Benders-type Branch-and-Cut Algorithms for Capacitated Facility Location with Single-Sourcing

Dieter Weninger* Laurence A. Wolsey†

November 25, 2022

Abstract

We consider the capacitated facility location problem with (partial) single-sourcing (CFLP-SS). A natural mixed integer formulation for the problem involves 0-1 variables x_j indicating whether faclility j is used or not and y_{ij} variables indicating the fraction of the demand of client i that is satisfied from facility j. When the x variables are fixed, the remaining problem is a transportation problem with single-sourcing. This structure suggest the use of a Benders' type decomposition algorithm. Here we present three possible variants. Applied to CFLP-SS they are compared computationally with a commercial solver (CPLEX) on the original formulation and a CPLEX version of Benders. Our results show that for CFLP-SS, when the percentage of clients requiring single-sourcing is less equal than 25%, the Benders' variants achieve a speedup of between 1.2 and 3.7.

Keywords: Integer programming, Benders' algorithm, Branch-and-cut, Mixed integer subproblems, Facilities planning and design

AMS 2010 Mathematics Subject Classification: 90C10, 90C11, 49M27, 65K05

1. Introduction

Our goal here is to compare algorithms for the capacitated facility location problem with (partial) single-sourcing. Specifically there are n potential facilities and m clients. Facility j has a capacity K_j and client i has a demand a_i. There is a fixed cost c_j of using facility j and h_{ij} is the cost of transporting a_i units from facility j to client i. The goal is to minimize the total cost while satisfying the demands subject to the capacity constraints and single-sourcing for the clients in $S \subseteq \{1, \ldots, m\}$. Letting $x_j = 1$ if facility j is opened and 0 otherwise and y_{ij} denote the fraction of the demand of client i satisfied from facility j, it is then straightforward to formulate the problem as a mixed integer linear programming problem.

Such a formulation is a special case of a mixed 0-1 integer program of the form

$$\min\{cx + hy : Ax + By \geq b, x \in \{0,1\}^M, y \in \mathbb{Z}_+^{N_1} \times \mathbb{R}_+^{N_2}\}.$$

In the last 20 years there has been a significant regain of interest in Benders' algorithm [4] that can treat problems with such structure. The basic idea, when the y-variables are all continuous, is to break up the problem into a mixed integer "Master Problem" in the (x, η)-variables in which η provides an underestimate of the corresponding optimal cost and a "Subproblem" in the y-variables in which x is fixed. Solution of the latter provides information to feed back to the Master Problem so as to improve the estimate provided by η. Thus Benders' approach allows one to largely treat the x and y-variables separately. When applied to mixed integer linear programs that rely on linear programming relaxations to obtain bounds, one interpretation is:

For a linear programming relaxation of the form $\min\{cx + hy : Ax + By \geq b, x \in [0,1]^M, y \in \mathbb{R}_+^N\}$, *Benders' algorithm is a method to solve such a linear program in which one iterates between solving an LP in the* (x, η)*-space (where* η *is a single additional variable) and an LP in the* y*-space.*

When some of the y-variables are integer, things are a little more difficult. However this viewpoint leads naturally to several possible branch-and-Benders'-cut algorithms. Here we describe three such algorithms and test their behavior on the CFLP-SS and compare with the solution by a standard MIP solver CPLEX and also with a CPLEX implementation of Benders.

*FAU Erlangen-Nürnberg, Analytics & Mixed-Integer Optimization, Cauerstr. 11, 91058 Erlangen, Germany, dieter.weninger@fau.de
†CORE, UCLouvain, 1348 Louvain-la-Neuve, Belgium, laurence.wolsey@uclouvain.be

The first algorithm, denoted BCxy (Branch-and-Cut in (x, y)-space), involves branching on both x and y-variables. In fact this closely resembles the standard MIP algorithm used in commercial MIP solvers, except that the linear programming relaxations are solved differently.

The other two algorithms avoid the need to branch in the (x, y)-space. The second algorithm, denoted BxLL (Branching in the x-space and adding Laporte-Louveaux (no-good cuts) [25]), is well-known (also a special case of Logical Benders, Hooker [24]). Here the y-space mixed integer problem needs to be solved to optimality. It can be seen as a simplified version of the next algorithm. For the third algorithm, denoted BxCy (Branching in x-space and Cutting in y-space), the price to pay is that the subproblems must now be solved at least partially as (mixed)-integer programs. Cuts for the subproblem in the y-space are lifted into valid cuts in the (x, y)-space.

The outline of the paper is as follows. In Section 2 we formulate CFLP-SS and briefly present Benders' approach. In Section 3 BCxy is presented and in Section 4 algorithms BxLL and BxCy are described. For BxCy it is shown how Gomory mixed integer cuts [22] in the y-space extend easily to give valid inequalities in the (x, y)-space. In Section 5 more specific details are given on the application of the three algorithms to CFLP-SS. Finally in Section 6 we present a computational comparison between these three Benders branch-and-cut variants, the commercial solver CPLEX [13] and CPLEX-Benders. Section 7 contains some conclusions.

We terminate this section by pointing out some important aspects when using Benders' approach and some related literature:

i) Since branch-and-cut algorithms have become standard, it is natural to modify the original approach of Benders involving repeated solution of a mixed integer Master problem. Instead the Master problem is solved once using branch-and-cut with the Benders' subproblem as the cut generation problem at each node of the branch-and-cut tree. Therefore instead of resolving the IP Master problem repeatedly as suggested by Benders, one now runs a single pass branch-and-cut algorithm.

ii) Given that the dual of the LP subproblem typically has multiple optimal solutions or unbounded rays, several ideas have been proposed so as to select solutions leading to "strong" Benders' cuts. These include Pareto-optimal cuts (Magnanti and Wong [26], Papadakos [29]), the use of different normalizations to bound the feasible region of the dual (Fischetti et al. [18]), the in-out approach (Ben-Ameur and Neto [3], Fischetti et al. [17]) and a partial re-optimization approach (Wentges [35]). A facet-generating approach is proposed in Conforti and Wolsey [8] and a recent generalization in Brandenberg and Stursberg [5].

iii) Until the cuts generated provide a reasonable approximation to the real cost of the continuous y-variables, the solutions of the Master Problem may be of little interest. Thus it is necessary to generate a good set of initial inequalities and then solve the linear programming relaxation of the Master before starting to branch on the x-variables that are fractional. Specific references are provided in the implementation section.

iv) When some or all y-variables are integer, the subproblem is now an integer or mixed integer program. So the standard Benders' y-variable subproblem here is no longer sufficient to provide dual information characterizing the optimal value of the integer subproblem. Various solutions have been proposed, many of them motivated by two-level stochastic programs with integer recourse. These include *no-good optimality and feasibility cuts* (Laporte and Louveaux [25]), solution of the integer subproblem using Gomory fractional cuts (Gade et at. [19]), lift-and-project cuts (Sen and Higle [32]) or the use of IP dual functions (Caroe and Tind [6]). Another approach is to incorporate some of the y-variables in the Master problem (Sen and Sherali [33]).

v) Numerous successful applications using Benders have been reported including multicommodity distribution design (Geoffrion and Graves [20]), simultaneous aircraft routing and crew scheduling (Cordeau et al. [9]), fixed-charge network design problems (Costa [12]), various facility location and covering problems (Fischetti et al. [16, 15]), stochastic three-level lot sizing (Gruson et al. [23]), balancing of assembly lines (Sikora [34]) and electricity market clearing problems (Ceyhan et al. [7]). Rahmaniani et al. [30] provides an extensive literature review of Benders' decomposition, including references to many more applications.

2. CFLP-SS and Benders' Approach

2.1 Formulation of CFLP-SS

First we formulate CFLP-SS as a linear mixed integer problem. Letting $x_j = 1$ if facility j is opened and 0 otherwise and y_{ij} denote the fraction of the demand of client i satisfied from facility j, it is then straightforward to formulate the problem as a mixed integer linear programming problem.

We consider the standard formulation

$$
\begin{aligned}
\min \quad & \sum_{j=1}^{n} c_j x_j + \sum_{i=1}^{m}\sum_{j=1}^{n} h_{ij} y_{ij} && && \text{(1a)}\\
& \sum_{j=1}^{n} y_{ij} = 1 && i = 1,\ldots,m && \text{(1b)}\\
& \sum_{i=1}^{m} a_i y_{ij} \le K_j x_j && j = 1,\ldots,n && \text{(1c)}\\
& y_{ij} \le x_j && i = 1,\ldots,m,\ \ j = 1,\ldots,n \qquad (\text{OP}^1) && \text{(1d)}\\
& \sum_{j=1}^{n} K_j x_j \ge \sum_{i=1}^{m} a_i && && \text{(1e)}\\
& x \in \{0,1\}^n, y \in \mathbb{R}_+^{mn}, && && \text{(1f)}\\
& y_{ij} \in \{0,1\} \text{ for } i \in SS. && && \text{(1g)}
\end{aligned}
$$

in which constraints (1b) ensure that all the demand of client i is satisfied, (1c) that the capacity of facility j is not exceeded, (1d) that client i can only be served from facility j if the facility is open and (1e) that there is sufficient global capacity available to satisfy all the demand at least when there is no single-sourcing. Here the constraints (1d) and (1e) are redundant, but it is well-known that they play an important role in strengthening the lower bounds obtained when considering linear programming relaxations of the problem.

2.2 Benders' Approach when $SS = \emptyset$

To introduce Benders' approach, we first consider the case with no single-sourcing, namely the problem (OP^1), (1a)-(1f) with $SS = \emptyset$. First the problem is rewritten as:

$$
\begin{aligned}
\zeta = \min \sum_{j=1}^{n} c_j x_j + \phi^1(x)\\
\sum_{j=1}^{n} K_j x_j \ge \sum_{i=1}^{m} a_i\\
x \in \{0,1\}^n,
\end{aligned}
$$

with associated subproblem ($\text{SP}^1(x^*)$) to evaluate $\phi^1(x)$ at a point x^*:

$$
\phi^1(x^*) = \{\min \sum_{ij} h_{ij} y_{ij} : \sum_{j=1}^{n} y_{ij} = 1 \ \forall\ i, \sum_{i=1}^{m} (-a_i) y_{ij} \ge -K_j x_j^* \ \forall\ j, -y_{ij} \ge -x_j^* \ \forall\ i,j,\ \ y \in \mathbb{R}_+^{mn}\}. \quad (\text{SP}^1(x^*))
$$

The corresponding dual LP is :

$$
\max\{\sum_{i=1}^{m} u_i^0 - \sum_{j=1}^{n} K_j x_j^* u_j^1 - \sum_{i=1}^{m}\sum_{j=1}^{n} x_j^* u_{ij}^2 : (u^0, u^1, u^2) \in U^1\} \qquad (\text{DSP}^1(x^*))
$$

where

$$
U^1 = \{(u^0, u^1, u^2) \in \mathbb{R}^m \times \mathbb{R}_+^n \times \mathbb{R}_+^{mn} : u_i^0 - a_j u_j^1 - u_{ij}^2 = h_{ij} \quad i = 1,\ldots,m, j = 1,\ldots,n\}.
$$

Thus the subproblem ($\text{SP}^1(x)$) can be rewritten as

$$
\begin{aligned}
\phi^1(x) &= \min \eta\\
\eta &\ge \sum_{i=1}^{m} u_i^{0s} - \sum_{j=1}^{n} K_j x_j u_j^{1s} - \sum_{i=1}^{m}\sum_{j=1}^{n} x_j u_{ij}^{2s} \qquad s = 1,\ldots,S\\
0 &\ge \sum_{i=1}^{m} v_i^{0r} - \sum_{j=1}^{n} K_j x_j v_j^{1r} - \sum_{i=1}^{m}\sum_{j=1}^{n} x_j u_{ij}^{2r} \qquad r = 1,\ldots,R
\end{aligned}
$$

where $\{(u^{0s}, u^{1s}, u^{2s})\}_{s=1}^{S}$, $\{(v^{0r}, v^{1r}, v^{2r})\}_{r=1}^{R}$ are the extreme points and extreme rays of U^1.

This representation of $\phi^1(x)$ leads to the Benders Master problem (BM^1) equivalent to (OP^1):

$$\begin{aligned}
\zeta = \min \sum_{j=1}^{n} c_j x_j + \eta & & \\
\sum_{j=1}^{n} K_j x_j &\geq \sum_{i=1}^{m} a_i & \\
\eta &\geq \sum_{i=1}^{m} u_i^{0s} - \sum_{j=1}^{n} K_j u_j^{1s} x_j - \sum_{i=1}^{m}\sum_{j=1}^{n} u_{ij}^{2s} x_j \quad s = 1, \dots, S & \qquad (\text{BM}^1)\\
0 &\geq \sum_{i=1}^{m} v_i^{0r} - \sum_{j=1}^{n} K_j v_j^{1r} x_j - \sum_{i=1}^{m}\sum_{j=1}^{n} v_{ij}^{2r} x_j \quad r = 1, \dots, R & \\
x \in \{0,1\}^n. & &
\end{aligned}$$

Now it is natural to work with a relaxation of the problem involving just a subset of the constraints associated to the extreme points and extreme rays. Given an optimal solution (x^*, η^*) of this relaxation, the subproblem ($\text{SP}^1(x^*)$) is then a cutting plane separation algorithm to generate missing inequalities, if any. Whereas Benders suggested resolving the relaxation of (BM^1) each time that new constraints/cuts were added, the modern approach is to view the subproblem as a way to generate missing constraints at each node of a branch-and-cut tree. In the following two sections we consider ways to deal with the integrality of some or all of the y-variables.

