Annals of Operations Research

Improving Benders decomposition via a non-linear cut selection procedure

--Manuscript Draft--

Manuscript Number:	ANOR-D-16-00238
Full Title:	Improving Benders decomposition via a non-linear cut selection procedure
Article Type:	Original Research
Keywords:	Benders decomposition; Lifting procedure; Stronger Benders cuts; Large scale optimization.
Corresponding Author:	Gilberto Miranda, Ph.D. Universidade Federal do Espirito Santo Sâo Mateus, Espírito Santo BRAZIL
Corresponding Author Secondary Information:	
Corresponding Author's Institution:	Universidade Federal do Espirito Santo
Corresponding Author's Secondary Institution:	
First Author:	Gilberto Miranda, Ph.D.
First Author Secondary Information:	
Order of Authors:	Gilberto Miranda, Ph.D.
	Ricardo Camargo, Ph.D.
	Fátima Lima, Ph.D.
Order of Authors Secondary Information:	
Funding Information:	
Abstract:	A non-linear lifting procedure is proposed to generate high density Benders cuts. The new denser cuts cover more master problem variables than traditional Benders cuts, shortening the required number of iterations to reach optimality, and speeding up the Benders decomposition algorithm. To lessen the intricacy stemmed from the non-linearity, a simple outer approximation linearization is devised to allow the use of linear programming tools to solve the additional Benders subproblem. Computational experiments show that the new cut selection procedure outperforms two existing alternatives — the Pareto-optimal cuts by Papadakos (2008), and the high density Benders cuts of Tang et al. (2013) — on solving two distinct location problems. Despite an extra computational effort, the worthwhile subproblem greatly reduces the number of Benders iterations when facing harder instances, yielding in remarkable time savings.

Powered by Editorial Manager® and ProduXion Manager® from Aries Systems Corporation

Manuscript

Click here to download Manuscript ibdnllp-gm.pdf

Annals of Operations Research manuscript No.
(will be inserted by the editor)

Improving Benders decomposition via a non-linear cut selection procedure

Gilberto de Miranda Jr. · Ricardo Saraiva de Camargo · Fátima Machado de Souza Lima

Received: date / Accepted: date

Abstract A non-linear lifting procedure is proposed to generate high density Benders cuts. The new denser cuts cover more master problem variables than traditional Benders cuts, shortening the required number of iterations to reach optimality, and speeding up the Benders decomposition algorithm. To lessen the intricacy stemmed from the non-linearity, a simple outer approximation linearization is devised to allow the use of linear programming tools to solve the additional Benders subproblem. Computational experiments show that the new cut selection procedure outperforms two existing alternatives — the Pareto-optimal cuts by Papadakos (2008), and the high density Benders cuts of Tang et al (2013) — on solving two distinct location problems. Despite an extra computational effort, the worthwhile subproblem greatly reduces the number of Benders iterations when facing harder instances, yielding in remarkable time savings.

Keywords Benders decomposition · Lifting procedure · Stronger Benders cuts · Large scale optimization

G. de Miranda Jr., R. S. de Camargo, and F. M. de Souza Lima would like to thank the agencies CNPq and FAPEMIG for grants 202765/2013-0, 302986/2012-0, 479682/2012-7, 305446/2010-0, 402651/2012-0, 480- 295/2012-3 and 02357/2012 which partially supported this research.

Gilberto de Miranda Jr.
E-mail: gilberto.miranda@ufes.br
Departamento de Matemática Aplicada
Universidade Federal do Espírito Santo - Brazil

Ricardo Saraiva de Camargo
E-mail: rcamargo@dep.ufmg.br
Departamento de Engenharia de Produção
Universidade Federal de Minas Gerais - Brazil

Fátima Machado de Souza Lima
E-mail: fatimasmlima@gmail.com
Departamento de Engenharia de Produção
Universidade Federal de Minas Gerais - Brazil

1 Introduction

Benders (1962) proposes a decomposition technique suitable for mixed integer programs (MIP) having a constraint matrix with a staircase structure in which a subset of the decision variables, referred to as complicating variables, couples a large scale mathematical program. Usually, when these variables are fixed, the attained subproblem (SP) can be decomposed into smaller, easy to solve subsystems. The method projects the variables of the large scale subproblem out, replacing them with a large number of constraints, potentially non-polynomial in size. These constraints referred to as Benders cuts (BC) involve only the complicating variables and an additional continuous variable used to select the best under-estimators of the SP's objective function value. The resulting mathematical program, called Benders master problem (MP), is proven equivalent to the original one, yielding thus the same optimal solutions.

Despite their number, the BCs can be relaxed, e.g. they can be temporarily disregarded, to be then later added to the MP on demand. This is possible since most of the BCs will not be binding at the optimal solution. Generating BCs *on the fly* requires a coordination strategy which iteratively solves the SP dual and the MP. The SP have the complicating variables temporarily fixed by the solution produced by the MP at each iteration. Whenever the dual SP is bounded, an optimality BC is produced; otherwise, a feasibility BC is generated. As the MP is a relaxation of the original problem, at each iteration it provides a lower bound (LB). An upper bound (UB) is easily obtained by composing the solution of the current bounded dual SP with the objective function part of the MP associated with the complicating variables. The procedure iterates until convergence of the LB and the overall best UB, when an optimal solution is then attained for the original problem.

Since its introduction, the Benders decomposition method has been successfully applied to a great range of different applications, e.g. the location of facilities (Geoffrion and Graves 1974; França and Luna 1982; Contreras et al 2011b,c,a; Tang et al 2013), the design of network systems (Magnanti and Wong 1984; Magnanti et al 1986; Costa 2005), the assignment of locomotives to passenger trains (Cordeau et al 2001), and the routing of aircrafts and the scheduling of crews in the aviation business (Mercier et al 2005; Papadakos 2009). Though most Benders algorithms are based on tailoring the method by using specific knowledge about the problem being addressed, some authors have provided general enhancements to the procedure.

Geoffrion (1972) extends the technique to some classes of mixed-integer nonlinear programs. McDaniel and Devine (1977) propose a warm-up phase in which the integer complicating variables are relaxed in the initial iterations. BCs are then generated from linear MPs' fractional solutions attained at a much lower computational effort, instead of resorting to solutions obtained from a branch-and-bound solver.

Magnanti and Wong (1981) introduce the concept of Pareto-optimal BCs. Whenever degeneracy occurs on the dual SPs, several alternative BCs with different strengths can be selected. A relationship of dominance among these cuts can be established by using a reference point to assess these strengths. This reference point is known as a Magnanti-Wong point. When a given cut is not dominated by other cuts with reference to a Magnanti-Wong point, then this cut is said to be a Pareto-optimal cut. Magnanti and Wong propose the solution of an additional, modified dual SP to produce Pareto-optimal BCs between iterations. This extra SP to be solved requires a given fixed Magnanti-Wong point and, due to its matrix structure, a great computational effort. It has also the disadvantage of being subject to numerical instabilities. To facilitate the separation of Pareto-Optimal cuts, Papadakos (2008) designs a scheme for updating the Magnanti-Wong point between cycles so that the additional dual SP to be solved is less pruned to numerical instabilities than the one originally devised by Magnanti and Wong.

All of the aforementioned enhancements imply that the induced primal SPs are feasible. For the cases with infeasible primal SPs, i.e. unbounded dual SPs, a feasibility BC needs to be generated. Mercier (2008); Fischetti et al (2010), and de Sá et al (2013) propose alternative selection procedures for better handling feasibility BCs than a textbook approach.

