k$, $\overline{x}^h_{ij} = 0$ for all $h > \min\{k, s\}$, and, in case of $k < s + 1$, $\overline{x}^k_{ij} = u_{ij} - \sum_{h<k} b^h_{ij}$. 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 [6].

3.3 Subgradient Optimization

Now, we give some details on the subgradient algorithm we use to solve the Lagrangian dual. For both relaxation types, it consists of maximizing the lower bound. That is, for the shortest path relaxation we have to maximize $L(\lambda,\nu)$ subject to $\lambda,\nu \leq 0$. And for the knapsack relaxation our task is simply to maximize $L(\mu)$. A popular way of solving these convex, piecewise linear maximization problems is to apply a subgradient search (see [1] for a general introduction). Several proposals regarding the specification of the general method have been made in the CNDP literature. Let s^t denote a subgradient in iteration t. We compute the new search direction by setting the search direction to $d^t = (s^t + \alpha d^{t-1})/(1+\alpha)$ [11].

3.4 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 can state the idea idea more formally, to ease the notation, we introduce some identifiers. For the shortest path relaxation, we define

$$\hat{f}_{ij} = f_{ij} + \lambda_{ij}u_{ij} + \sum_l \nu_{ij}^l b_{ij}^l \qquad \forall\,(i,j)\in E \tag{1}$$

$$\hat{B} = B - \sum_l L_{SP}^l(\lambda,\nu) \tag{2}$$

for Lagrangian multipliers $\lambda,\nu \leq 0$. And for the knapsack relaxation we set

$$\hat{f}_{ij} = f_{ij} + L_{KP}^{(i,j)}(\mu) \qquad \forall\,(i,j)\in E \tag{3}$$

$$\hat{B} = B + \mu^T d \tag{4}$$

for Lagrangian multipliers μ. Further, denote with L_R the current Lagrangian subproblem $L_{SP}(\lambda,\nu)$, or $L_{KP}(\mu)$, respectively.

Theorem 1. *Denote with $e_1,\dots,e_{|E|}$ 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

$$F = \text{argmin}_{0\leq u\leq |E|}\{\sum_{h\leq u} \hat{f}_{e_h} < \hat{B}\} \text{ and} \tag{5}$$

$$U = \text{argmax}_{0\leq u\leq |E|}\{\sum_{h\leq u} \hat{f}_{e_h} < \hat{B}\}. \tag{6}$$

b) And, in any improving solution (x,y), it holds that

$$F \leq \sum_{(i,j)\in E} y_{ij} \leq U. \tag{7}$$

Proof. a) It is sufficient to show that there exists an $1\leq l\leq |E|$ such that $\sum_{h\leq l}\hat{f}_{e_h} < \hat{B}$. Let $l = \text{argmin}_{0\leq h\leq |E|}\{\sum_{h\leq l}\hat{f}_{e_h}\}$. It holds, that

$$\sum_{h\leq l}\hat{f}_{e_h} = L_{SP}(\lambda,\nu) - \sum_{h\leq K} L_{SP}^h(\lambda,\nu) < B - \sum_{h\leq K} L_{SP}^h(\lambda,\nu) = \hat{B}. \tag{8}$$

Or, respectively,

$$\sum_{h\leq l}\hat{f}_{e_h} = L_{KP}(\mu) + \mu^T d < B + \mu^T d = \hat{B}. \tag{9}$$

b) Let $L^{>}[U] := L[\sum_{(i,j)\in E} y_{ij} > U]$ denote the optimization problem that evolves by adding the constraint $\sum_{(i,j)\in E} y_{ij} > U$ to a problem L. By the definition of U, we know that $L_R^{>}[U] > B$. Remind you, that we identify the name of an optimization problem with its optimal value. Now, assume that there exists a feasible solution (x, y) for L_{CNDP} with $|E| \geq \sum_{(i,j)\in E} y_{ij} > U$. Then, it holds

$$c^T x + f^T y \geq L_{CNDP}^{>}[U] \geq L_R^{>}[U] > B. \tag{10}$$

The case $\sum_{(i,j)\in E} y_{ij} < F$ follows analogously.