Note that because of the constraint $\sum_j K_j x_j \geq \sum_i a_i$, the subproblem ($\text{SP}^1(x^*)$) is always feasible and no extreme rays are generated in the case $SS = \emptyset$. However this will no longer be the case in the following sections when some y_{ij} variables are integer.

3. Algorithm BCxy

Here the integrality of some or all of the y-variables is treated by allowing branching on y-variables. However the linear programs that arise in solving the Master problem still only involve the x, η variables.

At a given node of the branch-and-cut tree for the Benders Master problem, we assume that $\ell^x \leq x \leq k^x$ and $y_{ij} \geq \ell_{ij} \geq 0$ and $y_{ij} \leq k_{ij} (\leq x_j)$ for all i, j.

For fixed x^*, the subproblem ($\text{SP}^2(x^*)$) at a node is now

$$\phi^2(x^*) = \{\min \sum_{ij} h_{ij} y_{ij} : \sum_{j=1}^{n} y_{ij} = 1 \ \forall\, i, \sum_{i=1}^{m} (-a_i) y_{ij} \geq -K_j x_j^* \ \forall\, j, -y_{ij} \geq -k_{ij}, y_{ij} \geq \ell_{ij} \ \forall\, i, j, y_{ij} \in \mathbb{Z}^1 \ i \in SS, j = 1, \dots, n\}.$$

with LP relaxation ($\text{SP}^2_{LP}(x^*)$) and optimal value $\phi^2_{LP}(x^*)$.

The dual of ($\text{SP}^2_{LP}(x^*)$) is

$$\max\{\sum_{i=1}^{m} u_i^0 - \sum_{j=1}^{n} K_j x_j^* u_j^1 - \sum_{i=1}^{m}\sum_{j=1}^{n} (k_{ij} u_{ij}^2 - \ell_{ij} u_{ij}^3) : (u^0, u^1, u^2, u^3) \in U^2\} \qquad (\text{DSP}^2(x^*))$$

where

$$U^2 = \{(u^0, u^1, u^2, u^3) \in \mathbb{R}^m \times \mathbb{R}^n_+ \times \mathbb{R}^{2mn}_+ : u_i^0 - a_j u_j^1 - (u_{ij}^2 - u_{ij}^3) = h_{ij} \quad i = 1, \dots, m, j = 1, \dots, n\}$$

and $\{(u^{0s}, u^{1s}, u^{2s}, u^{3s})\}_{s=1}^{S}$, $\{(v^{0r} v^{1r}, v^{2r}, v^{3r})\}_{r=1}^{R}$ are the extreme points and extreme rays of U^2.

The Benders' Master problem takes the form:

$$\zeta = \min \sum_{j=1}^{n} c_j x_j + \eta$$

$$\begin{aligned}
\sum_{j=1}^{n} K_j x_j &\geq \sum_{i=1}^{m} a_i \\
\eta &\geq \sum_{i=1}^{m} u_i^{0s} - \sum_{j=1}^{n} K_j u_j^{1s} x_j - \sum_{i=1}^{m} \sum_{j=1}^{n} (u_{ij}^{2s} k_{ij} - u_{ij}^{3s} \ell_{ij}) \quad s = 1, \dots, S \quad (\text{BM}^2) \qquad (2) \\
0 &\geq \sum_{i=1}^{m} v_i^{0r} - \sum_{j=1}^{n} K_j v_j^{1r} x_j - \sum_{i=1}^{m} \sum_{j=1}^{n} (u_{ij}^{2r} k_{ij} - v_{ij}^{3r} \ell_{ij}) \quad r = 1, \dots, R \qquad (3)
\end{aligned}$$

$$\ell^x \leq x \leq k^x$$
$$x \in \{0,1\}^n.$$

3.1 Outline of the BCxy Algorithm

Given the LP solution (x^*, η^*) of (BM^2) at a node in the branch-and-cut tree, we discuss the different possible outcomes in the subproblem.

1. If problem (DSP$^2(x^*)$) is unbounded, then the LP relaxation (SP$^2_{LP}(x^*)$) is infeasible. Let v^* be the associated unbounded extreme ray with $\sum_{i=1}^m v_i^{0*} - \sum_{j=1}^n K_j x_j^* v_j^{1*} - \sum_{i=1}^m \sum_{j=1}^n (k_{ij} v_{ij}^{2*} - \ell_{ij} v_{ij}^{3*}) > 0$.
   The corresponding feasibility cut (3) cutting off x^* is added to (BM^2).
2. If (DSP$^2(x^*)$) has a finite optimal value $\phi^2(x^*)$, let y^* be the optimal primal solution and u^* an optimal dual solution.
   i) If $\phi^2(x^*) > \eta^*$, an optimality cut (2) cutting off (x^*, η^*) is added to (BM^2).
   ii) If $\phi(x^*) = \eta^*$, the linear programming relaxation of (BM^2) at the node is solved with optimal value $\zeta^* = cx^* + \eta^*$. There are now three cases.

   a) If $x^* \in \{0,1\}^n, y_{ij}^* \in \{0,1\}$ for all i, j with $i \in SS$, a new feasible solution has been found. If best so far, the incumbent is updated. The node is pruned by optimality.

   b) If $x^* \notin \{0,1\}^n$, one branches on a variable with $x_j^* \notin \{0,1\}$. Two new nodes are created in the branch-and-cut tree for (BM^2).

   c) If $x^* \in \{0,1\}^n$, but $y_{ij}^* \notin \{0,1\}$ for some $i \in SS$ and $j \in \{1 \dots, n\}$, one branches on one such variable with $y_{ij}^* \notin \{0,1\}$. Two new nodes are created with updated constraint sets, one with $(y_{ij} \geq)\ell_{ij} = 1$ and the other with $(y_{ij} \leq) k_{ij} = 0$.

Thus branching takes place on x and y in the branch-and-cut tree, but the optimization of the current relaxation of (BM^2) at a node only involves x, η variables. Note that the set U^2 never changes so only the objective function in (DSP$^2(x^*)$) changes from one iteration to the next. It also follows that once an extreme point or extreme ray has been generated, it can be used in (BM^2) in any node of the branch-and-cut tree. By treating varying bounds in the subproblem, the set U^2 obtains larger sets of extreme points and extreme rays than the set U^1 derived in Section 2 when the set SS is empty. These provide additional cuts based on the problem structure for the (x, η)-space (BM^2) problem.

4. Algorithms BxLL and BxCy: Branching in x-space, Cutting from y-space

Here we present two Benders' branch-and-cut algorithms in which the branching in the Master problem only involves the x-variables. However the integrality of some y-variables in the subproblem allows us to generate additional cutting planes for addition to the Master problem. Thus we work with the original problem to which a set of valid inequalities $\tilde{\Pi} x + \Theta y \geq \tilde{\Pi}_0$ has been added. Here the bounds on y_{ij} in the subproblem do not vary.

Now the subproblem takes the form

$$\phi^3(x^*) = \min \sum_{ij} h_{ij} y_{ij}$$

$$\begin{array}{rcll} \sum_{j=1}^{n} y_{ij} & = & 1 & i = 1, \ldots m \\ \sum_{i=1}^{m} (-a_i) y_{ij} & \geq & -K_j x_j \quad j = 1, \ldots, n & (\mathrm{SP}^3(x^*)) \\ -y_{ij} & \geq & -x_j^* \quad i = 1, \ldots, m, j = 1, \ldots, n & \\ \Theta y & \geq & \tilde{\Pi}_0 - \tilde{\Pi} x^* & \\ y \in \mathbb{R}_+^{mn} & & y_{ij} \in \{0, 1\} \; i \in SS, j = 1, \ldots, n. & \end{array}$$

The dual of the LP relaxation is:

$$\max\{\sum_{i=1}^{m} u_i^0 - \sum_{j=1}^{n} K_j x_j^* u_j^1 - \sum_{i=1}^{m} \sum_{j=1}^{n} x_j^* u_{ij}^2 + \tilde{u}(\tilde{\Pi}_0 - \tilde{\Pi} x^*) : (u, \tilde{u}) \in U^3\} \qquad (\mathrm{DSP}^3(x^*))$$

where

$$U^3 = \{(u^0, u^1, u^2, \tilde{u}) \in \mathbb{R}^m \times \mathbb{R}_+^n \times \mathbb{R}_+^{mn} \times \mathbb{R}_+^T : u_i^0 - a_j u_j^1 - u_{ij}^2 + \tilde{u}\Theta_{ij} \geq h_{ij} \quad \forall\, i, j\}$$

and $\{(u^{0s}, u^{1s}, u^{2s}, \tilde{u}^s)\}_{s=1}^S$, $\{(v^{0r} v^{1r}, v^{2r}, \tilde{v}^r)\}_{r=1}^R$ are the extreme points and extreme rays of U^3.

The Benders Master problem (BM^3) now takes the form:

$$\zeta = \max \sum_{j=1}^{n} c_j x_j + \eta$$

$$\begin{array}{rcll} \sum_{j=1}^{n} K_j x_j & \geq & \sum_{i=1}^{m} a_i & \\ \eta & \geq & \sum_{i=1}^{m} u_i^{0s} - \sum_{j=1}^{n} K_j u_j^{1s} x_j - \sum_{i=1}^{m} \sum_{j=1}^{n} u_{ij}^{2s} x_j + \tilde{u}^s(\tilde{\Pi}_0 - \tilde{\Pi} x) \quad s = 1, \ldots, S \quad (\mathrm{BM}^3) & (4) \\ 0 & \geq & \sum_{i=1}^{m} v_i^{0r} - \sum_{j=1}^{n} K_j v_j^{1r} x_j - \sum_{i=1}^{m} \sum_{j=1}^{n} v_{ij}^{2r} x_j + \tilde{v}^r(\tilde{\Pi}_0 - \tilde{\Pi} x) \quad r = 1, \ldots, R & (5) \\ \ell^x & \leq & x \leq k^x & \\ x & \in & \{0, 1\}^n, \eta \in \mathbb{R}. & \end{array}$$

4.1 BxLL

Here the cuts used are very simple. When the solution x^* of (BM^3) at a node in the branch-and-cut tree lies in $\{0, 1\}^n$, the mixed integer subproblem ($\mathrm{SP}^3(x^*)$) is solved to optimality. If it is infeasible, a Laporte-Louveaux [25] or no-good cut:

$$\sum_{j: x_j^* = 0} x_j + \sum_{j: x_j^* = 1} (1 - x_j) \geq 1 \qquad (6)$$

is added to (BM^3) to ensure that the solution x^* will not appear again, and if feasible a no-good optimality cut

$$\eta \geq \phi^3(x^*) - (\phi^3(x^*) - L)(1 - \sum_{j: x_j^* = 0} x_j - \sum_{j: x_j^* = 1} (1 - x_j)) \qquad (7)$$

with L a lower bound on $\phi^3(x) : x \in \{0, 1\}^n$ is added guaranteeing that if x^* reappears, $\eta = \phi^3(x^*)$ is evaluated correctly. Note that in this case as the cuts do not involve y-variables, $\Theta = 0$ so there are no additional constraints in U^3 and ($\mathrm{SP}^3(x^*)$) from iteration to iteration. Thus the constraints added to (BM^3) are of the form $\tilde{\Pi} x \geq \tilde{\Pi}_0$.

4.2 BxCy

Here the idea is to use cuts generated in solving the mixed integer subproblem by a cutting plane algorithm so as to provide additional valid cuts for (BM^3). The main observation, first used in [19], is:

For some of the cutting planes used in solving MIPs, there is a "simple" lifting, so that a valid inequality in the y-space for a mixed-integer set $\{y \in \mathbb{Z}^p_+ \times \mathbb{R}^q_+ : By \geq b - Ax\}$ can be extended to give a valid inequality in the (x, y)-space for the set $\{(x, y) \in \{0, 1\}^n \times \mathbb{Z}^p_+ \times \mathbb{R}^q_+ : Ax + By \geq b\}$.

Here there are two types of cuts, standard Benders' type cuts in the (x, η)-variables that are added to the Benders' Master problem and cuts in (x, y)-space generated in the subproblems based on integrality of the y-variables. The latter are added to the original problem formulation and also to the y-space subproblem.

4.3 Outline of the BxCy Algorithm

We describe a branch-and-cut algorithm in the x-space. The integrality of the y-variables is only considered for points $x^* \in \{0, 1\}^n$. We define an iteration of the current Benders' Master problem (BM^3) at a node.

Solve the linear programming relaxation of (BM^3) with solution (x^*, η^*) and $\zeta_{LP} = cx^* + \eta^*$.
Solve (SP$^3_{LP}(x^*)$) with optimal solution y^*, dual solution $(u^*, \tilde{u}^*)$, $\phi^3_{LP}(x^*) = hy^*$. There are seven cases.

1. $x^* \notin \{0, 1\}^n$. $\eta^* = \phi^3_{LP}(x^*)$.
   The linear programming relaxation of (BM^3) is solved. Branch on a variable x_j with $x^*_j \notin \{0, 1\}$.
   Create two new nodes.

2. $x^* \notin \{0, 1\}^n$. $\eta^* < \phi(x^*)$.
   Add a standard Benders' infeasibility cut (5) or optimality cut (4) to (BM^3) and return to to (BM^3).

3. $x^* \in \{0, 1\}^n$. $y^*_{ij} \in \{0, 1\}$ for all i, j with $i \in SS$. $\eta^* < \phi^3_{LP}(x^*)$.
   A feasible solution is found. If best so far, update incumbent. Add an optimality cut (4) to (BM^3) and return to (BM^3).