Recently Tang et al (2013) introduced a procedure which greatly speeds up the method by producing high density, strong cuts by lifting Pareto-optimal BCs by solving an auxiliary problem, besides the traditional SP. This additional SP produces BCs which cover more complicating variables by non-zero coefficients than Pareto-optimal BCs. By improving cuts from Pareto-optimal BCs, Tang et al implicitly question the definition itself of Pareto-optimality BCs, while opening a new venue of possibilities to design alternative cut selection strategies. Inspired by Tang et al's results, in this study high density BCs are lifted from Pareto-optimal BCs by a non-linear lifting auxiliary program. These new BCs greatly accelerate the Benders decomposition algorithm.

To efficiently solve the non-linear subproblem, a linearization akin to the Outer-Approximation technique (Duran and Grossmann 1986b,a; Fletcher and Leyfer 1994) is used. The new proposed Benders decomposition algorithm framework is tested on two different problems to assess its performance: (i) the uncapacitated facility location problem (Cornuéjols et al 1990); and (ii) the logistics facility location problem with capacity expansions (Tang et al 2013) . The attained results suggest that the new BC selection scheme clearly outperforms previous enhancements — the Pareto-optimal cuts by Papadakos (2008) and Magnanti and Wong (1981), and the high density Benders cuts of Tang et al (2013) — when harder instances are addressed. The remainder of this paper is organized as follows: §2 presents the Benders decomposition algorithm, while §3 show the new non-linear procedure and its solution technique. The attained computational results are reported in §4, with final remarks and suggestions for future works done in §5.

2 Benders decomposition method

For sake of presentation as two different problems are here addressed, the Benders decomposition method is presented in general matrix form. Let a mathematical programming problem have a subset of its decision variables considered to be *complicating*, e.g. y, and another subset considered to be of large scale, e.g. x. Further, let its whole matrix system have a stair-case structure. Given these assumptions, let this mathematical program be written as:

$$\begin{aligned}
\min\ & f^T y + c^T x \\
\text{s.t.:}\ & Dy + Ax \geq b \\
& y \in Y \\
& x \geq 0
\end{aligned}$$

without loss of generality, where f, c, and b, and A and D are column vectors and matrices of appropriate sizes, respectively. Y is a set of feasible points involving constraints having only the variables y.

To project the large scale variables x out, one first parametrizes the complicating variables to obtain the following equivalent problem:

$$\begin{aligned}
\min\ & f^T y + \min\{c^T x : Ax \geq b - Dy, x \geq 0\} \\
\text{s.t.:}\ & y \in Y
\end{aligned}$$

Note that the inner minimization problem is a linear program involving only the large scale continuous variables x. Its feasibility space changes for every single value of y, but its dual feasibility space does not. So after associating the dual variables $u \geq 0$ to the constraints of the inner subproblem, the following min-max problem is attained:

$$\begin{aligned}
\min\ & f^T y + \max\{(b - Dy)^T u : A^T u \leq c, u \geq 0\} \\
\text{s.t.:}\ & y \in Y
\end{aligned}$$

By the Finite Basis Theorem for Polyhedra, the polyhedron $\mathbf{U} = \{u : A^T u \leq c, u \geq 0\}$ is finitely generated by its sets of extreme points $\mathcal{U}$ and extreme rays $\mathcal{R}$, allowing the whole problem to be re-written as:

$$\begin{aligned}
\min\ & f^T y + \max\{(b - Dy)^T u : u \in \mathcal{U}\} \\
\text{s.t.:}\ & 0 \geq (b - Dy)^T u && \forall\ u \in \mathcal{R} \\
& y \in Y
\end{aligned}$$

As the supremum is the least upper bound, it is then possible to replace the inner-maximization subproblem by constraints to obtain what is known as the Benders MP, provided that an extra variable η is added to select the best SP dual solution:

$$\begin{aligned}
\min\ & f^T y + \eta \\
\text{s.t.:}\ & \eta \geq (b - Dy)^T u && \forall\ u \in \mathcal{U} \\
& 0 \geq (b - Dy)^T u && \forall\ u \in \mathcal{R} \\
& y \in Y
\end{aligned}$$

where constraints $\eta \geq (b-Dy)^T u$ and $0 \geq (b-Dy)^T u$ are referred to as optimality and feasibility BCs, respectively. The Benders MP is equivalent to the original problem, i.e. it attains the same optimal solution, but with fewer variables and many more constraints.

As most of the BCs will not be binding at an optimal solution, Benders (1962) suggests an approach of relaxing all of them first, and then generate them on demand. This requires solving the relaxed MP iteratively along the solution of the dual SP:

$$\begin{aligned} \phi(u) = \max\ & (b - D\bar{y})^T u \\ \text{s.t.:}\ & A^T u \leq c \\ & u \geq 0 \end{aligned}$$

The relaxed MP is responsible for fixing different values for the complicating variables $y = \bar{y}$, $y \in Y$, while BCs are generated from the dual Benders SP. If the dual Benders SP is bounded, then an optimality BC is obtained; otherwise, a feasibility BC is used to handle the unbounded direction.

A basic outline of a Benders decomposition algorithm is illustrated in Algorithm 1 in which the BCs are generated on demand. The UB and LB are initially set to infinity and zero, respectively. The method then alternates between the solution of the MP and the SP. While the MP updates the LB value and proposes new values for $\bar{y}$; the dual SP can render optimality BCs when $\phi(u)$ is bounded or feasibility BCs otherwise. Whenever the dual SP is bounded, the UB can be also updated. The algorithm iterates until convergence of the UB and LB. Note that the LB is monotonic non-decreasing. After a finite number of iterations, Algorithm 1 finds an optimal solution, if one exists, or shows that none exists Benders (1962).

Algorithm 1 Benders Algorithm

```
UB ← +∞, LB ← 0, h ← 0
while (UB ≠ LB) do
    LB ← SOLVE MP
    if (MP is infeasible) then
        STOP, return infeasible
    end if
    φ(u) ← SOLVE SP
    if (φ(u) < ∞) then
        ADD optimality BCs to MP
        UB ← min{UB, φ(u) + (LB − η)}
    else
        ADD feasibility BCs to MP
    end if
    h ← h + 1
end while
```

2.1 Magnanti and Wong Pareto-optimal cuts

Magnanti and Wong (1981) show that whenever the dual SP is bounded and degenerate, it is possible to generate several alternative optimality BCs with different strengths from multiple optimal solutions. Dual degeneracy is often common when addressing problems with embedded network flows and/or transportation structures. Magnanti and Wong propose establishing a relationship of dominance among these cuts by using a reference point to assess the different strengths. This reference point is nowadays known as a Magnanti-Wong point (y^{MW}), and it is defined as:

Definition 1 A point y^{MW} is a Magnanti-Wong point (i) if y^{MW} is not one of the points that spans the set of all possible projections of the original problem into the variables of the Benders MP; (ii) if the primal Benders subproblem is feasible for y^{MW}; (iii) and if the dual feasible solution set is nonempty.