□

Theorem1 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.

3.5 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 λ and ν in the current shortest path subproblem, a value $l \in \{0, 1\}$ and any edge $(i, j) \in E$, we can set

$$y_{ij} = l \quad \text{if } (2l-1)(f_{ij} + \lambda_{ij} u_{ij} + \sum_l \nu_{ij}^l b_{ij}^l) > B - L_{SP}(\lambda, \nu) \tag{11}$$

Analogously, 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

$$y_{ij} = l \quad \text{if } (2l-1)(f_{ij} + L_{KP}^{(i,j)}(\mu)) > B - L_{KP}(\mu) \tag{12}$$

Now, we have two variable fixing algorithms at hand for 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 [16], 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.

Let us assume we decide to use the shortest path relaxation. Given the current Lagrangian multipliers λ and ν, in every iteration we need to solve K shortest path problems. Using the well known Dijkstra algorithm for that purpose, we do not only get the optimal objective of each problem, we also get the shortest path distances $\overline{\mu}_i^l$ of each node i for free. It is known for a long time that those distances are optimal dual values in the corresponding LP. The idea of the coupling method consists of using these duals as Lagrangian multipliers for the knapsack subproblem next. That is, we consider $L_{KP}(\overline{\mu})$. Like that, we can apply the variable fixing algorithms for both subproblems. Note, that for optimal Lagrangian multipliers λ and ν, the shortest path distances $\overline{\mu}$ in combination with the multipliers are also optimal for the dual of L_{CNDP}.

When using the knapsack relaxation, the situation is slightly more complicated, because 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 3.2. 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 $L_{KP}(\mu)$.*

Proof. It is sufficient to show that the value $\overline{\lambda}_{ij}$ and the vector $\overline{\nu}_{ij} = (\overline{\nu}^1_{ij}, \ldots, \ \overline{\nu}^K_{ij})$ give an optimal dual solution for $L^{(i,j)}_{KP}(\mu)$ for all $(i,j) \in E$.

The dual of $L^{(i,j)}_{KP}(\mu)$ is the following linear program:

$$\begin{array}{ll} \text{Maximize} & D^{(i,j)}_{KP}(\mu) = u_{ij}\lambda_{ij} + \sum_l b^l_{ij}\nu^l_{ij} \\ \text{subject to} & \lambda_{ij} + \nu^l_{ij} \leq \overline{c}^l_{ij} \qquad \forall\, 1 \leq l \leq K \\ & \qquad \lambda,\, \nu \leq 0 \end{array}$$

First, we show that our solution is dual feasible:

$\mathbf{k < s + 1:}$ It holds, that $\overline{\lambda}_{ij} = \overline{c}^k_{ij} \leq 0$, because $k \leq s$. Also, if $l < k$, then $\overline{\nu}^l_{ij} = \overline{c}^l_{ij} - \overline{c}^k_{ij} \leq 0$, because $\overline{c}^l_{ij} \leq \overline{c}^k_{ij}$. And if $l \geq k$, then $\overline{\nu}^l_{ij} = 0$. Thus, all values are non-positive.
Further, if $l < k$, it holds that $\overline{\lambda_{ij}} + \overline{\nu}^l_{ij} = \overline{c}^k_{ij} + \overline{c}^l_{ij} - \overline{c}^k_{ij} = \overline{c}^l_{ij}$. And for $l \geq k$, it holds $\overline{\lambda_{ij}} + \overline{\nu}^l_{ij} = \overline{c}^k_{ij} \leq \overline{c}^l_{ij}$.
$\mathbf{k = s + 1:}$ It holds, that $\overline{\lambda}_{ij} = 0$ and $\overline{\nu}^l_{ij} = \overline{c}^l_{ij} \leq 0$ for all $l < k$, because this implies $l \leq s$. Further, $\overline{\nu}^l_{ij} = 0$ for all $l \geq k$. Thus, all values are non-positive.
And, for $k = s + 1$, we have that $\overline{\lambda}_{ij} + \overline{\nu}^l_{ij} = \overline{c}^l_{ij}$ for all $1 \leq l \leq K$.