4. $x^* \in \{0, 1\}^n$. $y^*_{ij} \in \{0, 1\}$ for all i, j with $i \in SS$. $\eta^* = \phi^3_{LP}(x^*) = \phi^3(x^*)$.
   A feasible solution is found. If best so far, update the incumbent. Prune the node of (BM^3) by optimality.

5. $x^* \in \{0, 1\}^n$. $y^*_{ij} \notin \{0, 1\}$ for some i, j with $i \in SS$. $\eta^* = \phi^3_{LP}(x^*) \leq \phi^3(x^*)$.
   Generate a cut $\theta y \geq \pi_0$ cutting off y^* and an (x, y)-space cut $\tilde{\pi}x + \theta y \geq \tilde{\pi}_0$ cutting off the point (x^*, y^*).
   Add the cut to (SP$^3(x^*)$) and (BM^3) respectively. Return to (BM^3).

6. $x^* \in \{0, 1\}^n$. $y^*_{ij} \notin \{0, 1\}$ for some i, j with $i \in SS$. $\eta^* < \phi^3_{LP}(x^*)$.
   Add an infeasibility (5) or optimality cut (4) to (BM^3) as in case 2. Return to (BM^3).

7. If no suitable cuts can be found for (SP$^3_{LP}(x^*)$) and no suitable optimality cut can be returned, solve (SP$^3(x^*)$) to optimality, and return a no-good feasibility (6) or no-good optimality cut (7) to (BM^3), where L is now a lower bound on $\min\{\phi^3(x) : x \in \{0, 1\}^n\}$.

Here we consider how to generate an (x, y)-space cut from the subproblem (SP$^3(x^*)$) in case 5.
For ease of presentation, we introduce the additional variables $\bar{x}_j = 1 - x_j$ for $j = \{1, \ldots, n\}$. Let $N_0 = \{j : x^*_j = 0\}$, $N_1 = \{j : x^*_j = 1\}$ and $\tilde{\Pi}_0 = \Pi_0 - \sum_{j \in N_1} \Pi_j$.
(SP$^3(x^*)$) can now be rewritten as:

$$
\begin{array}{lcllr}
\min \sum_{i,j} h_{ij} y_{ij} & & & & \\
\sum_{j=1}^{n} y_{ij} & = & 1 & & i = 1 \ldots, m \\
-\sum_{i=1}^{m} a_j y_{ij} & \geq & 0 & [-K_j x^*_j] \ \text{ if } x^*_j = 0 & j = 1, \ldots, n \\
-\sum_{i=1}^{m} a_j y_{ij} & \geq & -K_j & [+K_j \bar{x}^*_j] \ \text{ if } x^*_j = 1 & j = 1, \ldots, n \\
-y_{ij} & \geq & 0 & [-x_j] \text{ if } \ x^*_j = 0 & i = 1, \ldots, m, j = 1, \ldots, n \\
-y_{ij} & \geq & -1 & [+\bar{x}^*_j] \text{ if } \ x^*_j = 1 & i = 1, \ldots, m, j = 1, \ldots, n \\
\Theta y & \geq & \tilde{\Pi}_0 & [-\sum_{j \in N_0} \tilde{\Pi}_j x_j + \sum_{j \in N_1} \tilde{\Pi}_j \bar{x}_j] & \\
\bar{x} & = & 1 - x & & \\
x & = & x^* & & \\
y & \in & \mathbb{R}^{mn}_+. & &
\end{array}
$$

Note that the terms on the right hand side between the brackets [.] are all of value 0 when $x = x^*$. Thus to obtain the requirements vector for $(\mathrm{SP}^3(x^*))$, the terms in brackets can be ignored.

Suppose that the optimal solution y^* of $(\mathrm{SP}^3_{LP}(x^*))$ has some single-sourcing variable y^*_{ij} that is fractional. Generate a valid inequality $\theta y \geq \pi_0$ for $(\mathrm{SP}^3(x^*))$ cutting off y^* and extend it to a valid inequality $\theta y \geq \tilde{\pi}_0 - \sum_{j\in N_0} \tilde{\pi}_j x_j + \sum_{j\in N_1} \tilde{\pi}_j \bar{x}_j$ for the original problem cutting off (x^*, y^*).

Below we describe how to lift the Gomory mixed integer cut [22] that we use in the implementation of BxCy. For ease of description, we suppose that the cut is the first cut generated for $(\mathrm{SP}^3(x^*))$.

4.4 The Gomory Mixed Integer Cut for the Subproblem

Solve the linear programming relaxation of $(\mathrm{SP}^3(x^*))$ with solution y^*. Suppose that $y^*_{pq} \notin \{0,1\}$ for some p, q with $p \in SS$. The corresponding row of the optimal LP tableau takes the form:

$$y_{pq} + \sum_{ij\in N_{SS}} \beta_{ij} y_{ij} + \sum_{k\in K} \gamma_k z_k = \beta_0 - \sum_{j\in N_0} \alpha_j x_j + \sum_{j\in N_1} \alpha_j (1 - x_j), x = x^*$$

where N_{SS} denotes the nonbasic y_{ij}-variables with $i \in SS$ and z_k for $k \in K$ contains the nonbasic real variables y_{ij} with $i \notin SS$ and slack variables from the constraints.

The basic variable y_{pq} in the row takes value $\beta_0 \in (0, 1)$. Let $f_0 = \beta_0$, $f_{ij} = \beta_{ij} - \lfloor \beta_{ij} \rfloor$, $g_j = \alpha_j - \lfloor \alpha_j \rfloor$ for $j \in N_0$, $g_j = -\alpha_j - \lfloor -\alpha_j \rfloor$ for $j \in N_1$.

Then the lifted Gomory mixed integer cut [22]

$$\sum_{ij\in N_{SS}: f_{ij}\leq f_0} f_{ij} y_{ij} + \sum_{ij\in N_{SS}: f_{ij}> f_0} \frac{f_0(1 - f_{ij})}{1 - f_0} y_{ij} + \sum_{k\in K:\gamma_k\geq 0} \gamma_k z_k - \sum_{k\in K:\gamma_k< 0} \frac{f_0}{1 - f_0} \gamma_k z_k$$

$$\geq f_0 - \sum_{j\in N_0: g_j\leq f_0} g_j x_j - \sum_{j\in N_0: g_j> f_0} \frac{f_0(1 - g_j)}{1 - f_0} x_j - \sum_{j\in N_1: g_j\leq f_0} g_j (1 - x_j) - \sum_{j\in N_1: g_j> f_0} \frac{f_0(1 - g_j)}{1 - f_0} (1 - x_j)$$

is the cut that is added to (BM^3).

Note that as $x \in \{0, 1\}^n$, it is always possible to lift a valid inequality $\theta y \geq \pi_0$ into a valid inequality $\theta y \geq \tilde{\pi}_0 - \tilde{\pi} x$.

5. Implementation Details

In this section we describe some implementation details of BCxy, BxLL and BxCy. Our algorithms have been implemented in C and make use of the IBM ILOG CPLEX 20.1.0.0 callable library.

5.1 Benders' Initialization

Starting from the formulation (1a)-(1g) of CFLP-SS in Section 2, the initial Benders Master Problem (BM) is:

$$\zeta = \min \quad cx + \eta \tag{8a}$$

$$\sum_{j=1}^{n} K_j x_j \geq \sum_{i=1}^{m} a_i \tag{8b}$$

$$x \in [0, 1]^n, \eta \in \mathbb{R}^1. \tag{8c}$$

The initial separation problem $(\mathrm{SP}(x^*))$ is given in (2) with the bounds are $\ell_{ij} = 0$ and $k_{ij} = x^*_j$ for all i, j.

5.2 Generation of initial Cuts

In order for BCxy, BxLL and BxCy to run efficiently, it is necessary to generate initial cuts that define a strong bound at the root node. For this purpose, a Benders approach can be used in which the integrality of the variables is relaxed. However, a usual Benders approach often converges too slowly and makes the overall method simply impractical. In Fischetti et al. [15], however, a fast method based on an in-out variant in the spirit of Ben-Ameur and Neto [3] is proposed that achieves strong bounds. This in-out approach is aimed at quickly determining a small set of Benders cuts that brings the master LP relaxation objective function value as close as possible to the optimal value of the LP relaxation of the original problem. At each cut loop iteration, we have two points in the space of the x-variables: the optimal solution $x^* \in [0,1]^n$ of the current master LP, and a stabilizing point $\tilde{x} \in [0,1]^n$ inside the feasible region of the master LP. At the beginning the stabilizing point is initialized by setting $\tilde{x}_j = 1$ for $j = 1, \dots, n$. At each iteration, we move $\tilde{x}$ towards x^* by setting $\tilde{x} = \alpha\tilde{x} + (1-\alpha)x^*$ and then perform Benders cut separation to the point $\lambda x^* + (1-\lambda)\tilde{x}$, where $\alpha \in (0,1]$ and $\lambda \in (0,1]$. Initially we use $\alpha = 0.5$ and $\lambda = 0.1$.

When using Benders method to determine initial cuts, it is often the case that the lower bound improves quickly at the beginning, but with an increasing number of iterations the lower bound improves only slightly. Therefore, one tries to stop the separation of cuts if the bound does not improve sufficiently. Let ζ_{current} be the objective function value of the master LP at the current iteration and ζ_{last} the value of the master LP of the previous iteration. At the beginning we assume $\zeta_{\text{last}} = 0$. The lower bound is judged to have improved if $|\zeta_{\text{current}} - \zeta_{\text{last}}| > \varepsilon \cdot |\zeta_{\text{current}}|$ holds for $\varepsilon = 10^{-6}$. If for 5 consecutive iterations the lower bound has not improved, the parameter λ is set to 1 and the cut generation continues. If after another 5 consecutive iterations the lower bound has not improved, we stop cut separation. Then the master LP is solved one last time and all cuts with a positive slack value greater than $\gamma + \gamma \cdot |b|$, where b is the right hand side of the Benders cut and $\gamma = 10^{-4}$, are removed. Such a cut purging is also done at every fifth iteration of the master LP in order to generate as few cuts as possible and accelerate computation.

Separating ordinary Benders cuts as described in Wolsey [37] usually results in a lower bound not being improved fast enough. We therefore decided to separate another class of Benders cuts as proposed in [15]. For this purpose, the solution x^* of the master LP is taken and the separation problem is fixed. Then the separation problem $(\text{SP}^1(x^*))$ is solved to obtain values of the dual variables. Let $u^* \in \mathbb{R}^m$ be the vector of dual variables associated to the assignment constraints $\sum_{j=1}^n y_{ij} = 1$. The idea is now to recompute optimal reduced costs by solving a series of n continuous knapsack problems

$$\text{KP}^j_{u^*} := \min\left\{ \sum_{i=1}^m (h_{ij} - u_i^*)z_i : \sum_{i=1}^m a_i z_i \le K_j, z \in [0,1]^m \right\}, \tag{9}$$

where $j \in \{1, \dots, n\}$. The knapsack problems (9) can be solved rapidly by a sorting approach.

Using the solutions of the knapsack problems (9) we determine Benders cuts of the form

$$\eta \ge \sum_{i=1}^m u_i^* + \sum_{j=1}^n \text{KP}^j_{u^*} x_j. \tag{10}$$

More details on the derivation of cuts (10) can be found in Cornuéjols et al. [11] and Fischetti et al. [15].

Once we have solved the separation problem $(\text{SP}^1(x^*))$, we remember the optimal basis and reload it the next time we solve $(\text{SP}^1(x^*))$. This usually saves iterations when using the simplex algorithm.

5.3 BCxy

As described in Section 3, for BCxy branching is performed on both the x and y-variables. We always branch preferably on the x-variables and only branch on y-variables when the x-variables take on integer values. For the x-variables we use “full strong branching” [1] as branching rule. Full strong branching is often very costly, but provides the best local branching decision. However, the master LP is usually solvable quite quickly and the effort for full strong branching pays off. The situation for the separation subproblem is different, because solving the corresponding LPs is usually more elaborate. In addition, the number of integer variables in the subproblem is significantly larger than in the master problem, especially if many clients demand single-sourcing. For simplicity we use “most infeasible branching” [1] as branching rule for the y-variables.

If $x^* \in \{0,1\}^n$, but $y^*_{ij} \notin \{0,1\}$ for some $i \in SS$ and $j \in \{1 \dots, n\}$, one branches on a variable with $y^*_{ij} \notin \{0,1\}$ (see case c). In the branch $y_{ij} = 1$ we add additionally $x_j = 1$ in order to avoid infeasible subproblems in the further course of the solution process.

In each iteration of BCxy, we choose a node q with the smallest bound and separate cuts of type (10) (see Section 3.1 case 2i). If the node is solved (see Section 3.1 case 2ii), it is then checked whether all integer variables have integer values (case a) and, if so, the incumbent can be updated, or branching is performed on an x-variable (case b) or on a y-variable (case c). Let ζ^q_{current} denote the objective function value of the master LP of node q at the current iteration and ζ^q_{last} be the value of the master LP of node q at the previous iteration. From the second iteration on, cuts for node q are only separated as long as $\zeta^q_{\text{current}}/\zeta^q_{\text{last}} - 1 > \beta$ with $\beta = 1.5 \cdot 10^{-4}$.