Given a Magnanti-Wong point y^{MW}, and the optimal dual values $\bar{u}$ associated to the optimal solution $\phi(\bar{u})$ of the dual SP for $\bar{y}$, Magnanti and Wong (1981) formulate an additional SP, called a Magnanti-Wong dual SP, that generates a Pareto-optimal cut regarding to this reference point (y^{MW}). A cut that is not dominated by any other cut is named a Pareto-optimal BC. The Magnanti-Wong dual SP is written as:

$$
\begin{aligned}
\max\ & (b - Dy^{MW})^T u \\
\text{s.t.: }& (b - D\bar{y})^T u = \phi(\bar{u}) \\
& A^T u \leq c \\
& u \geq 0
\end{aligned}
$$

Algorithm 2 outlines the Magnanti-Wong dual SP embedded into the Benders decomposition algorithm. Note that a Magnanti-Wong point y^{MW} is required to be obtained at each iteration for the method to be the most effective. Further, constraint $(b - D\bar{y})^T u = \phi(\bar{u})$ is also reputed to make the whole Magnanti-Wong SP subject to numerical instabilities, sometimes delaying convergence.

2.2 Papadakos Pareto-optimal cuts

Using the Magnanti and Wong's BC generation scheme is not straightforward. Finding a new Magnanti-Wong point at each iteration and handling the numerical instabilities of the Magnanti-Wong dual SP pose as great challenges. Circumventing these issues, Papadakos (2008) shows that, when the search direction of the dual SP objective function is judiciously modified at each iteration, Pareto-optimal cuts can be generated even in the absence of constraint $(b - D\bar{y})^T u = \phi(\bar{u})$.

Papadakos shows that a valid new Magnanti-Wong point can be obtained by doing a linear combination of y^{MW} with the current Benders MP $\bar{y}^h$ solution value at iteration h or:

$$y^{MW} = (1 - \lambda)y^{MW} + \lambda\bar{y}^h$$

Algorithm 2 Magnanti-Wong Algorithm

```
UB ← +∞, LB ← 0, h ← 0
while (UB ≠ LB) do
    LB ← SOLVE MP
    if (MP is infeasible) then
        STOP, return infeasible
    end if
    φ(u) ← SOLVE SP
    if (φ(u) < ∞) then
        ADD optimality BCs to MP
        FIND a y^MW
        SOLVE Magnanti-Wong SP
        ADD optimality BCs to MP
        UB ← min{UB, φ(u) + (LB − η)}
    else
        ADD feasibility BCs to MP
    end if
    h ← h + 1
end while
```

where $0 < \lambda < 1$. He empirically demonstrates that the most effective value for λ is $1/2$, if all properties of Definition 1 are respected for the initial y^{MW}.

Papadakos modifies the Magnanti and Wong dual SP written it as:

$$\begin{aligned} \phi(u^{MW}) = \max\ & (b - Dy^{MW})^T u \\ \text{s.t.: }\ & A^T u \leq c \\ & u \geq 0 \end{aligned}$$

Papadakos also inverts the solution order of the MP and the Papadakos SP as illustrated in Algorithm 3. Instead of starting the loop with the solution of the Benders MP, the Papadakos dual SP is solved first given that a new y^{MW} is provided. A Pareto-optimal BC is then added to the MP which is then solved to produce a new LB and new values for $\bar{y}$. After that, the method is executed in the same way, but now with the presence of the updating scheme for the Magnanti-Wong point when the dual SP is bounded. Notice that the Papadakos dual SP can only be called in those iterations in which the y^{MW} was previously updated. Further a valid initial y^{MW} can be found by linearly combining feasible solutions for the original problem.

2.3 Tang et al lifting procedure

Observing that many of the MP's variables are not covered by non-zero coefficients in a Pareto-optimal BC, Tang et al (2013) propose a lifting procedure to generate denser cuts to accelerate the performance of the Benders decomposition algorithm. These new denser cuts, referred to as high density Pareto optimal Benders cuts, further restrict the MP's feasibility space, improving the convergence rate of the method.

For instance, given the optimality BC, $\eta + u^T Dy \geq b^T u,\ \forall\ u \in \mathcal{U}$, the idea is to have the largest possible number of coefficients of $u^T D$ with non-zero values

Algorithm 3 Papadakos Algorithm

$UB \leftarrow +\infty$, $LB \leftarrow 0$, $h \leftarrow 0$, $\lambda = 1/2$
FIND a y^{MW}
while ($UB \neq LB$) **do**
  if (y^{MW} is new) **then**
    SOLVE Papadakos SP
    ADD optimality BCs to MP
  end if
  $LB \leftarrow$ **SOLVE** MP
  if (MP is infeasible) **then**
    STOP, return *infeasible*
  end if
  $\phi(u) \leftarrow$ **SOLVE** SP
  if ($\phi(u) < \infty$) **then**
    ADD optimality BCs to MP
    UPDATE $y^{MW} = (1-\lambda)y^{MW} + \lambda\bar{y}^h$
    $UB \leftarrow \min\{UB, \phi(u) + (LB - \eta)\}$
  else
    ADD feasibility BCs to MP
  end if
  $h \leftarrow h + 1$
end while

and with the desirable feature of being within the same scale order. If the non-zero coefficients differ significantly in their magnitude, the ones with smaller values will most likely not be covered in practice.

To generate these coefficients, i.e. the values for the u dual variables, Tang et al (2013) suggest the solution of an additional SP at each iteration which has a bounded dual solution:

$$\begin{aligned} \max\ & \hat{c}^T u \\ \text{s.t.:}\ & (b - D\bar{y})^T u = \phi(\bar{u}) \\ & A^T u \leq c \\ & u \geq 0 \end{aligned}$$

This problem is similar to the Magnanti-Wong dual SP, but instead of using a coefficient vector based on a Maganti-Wong point (e.g. $(b - Dy^{MW})$) in the objective function, it employs a vector $\hat{c}$ with coefficients judiciously chosen. Tang et al construct a scaled, equilibrated $\hat{c}$ from the values of the matrices D and b or:

$$\hat{c}_i = \frac{1}{b_i} - \frac{1}{\sum_{j=1}^{m} d_{ij}} \qquad \forall\ i = 1, \ldots, n$$

assuming that matrices D and b have sizes equal to $n \times m$, and $n \times 1$, and coefficients equal to b_i and d_{ij}, respectively. Whenever b_i and $\sum_j^m d_{ij}$ are equal to zero, then the associated coefficient $\hat{c}_i$ is set to zero. By using $\hat{c}$, lifted BCs can obtained with a large number of non-zero, same scale coefficients. Aforementioned scheme can only be employed for those iterations with bounded dual SPs, and for those problems having SPs with dual degeneracy (i.e. multiple optimal solutions).

3 The non-linear cut selection procedure

Inspired by Tang et al (2013)'s results on lifting Pareto-optimal BCs, a new lifting procedure is here proposed based on a simple, but effective idea. Tang et al use an auxiliary SP with an objective function with intricate coefficients $\hat{c}$, and an equality constraint $(b - D\bar{y})^T u = \phi(\bar{u})$ that forces the new cut to have the same direction than the Pareto-optimal BC being lifted, but with different cut coefficients. The expectation is to have the $\hat{c}$ coefficients inducing a dual solution u with more nonzero values, covering hence more MP's variables when assembling the new BC.

Imposing the lifted new cut to have the same direction of the BC being lifted (equality constraint), and having $\hat{c}$ with different magnitude coefficients may lessen the degrees of freedom that the dual variables can assume, preventing thus an utmost possible coverage of the MP's variables. A more interesting idea is to trade-off a greater coverage at the expense of accepting some small deflection (error) from the direction of the Pareto optimal BC being lifted. This can be achieved by having a dual SP with all the dual variable coefficients in the objective function equal to one, and by having an inequality instead of an equality. Further a penalized artificial error variable can be set to control the accepted deviation degree from the Pareto optimal BC being lifted.