To prove the optimality of the dual solution (and, at the same time, also for the primal solution we gave in Section 3.2), we show that the two objective values for $\overline{x}_{ij}$ in $L^{(i,j)}_{KP}(\mu)$ and $\overline{\lambda}_{ij}, \overline{\nu}_{ij}$ in $D^{(i,j)}_{KP}(\mu)$ match each other:

$\mathbf{k < s + 1:}$ $\sum_l \overline{c}^l_{ij}\overline{x}^l_{ij} = \sum_{l<k} \overline{c}^l_{ij} b^l_{ij} + \overline{c}^k_{ij}(u_{ij} - \sum_{l<k} b^l_{ij}) = \sum_{l<k} b^l_{ij}(\overline{c}^l_{ij} - \overline{c}^k_{ij}) + u_{ij}\overline{c}^k_{ij} = \sum_l b^l_{ij}\nu^l_{ij} + u_{ij}\lambda_{ij}$.
$\mathbf{k = s + 1:}$ $\sum_l \overline{c}^l_{ij}\overline{x}^l_{ij} = \sum_{l \leq s} \overline{c}^l_{ij} b^l_{ij} = \sum_l b^l_{ij}\nu^l_{ij} + u_{ij}\lambda_{ij}$. □

To summarize: if we choose the shortest path relaxation, in every Lagrangian subproblem we solve K shortest path subproblems and achieve a lower bound on the CNDP. If that bound is worse than the best known or even optimal objective value B, we can prune the current choice point. Otherwise, we fix variables according to Equation 11. Then, we set up $L_{KP}(\overline{\mu})$, where $\overline{\mu}$ are the shortest path distances, and fix variables according to Equation 12.

If, instead, 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. Again, if that bound is worse than B, we can prune the current choice point. Otherwise, we fix variables according to Equation 12. Then, we set up $L_{SP}(\overline{\lambda}, \overline{\nu})$, where $\overline{\lambda}$ and $\overline{\nu}$ are the dual values of $L_{KP}(\mu)$ in Theorem 2, and fix variables according to Equation 11.

Of course, many variants of the algorithm sketched in the above can be thought of. For example, we may find that applying both variable fixing algorithms in every Lagrangian subproblem is too time consuming and does not pay off. Then, we can introduce frequency parameters that control the percentage of Lagrangian subproblems for which either of the two variable fixing algorithms is applied. As an extreme choice, we may for example decide to apply variable fixing for the optimal Lagrangian multipliers only. 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,*

$$L_{SP}(\overline{\lambda}, \overline{\nu}) \geq L_{KP}(\mu). \tag{13}$$

Proof. Denote with $\overline{\kappa} \leq 0$ optimal dual values for the upper bound constraints of y in $L_{KP}(\mu)$. The vectors μ and $\overline{\kappa}$ are dual feasible for $L_{SP}(\overline{\lambda}, \overline{\nu})$, because $\overline{\lambda}$, $\overline{\nu}$ and $\overline{\kappa}$ are primal optimal for $L_{KP}(\mu)$, and thus

$$\mu_j^k - \mu_i^k \leq c_{ij}^k - \lambda_{ij} - \nu_{ij}^k \qquad \forall\,(i,j) \in E,\ 1 \leq k \leq K, \quad \text{and} \tag{14}$$

$$-\overline{\kappa}_{ij} \leq f_{ij} - u_{ij}\lambda_{ij} - \sum_k b_{ij}^k \nu_{ij}^k \qquad \forall\,(i,j) \in E \tag{15}$$