5.4 BxLL

We would just like to make two small remarks here about creating the cuts described in Section 4.1. First, we determine a lower bound L on $\phi^3(x) : x \in \{0,1\}^n$ by solving the subproblem with fixation $x_j = 1$ for $j \in \{1, \dots, n\}$ once at the beginning. Second, the cuts generated during the solution process are globally valid and we therefore add them to all nodes of the branch-and-cut tree.

5.5 BxCy

The publicly available source code of the solver SCIP [31] contains several examples that demonstrate its usage. One example provides an implementation of a Gomory mixed integer cut separator. This separator modifies the generated cuts and checks their numerical properties before adding them to the LP relaxation. Details of this can be found in Cornuéjols et al. [10]. We took this code as the basis for our implementation of lifted Gomory mixed integer cuts for BxCy.

In the implementation of BxCy, branching is primarily performed on fractional x-variables. Only if all binary variables of the master problem take integer values lifted Gomory mixed integer cuts are generated.

Let ζ^q_{current} denote the objective function value of the master LP of node q at the current iteration and ζ_{inc} be the value of the incumbent. Within a node, lifted cuts are separated only as long as $|\zeta^q_{\text{current}} - \zeta_{\text{inc}}|/|\zeta_{\text{inc}}| > \delta$ with $\delta = 1.5 \cdot 10^{-4}$ is valid.

5.6 Heuristics

For BCxy, BxLL and BxCy we use heuristics to construct a feasible solution and also to constantly improve an incumbent. After generating some initial cuts, we first run a construction heuristic that uses an optimal solution of the LP relaxation of the master problem. All x-variables that take an integer value are fixed to this value in the subproblem and fractional x-variables are fixed to 1 in the subproblem. This procedure together with constraint (8b) in the master problem ensure that the subproblem is feasible. The subproblem is then solved as a mixed binary problem. This heuristic usually finds a feasible solution fairly quickly.

At each node, we use the improvement heuristic relaxation induced neighborhood search (RINS) [14]. Unlike our construction heuristic, RINS requires not only an LP relaxation as input, but also a feasible solution. However, such a feasible solution is always present in our case due to the application of our construction heuristic. Technically, we use a variant of RINS adapted to Benders decomposition. First, the master problem is treated with RINS. Then the master solution x^* is used to define the subproblem. For the subproblem, we first solve its LP relaxation to obtain y^*. If

$$\sum_{j=1}^{n} c_j x_j^* + \sum_{i=1}^{m} \sum_{j=1}^{n} h_{ij} y_{ij}^* < \zeta_{\text{inc}} - \kappa,$$

with $\kappa = 0.1$, then the subproblem is finally solved as a mixed binary problem. It can be observed that this adaptation of RINS mostly generates good solutions at the beginning of the branching process and does not often find solutions deeper in the branch-and-cut tree.

6. Computational Study

In this section we report on computational experiments of the proposed algorithms on CFLP-SS instances. As benchmark instances we used three different test sets:

- `AB`: The first test set consists of a selection of instances from Avella and Boccia [2], where we limited ourselves to those twenty instances with 300 facilities and 300 clients.
- `OR`: The second test set is a subset of non-trivial instances from the OR-Library [28], which includes twelve instances with 100 facilities and 1000 clients.
- `GK`: The third test set is a subset of ten instances from Görtz and Klose [21]. Here we have selected the instances with 100 facilities and 200 clients as well as the instances with 100 facilities and 500 clients. In addition, we have limited ourselves to choosing the instances with a scaling factor of 10.

We created five different versions of the selected instances, in which every 10th, every 5th, every 4th, every 2nd and every client demands single-sourcing. This information is passed on to the algorithms BCxy, BxCy and BxLL by means of an additional `*.siso` file. A `*.siso` file contains a 1 or a 0 for each client, where a 1 means that single-sourcing is required for this client. We always started with the first client for the assignment of single-sourcing and all clients are treated in the order in which they appear in the instance. Together, this results in a test scope of 210 instances. Some details of the three test sets can be seen in Table 1, where n is the number of facilities, m is the number of clients, "#Cons" ist the

Test set	Single-sourcing	n	m	#Cons	#Vars	x #Bin	y #Con	y #Bin
AB	Every 10th client	300	300	90600	90300	300	81000	9000
	Every 5th client	300	300	90600	90300	300	72000	18000
	Every 4th client	300	300	90600	90300	300	67500	22500
	Every 2nd client	300	300	90600	90300	300	45000	45000
	Every client	300	300	90600	90300	300	0	90000
OR	Every 10th client	100	1000	101100	100100	100	90000	10000
	Every 5th client	100	1000	101100	100100	100	80000	20000
	Every 4th client	100	1000	101100	100100	100	75000	25000
	Every 2nd client	100	1000	101100	100100	100	50000	50000
	Every client	100	1000	101100	100100	100	0	100000
GK	Every 10th client	100	200	20300	20100	100	18000	2000
	Every 5th client	100	200	20300	20100	100	16000	4000
	Every 4th client	100	200	20300	20100	100	15000	5000
	Every 2nd client	100	200	20300	20100	100	10000	10000
	Every client	100	200	20300	20100	100	0	20000
	Every 10th client	100	500	50600	50100	100	45000	5000
	Every 5th client	100	500	50600	50100	100	40000	10000
	Every 4th client	100	500	50600	50100	100	37500	12500
	Every 2nd client	100	500	50600	50100	100	25000	25000
	Every client	100	500	50600	50100	100	0	50000

Table 1: Details about the three test sets

number of constraints, "#Vars" is the number of variables, "#Bin" is the number of binary variables and "#Con" is the number of continuous variables. The data in Table 1 refer to formulation (1a)-(1g), but without (1e).

We have compared BCxy, BxCy and BxLL with the general purpose branch-and-cut of CPLEX 20.1.0.0 and the Benders decomposition in CPLEX. CPLEX is always operated with default parameters in all tests. The CPLEX Benders decomposition framework allows for three different strategies. Strategies 1 (USER) and 2 (WORKERS) are not applicable in our case, since binary variables are present in the subproblem. Only strategy 3 (FULL) can be used, where CPLEX automatically decomposes the model, ignoring any annotations you may have supplied. It should be noted, however, that strategy 3 is also no longer applicable as soon as the problem contains only integer variables, which is the case, for example, when single-sourcing is required for each client. In the following, we denote by CPLEX the general purpose branch-and-cut algorithm of CPLEX and by CPLEX-B the Benders decomposition framework of CPLEX with strategy 3. CPLEX receives as input an LP file containing the model (1a)-(1g), but without the inequality (1e). BCxy, BxCy and BxLL get for every instance a `*.plc` file and a `*.siso` file as input. The original instance is passed by the `*.plc` file in a format that was also used in [2]. For all instances, BCxy, BxCy and BxLL used the same parameters, which were described in Section 5. We have used a time limit of 600 seconds. The computational study was performed on a machine with Intel i7–6820HQ CPU @ 2.70GHz and 8GB RAM, and operating system 64Bit Linux Ubuntu 17.10.

After an instance is read in, an LP relaxation is first solved using the Benders in-out approach as described in Section 5. The goal is not only to determine a good bound for BCxy, BxCy or BxLL quickly, but also with as few cuts as possible. A large number of remaining cuts would significantly slow down the subsequent solution process. Table 2 shows the determined bounds and runtimes of CPLEX and the Benders in-out approach on the AB instances. The two columns labeled "Opt" show the objective function values of the optimal solutions obtained and the columns labeled "T [s]" indicate the required time in seconds. The last column "#BC" contains the number of Benders cuts, which are left after the final cut purging. It can be seen that the Benders in-out approach works very efficiently compared to the dual simplex algorithm of CPLEX.

The results of the computational study over all 210 instances are presented in compact form in Table 3. The first column of this table shows the corresponding test set and the second column shows the proportion of single-sourcing. Two columns are shown for each of the five methods examined. The column labelled "GM" contains the geometric mean over all runtimes achieved for this method. Column "TO" shows the number of timeouts that occurred. If at least one timeout occurred for a method, then no geometric mean of the runtimes is shown. The two dashes for CPLEX-B indicate that this method is not applicable if single-sourcing is required for each client. The detailed results across all 210 instances are shown in Appendix A.

From Table 3 it can be seen for test set AB that BCxy, BxCy, and BxLL are comparable to CPLEX-B and superior to CPLEX up to 25 percent single-sourcing. For 50 or 100 percent single-sourcing, CPLEX generates fewer timeouts as

Instance	CPLEX Relaxation		Benders in-out		
	Opt	T [s]	Opt	T [s]	#BC
i300-1	16 292.00	17.0	16 292.00	1.8	28
i300-2	15 862.22	13.2	15 862.21	1.9	35
i300-3	15 414.95	14.1	15 414.94	2.6	34
i300-4	17 926.00	15.3	17 926.00	1.7	36
i300-5	17 962.80	16.5	17 962.80	2.3	37
i300-6	11 177.55	9.9	11 177.50	4.7	47
i300-7	11 307.61	6.7	11 307.47	3.6	29
i300-8	11 331.09	6.2	11 331.09	2.7	41
i300-9	10 787.82	5.9	10 787.81	3.8	49
i300-10	11 174.34	8.6	11 174.34	3.2	44
i300-11	9977.14	3.2	9977.11	3.0	39
i300-12	9300.66	2.6	9300.65	2.5	38
i300-13	9980.74	5.6	9980.74	3.6	43
i300-14	9669.64	4.6	9669.63	4.0	50
i300-15	9788.41	2.9	9788.41	2.8	42
i300-16	9096.07	1.7	9096.07	2.1	43
i300-17	9133.87	1.7	9133.87	3.1	46
i300-18	9533.72	3.2	9533.69	4.1	53
i300-19	9013.60	2.3	9013.60	2.7	43
i300-20	9044.75	2.3	9044.74	2.3	33

Table 2: Determination of initial cuts

Test set	Single-sourcing	CPLEX		CPLEX-B		BCxy		BxCy		BxLL	
		GM	TO	GM	TO	GM	TO	GM	TO	GM	TO
`AB`	Every 10th client	49.4	0	28.9	0	32.2	0	32.3	0	33.2	0
	Every 5th client	52.9	0	42.8	0	40.7	0	39.5	0	40.2	0
	Every 4th client	65.4	0	51.4	0	54.7	0	49.5	0	40.3	0
	Every 2nd client		1		9		7		5		3
	Every client		10	-	-		18		17		13
`OR`	Every 10th client		3	32.8	0	29.1	0	30.0	0	29.7	0
	Every 5th client		3		1	28.2	0	29.1	0	29.7	0
	Every 4th client		3		1	30.3	0	31.2	0	30.5	0
	Every 2nd client		3		3	29.1	0	29.2	0	30.7	0
	Every client		3	-	-		1	30.1	0	31.9	0
`GK`	Every 10th client	38.3	0	8.5	0	10.3	0	10.6	0	10.9	0
	Every 5th client		1	15.9	0	12.6	0	11.2	0	11.6	0
	Every 4th client		1	23.1	0	11.2	0	11.0	0	12.4	0
	Every 2nd client		2		2	12.7	0	12.4	0	12.0	0
	Every client		3	-	-		7		7	19.1	0

Table 3: Summary of results across all test sets

BCxy, BxCy and BxLL. CPLEX-B generates the most timeouts for 50 percent single-sourcing and, as already mentioned, is no longer usable for 100 percent single-sourcing. It is also apparent that on these instances BxCy and BxLL perform slightly better than BCxy.

Looking at the test set `OR`, it can be seen that CPLEX consistently has problems solving three instances. From the Tables 9 to 13 shown in Appendix A, it can be seen that the three instances are capa8000, capa10000 and capa12000. Overall, it can be concluded that BCxy, BxCy, and BxLL behave similarly, but all three variants are superior to both CPLEX and CPLEX-B.

Based on the results in Table 3, it can be seen for test set `GK` that BCxy, BxCy and BxLL provide better results than CPLEX up to 50 percent single-sourcing. Only at 100 percent single-sourcing the situation changes and CPLEX generates fewer timeouts as BCxy and BxCy. For 10 percent single-sourcing, CPLEX-B still does very well, but with increasing integrality in the subproblem, CPLEX-B deteriorates faster than BCxy, BxCy, and BxLL.

In summary, on these three test sets up to 25 percent single-sourcing BCxy, BxCy and BxLL always perform better than the general purpose branch-and-cut of CPLEX. Considering the geometric mean, speedups of 1.2 to 1.6 are achieved on the test set `AG`. For the instances with 10 percent single-sourcing of the test set `GK`, even a maximum speedup of 3.7 is reached. At 50 percent single-sourcing the results are mixed and at 100 percent single-sourcing CPLEX produces fewer timeouts. CPLEX-B has the disadvantage that an application for instances with 100 percent single-sourcing is not possible. This positive result for our proposed methods is also particularly noteworthy because no separability of the subproblem occurs.

7. Conclusion

Above we have examined the behaviour of different Benders based branch-and-cut algorithms in solving CFLP-SS with different levels of single-sourcing, namely problems in which the Benders' subproblem is a mixed integer program. The computational results indicate that for instances of CFLP-SS with up to 25% single-sourcing the different Benders variants are superior to a general purpose branch-and-cut solver. Speedups of at least 1.2 with a maximum of 3.7 can be achieved. In addition the algorithms presented give better results than the CPLEX Benders decomposition framework with strategy 3. Overall, BxCy seems to perform slightly better than BCxy and neither method dominates BxLL. The test instances contain between 2000 and 100000 0-1 subproblem variables. This suggests that other mixed integer problems of the form $\min\{cx + hy : Ax + By \geq b, x \in \{0,1\}^M, y \in \mathbb{Z}_+^{N_1} \times \mathbb{R}_+^{N_2}\}$ may be solved effectively using BxCy, BxLL or BxCy as well.