Minimizing this penalized error while maximizing the dual values can generate a lifted BC with more covered MP's variables. The aforementioned ideas can be further enhanced if a non-linear penalty is adopted, instead of a linear one. A non-linear function for the penalized error may induce more basic dual variables different than zero than a linear approach. Hence here a non-linear programming cut selection strategy is proposed to generate high density BCs. The devised non-linear SP has homogeneous coefficients (e.g. vector $\mathbb{1}$) for the dual variables u on the objective function, and an error variable $\xi \geq 0$ to compute the deviation from the direction of the Pareto optimal BC being lifted. A scalar factor $\omega > 0$ is used to weight the adopted quadratic non-linear function of ξ to be minimized. The proposed lifting non-linear SP is written as:

$$
\begin{aligned}
&\min\ \omega\xi^2 - \mathbb{1}^T u \\
&\text{s.t.: } (b - Dy^{MW})^T u + \xi \geq \phi(\bar{u}^{MW}) \\
&\qquad A^T u \leq c \\
&\qquad u \geq 0 \\
&\qquad \xi \geq 0
\end{aligned}
$$

This is a convex quadratic non-linear problem with linear constraints that can be handily solved by different methods (e.g. Frank and Wolfe (1956); Bertsekas (1982); Duran and Grossmann (1986b) and Boggs and Tolle (2000)). However note that the ξ variables are not coupled to the y variables on the objective function, being thus independent of the current values of the complicating variables. The quadratic term can then be replaced by a piecewise linear approximation (see Figure 1), judiciously constructed by several supporting hyperplanes generated from different ξ values, given rise to the following constraints:

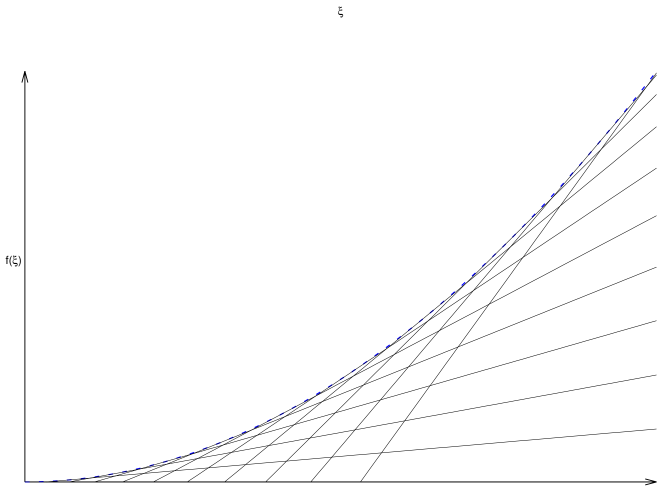


Fig. 1 A piecewise linear approximation of a convex quadratic function.

$$\varepsilon \geq \omega(\bar{\xi}^t)^2 + 2\omega\bar{\xi}^t(\xi - \bar{\xi}^t) \qquad \forall t = 1, \dots, \tau$$

where τ is the number of points chosen to be spanned over the ξ axis, and $\varepsilon \geq 0$ is an auxiliary variable used to approximate the quadratic function. A larger value for τ, along with well spanned values for $\bar{\xi}^t$, better fits the function but at the cost of having a larger linear program:

$$\begin{aligned}
\min\ & \varepsilon - \mathbb{1}^T u \\
\text{s.t.: } & (b - Dy^{MW})^T u + \xi \geq \phi(\bar{u}^{MW}) \\
& \varepsilon \geq \omega(\bar{\xi}^t)^2 + 2\omega\bar{\xi}^t(\xi - \bar{\xi}^t) && \forall t = 1, \dots, \tau \\
& A^T u \leq c \\
& u \geq 0 \\
& \xi \geq 0 \\
& \varepsilon \geq 0
\end{aligned}$$

This is a linear formulation which can be easily solved by a Simplex method.

The improved Benders decomposition is outlined in Algorithm 4. It differs from Algorithm 3 by having the solution of the lifting SP rightly after solving the Papadakos SP. As suggested by the computational experiments, the extra effort for solving an additional SP is worthwhile when harder instances are addressed, given the number of iterations saved, and the strength of the BCs generated.

Algorithm 4 The Nonlinear Cut Selection Algorithm

$UB \leftarrow +\infty$, $LB \leftarrow 0$, $h \leftarrow 0$, $\lambda = 1/2$
FIND a y^{MW}
while $(UB \neq LB)$ **do**
  if (y^{MW} is new) **then**
    SOLVE Papadakos SP
    SOLVE Lifting SP
    ADD optimality HDBCs to MP
  end if
  $LB \leftarrow$ **SOLVE** MP
  if (MP is infeasible) **then**
    STOP, return *infeasible*
  end if
  $\phi(u) \leftarrow$ **SOLVE** SP
  if $(\phi(u) < \infty)$ **then**
    ADD optimality BCs to MP
    UPDATE $y^{MW} = (1-\lambda)y^{MW} + \lambda\bar{y}^h$
    $UB \leftarrow \min\{UB, \phi(u) + (LB - \eta)\}$
  else
    ADD feasibility BCs to MP
  end if
  $h \leftarrow h + 1$
end while

4 Numerical examples

The lifting proposed Benders decomposition algorithm is compared with two other cut selection strategies (Papadakos (2008) and Tang et al (2013)) on solving two different problems: (i) the logistics facility location problem with capacity expansions (Tang et al 2013); and (ii) the uncapacitated facility location problem (Cornuéjols et al 1990). All algorithms were implemented using IBM CPLEX 12.5 on a *Dell PowerEdge T410* workstation, equipped with one *Intel Xeon E5-2600v2* processor and 24 Gbytes of RAM memory. Only one thread was allowed to be activated. The adopted test sets for each experiment are later on explained.

4.1 Logistics facility location problem with capacity expansions

Tang et al (2013) study a logistic facility location problem in which suppliers in set J provide products in set L to clients of set I through existing facilities in set K_0 or via facilities to be selected and installed from set K_1 of potential candidates ($K = K_0 \cup K_1$). Each client $j \in J$ has a demand D_{jl} for product $l \in L$ which can be partially served by a supplier $i \in I$ with capacity S_{il} for product $l \in L$. An opened facility $k \in K$ has a minimum $\mathcal{M}_k^L$ required throughput and a maximum $\mathcal{M}_k^M$ one, before being allowed to expand its capacity. The maximum capacity expansion allowed is R_k units for facility $k \in K$ at the unitary cost e_k. A setup cost f_k is incurred whenever a facility $k \in K_1$ is installed; while a saving cost p_k is attained for closing an existing facility $k \in K_0$. The handling cost per unit of product at facility $k \in K$ is given by v_k; and the unitary transportation cost for product $l \in L$ from supplier $i \in I$ to client $j \in J$, through facility $k \in K$, is represented by c_{ikjl}.