Any dual feasible solution for a minimization problem determines a valid lower bound on that problem. Denote with D the dual of an optimization problem L. Then,

$$L_{SP}(\overline{\lambda}, \overline{\nu}) = D_{SP}(\overline{\lambda}, \overline{\nu}) \geq \mu^T d + 1^T \overline{\kappa} = D_{KP}(\mu) = L_{KP}(\mu). \tag{16}$$

$\square$

We can use this Theorem to relax Inequality 11, 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 too ineffective, and therefore this idea cannot be used to improve on the running time of a CNDP solver.

Another important choice is to decide when the effects of the variable fixing algorithms should be made visible to the subgradient algorithm. Obviously, when we fix variables during the process of finding optimal Lagrangian multipliers, we are actually changing the problem. Thus, with respect of the convergence of the iterative procedure, we have to reset. The most defensive strategy is to start from scratch (with the current Lagrangian multipliers as starting point) and to set the step length coefficient to its initial setting. However, we may consider to allow only a little more flexibility by scaling the factor by a fixed percentage. A totally different approach would be not to incorporate the changes in the variable bounds during the optimization process, but to restart the whole procedure (if any variable fixing has taken place) after optimal Lagrangian multipliers have been found. We will evaluate different parameter settings in the numerics.

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 using a column generation approach we 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 MMCF solutions on a subset of the edges and deassigning all arcs that carry no flow. For further details, we refer to [11].

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, and in the optimality proof.

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 their results. 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 [6, 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 [6]. 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. The exact details about the benchmark data is given in the Apendix. 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), that fix variables statically (STAT) after optimal Lagrangian multipliers have been computed, or dynamically (DYN) during the optimization of the Lagrangian dual; 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. 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. Table 2 shows a comparison of two different strategies for the selection of the branching variable. Comparing the two variants, the table gives the

	BR0 $\rightarrow$ BR0-CIT	BR1 $\rightarrow$ 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%

Fig. 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.

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.

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 results 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 somehow 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.

	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%

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

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. This effect is caused by the additional effort that is necessary to sort the edges with respect to the current reduced costs.

When looking at the runs 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.

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 have it prove its optimality only.

	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%

Fig. 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.

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 that the one based on the knapsack relaxation: Recall from Section 3.5, that the shortest path relaxation value consists of two values, one for the shortest paths routings, and the other one for the design variables. But the variable fixing algorithm only incorporates the latter costs and does not take into account that the removal of edges may generally increase the routing costs as well. Therefore, using shortest path variable fixing as described in Section 3.5 is ineffective and does not pay off.

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.

In Table 4, we compare the strategy to fix variables during the optimization of the Lagrangian dual with the fixing of variables after optimal multipliers have been found only. We see that a fixing "on the fly" is slightly favorable. Our experience also shows that the subgradient method is very robust with respect to variable fixing, and we can afford not to reset the step length after variables have been fixed in the Lagrangian subproblem without loosing convergence in practice.

	DYN $\rightarrow$ STAT
time	**115.9%**
min	36.95%
max	229.2%
variance	14.1%
nodes	**123.2%**
min	29.72%
max	233.8%
variance	20.8%

Fig. 4. Optimality proofs: comparison of two strategies for knapsack relaxation with fixation based on knapsack relaxation: on-the-fly fixing vs. fixing after Lagrange.

Finally, 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%

Fig. 5. Comparison of CPLEX branch-and-cut algorithm against knapsack relaxation with fixation based on knapsack relaxation and cardinality interval tightening.

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.