There is room for considerable improvements in the implementations. For BCxy a first step might be to test for more efficient branching rules. For BxCy, the cut that was most violated by the current node solution was used. However, it would also make sense to look at whether the cuts should still be differentiated according to their density and deviation in the magnitude of the coefficients. Currently, purging of Benders cuts separated within each node is inactive. However, purging could have a positive impact given the relatively high number of Benders cuts generated as shown in the tables. Another possibility would be the use of other lifted cutting planes in BxCy. For instance the lifting of mixed integer rounding inequalities as described in [36].

Acknowledgments

We would like to thank Matteo Fischetti, who supported us to efficiently implement Benders in-out approach to generate initial cuts.

References

[1] T. Achterberg, T. Koch and A. Martin. Branching rules revisited. *Operations Research Letters*, Volume 33, 42–54, 2005.

[2] P. Avella and M. Boccia. A cutting plane algorithm for the capacitated facility location problem. *Computational Optimization and Applications*, 43(1):39–65, 2009.

[3] W. Ben-Ameur and J. Neto. Acceleration of cutting-plane and column generation algorithms: Applications to network design. *Networks*, 49:3–17, 2007.

[4] J. F. Benders. Partitioning procedures for solving mixed variables programming problems. *Numerische Mathematik*, 4:238–252, 1962.

[5] R. Brandenberg and P. Stursberg. Refined cut selection for Benders' decompostion applied to network capacity expansion problems. *Mathematical Methods of Operations Research*, 94:383-412, 2021.

[6] C. C. Caroe and J. Tind. L-shaped decomposition of two-stage stochastic programs with integer recourse. *Mathematical Programming*, 83:451–464, 1998.

[7] G. Ceyhan, M. Köksalan and B. Lokman. Extensions for Benders cuts and new valid inequalities for solving the European day-ahead electricity market clearing problem efficiently. *European Journal of Operational Research*, vol. 300, issue 2, 713-726, 2022.

[8] M. Conforti and L. A. Wolsey. "Facet" separation with one linear program. *Mathematical Programming*, 178:361–380, 2019.

[9] J.-F. Cordeau, G. Stojkovic, F. Soumis and J. Desrosiers. Benders decomposition for simultaneous aircraft routing and crew scheduling. *Transportation Science*, 35:375–388, 2001.

[10] G. Cornuéjols, F. Margot and G. Nannicini. On the safety of Gomory cut generators. *Mathematical Programming Computation*, 5:345–395, 2013.

[11] G. Cornuéjols, R. Sridharan and J. Thizy. A comparison heuristic and relaxations for the capacitated plant location problem. *European Journal of Operational Research*, 50(3):280-297, 1991

[12] A. M. Costa. A survey on Benders' decomposition applied to fixed-charge network design problems. *Computers & OR*, 32:1429–1450, 2005.

[13] CPLEX. *IBM ILOG CPLEX Optimization Studio CPLEX User's Manual*, 2019.

[14] E. Danna, E. Rothberg and C. Le Pape. Exploring relaxation induced neighborhoods to improve MIP solutions. *Mathematical Programming*, 102(1):71–90, 2004.

[15] M. Fischetti, I. Ljubić and M. Sinnl. Benders decomposition without separability: a computational study for capacitated facility location problems. *European Journal of Operational Research*, 253:557–569, 2016.

[16] M. Fischetti, I. Ljubić and M. Sinnl. Redesigning Benders decomposition for large scale facility location. *Management Science*, 63(7):2146–2162, 2017.

[17] M. Fischetti and D. Salvagnin. An in-out approach to disjunctive optimization. In A. Lodi, M. Milano, and P. Toth, editors, *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*, pages 136–140, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

[18] M. Fischetti, D. Salvagnin and A. Zanette. A note on the selection of Benders' cuts. *Mathematical Programming, Ser. B*, 124:175–182, 2010.

[19] D. Gade, S. Küçükyavuz and S. Sen. Decomposition algorithms with parametric Gomory cuts for two-stage stochastic integer programs. *Mathematical Programming*, 144(1-2), 2014.

[20] A. M. Geoffrion and G. W. Graves. Multicommodity distribution design by Benders decomposition. *Management Science*, 20:822–844, 1974.

[21] S. Görtz and A. Klose. A simple but usually fast branch-and-bound algorithm for the capacitated facility location problem. *INFORMS Journal on Computing*, 24(4):597–610, 2012.

[22] R. E. Gomory. An algorithm for the mixed integer problem. *Rand Corporation*, (RM-2597-PR), 1960.

[23] M. Gruson, J. Cordeau and R. Jans. Benders decomposition for a stochastic three-level lot sizing and replenishment problem with a distribution structure. *European Journal of Operational Research*, vol. 291, issue 1, 206-217, 2021.

[24] J. Hooker and G. Ottosson, Logic-based Benders decomposition. *Mathematical Programming*, 96, 33–60 (2003).

[25] G. Laporte and F. V. Louveaux. The integer L-shaped method for stochastic integer programs with complete recourse. *Operations Research Letters*, 13:133–142, 1993.

[26] T. L. Magnanti and R. T. Wong. Accelerating Benders decomposition: Algorithmic enhancement and model selection criteria. *Operations Research*, 29:464–484, 1981.

[27] G. L. Nemhauser and L. A. Wolsey. A recursive procedure to generate all cuts for 0–1 mixed integer programs. *Mathematical Programming*, 46(1):379–390, 1990.

[28] OR-Library. J. E. Beasley. A collection of test data sets for a variety of Operations Research (OR) problems, `URL: http://people.brunel.ac.uk/ mastjjb/jeb/orlib/capinfo.html`, 2022.

[29] N. Papadakos. Practical enhancements to the Magnanti-Wong method. *Operations Research Letters*, 36:444–449, 2008.

[30] R. Rahmaniani, T. G. Crainic, M. Gendreau, and W. Rei. The Benders decomposition algorithm: A literature review. *European Journal of Operational Research*, 259:801–817, 2017.

[31] SCIP. K. Bestuzheva, M. Besançon, W. Chen, A. Chmiela, T. Donkiewicz, J. van Doornmalen, L. Eifler, O. Gaul, G. Gamrath, A. Gleixner, L. Gottwald, C. Graczyk, K. Halbig, A. Hoen, C. Hojny, R. van der Hulst, T. Koch, M. Lübbecke, S. J. Maher, F. Matter, E. Mühmer, B. Müller, M. E. Pfetsch, D. Rehfeldt, S. Schlein, F. Schlösser, F. Serrano, Y. Shinano, B. Sofranac, M. Turner, S. Vigerske, F. Wegscheider, P. Wellner, D. Weninger and J. Witzig. The SCIP Optimization Suite 8.0. Technical report, Optimization Online, December 2021.

[32] S. Sen and J. L. Higle. The C3 theorem and a D2 algorithm for large scale stochastic mixed-integer programming: Set convexification. *Mathematical Programming*, 104(1):1–20, 2005.

[33] S. Sen and H. D. Sherali. Decomposition with branch-and-cut approaches for two-stage stochastic mixed-integer programming. *Mathematical Programming*, 106(2):203–223, 2006.

[34] C. G. S. Sikora. Benders decomposition for the balancing of assembly lines with stochastic demand. *European Journal of Operational Research*, vol. 292, issue 1, 108-124, 2021

[35] P. Wentges. Accelerating Benders' decomposition for the capacitated facility location problem. *Mathematical Methods of Operations Research*, 44(2):267–290, 1996.

[36] D. Weninger and L. A. Wolsey. Benders' algorithm with mixed integer subproblems. CORE Discussion Paper DP2019/20, UClouvain, Louvain-la-Neuve, Belgium.

[37] L. A. Wolsey. Integer programming. John Wiley & Sons, 2021

Appendices

A. Detailed results

To be able to interpret the following tables, the identifiers of the columns should be explained first. The column named "Instance" contains the name of the instance. Columns labeled with "Opt" contain the achieved objective function value and the columns marked with "T [s]" list the runtime. The columns denoted by "#BC" (Benders cuts) list the number of Benders cuts generated in all nodes together, not counting the cuts shown in Table 2 for the initial LP relaxation. The columns labeled "#MB" (master branchings) represent the number of branchings on the x-variables. The number of branchings on the y-variables are labeled by "#SB" (subproblem branchings). The column denoted by "#LC" (lifted cuts) lists the number of lifted Gomory mixed integer cuts and the column labeled by "#LL" (Laporte-Louveaux cuts) contains the number of generated "no-good" Laporte-Louveaux cuts. If no timeout occurred for a method within one table, the geometric mean of the runtimes is displayed in the penultimate line labeled with "GM". If a timeout occurred, no geometric mean of the runtimes is displayed for this method. The last line denoted by "TO" shows the number of timeouts.

`AB` instances

	CPLEX		CPLEX-B		BCxy					BxCy					BxLL				
Instance	Opt	T [s]	Opt	T [s]	Opt	T [s]	#BC	#MB	#SB	Opt	T [s]	#BC	#MB	#LC	Opt	T [s]	#BC	#MB	#LL
i300-1	16 351.7	90.6	16 351.7	20.1	16 351.7	32.8	384	493	2	16 351.7	32.4	388	492	1	16 351.7	34.1	378	491	1
i300-2	15 951.2	204.1	15 951.3	56.2	15 951.2	120.6	1823	1047	11	15 951.2	123.9	1826	1048	5	15 951.2	121.3	1802	1042	4
i300-3	15 477.9	139.9	15 477.9	31.4	15 477.9	29.5	557	189	2	15 477.9	36.9	575	207	4	15 477.9	30.9	553	189	1
i300-4	17 991.6	129.0	17 992.1	24.0	17 991.6	37.9	691	505	1	17 991.6	38.6	691	503	1	17 991.6	39.4	686	502	1
i300-5	18 040.3	171.3	18 040.2	41.1	18 040.1	81.7	1263	761	14	18 040.1	75.1	1238	756	1	18 040.1	76.7	1226	748	1
i300-6	11 252.5	97.7	11 252.5	152.1	11 252.5	168.9	2882	841	7	11 252.5	167.2	2869	838	1	11 252.5	184.8	2866	836	1
i300-7	11 394.0	143.3	11 394.0	134.4	11 394.0	192.4	4177	1331	5	11 394.0	193.4	4168	1331	2	11 394.0	204.7	4159	1322	2
i300-8	11 378.1	15.7	11 378.1	12.5	11 378.1	16.3	246	95	3	11 378.1	15.1	239	92	1	11 378.1	16.2	238	92	1
i300-9	10 879.0	108.6	10 879.0	130.3	10 879.0	151.0	3196	814	1	10 879.0	150.7	3193	812	1	10 879.0	156.6	3192	812	1
i300-10	11 234.2	23.9	11 234.2	32.8	11 234.2	14.1	173	53	4	11 234.2	12.5	161	47	1	11 234.2	13.8	160	47	1
i300-11	10 024.7	21.5	10 024.6	25.1	10 024.6	15.4	251	73	2	10 024.6	14.6	245	70	1	10 024.6	15.2	244	70	1
i300-12	9338.9	48.9	9338.9	30.6	9338.9	22.5	419	218	2	9338.9	23.0	421	219	1	9338.9	23.8	411	212	2
i300-13	10 058.5	197.2	10 058.5	73.0	10 058.5	134.0	3335	869	0	10 058.5	134.4	3335	869	0	10 058.5	146.4	3335	869	0
i300-14	9702.6	41.9	9702.6	12.5	9702.6	26.3	374	160	6	9702.6	29.7	376	170	4	9702.6	22.8	350	148	2
i300-15	9843.9	25.8	9843.9	10.8	9843.9	14.3	201	93	1	9843.9	13.9	195	88	1	9843.9	13.9	195	90	1
i300-16	9158.8	25.7	9158.8	9.0	9158.8	7.8	126	41	0	9158.8	7.8	126	41	0	9158.8	8.2	126	41	0
i300-17	9171.8	13.0	9171.8	12.5	9171.8	35.3	732	313	0	9171.8	35.4	732	313	0	9171.8	37.1	732	313	0
i300-18	9553.7	21.1	9553.7	48.6	9553.7	18.3	248	114	0	9553.7	18.4	248	114	0	9553.7	18.8	248	114	0
i300-19	9053.7	19.0	9053.7	18.5	9053.7	13.5	223	79	0	9053.7	13.6	223	79	0	9053.7	14.3	223	79	0
i300-20	9046.3	7.7	9046.3	5.7	9046.3	3.0	5	4	0	9046.3	3.0	5	4	0	9046.3	3.7	5	4	0
GM		49.4		28.9		32.2					32.3					33.2			
TO		0		0		0					0					0			

Table 4: Comparison on the AB instances for every 10th client requesting single-sourcing.