The problem consists of determining which existing facilities are to remain open or not; which new facilities are to be installed; which expansions to be made; and which routes are to be established to serve the client demands for each product. The problem objective is to minimize the total costs involved with the facility installations, capacity expansions, and with the product handling and transport, as well as to maximize the savings from closing existing facilities. Tang et al (2013) adopt the following decision variables to model the problem: $z_k \in \{0,1\}$ is equal to 1 if facility $k \in K_1$ is installed or 0, otherwise; $u_{kj} \in \{0,1\}$ is equal to 1 if client $j \in J$ is served by facility $k \in K$ or 0, otherwise; $x_{ikjl} \geq 0$ represents the amount of product $l \in L$ routed from supplier $i \in I$ through facility $k \in K$ to client $j \in J$; and $s_k \geq 0$ is the amount of capacity expansion for facility $k \in K$. Tang et al model the aforementioned problem as:

$$\begin{aligned}
\min \sum_{k\in K} e_k s_k &+ \sum_{k\in K_1} f_k z_k - \sum_{k\in K_0} p_k(1-z_k) \\
&+ \sum_{k\in K}\sum_{j\in J}\sum_{l\in L} v_k D_{jl} u_{kj} \\
&+ \sum_{k\in K}\sum_{j\in J}\sum_{l\in L}\sum_{i\in I} c_{ikjl} x_{ikjl} && (1)\\
\text{s.t.: } & \sum_{k\in K}\sum_{j\in J} x_{ikjl} \leq S_{il} && \forall i \in I, l \in L && (2)\\
& \sum_{i\in I} x_{ikjl} = D_{jl} u_{kj} && \forall k \in K, j \in J, l \in L && (3)\\
& \mathcal{M}_k^L z_k \leq \sum_{j\in J}\sum_{l\in L} D_{jl} u_{kj} \leq \mathcal{M}_k^U z_k + s_k && \forall k \in K && (4)\\
& s_k \leq \mathcal{R}_k z_k && \forall k \in K && (5)\\
& \sum_{k\in K} u_{kj} = 1 && \forall j \in J && (6)\\
& x_{ikjl} \geq 0 && \forall i \in I, k \in K, j \in J, l \in L && (7)\\
& s_k \geq 0 && \forall k \in K && (8)\\
& z_k \in \{0,1\} && \forall k \in K && (9)\\
& u_{kj} \in \{0,1\} && \forall k \in K, j \in J && (10)
\end{aligned}$$

The objective function (1) minimizes the total cost consisted of the capacity expansion, the transportation, the installation, and the handling costs, as well as the savings from closing the existing facilities. Constraints (2) are supply capacity constraints for supplier $i \in I$ and product $l \in L$. Constraints (3) guarantee that the total amount of product shipped from all suppliers to customer $j \in J$ via facility $k \in K$ is equal to the demand of customer $j \in J$ for product $l \in L$. Constraints (4) enforce the minimum and maximum required throughput allowed for each facility. Constraints (4) also allows for capacity expansions. Constraints (5) limits the capacity expansions. Constraints (6) assign customers to facilities. Finally, constraints (7)-(10) are the decision variables domain.

A Benders decomposition algorithm can be applied to formulation (1)-(10) if variables z and u are considered as complicating. Hence for fixed values of

$z = \bar{z}$ and $u = \bar{u}$, and after associating the dual variables $\mu_{il} \leq 0$, $r_{kjl} \in \mathbb{R}$, $\phi_k \geq 0$ and $\alpha_k \leq 0$ to constraints (2)-(5), respectively, the implied dual SP can be split into two different smaller dual SPs: One is related to the transportation and routing of products from the suppliers to the clients; and the other is associated to the capacity expansion of the facilities. The former is written as the dual linear program (11)-(14); while the latter is given by the linear model (15)-(18).

$$\max \sum_{i \in I} \sum_{l \in L} S_{il} \mu_{il} + \sum_{k \in K} \sum_{j \in J} \sum_{l \in L} D_{jl} \bar{u}_{kj} r_{kjl} \tag{11}$$

$$\text{s.t.: } \mu_{il} + r_{kjl} \leq c_{ikjl} \qquad \forall i \in I, k \in K, j \in J, l \in L \tag{12}$$

$$\mu_{il} \leq 0 \qquad \forall i \in I, l \in L \tag{13}$$

$$r_{kjl} \in \mathbb{R} \qquad \forall k \in K, j \in J, l \in L \tag{14}$$

$$\max \sum_{k \in K_0} \left(\left(\sum_{j \in J} \sum_{l \in L} D_{jl} \bar{u}_{kj} - \mathcal{M}_k^U \bar{z}_k \right) \phi_k + R_k \bar{z}_k \alpha_k \right) \tag{15}$$

$$\text{s.t.: } \phi_k + \alpha_k \leq e_k \qquad \forall k \in K_0 \tag{16}$$

$$\phi_k \geq 0 \qquad \forall k \in K_0 \tag{17}$$

$$\alpha_k \leq 0 \qquad \forall k \in K_0 \tag{18}$$

From the objective functions (11) and (15), BCs can be generated at each iteration of the algorithm. Differently from §4.2, in which the dual SPs are always feasible and finite, the linear SPs (11)-(14) and (15)-(18) can now be bounded or unbounded depending on the values of $\bar{z}$ and $\bar{u}$. Thus optimality ((20) and (22)) and feasibility ((21) and (23)) BCs may be assembled along the iterations of the algorithm. These cuts are added to the MP which can be written as:

$$\min \sum_{k\in K}\sum_{j\in J}\sum_{l\in L} v_k D_{jl} u_{kj} + \eta + \xi + \sum_{k\in K_1} f_k z_k - \sum_{k\in K_0} p_k(1-z_k) \tag{19}$$

$$\text{s.t.: } \eta \geq \sum_{i\in I}\sum_{l\in L} S_{il}\bar{\mu}^h_{il} + \sum_{k\in K}\sum_{j\in J}\sum_{l\in L} D_{jl} u_{kj} \bar{r}^h_{kjl} \qquad h = 1,\dots,H \tag{20}$$

$$0 \geq \sum_{i\in I}\sum_{l\in L} S_{il}\bar{\mu}^h_{il} + \sum_{k\in K}\sum_{j\in J}\sum_{l\in L} D_{jl} u_{kj} \bar{r}^h_{kjl} \qquad h = 1,\dots,\mathcal{R} \tag{21}$$

$$\xi \geq \sum_{k\in K_0}\left(\sum_{j\in J}\sum_{l\in L} D_{jl}u_{kj} - \mathcal{M}^U_k z_k\right)\bar{\phi}_k + \sum_{k\in K_0} R_k z_k \bar{\alpha}_k \qquad h = 1,\dots,H \tag{22}$$

$$0 \geq \sum_{k\in K_0}\left(\sum_{j\in J}\sum_{l\in L} D_{jl}u_{kj} - \mathcal{M}^U_k z_k\right)\bar{\phi}_k + \sum_{k\in K_0} R_k z_k \bar{\alpha}_k \qquad h = 1,\dots,\mathcal{R} \tag{23}$$

$$\sum_{k\in K} u_{kj} = 1 \qquad \forall j \in J \tag{24}$$

$$\sum_{j\in J}\sum_{l\in L} D_{jl}u_{kj} \geq \mathcal{M}^L_k z_k \qquad \forall k \in K_0 \tag{25}$$

$$\mathcal{M}^L_k z_k \leq \sum_{j\in J}\sum_{l\in L} D_{jl}u_{kj} \leq \mathcal{M}^U_k z_k \qquad \forall k \in K_1 \tag{26}$$

$$z_k \in \{0,1\} \qquad \forall k \in K \tag{27}$$

$$u_{kj} \in \{0,1\} \qquad \forall k \in K, j \in J \tag{28}$$

$$\eta \geq 0 \tag{29}$$

$$\xi \geq 0 \tag{30}$$

variables η and ξ sub-estimate the transportation and expansion costs, respectively; and H and R are the maximum number of iterations generating optimality and feasibility BCs. The following valid inequalities proposed by Tang et al (2013) are incorporated to the MP (19)-(30) to restrict it even further:

$$\sum_{j\in J}\sum_{l\in L} D_{jl}u_{kj} \leq z_k(\mathcal{M}^U_k + R_k) \qquad \forall k \in K_0 \tag{31}$$

$$\sum_{k\in K_0} z_k(\mathcal{M}^U_k + R_k) + \sum_{k\in K1} \mathcal{M}^U_k z_k \leq \sum_{j\in J}\sum_{l\in L} D_{jl} \tag{32}$$

$$\sum_{j\in J} u_{kj} \leq |J| z_k \qquad \forall k \in K \tag{33}$$

Constraints (31) ensure that the total demands assigned to a facility does not exceed its maximum allowed capacity even after being expanded. Constraint