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. A. Atamtürk, D. Rajan. On Splittable and Unsplittable Capacitated Network Design Arc-Set Polyhedra. To appear in *Mathematical Programming*, 2001.
3. D. Bienstock, O. Günlük, S. Chopra, C.Y. Tsai. Mininum cost capacity installation for multicommodity flows. *Mathematical Programming*, 81:177-199, 1998.
4. D. Bienstock. Experiments with a network design algorithm using epsilon-approximate linear programs. CORC Report 1999-4.
5. L.W. Clarke, P. Gong. Capacitated Network Design with Column Generation. Research Report. Georgia Institute of Technology, Atlanta, USA, 1998.
6. 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.
7. A. Frangioni. Dual Ascent Methods and Multicommodity Flow Problems. Ph.D. Dissertation TD 97-5, Dip. di Informatica, Univ. di Pisa, 1997.
8. O. Günlük. A branch-and-cut algorithm for capacitated network design problems. *Mathematical Programming*, 86(1): 17-39, 1999.
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. T.L. Magnanti and R.T. Wong. Network design and transportation planning: Models and algorithms. *Transportation Science*, 18: 1-55, 1984.
13. 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.
14. S. Martello and P. Toth. Knapsack Problems – Algorithms and Computer Implementations. *Wiley Interscience*, 1990.
15. G.L. Nemhauser, L.A. Wolsey. *Integer and Combinatorial Optimization.* Wiley Interscience, 1988.
16. 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.
17. V. Sridhar and J.S. Park. Benders-and-cut algorithm for fixed-charge capacitated network design problem. *European Journal of Operational Research*, 125(3), 622-632, 2000.

7 Appendix

	nodes	arcs	comm	fixed cost ratio	capacity ratio
A1	12	50	50	0.02	1.5
A2	12	50	50	0.04	1.5
A3	12	50	50	0.02	3
A4	12	50	50	0.04	3
B1	12	100	50	0.03	3
B2	12	100	50	0.06	3
B3	12	100	50	0.03	6
B4	12	100	50	0.06	6
C1	12	50	100	0.06	3
C2	12	50	100	0.12	3
C3	12	50	100	0.06	6
C4	12	50	100	0.12	6
D1	12	100	100	0.01	1.5
D2	12	100	100	0.015	1.5
D3	12	100	100	0.01	3
D4	12	100	100	0.015	3
E1	18	120	60	0.024	4.3
E2	18	120	60	0.048	4.3
E3	18	120	60	0.024	8.6
E4	18	120	60	0.048	8.6
F1	18	240	60	0.016	6
F2	18	240	60	0.020	6
F3	18	240	60	0.016	8
F4	18	240	60	0.020	8
G1	18	120	120	0.012	2.15
G2	18	120	120	0.015	2.15
G3	18	120	120	0.012	4.3
G4	18	120	120	0.015	4.3
H1	18	240	120	0.012	1.5
H2	18	240	120	0.015	1.5
H3	18	240	120	0.012	3
H4	18	240	120	0.015	3
I1	24	220	80	0.008	2.65
I2	24	220	80	0.012	2.65
I3	24	220	80	0.008	5.3
I4	24	220	80	0.012	5.3
J1	24	440	80	0.008	5.3
J2	24	440	80	0.012	5.3
J3	24	440	80	0.008	10.6
J4	24	440	80	0.012	10.6
K1	24	220	160	0.004	2
K2	24	220	160	0.006	2
K3	24	220	160	0.004	4
K4	24	220	160	0.006	4
L1	24	440	160	0.002	2
L2	24	440	160	0.004	2
L3	24	440	160	0.002	4
L4	24	440	160	0.004	4