	CPLEX		CPLEX-B		BCxy					BxCy					BxLL				
Instance	Opt	T [s]	Opt	T [s]	Opt	T [s]	#BC	#MB	#SB	Opt	T [s]	#BC	#MB	#LC	Opt	T [s]	#BC	#MB	#LL
i300-1	16 355.9	84.5	16 355.9	34.7	16 355.9	97.0	534	620	41	16 355.9	97.0	529	622	12	16 355.9	67.3	448	600	4
i300-2	15 951.8	198.2	15 952.2	192.2	15 951.8	140.5	1879	1037	25	15 951.8	142.0	1860	1041	8	15 951.8	150.7	1820	1033	4
i300-3	15 480.1	197.7	15 480.1	94.7	15 480.1	38.3	636	230	11	15 480.1	40.4	632	249	4	15 480.1	37.8	611	225	1
i300-4	17 993.5	111.2	17 993.9	39.0	17 993.5	45.7	789	582	6	17 993.5	55.2	810	604	5	17 993.5	51.2	780	581	1
i300-5	18 041.4	139.4	18 041.4	120.7	18 041.4	141.2	1494	815	103	18 041.4	102.4	1373	856	6	18 041.4	101.5	1271	788	2
i300-6	11 252.9	128.3	11 252.9	347.8	11 252.9	179.4	2977	879	9	11 252.9	178.0	2962	880	1	11 252.9	205.7	2955	872	1
i300-7	11 396.9	197.4	11 397.3	377.4	11 396.9	229.0	4726	1482	30	11 396.9	245.7	4760	1558	9	11 396.9	228.6	4635	1440	3
i300-8	11 378.1	15.5	11 378.1	16.3	11 378.1	17.1	246	95	3	11 378.1	15.7	239	92	1	11 378.1	16.8	238	92	1
i300-9	10 879.7	102.9	10 879.7	203.0	10 879.7	157.9	3305	859	3	10 879.7	159.7	3308	863	3	10 879.7	169.2	3297	855	2
i300-10	11 234.2	26.9	11 234.2	32.1	11 234.2	15.0	173	53	4	11 234.2	13.1	161	47	1	11 234.2	14.3	160	47	1
i300-11	10 025.9	40.1	10 025.9	25.3	10 025.9	18.9	271	81	7	10 025.9	16.9	259	79	2	10 025.9	17.0	253	74	1
i300-12	9338.9	31.8	9338.9	11.8	9338.9	24.2	424	220	4	9338.9	23.5	420	219	1	9338.9	23.6	411	212	2
i300-13	10 058.7	196.7	10 058.7	100.6	10 058.8	138.9	3394	865	0	10 058.8	138.6	3394	865	0	10 058.8	156.7	3394	865	0
i300-14	9702.9	51.9	9702.9	23.1	9702.9	31.8	408	182	11	9702.9	30.8	395	186	4	9702.9	27.2	362	158	2
i300-15	9845.0	30.6	9845.0	31.3	9845.0	19.3	233	108	7	9845.0	17.1	216	99	2	9845.0	17.3	209	97	2
i300-16	9159.0	26.0	9159.0	10.1	9159.0	9.2	128	44	0	9159.0	9.1	128	44	0	9159.0	10.9	128	44	0
i300-17	9171.8	17.3	9171.8	17.0	9171.8	35.8	730	313	0	9171.8	35.6	730	313	0	9171.8	41.5	730	313	0
i300-18	9555.7	22.0	9555.4	23.6	9555.4	22.7	310	150	2	9555.4	22.8	307	152	1	9555.4	24.6	301	146	1
i300-19	9054.8	24.3	9054.8	19.0	9054.8	16.5	257	88	1	9054.8	16.2	254	86	1	9054.8	18.6	253	86	1
i300-20	9048.1	5.5	9048.1	6.1	9048.1	5.9	20	16	2	9048.1	5.4	15	13	2	9048.1	5.2	11	10	1
GM		52.9		42.8		40.7					39.5					40.2			
TO		0		0		0					0					0			

Table 5: Comparison on the AB instances for every 5th client requesting single-sourcing.

	CPLEX		CPLEX-B		BCxy					BxCy					BxLL				
Instance	Opt	T [s]	Opt	T [s]	Opt	T [s]	#BC	#MB	#SB	Opt	T [s]	#BC	#MB	#LC	Opt	T [s]	#BC	#MB	#LL
i300-1	16 354.7	99.7	16 354.4	31.0	16 354.0	91.4	566	572	73	16 354.0	56.8	458	565	2	16 354.0	49.5	414	557	2
i300-2	15 955.2	222.8	15 955.2	312.2	15 955.2	297.0	2489	1363	118	15 955.2	224.2	2260	1307	20	15 955.2	168.5	2243	1344	5
i300-3	15 482.0	358.6	15 482.0	237.3	15 482.0	537.7	1402	283	350	15 482.0	142.3	759	301	24	15 482.0	60.3	700	258	2
i300-4	18 000.3	181.3	18 000.4	290.8	18 000.3	456.3	1711	967	252	18 000.3	306.2	1258	931	42	18 000.3	153.2	1227	971	5
i300-5	18 041.6	190.2	18 041.6	245.4	18 041.6	104.6	1430	845	34	18 041.6	104.6	1441	880	6	18 041.6	83.0	1357	829	2
i300-6	11 258.1	273.2	11 258.1	331.7	11 258.1	275.3	4098	1258	49	11 258.1	294.8	4161	1392	11	11 258.1	241.5	3974	1207	3
i300-7	11 395.5	212.9	11 395.5	262.4	11 395.5	200.6	4395	1345	6	11 395.5	221.5	4434	1395	5	11 395.5	199.4	4371	1333	2
i300-8	11 386.2	45.1	11 386.2	33.6	11 386.2	35.3	452	172	23	11 386.2	35.1	427	183	5	11 386.2	22.8	398	155	1
i300-9	10 879.6	90.3	10 879.6	310.9	10 879.6	184.8	3939	1008	3	10 879.6	185.0	3934	1004	2	10 879.6	186.1	3930	1003	2
i300-10	11 236.6	30.5	11 236.6	31.6	11 236.6	20.2	215	60	11	11 236.6	28.8	211	66	12	11 236.6	13.9	183	47	1
i300-11	10 025.3	16.7	10 025.3	15.9	10 025.4	15.8	259	72	1	10 025.4	15.6	256	70	1	10 025.4	15.4	255	70	1
i300-12	9337.0	45.4	9337.0	15.3	9337.0	17.2	337	165	0	9337.0	17.4	337	165	0	9337.0	17.8	337	165	0
i300-13	10 059.9	65.0	10 059.9	242.9	10 059.9	148.8	3567	907	3	10 059.9	145.4	3558	904	1	10 059.9	147.3	3556	902	1
i300-14	9702.7	64.9	9702.7	21.5	9702.7	28.2	389	167	7	9702.7	30.4	392	174	4	9702.7	24.2	366	156	2
i300-15	9846.8	50.4	9846.8	17.7	9846.8	21.7	255	122	7	9846.8	22.3	249	125	4	9846.8	17.3	231	110	2
i300-16	9159.7	19.0	9159.7	9.7	9159.8	11.3	144	56	2	9159.8	10.5	140	54	1	9159.7	10.4	137	53	2
i300-17	9172.1	37.4	9172.1	16.8	9172.1	36.6	767	305	0	9172.1	38.6	767	305	0	9172.1	37.3	767	305	0
i300-18	9555.4	24.7	9555.4	14.5	9555.4	26.3	327	162	7	9555.4	24.0	311	154	3	9555.4	21.7	301	146	1
i300-19	9053.7	46.1	9053.7	13.7	9053.7	13.6	223	79	0	9053.7	13.7	223	79	0	9053.7	14.1	223	79	0
i300-20	9046.4	8.0	9046.4	4.9	9046.4	3.1	5	4	0	9046.4	3.0	5	4	0	9046.4	3.5	5	4	0
GM		65.4		51.4		54.7					49.5					40.3			
TO		0		0		0					0					0			

Table 6: Comparison on the AB instances for every 4th client requesting single-sourcing.

	CPLEX		CPLEX-B		BCxy					BxCy					BxLL				
Instance	Opt	T [s]	Opt	T [s]	Opt	T [s]	#BC	#MB	#SB	Opt	T [s]	#BC	#MB	#LC	Opt	T [s]	#BC	#MB	#LL
i300-1	16 366.5	155.0	16 366.7	600.0	16 366.4	600.2	998	672	500	16 366.4	507.4	958	811	70	16 366.4	133.2	633	812	6
i300-2	15 963.3	364.6	15 966.8	600.0	15 963.3	601.4	2631	1518	104	15 963.3	601.4	2502	1495	26	15 963.3	600.3	2959	1886	7
i300-3	15 501.1	335.8	15 511.3	600.0	15 505.2	602.6	1188	445	102	15 505.2	601.7	1448	751	22	15 501.1	504.4	2108	1306	8
i300-4	18 014.1	220.5	18 023.9	600.0	18 014.0	600.3	1339	950	77	18 014.0	600.9	1315	973	22	18 014.0	600.7	1431	1091	5
i300-5	18 059.0	306.5	18 070.9	600.0	18 058.7	600.3	2347	1357	264	18 058.7	601.0	2603	1855	30	18 058.7	600.0	3395	2446	8
i300-6	11 269.2	600.0	11 284.5	600.0	11 268.5	600.3	6538	2075	196	11 268.5	600.0	6775	2582	25	11 268.5	458.0	6993	2216	5
i300-7	11 398.1	269.2	11 408.6	600.0	11 398.1	307.8	4814	1537	97	11 398.1	303.2	4725	1621	17	11 398.1	272.0	4548	1435	4
i300-8	11 395.7	32.7	11 395.4	327.8	11 395.4	600.3	1643	557	765	11 395.4	107.1	725	340	20	11 395.4	46.5	614	259	3
i300-9	10 883.3	164.6	10 896.4	600.0	10 883.3	249.8	4099	1086	56	10 883.3	240.3	4032	1126	11	10 883.3	197.5	3958	1051	2
i300-10	11 240.8	48.7	11 240.8	124.5	11 240.8	214.9	938	241	303	11 240.8	41.1	277	108	13	11 240.8	20.5	244	83	1
i300-11	10 027.2	11.5	10 027.2	50.0	10 027.3	45.9	385	122	32	10 027.3	27.2	312	97	7	10 027.3	18.4	290	79	1
i300-12	9340.3	23.2	9340.3	26.9	9340.3	43.5	519	262	20	9340.3	29.4	452	223	2	9340.3	27.2	442	217	3
i300-13	10 061.8	132.1	10 078.7	600.0	10 061.8	169.7	3776	980	17	10 061.8	180.5	3796	1026	9	10 061.8	161.1	3738	969	2
i300-14	9702.9	40.9	9702.9	32.0	9702.9	33.3	405	168	13	9702.9	36.3	402	183	6	9702.9	25.2	370	154	2
i300-15	9851.6	23.9	9851.6	51.6	9851.6	65.9	482	239	40	9851.6	41.7	371	190	7	9851.6	29.5	337	163	6
i300-16	9161.8	17.0	9161.8	17.8	9161.8	26.0	214	89	15	9161.8	16.9	173	69	4	9161.8	12.4	151	51	3
i300-17	9173.3	19.9	9173.3	26.3	9173.3	46.4	859	343	6	9173.3	45.5	851	347	3	9173.3	42.5	839	336	2
i300-18	9557.4	41.0	9557.1	33.6	9557.1	38.1	414	177	25	9557.1	32.1	374	187	5	9557.1	25.0	359	172	1
i300-19	9053.8	18.9	9053.8	16.2	9054.1	16.3	251	85	0	9054.1	16.2	251	85	0	9054.1	19.1	251	85	0
i300-20	9047.0	7.8	9047.0	9.5	9047.0	7.4	24	19	3	9047.0	5.0	11	10	1	9047.0	4.8	10	10	1
GM																			
TO		1		9		7					5					3			

Table 7: Comparison on the AB instances for every 2nd client requesting single-sourcing.

	CPLEX		**BCxy**					**BxCy**					**BxLL**				
Instance	Opt	T [s]	Opt	T [s]	#BC	#MB	#SB	Opt	T [s]	#BC	#MB	#LC	Opt	T [s]	#BC	#MB	#LL
i300-1	16 730.7	600.0	16 881.3	602.8	474	198	0	16 881.3	601.4	474	198	0	16 881.3	600.3	474	197	0
i300-2	16 466.4	600.0	17 122.0	602.9	532	192	0	17 122.0	602.1	532	192	0	17 122.0	602.6	531	191	0
i300-3	15 888.8	600.0	17 452.3	601.1	507	191	0	17 452.3	601.1	511	192	0	17 452.3	602.5	511	192	0
i300-4	18 747.7	600.0	19 560.1	602.6	405	194	0	19 560.1	602.0	405	194	0	19 560.1	600.1	404	193	0
i300-5	18 529.7	600.0	20 610.5	600.7	484	192	0	20 610.5	602.9	486	193	0	20 610.5	600.9	484	192	0
i300-6	11 350.5	600.0	11 326.4	600.4	3033	816	0	11 326.4	602.4	3051	818	0	11 326.4	601.1	3033	816	0
i300-7	11 490.7	600.0	11 472.4	602.6	2042	558	0	11 472.4	601.4	2068	565	0	11 472.4	601.7	2043	559	0
i300-8	11 450.0	530.9	11 449.7	600.5	1140	366	147	11 449.7	601.2	1197	467	9	11 449.7	600.0	1001	307	1
i300-9	10 954.2	600.0	10 932.9	600.0	4167	1023	0	10 932.9	602.2	4228	1042	0	10 932.9	600.8	4287	1055	0
i300-10	11 324.8	600.0	11 331.3	600.7	602	177	51	11 331.3	600.1	782	251	11	11 324.3	600.2	952	256	4
i300-11	10 047.4	19.9	10 046.9	600.3	1320	617	462	10 046.9	600.8	742	293	100	10 046.9	61.9	616	218	11
i300-12	9360.0	149.5	9359.6	600.2	1929	890	519	9359.6	601.5	1405	675	104	9359.6	122.5	1348	595	26
i300-13	10 112.0	600.0	10 103.5	600.6	5579	1258	0	10 103.5	602.9	5429	1302	1	10 103.5	600.2	5532	1299	1
i300-14	9738.0	152.9	9738.0	600.2	1593	589	361	9738.0	600.3	2476	982	41	9738.0	600.4	2491	983	48
i300-15	9902.3	50.1	9902.3	600.4	1675	640	275	9902.3	604.1	2238	896	41	9902.3	600.1	2560	1102	81
i300-16	9168.1	21.1	9168.1	550.3	1838	672	613	9168.1	31.0	211	95	10	9168.1	16.3	167	62	5
i300-17	9181.1	22.8	9181.1	600.1	2902	1156	763	9181.1	140.5	1686	669	21	9181.1	91.6	1556	566	9
i300-18	9582.0	64.0	9582.0	600.3	2554	976	793	9582.0	603.7	3011	1128	91	9582.0	262.9	2834	1097	36
i300-19	9062.5	27.5	9062.2	450.5	1789	598	501	9062.2	57.8	544	216	10	9062.2	37.8	497	180	4
i300-20	9077.9	11.8	9077.9	600.2	1178	438	703	9077.9	604.7	299	114	117	9077.9	38.0	256	130	10
GM																	
TO		10		18					17					13			

Table 8: Comparison on the `AB` instances for every client requesting single-sourcing.