(32) guarantees that the installed capacity is at least the total demand. Finally, constraints (33) couple variables u with z, being activation constraints.

Tang et al (2013) propose 10 classes of instances with 20 instances each and parameters randomly generated following uniform distributions with $c_{ikjl} \in [1, 10]$, $D_{jl} \in [20, 100]$, $v_k \in [5, 10]$, for all $k \in K$, $S_{il} \in [D_l/|I|, 2D_l/|I|]$, $\mathcal{M}_k^U \in [4W_T/5|K_0|, 6W_T/5|K_0|]$ and $\mathcal{M}_k^L \in [\mathcal{M}_k^U/3, 2\mathcal{M}_k^U/5]$, for all $k \in K_0$, where $W_T = [2/5D, 2/3D]$ and D is the total demand for all products. Further $f_k \in [2\mathcal{M}_k^U, 5\mathcal{M}_k^U]$, $\mathcal{M}_k^U \in [N_T/|K_1|, N_T/2]$ and $\mathcal{M}_k^L \in [\mathcal{M}_k^U/3, 2\mathcal{M}_k^U/5]$, for all $k \in K_1$, where $N_T = D - W_T$, and $e_k \in [2f_k/\mathcal{M}_k^U, 4f_k/\mathcal{M}_k^U]$ $R_k \in [2\mathcal{M}_k^U/5, \mathcal{M}_k^U/2]$, and $p_k \in [4f_k/5, f_k]$, for all $k \in K_0$. Table 1 shows the set sizes for each problem class.

Table 1 Set sizes of the generated problem classes.

class	$\|I\|$	$\|K_0\|$	$\|K_1\|$	$\|J\|$	$\|L\|$
1	5	2	3	10	3
2	5	3	5	30	5
3	5	4	6	50	5
4	10	4	6	50	5
5	5	8	10	30	8
6	6	4	5	40	8
7	5	8	10	40	10
8	10	6	9	40	10
9	12	10	15	50	15
10	20	5	5	60	30

The 200 instances were solved by the three Benders decomposition algorithms (Papadakos (2008), Tang et al (2013) and the new Benders cut selection scheme) and by the optimization solver CPLEX with a time limit of 10 hours or 36,000 seconds. To evaluate and compare the performance of the methods on the test set, the benchmarking procedure proposed by Dolan and Moré (2002) was adopted.

The benchmarking procedure profiles the performance of each optimization method by computing a cumulative distribution function over a given performance metric. Dolan and Moré have suggested the ratio of the computational time of a solver over the best time of all of the solvers as a performance metric for each instance. In other words, a performance ratio r_{ps} of an instance p by solver s is measured as $r_{ps} = \frac{t_{ps}}{\min\{t_{p\gamma} : \gamma \in \mathbb{S}\}}$, in which t_{ps} is the solution time of solver s for problem p, and $\mathbb{S}$ is a set of solvers. The closest to 1, the faster the solution method; while when it is equal to 1, it indicates the fastest solver for that instance.

By using this performance ratios, Dolan and Moré have shown how to estimate then the probability $\rho_s(\tau)$ of a solver $s \in \mathbb{S}$ to have a performance ratio r_{ps} within a factor $\tau \in \mathbb{R}$ of the best possible ratio or $\rho_s(\tau) = \frac{|\{p \in \mathbb{P} : r_{ps} \le \tau\}|}{\mathbb{n}}$, in which $\mathbb{P}$ is the set of instances and $\mathbb{n} = |\mathbb{P}|$. $\rho_s(\tau)$ is a cumulative distribution function of the performance ratio known as performance profile that when plotted it helps visualize which solver performs better than the others. Here the number of Benders iterations were also benchmarked.

The performance profile for the current solvers on the test set can be seen in Figure 2. The new scheme is the only method able to solve all the instances within the given time limit. Notwithstanding a comparable performance with the other Benders cut selection schemes for most problems, even though it requires the solution of an additional subproblem per iteration, the new scheme achieved the overall best performance. It has a larger probability of solving more instances faster than the other methods for a smaller value of τ.

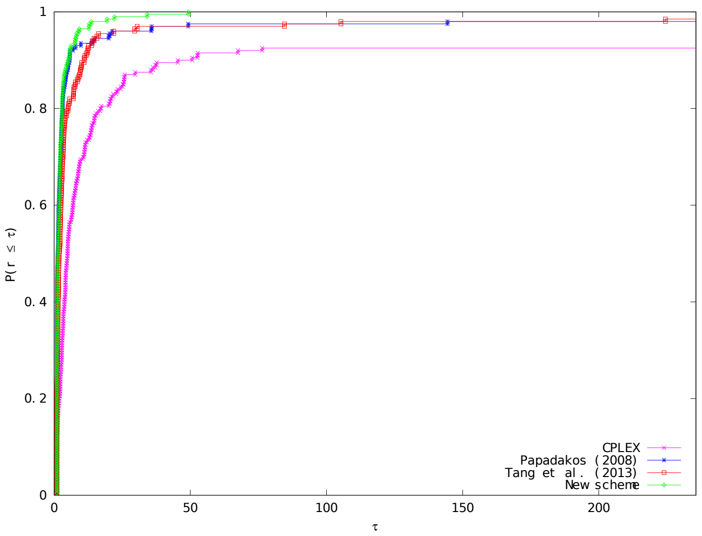


Fig. 2 Running time performance profile.

The strength of the Benders cuts of the new scheme can be verified on the performance profile shown in Figure 3 for the number of Benders iterations required to reach optimality. The new scheme attained the smallest number of Benders cycles for all instances.

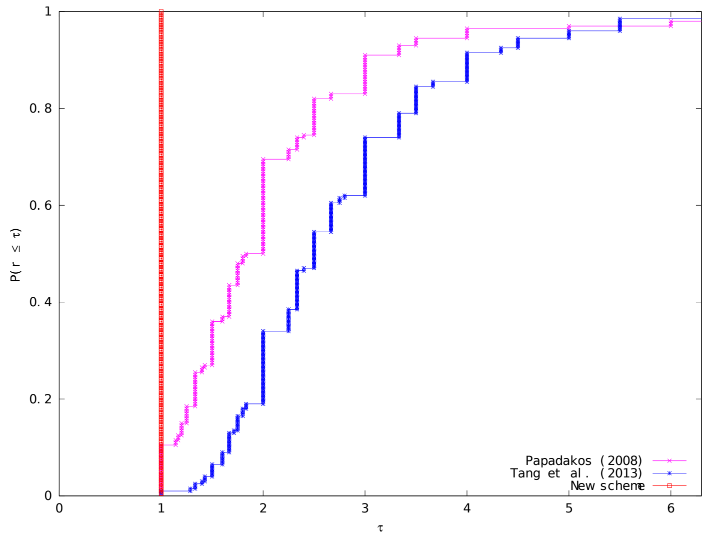


Fig. 3 Iteration count performance profile.