Table 1. PAD instances for the CNDP

	CPLEX		KP-BR0-KF-CIT		KP-BR1-KF-CIT		KF-BR0-KF-SF-CIT		KP-BR0-KF		KP-BR1-KF	
	time	nodes	time	nodes	time	nodes	time	nodes	time	nodes	time	nodes
A1	0,17	1	0.79	1	0.79	1	1.07	1	3.74	73	35.83	1225
A2	0.88	1	1.96	7	3.92	29	2.48	5	5.01	277	60.52	2637
A3	0.34	1	1.06	3	2.09	15	1.28	3	1.23	83	10.09	479
A4	1.46	2	3.66	25	3.26	23	4.36	25	10.16	273	61.42	1597
B1	16.97	4	4.37	15	5.67	25	5.57	15	6.04	233	153.02	7225
B2	54.78	16	32.03	171	32.14	183	40.46	171	49.89	1197	292.25	8551
B3	38.3	4	9.97	47	18.69	109	46.47	307	16.86	497	366.17	12811
B4	13.03	2	2.93	9	4.85	21	3.49	9	62.01	1229	311.86	7643
C1	18.72	5	10.67	25	7.83	19	12.93	25	24.49	183	350.12	3639
C2	249.71	115	1216.7	3725	1652.7	5411	1638.3	3747	710.06	3491	1255.9	6897
C3	34.65	23	51.5	155	123.04	419	66.38	155	36.01	525	211.73	2637
C4	268.09	136	942.58	2713	775.46	2429	1143.8	2627	610.53	3369	711.41	4963
D1	35.82	2	15.87	37	147.31	499	17.87	35	51.14	325	1745	13975
D2	114.34	12	186.6	459	239.7	619	634.99	1281	564.06	3875	8100.6	59105
D3	46.6	6	36.07	89	142.76	401	44.19	89	127.59	1191	1700.2	12777
D4	255.75	89	1193.2	2859	1549.6	4269	1493.1	2851	805.7	3679	3169.9	20881
E1	205.31	60	1031.4	3665	1221.9	5055	996.26	2777	292.14	3035	3661.9	33321
E2	556.73	166	4195.4	15387	1296.8	5471	5532	14967	4776.7	26827	3515.7	31313
E3	127.33	101	206.58	819	199.55	835	290.25	873	192.24	1363	7458.8	70765
E4	2213.17	491	6402.7	26351	13788	61661	8216.1	25009	14400	84956	14400	106406
F1	100.76	6	48.72	275	23.1	43	61.86	275	72.38	1607	4489.3	57957
F2	1183.03	233	4469.4	10425	2808.9	7221	4933.5	8623	2379.9	10415	5910	55981
F3	396.71	56	1515.6	3575	14400	36850	2112.8	4009	1119.2	7509	14400	70873
F4	212.68	9	61.61	333	45.84	105	76.06	333	146.58	1529	3321.4	50533
G1	217.76	16	689.26	1323	434.44	711	1402.9	1785	322.94	1143	6404.4	41527
G2	680.44	104	1041.7	1663	4057.7	7227	1337.3	1647	1273.6	6241	5259.2	32623
G3	213.1	12	336	543	1984	3443	420.06	533	1572.6	7161	5461.9	29205
G4	1321.32	223	8845.4	13971	4359.7	7417	11460	13855	9977.8	28149	6788	25287
H1	53.62	2	23.18	15	104.2	143	28.59	15	210.42	503	11000	39049
H2	697.95	18	14401	14200	1451.3	1607	14401	11352	1149.7	5883	14400	60310
H3	11.43	2	14.99	9	40.29	53	19.96	11	161.56	381	12029	49401
H4												
I1	6.67	1	11.17	11	11.58	11	11.23	9	81.12	267	4084.3	51839
I2	321.2	30	877.48	1411	705.97	1145	1647.63	1367	449.38	1469	14400	50915
I3	371.13	45	1115	1833	13622	26157	1873	1823	1591	7013	14400	62284
I4	647.12	44	2602.8	4523	6052	10739	3450.5	4493	2337.5	8231	14400	67019
J1	203.48	2	47.93	35	204.23	279	125.52	237	297.23	2861	14401	62841
J2	2385.94	107	7947.9	7783	6008.4	6965	10938	7719	2760.5	7523	14400	32695
J3	456.17	42	1747.3	2029	6274.5	8747	2530.4	2413	1279.3	2655	14400	43119
J4	8237.09	246	14401	13566	14401	16799	14402	9554	14400	22950	14400	29871
K1	664.41	22	622.91	401	2883.5	1993	802.16	387	641.36	781	14400	30492
K2												
K3												
K4												
L1	3178.2	88	14403	5289	14401	6305	14401	3678	14401	8591	14400	13895
L2												
L3												
L4												