`OR` instances

	CPLEX		CPLEX-B		BCxy					BxCy					BxLL				
Instance	Opt	T [s]	Opt	T [s]	Opt	T [s]	#BC	#MB	#SB	Opt	T [s]	#BC	#MB	#LC	Opt	T [s]	#BC	#MB	#LL
capa8000	19 241 444.1	600.0	19 240 811.8	37.1	19 240 811.8	164.2	3162	1045	0	19 240 811.8	168.7	3162	1045	0	19 240 811.8	170.0	3162	1045	0
capa10000	18 438 030.4	600.0	18 438 030.4	56.7	18 438 030.4	402.5	7605	5867	0	18 438 030.4	433.7	7605	5867	0	18 438 030.4	402.6	7605	5867	0
capa12000	17 765 189.3	600.1	17 765 189.3	37.5	17 765 189.3	26.4	462	237	0	17 765 189.3	27.3	462	237	0	17 765 189.3	27.6	462	237	0
capa14000	17 160 424.7	13.3	17 160 424.7	4.1	17 160 424.7	2.7	0	0	0	17 160 424.7	2.8	0	0	0	17 160 424.7	3.5	0	0	0
capb5000	13 656 379.2	314.3	13 656 379.2	52.6	13 656 379.2	46.5	505	161	0	13 656 379.2	50.1	505	161	0	13 656 379.2	45.8	505	161	0
capb6000	13 361 927.6	381.8	13 361 927.6	143.8	13 361 927.6	64.8	831	245	0	13 361 927.6	67.9	831	245	0	13 361 927.6	63.2	831	245	0
capb7000	13 198 555.9	266.5	13 198 555.9	91.0	13 198 555.9	45.3	599	186	0	13 198 555.9	49.9	599	186	0	13 198 555.9	42.2	599	186	0
capb8000	13 082 517.2	61.7	13 082 517.2	10.0	13 082 517.2	9.6	24	18	0	13 082 517.2	9.7	24	18	0	13 082 517.2	9.3	24	18	0
capc5000	11 646 596.4	210.2	11 646 596.4	267.0	11 646 596.4	45.5	651	189	0	11 646 596.4	48.5	651	189	0	11 646 596.4	42.5	651	189	0
capc5750	11 570 339.7	97.4	11 570 339.7	68.1	11 570 339.7	37.1	520	186	0	11 570 339.7	35.9	520	186	0	11 570 339.7	34.1	520	186	0
capc6500	11 518 743.2	35.9	11 518 743.2	8.5	11 518 743.2	8.5	23	18	0	11 518 743.2	8.3	23	18	0	11 518 743.2	10.0	23	18	0
capc10000	11 505 593.7	10.9	11 505 593.7	4.5	11 505 593.7	4.2	7	4	0	11 505 593.7	4.1	7	4	0	11 505 593.7	4.3	7	4	0
GM				32.8		29.1					30.0					29.7			
TO		3		0		0					0					0			

Table 9: Comparison on the `OR` instances for every 10th client requesting single-sourcing.

	CPLEX		CPLEX-B		BCxy					BxCy					BxLL				
Instance	Opt	T [s]	Opt	T [s]	Opt	T [s]	#BC	#MB	#SB	Opt	T [s]	#BC	#MB	#LC	Opt	T [s]	#BC	#MB	#LL
capa8000	19 242 030.7	600.0	19 252 027.1	600.0	19 240 811.8	166.1	3162	1045	0	19 240 811.8	166.4	3162	1045	0	19 240 811.8	168.4	3162	1045	0
capa10000	18 438 145.1	600.0	18 438 030.4	145.7	18 438 030.4	403.1	7605	5867	0	18 438 030.4	403.2	7605	5867	0	18 438 030.4	409.2	7605	5867	0
capa12000	17 765 189.3	600.1	17 765 189.3	42.6	17 765 189.3	26.4	462	237	0	17 765 189.3	27.2	462	237	0	17 765 189.3	27.2	462	237	0
capa14000	17 160 424.7	12.1	17 160 424.7	7.1	17 160 424.7	2.7	0	0	0	17 160 424.7	2.8	0	0	0	17 160 424.7	3.4	0	0	0
capb5000	13 656 379.2	303.4	13 656 379.2	195.7	13 656 379.2	46.7	505	161	0	13 656 379.2	47.3	505	161	0	13 656 379.2	48.2	505	161	0
capb6000	13 361 927.6	264.2	13 361 927.6	504.7	13 361 927.6	63.3	831	245	0	13 361 927.6	66.0	831	245	0	13 361 927.6	64.6	831	245	0
capb7000	13 198 566.5	213.5	13 198 566.5	58.3	13 198 566.5	43.0	599	186	0	13 198 566.5	45.8	599	186	0	13 198 566.5	42.9	599	186	0
capb8000	13 082 517.2	56.7	13 082 517.2	17.0	13 082 517.2	9.0	24	18	0	13 082 517.2	10.2	24	18	0	13 082 517.2	9.7	24	18	0
capc5000	11 646 596.4	219.3	11 646 596.4	32.2	11 646 596.4	42.0	651	189	0	11 646 596.4	44.3	651	189	0	11 646 596.4	42.7	651	189	0
capc5750	11 570 339.7	91.7	11 570 339.7	38.9	11 570 339.7	34.4	520	186	0	11 570 339.7	35.1	520	186	0	11 570 339.7	34.8	520	186	0
capc6500	11 518 743.2	26.3	11 518 743.2	11.2	11 518 743.2	8.0	23	18	0	11 518 743.2	8.0	23	18	0	11 518 743.2	8.5	23	18	0
capc10000	11 505 593.7	11.3	11 505 593.7	5.9	11 505 593.7	4.1	7	4	0	11 505 593.7	4.0	7	4	0	11 505 593.7	4.5	7	4	0
GM						28.2					29.1					29.7			
TO		3		1		0					0					0			

Table 10: Comparison on the `OR` instances for every 5th client requesting single-sourcing.

	CPLEX		CPLEX-B		BCxy					BxCy					BxLL				
Instance	Opt	T [s]	Opt	T [s]	Opt	T [s]	#BC	#MB	#SB	Opt	T [s]	#BC	#MB	#LC	Opt	T [s]	#BC	#MB	#LL
capa8000	19 262 899.5	600.0	19 240 811.8	235.6	19 240 811.8	176.5	3162	1045	0	19 240 811.8	171.7	3162	1045	0	19 240 811.8	175.5	3162	1045	0
capa10000	18 438 030.4	600.1	18 441 663.7	600.0	18 438 030.4	410.8	7605	5867	0	18 438 030.4	514.5	7605	5867	0	18 438 030.4	391.1	7605	5867	0
capa12000	17 765 189.3	600.0	17 765 189.3	494.9	17 765 189.3	26.1	462	237	0	17 765 189.3	29.2	462	237	0	17 765 189.3	26.9	462	237	0
capa14000	17 160 509.1	11.4	17 160 850.4	12.0	17 160 509.1	2.8	0	0	0	17 160 509.1	3.1	0	0	0	17 160 509.1	3.4	0	0	0
capb5000	13 656 490.2	286.7	13 656 490.2	246.4	13 656 490.2	46.7	499	158	0	13 656 490.2	46.8	499	158	0	13 656 490.2	46.1	499	158	0
capb6000	13 361 976.2	374.6	13 361 964.4	140.8	13 362 003.8	69.7	834	247	0	13 362 003.8	66.1	834	247	0	13 362 003.8	65.1	834	247	0
capb7000	13 198 555.9	277.8	13 198 555.9	77.2	13 198 555.9	46.5	599	186	0	13 198 555.9	44.3	599	186	0	13 198 555.9	42.3	599	186	0
capb8000	13 082 531.7	114.4	13 082 531.7	14.7	13 082 531.7	10.9	24	18	0	13 082 531.7	10.5	24	18	0	13 082 531.7	11.5	24	18	0
capc5000	11 646 726.4	309.1	11 646 628.4	282.4	11 646 729.7	45.6	649	188	0	11 646 729.7	45.1	649	188	0	11 646 729.7	45.1	649	188	0
capc5750	11 570 339.7	159.5	11 570 339.7	54.4	11 570 339.7	35.7	520	186	0	11 570 339.7	41.7	520	186	0	11 570 339.7	34.4	520	186	0
capc6500	11 518 747.9	30.0	11 518 747.9	13.3	11 518 747.9	9.6	23	18	0	11 518 747.9	8.9	23	18	0	11 518 747.9	9.3	23	18	0
capc10000	11 505 593.7	26.2	11 505 593.7	6.6	11 505 593.7	4.4	7	4	0	11 505 593.7	4.5	7	4	0	11 505 593.7	4.9	7	4	0
GM						30.3					31.2					30.5			
TO		3		1		0					0					0			

Table 11: Comparison on the `OR` instances for every 4th client requesting single-sourcing.

	CPLEX		CPLEX-B		BCxy					BxCy					BxLL				
Instance	Opt	T [s]	Opt	T [s]	Opt	T [s]	#BC	#MB	#SB	Opt	T [s]	#BC	#MB	#LC	Opt	T [s]	#BC	#MB	#LL
capa8000	19 293 470.6	600.0	19 289 253.2	600.0	19 240 811.8	162.2	3162	1045	0	19 240 811.8	162.2	3162	1045	0	19 240 811.8	164.1	3162	1045	0
capa10000	18 439 034.8	600.1	18 438 065.8	123.5	18 438 054.4	390.8	7549	5824	0	18 438 054.4	391.0	7549	5824	0	18 438 054.4	415.7	7549	5824	0
capa12000	17 765 189.3	600.1	17 770 121.4	600.0	17 765 189.3	26.3	462	237	0	17 765 189.3	26.2	462	237	0	17 765 189.3	26.9	462	237	0
capa14000	17 160 524.4	14.4	17 160 524.4	12.3	17 160 524.4	2.9	0	0	0	17 160 524.4	2.9	0	0	0	17 160 524.4	3.5	0	0	0
capb5000	13 656 697.7	333.6	13 656 697.7	111.2	13 656 692.8	49.1	514	160	0	13 656 692.8	49.0	514	160	0	13 656 692.8	48.5	514	160	0
capb6000	13 362 037.3	323.7	13 361 977.8	248.9	13 362 285.0	65.5	846	246	0	13 362 285.0	65.4	846	246	0	13 362 285.0	66.3	846	246	0
capb7000	13 198 628.4	161.1	13 199 305.3	171.1	13 198 628.4	42.3	600	186	0	13 198 628.4	42.3	600	186	0	13 198 628.4	43.4	600	186	0
capb8000	13 082 748.6	51.1	13 083 276.9	31.8	13 082 753.6	10.3	25	18	0	13 082 753.6	10.4	25	18	0	13 082 753.6	11.3	25	18	0
capc5000	11 646 747.5	234.4	11 651 470.9	600.0	11 646 754.5	43.9	649	188	0	11 646 754.5	43.6	649	188	0	11 646 754.5	44.8	649	188	0
capc5750	11 570 339.7	79.5	11 570 339.7	71.9	11 570 339.7	33.9	520	186	0	11 570 339.7	33.9	520	186	0	11 570 339.7	34.6	520	186	0
capc6500	11 518 844.8	34.5	11 518 844.8	25.5	11 518 844.8	9.0	23	18	0	11 518 844.8	8.9	23	18	0	11 518 844.8	9.5	23	18	0
capc10000	11 505 593.7	20.6	11 505 593.7	11.4	11 505 593.7	4.2	7	4	0	11 505 593.7	4.2	7	4	0	11 505 593.7	4.6	7	4	0
GM						29.1					29.2					30.7			
TO		3		3		0					0					0			

Table 12: Comparison on the `OR` instances for every 2nd client requesting single-sourcing.