The achieved results employed at the generation of the performance profiles shown in Figures 2 and 3 are reported in Tables 4-6 in the Appendix B and summarized in Table 2. The latter table allows a more detailed analysis of the results by showing for each class: the number of unsolved instances; the number of fastest runnings; the average number of iterations; and the average performance ratio. The number of total variables are reported as a guidance on the complexity of each class.

For small size instances, the burden spent on solving an additional subproblem to render a lifted Benders cuts may not be worthed the effort as can be seen for classes 4 and 5; though the method was competitive for the small classes 1 and 2. The new scheme however will most likely delay the solving process, despite the number of iterations that can save. Nevertheless for more complex instances as in class 9 and 10, the new cut selection scheme presents itself as a valuable tool as it provided the best performance ratios.

It is important to remark that the higher density of the new scheme Benders cuts may interfere with the branching rules of CPLEX which may slow down the solution of each master problem instead of speeding up. This can be inferred from Table 6 for class 8, instances numbered $1, 11, 12, 15$, and $17-19$. Although the new scheme greatly abbreviates the required number of iterations for these instances, the method is unable to solve the problems faster. On the contrary, there is a further increase on the solution time, certainly stemmed from the burden of handling the denser cuts. The authors believe that there was unfavorable interference on the branching rules adopted by CPLEX during the solution of the master problem for those cases.

In the next section, the new scheme is evaluated on a different problem: the uncapacitated facility location problem. This is a classical problem of the literature usually adopted to assess the performance of Benders cuts selection procedures.

Table 2 Result comparison of the three Benders decomposition schemes and CPLEX.

class	# vars	CPLEX	Papadakos (2008)	Tang et al (2013)	New scheme
Number of unsolved instances within 36,000 secs.					
1	807				
2	6,251				
3	13,014				
4	25,514				
5	22,166				
6	17,653				
7	36,746	1	1		
8	60,621				
9	226,285	4	2		
10	360,615	10	1	3	
sum		15	4	3	
Number of fastest runnings for each class.					
1	807	6	5	3	6
2	6,251	11	3	2	4
3	13,014		7	4	9
4	25,514		9	4	7
5	22,166	6	7	2	5
6	17,653		11	5	4
7	36,746		5	7	8
8	60,621		10	6	4
9	226,285		7	5	8
10	360,615		8	4	8
sum		23	72	42	63
Average number of iterations for each class.					
1	807		4.00	7.55	3.10
2	6,251		5.20	6.50	2.85
3	13,014		4.90	6.75	2.45
4	25,514		5.20	7.05	3.10
5	22,166		5.25	9.60	3.05
6	17,653		5.50	7.70	3.40
7	36,746		6.58	8.95	3.65
8	60,621		8.65	9.75	5.05
9	226,285		8.19	10.39	3.40
10	360,615		15.80	7.84	3.94
average			6.42	8.18	3.39
Average performance ratio for each class.					
1	807	5.65	4.64	9.63	3.73
2	6,251	2.74	2.10	4.56	2.84
3	13,014	3.69	2.21	2.66	1.34
4	25,514	9.51	1.96	2.45	2.20
5	22,166	2.30	1.64	3.99	2.35
6	17,653	8.58	3.34	3.83	1.93
7	36,746	8.46	1.86	3.35	1.83
8	60,621	16.55	1.74	2.77	2.95
9	226,285	14.20	3.83	14.61	2.84
10	360,615	26.62	13.33	8.48	5.89
average		8.84	3.62	5.59	2.79

4.2 Uncapacitated Facility location problem

The uncapacitated facility location problem is one of the most studied problems in location sciences (Drezner 1995; Drezner and Hamacher 2004; Farahani and

Hekmatfar 2009), being often used in textbooks (Lasdon 1972; Martin 1999) to illustrate the applicability of Benders decomposition algorithms. The problem consists of selecting an unknown number of facilities from a set J of candidates to supply the demand of the clients of a set I at minimum total cost. The total cost is composed of the facility setup and demand transportation costs. Each facility $j \in J$ has an installation cost of $f_j > 0$, while each client $i \in I$ has a demand $d_i > 0$. The transportation cost from a client $i \in I$ to a facility $j \in J$ is given by c_{ij}. With the aid of integer decision variables $y_j \in \{0,1\}$ to represent if facility $j \in J$ is opened ($y_j = 1$) or not ($y_j = 0$), and continuous decision variables $x_{ij} \geq 0$ to indicate the percentage of the demand of $i \in I$ served by a facility at $j \in J$, the uncapacitated facility location problem can be written as:

$$\min \sum_{j \in J} f_j y_j + \sum_{i \in I} \sum_{j \in J} d_i c_{ij} x_{ij} \tag{34}$$

$$\text{s.t.: } \sum_{j \in J} x_{ij} = 1 \qquad \forall i \in I \tag{35}$$

$$x_{ij} \leq y_j \qquad \forall i \in I, j \in J \tag{36}$$

$$x_{ij} \geq 0 \qquad \forall i \in I, j \in J \tag{37}$$

$$y_j \in \{0,1\} \qquad \forall j \in J \tag{38}$$

The objective function (34) minimizes the total cost. Constraints (35) assure that demand of $i \in I$ is fully served by facilities. Constraints (36) are activation ones: a client $i \in I$ can only be attended by a facility at $j \in J$ if this facility is opened. The domain for the values of decision variables is ensured by constraints (37) and (38). Note that problem (34)-(38) has a staircase matrix structure, being the y variables considered to be complicating ones. Thus the whole problem is amenable to Benders decomposition algorithms.

For fixed values of $y = \bar{y}$ and after associating the dual variables $\rho_i \in \mathbb{R}$ and $u_{ij} \geq 0$ to constraints (35) and (36), respectively, the following dual linear problem is obtained:

$$\max \sum_{i \in I} \rho_i - \sum_{i \in I} \sum_{j \in J} \bar{y}_j u_{ij} \tag{39}$$

$$\text{s.t.: } \rho_i - u_{ij} \leq d_i c_{ij} \qquad \forall i \in I, j \in J$$

$$\rho_i \in \mathbb{R} \qquad \forall i \in I$$

$$u_{ij} \geq 0 \qquad \forall i \in I, j \in J$$

From the dual objective function (39), BCs on the form (40) can be generated to be added the MP given as:

$$
\begin{aligned}
&\min \sum_{j \in J} f_j y_j + \eta \\
&\text{s.t.: } \eta \geq \sum_{i \in I} \bar{\rho}_i^h - \sum_{i \in I} \sum_{j \in J} y_j \bar{u}_{ij}^h && \forall h = 1, \dots, H && (40) \\
&\sum_{j \in J} y_j \geq 1 && && (41) \\
&y_j \in \{0, 1\} && \forall j \in J \\
&\eta \geq 0
\end{aligned}
$$

where variable $\eta \geq 0$ sub-estimates the transportation costs; H is the maximum number of Benders iterations; and constraint (41) prevents the generation of feasibility BCs.