Table 2. SOLVE: Solving benchmark data with CPLEX and CNDP solver. Empty lines mark instances that cannot be solved within 4 hours.

	CPLEX		KP-BR0-KF-CIT		KP-BR0-KF-SF-CIT		KP-BR0-KF-CIT-STAT	
	time	nodes	time	nodes	time	nodes	time	nodes
A1	0.22	1	0.57	5	0.56	1	0.42	1
A2	2.22	1	1.05	5	1.56	5	1.14	9
A3	0.3	1	0.56	3	0.73	3	0.54	5
A4	2.72	2	1.33	15	2.01	15	1.19	17
B1	13.31	3	2.08	9	2.75	9	1.97	11
B2	35.06	10	9.29	65	10.33	49	12.53	111
B3	31.57	4	3.67	23	6.16	29	3.8	33
B4	10.8	2	2.08	13	3.19	13	1.82	11
C1	15.85	5	3.97	13	5.19	13	3.69	13
C2	188.7	108	335.5	1269	369.25	951	324.21	1375
C3	25.34	30	14.78	59	20.63	57	17.71	83
C4	207.89	171	180.83	695	243.96	667	197.14	849
D1	28.76	2	7.52	23	10.48	23	7.93	35
D2	89.38	12	20.46	71	23.35	53	27.53	119
D3	39.65	6	10.32	39	15.02	35	10.49	47
D4	184.11	98	144.8	503	214.3	503	253.2	1089
E1	124.11	48	162.18	831	214.96	721	128.22	777
E2	424.34	174	381.09	1865	1034.1	3363	783.38	4357
E3	118.84	110	54.93	317	101.75	415	56.33	399
E4	2287.51	620	407.4	2007	718.85	2155	525.75	3105
F1	71.26	2	21.15	75	30.61	73	6.51	19
F2	1232.38	330	628.72	1969	1443.9	2837	573.91	2021
F3	277.27	59	216.22	679	496.18	1013	204.96	747
F4	181.01	9	17.44	47	20.82	35	13.92	43
G1	172.21	21	39.33	81	55.13	75	46.03	119
G2	392.59	118	273.09	583	402.48	571	276.24	711
G3	164.43	10	39.87	85	58.42	83	37.93	95
G4	1702.83	374	1658.2	3581	2452.4	3533	1873.2	4775
H1	48.06	2	24.31	37	32.85	31	10.18	11
H2	513.89	18	194.87	231	277.41	231	204.78	313
H3	6.97	1	13.47	17	15.02	11	11.32	17
H4								
I1	6.67	1	8.14	13	10.99	11	6.49	13
I2	250.84	28	114.4	237	183.07	253	105.99	263
I3	270.82	45	402.72	911	613.04	887	607.29	1685
I4	481.39	56	899.37	1811	1384.5	1849	827.19	2045
J1	177.64	2	72.85	105	37.79	31	41.41	63
J2	1679.39	73	687.56	835	1039	821	1047	1479
J3	409.79	42	537.18	871	663.1	701	389.02	705
J4	4825.17	228	5326.8	6211	6802.9	5209	7707	10591
K1	616.44	22	317.44	265	307.71	153	106.4	101
K2								
K3								
K4								
L1	2470.9	102	14402	6395	14400	3802	14403	6644
L2								
L3								
L4								

Table 3. OPT: Optimality proof with CPLEX and CNDP solver. Empty lines mark instances that cannot be solved within 4 hours.