	CPLEX		BCxy					BxCy					BxLL				
Instance	Opt	T [s]	Opt	T [s]	#BC	#MB	#SB	Opt	T [s]	#BC	#MB	#LC	Opt	T [s]	#BC	#MB	#LL
capa8000	19 292 387.2	600.0	19 241 046.3	163.1	3163	1046	0	19 241 046.3	162.7	3163	1046	0	19 241 046.3	181.5	3163	1046	0
capa10000	18 538 339.8	600.0	18 438 593.9	420.7	7582	5852	0	18 438 593.9	390.7	7582	5852	0	18 438 593.9	406.4	7582	5852	0
capa12000	17 765 189.3	600.0	17 765 189.3	28.6	462	237	0	17 765 189.3	26.2	462	237	0	17 765 189.3	27.2	462	237	0
capa14000	17 160 903.5	14.2	17 160 801.2	2.9	0	0	0	17 160 801.2	2.9	0	0	0	17 160 801.2	3.8	0	0	0
capb5000	13 657 904.1	296.1	13 657 834.2	285.9	1181	178	322	13 657 834.2	51.8	533	167	1	13 657 693.5	51.1	509	158	1
capb6000	13 362 795.6	281.4	13 362 626.4	61.0	690	203	4	13 362 626.4	58.0	681	197	1	13 362 626.4	58.4	680	197	1
capb7000	13 199 667.1	205.8	13 199 598.8	90.0	755	212	62	13 199 598.8	44.9	623	199	1	13 199 598.8	45.5	622	199	1
capb8000	13 083 601.6	107.2	13 083 836.0	600.3	1660	141	978	13 083 836.0	12.1	30	21	1	13 083 836.0	12.3	29	21	1
capc5000	11 647 831.2	250.4	11 647 416.3	52.4	672	206	8	11 647 416.3	45.2	653	194	1	11 647 416.3	46.0	652	194	1
capc5750	11 570 437.1	90.4	11 570 437.1	37.7	513	188	0	11 570 437.1	35.6	513	188	0	11 570 437.1	36.4	513	188	0
capc6500	11 519 316.8	45.2	11 519 418.3	14.5	42	31	5	11 519 418.6	10.5	29	26	1	11 519 418.6	10.8	28	26	1
capc10000	11 505 593.7	16.9	11 505 593.7	4.1	7	4	0	11 505 593.7	4.1	7	4	0	11 505 593.7	4.9	7	4	0
GM									30.1					31.9			
TO		3		1					0					0			

Table 13: Comparison on the `OR` instances for every client requesting single-sourcing.

GK instances

	CPLEX		CPLEX-B		BCxy					BxCy					BxLL				
Instance	Opt	T [s]	Opt	T [s]	Opt	T [s]	#BC	#MB	#SB	Opt	T [s]	#BC	#MB	#LC	Opt	T [s]	#BC	#MB	#LL
T200x100-10-1	13 997.7	35.3	13 997.7	5.1	13 997.7	6.9	419	192	0	13 997.7	6.9	419	192	0	13 997.7	7.0	419	192	0
T200x100-10-2	14 231.7	17.4	14 231.7	3.7	14 231.7	3.0	120	31	0	14 231.7	3.0	120	31	0	14 231.7	3.1	120	31	0
T200x100-10-3	13 903.3	7.5	13 903.3	1.1	13 903.3	0.7	0	0	0	13 903.3	0.7	0	0	0	13 903.3	0.7	0	0	0
T200x100-10-4	14 091.5	55.2	14 091.5	7.0	14 091.5	6.7	479	245	0	14 091.5	6.7	479	245	0	14 091.5	6.9	479	245	0
T200x100-10-5	14 044.5	44.9	14 044.5	6.3	14 044.5	8.2	380	210	0	14 044.5	8.3	380	210	0	14 044.5	8.3	380	210	0
T500x100-10-1	23 458.4	92.4	23 458.4	88.9	23 458.4	227.6	5920	1635	0	23 458.4	261.9	5920	1635	0	23 458.4	248.0	5920	1635	0
T500x100-10-2	23 254.9	10.3	23 254.9	3.5	23 254.9	3.1	1	0	0	23 254.9	3.1	1	0	0	23 254.9	3.2	1	0	0
T500x100-10-3	23 544.7	307.0	23 544.7	77.9	23 544.7	98.4	2577	918	0	23 544.7	107.4	2577	918	0	23 544.7	110.0	2577	918	0
T500x100-10-4	22 883.9	42.2	22 883.9	11.7	22 883.9	18.8	420	159	0	22 883.9	19.4	420	159	0	22 883.9	21.2	420	159	0
T500x100-10-5	22 489.5	48.0	22 489.5	7.6	22 489.5	13.3	141	63	0	22 489.5	14.3	141	63	0	22 489.5	14.4	141	63	0
GM		38.3		8.5		10.3					10.6					10.9			
TO		0		0		0					0					0			

Table 14: Comparison on the GK instances for every 10th client requesting single-sourcing.

	CPLEX		CPLEX-B		BCxy					BxCy					BxLL				
Instance	Opt	T [s]	Opt	T [s]	Opt	T [s]	#BC	#MB	#SB	Opt	T [s]	#BC	#MB	#LC	Opt	T [s]	#BC	#MB	#LL
T200x100-10-1	13 997.7	27.9	13 997.7	21.9	13 997.7	6.8	419	193	0	13 997.7	6.9	419	193	0	13 997.7	7.8	419	193	0
T200x100-10-2	14 231.7	15.7	14 231.7	7.3	14 231.7	3.1	120	31	0	14 231.7	3.0	120	31	0	14 231.7	3.4	120	31	0
T200x100-10-3	13 904.0	5.2	13 904.0	2.0	13 904.0	0.7	0	0	0	13 904.0	0.7	0	0	0	13 904.0	0.9	0	0	0
T200x100-10-4	14 091.5	59.0	14 091.5	16.5	14 091.5	6.7	479	245	0	14 091.5	6.8	479	245	0	14 091.5	7.5	479	245	0
T200x100-10-5	14 046.1	44.5	14 045.3	7.3	14 045.3	8.5	384	211	1	14 045.3	8.6	383	211	1	14 045.3	8.8	382	211	1
T500x100-10-1	23 458.6	89.3	23 460.0	153.3	23 458.6	305.1	6228	1664	0	23 458.6	269.0	6228	1664	0	23 458.6	259.7	6228	1664	0
T500x100-10-2	23 254.9	9.7	23 254.9	5.1	23 254.9	4.5	1	0	0	23 254.9	3.2	1	0	0	23 254.9	3.5	1	0	0
T500x100-10-3	23 581.2	600.0	23 545.2	244.0	23 545.2	150.4	2608	921	0	23 545.2	116.9	2608	921	0	23 545.2	106.3	2608	921	0
T500x100-10-4	22 883.9	53.8	22 883.9	13.0	22 883.9	27.7	420	159	0	22 883.9	23.7	420	159	0	22 883.9	20.1	420	159	0
T500x100-10-5	22 490.3	56.3	22 490.3	10.9	22 490.3	21.7	146	71	0	22 490.3	16.8	146	71	0	22 490.3	15.8	146	71	0
GM				15.9		12.6					11.2					11.6			
TO		1		0		0					0					0			

Table 15: Comparison on the GK instances for every 5th client requesting single-sourcing.

	CPLEX		CPLEX-B		BCxy					BxCy					BxLL				
Instance	Opt	T [s]	Opt	T [s]	Opt	T [s]	#BC	#MB	#SB	Opt	T [s]	#BC	#MB	#LC	Opt	T [s]	#BC	#MB	#LL
T200x100-10-1	14 001.0	32.1	14 001.0	20.7	14 001.0	7.7	465	215	2	14 001.0	7.9	475	228	1	14 001.0	8.1	457	212	1
T200x100-10-2	14 231.7	16.3	14 231.7	5.4	14 231.7	3.0	120	31	0	14 231.7	3.0	120	31	0	14 231.7	3.5	120	31	0
T200x100-10-3	13 902.7	6.0	13 902.7	2.2	13 902.7	0.6	0	0	0	13 902.7	0.6	0	0	0	13 902.7	0.8	0	0	0
T200x100-10-4	14 091.5	286.0	14 091.5	27.4	14 091.5	7.1	479	245	0	14 091.5	6.8	479	245	0	14 091.5	7.6	479	245	0
T200x100-10-5	14 045.0	53.3	14 045.0	15.9	14 045.0	9.0	382	211	0	14 045.0	8.2	382	211	0	14 045.0	9.0	382	211	0
T500x100-10-1	23 458.4	86.5	23 458.4	350.7	23 458.4	239.2	6169	1628	0	23 458.4	240.3	6169	1628	0	23 458.4	276.2	6169	1628	0
T500x100-10-2	23 255.7	9.9	23 255.7	6.9	23 255.7	4.4	11	8	1	23 255.7	4.2	10	8	1	23 255.7	4.6	9	8	1
T500x100-10-3	23 545.2	600.0	23 545.2	263.1	23 545.2	108.9	2695	973	0	23 545.2	108.4	2695	973	0	23 545.2	128.4	2695	973	0
T500x100-10-4	22 884.2	35.4	22 884.2	33.8	22 884.2	19.7	417	158	0	22 884.2	19.8	417	158	0	22 884.2	22.3	417	158	0
T500x100-10-5	22 489.7	50.3	22 489.7	18.9	22 489.7	14.3	143	63	0	22 489.7	14.3	143	63	0	22 489.7	15.8	143	63	0
GM				23.1		11.2					11.0					12.4			
TO		1		0		0					0					0			

Table 16: Comparison on the GK instances for every 4th client requesting single-sourcing.

	CPLEX		CPLEX-B		BCxy					BxCy					BxLL				
Instance	Opt	T [s]	Opt	T [s]	Opt	T [s]	#BC	#MB	#SB	Opt	T [s]	#BC	#MB	#LC	Opt	T [s]	#BC	#MB	#LL
T200x100-10-1	14 001.5	39.1	14 001.5	104.8	14 001.5	8.5	483	219	8	14 001.5	9.0	481	233	2	14 001.5	8.2	463	216	1
T200x100-10-2	14 234.5	18.9	14 234.1	16.2	14 234.4	3.9	141	38	4	14 234.4	3.7	141	41	2	14 234.4	3.5	133	35	1
T200x100-10-3	13 903.8	5.5	13 903.8	6.9	13 903.8	0.7	0	0	0	13 903.8	0.7	0	0	0	13 903.8	0.8	0	0	0
T200x100-10-4	14 092.3	600.0	14 092.4	150.1	14 092.4	7.3	498	281	1	14 092.4	7.7	505	286	1	14 092.4	7.3	493	281	1
T200x100-10-5	14 046.4	222.1	14 046.4	63.6	14 046.4	8.8	413	207	3	14 046.4	8.7	409	208	1	14 046.4	8.6	407	208	1
T500x100-10-1	23 459.0	79.2	23 460.3	600.0	23 459.8	283.9	6275	1666	5	23 459.8	278.2	6260	1660	1	23 459.8	237.6	6259	1660	1
T500x100-10-2	23 256.2	11.0	23 255.7	25.6	23 255.7	4.5	11	8	1	23 255.7	4.3	10	8	1	23 255.7	4.7	9	8	1
T500x100-10-3	23 730.4	600.0	23 605.2	600.0	23 545.4	127.9	2683	963	0	23 545.4	110.0	2683	963	0	23 545.4	106.8	2683	963	0
T500x100-10-4	22 884.2	37.6	22 884.2	50.7	22 884.2	22.3	417	158	0	22 884.2	24.0	417	158	0	22 884.2	19.9	417	158	0
T500x100-10-5	22 492.3	45.0	22 492.5	45.6	22 492.4	19.9	175	81	2	22 492.4	18.3	169	76	2	22 492.4	19.0	167	76	2
GM						12.7					12.4					12.0			
TO		2		2		0					0					0			

Table 17: Comparison on the GK instances for every 2nd client requesting single-sourcing.

	CPLEX		BCxy					BxCy					BxLL				
Instance	Opt	T [s]	Opt	T [s]	#BC	#MB	#SB	Opt	T [s]	#BC	#MB	#LC	Opt	T [s]	#BC	#MB	#LL
T200x100-10-1	14 008.3	383.4	14 008.3	298.5	3041	566	1133	14 008.3	600.7	808	266	473	14 008.3	11.8	546	245	4
T200x100-10-2	14 248.3	242.7	14 248.3	600.0	7205	819	3976	14 248.3	600.1	645	83	464	14 248.3	5.1	183	54	1
T200x100-10-3	13 921.9	5.8	13 922.6	0.8	0	0	0	13 922.6	0.7	0	0	0	13 922.6	0.9	0	0	0
T200x100-10-4	14 102.4	600.0	14 102.4	53.1	1112	397	255	14 102.4	12.1	645	386	5	14 102.4	9.6	588	367	1
T200x100-10-5	14 073.0	600.0	14 060.8	600.1	5335	337	3908	14 060.8	600.8	996	283	485	14 060.8	12.0	541	267	1
T500x100-10-1	23 473.9	182.9	23 473.9	600.2	9332	2209	1075	23 473.9	601.4	8537	2353	102	23 473.9	416.4	8487	2215	3
T500x100-10-2	23 274.3	29.6	23 274.2	600.1	1977	251	1631	23 274.2	601.3	273	73	204	23 274.2	8.0	29	22	2
T500x100-10-3	23 733.4	600.0	23 568.6	600.1	5357	1524	943	23 568.6	602.9	4642	1548	136	23 568.5	243.4	4655	1438	2
T500x100-10-4	22 891.9	53.3	22 892.2	600.1	2921	234	2142	22 892.2	49.0	607	224	11	22 892.2	38.7	577	198	1
T500x100-10-5	22 502.9	209.3	22 502.9	600.3	2027	188	1709	22 502.9	605.4	477	175	168	22 502.9	31.0	275	126	2
GM														19.1			
TO		3		7					7					0			

Table 18: Comparison on the GK instances for every client requesting single-sourcing.