Three classes created by Kochetov and Ivanenko (2005) were used to compare the performance of the new Benders cut generation scheme with the Papadakos (2008)'s version. The Tang et al (2013)'s variant was not included in the comparison, since it presented the smallest performance profile probability on the previous studied problem in the former section. Kochetov and Ivanenko's instance sets are highly artificial, created to have large integrality gaps so that they can pose as a challenge for exact methods.

Each class has 30 instances with 100 nodes each. While the demand vector d is assumed to be unitary for these classes, their cost matrices c were randomly generated according to a rule for each class. While, for class A, l elements of each column of c are randomly selected to have non-infinity values; for class B, l entries of each row are randomly chosen. Class C has each row and column with exactly l randomly selected coefficients with non-infinity values. For all classes, the assigned non-infinity values are uniformly chosen within the interval $[0, 4]$. Classes A, B and C have average integrality gaps of 25.6%, 21.2% and 28.4%, respectively.

Once again the performance profile procedure of Dolan and Moré (2002) was used to assess the methods. Figure 4 shows the plots of the performance ratios for the achieved results reported in Tables 7-8 in the Appendix A. The integrality gaps are reported as a guidance on the complexity of each instance.

The new scheme attained a running time performance comparable to the Papadakos (2008) algorithm, except for 8 instances on which it is outperformed. A more detailed performance ratio analysis, one for each class, is plotted in Figure 5 and summarized in Table 3.

The new scheme performed better than the Papadakos (2008) algorithm for all instances in class C (see Figure 5(c)) with the larger integrality gaps, assumed thus to be most difficult testbed. However for classes A and B with smaller integrality gaps (see Figures 5(a) and 5(b)), the new scheme was competitive for some instances, but being outperformed for others. For some of these problems, the new scheme had unfavorable poor performance. Observing Table 7, these instances are instances numbered 332, 1232, 3032, 3132 of class A, and 931, 1031 and 1531 of class B.

An interesting case is instance 931 of class B in which the new scheme took 13.81 more time to reach optimality than the Papadakos (2008) algorithm. Once again it is here surmised that for cases like this, the higher density of the new

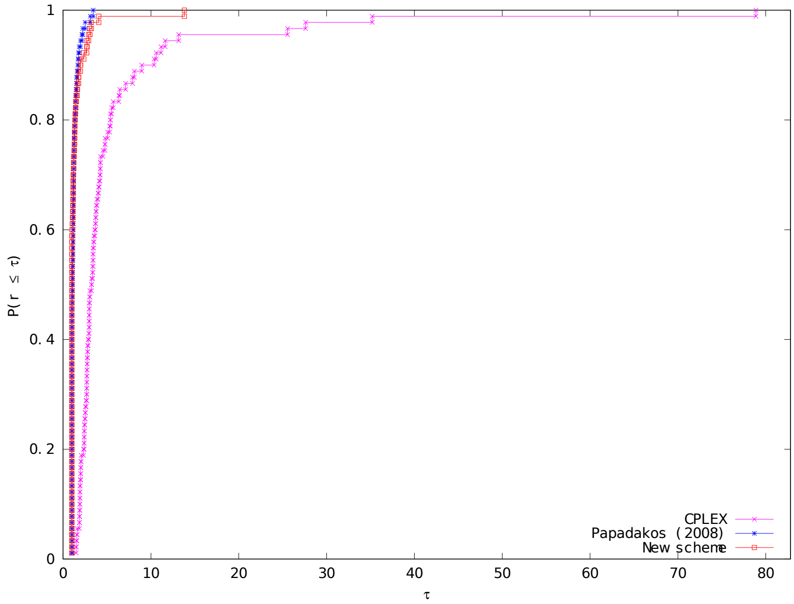


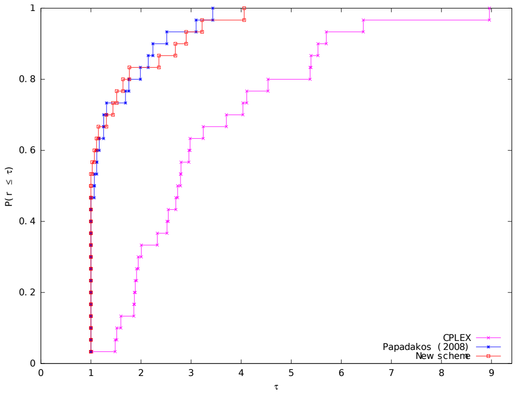
Fig. 4 Running time performance profile.

Table 3 Result comparison of the three Benders decomposition schemes and CPLEX.

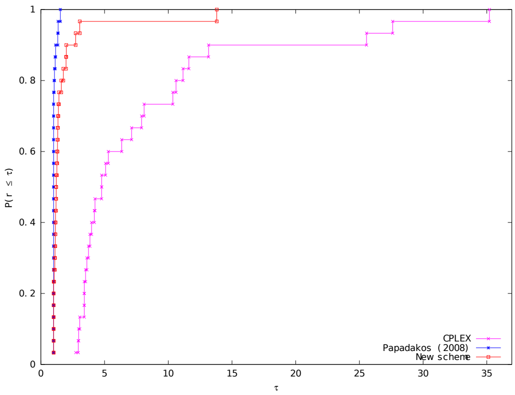
class	linear gap [%]	CPLEX	Papadakos (2008)	New scheme
	Number of fastest runnings for each class.			
1	25.52		14	16
2	21.75		23	7
3	28.41		1	29
	sum		38	52
	Average number of iterations for each class.			
1	25.52		6.30	2.90
2	21.75		4.87	3.53
3	28.41		6.20	3.10
	average		5.70	3.18
	Average running time ratio for each class.			
1	25.52	3.28	1.41	1.44
2	21.75	8.13	1.05	1.79
3	28.41	5.30	1.25	1.00
	average	5.57	1.24	1.41

scheme Benders cuts may have interfered with the branching rules of CPLEX. This most likely may have slowed down the solution of master problems, even though the new scheme consistently reduced the number of Benders iterations required to achieve optimality as can be seen in Figure 6.

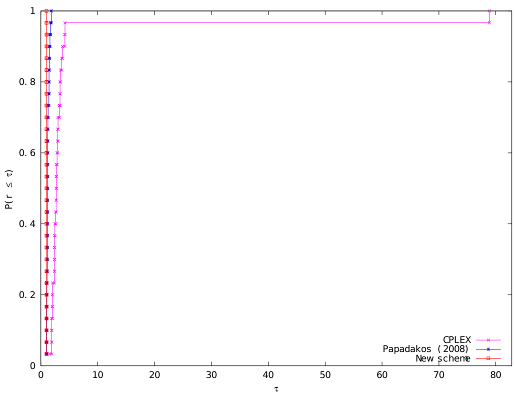
To further illustrate the performance comparisons, and also to clarify the BC selection schemes' suitability for arbitrary instances, the performance profiling of the computational experiments was set to account two key observations: (i) whenever instances are well handled by CPLEX, assumed thus to be easy, then there is



(a) Running time performance profile for class A.



(b) Running time performance profile for class B.



(c) Running time performance profile for class C.

Fig. 5 Performance profile.

no much advantage to use the new scheme. However it is clear that the new scheme will be more advantageous whenever the Papadakos (2008) algorithm displays convergence issues. This will most likely happen when solving harder instances. (ii)

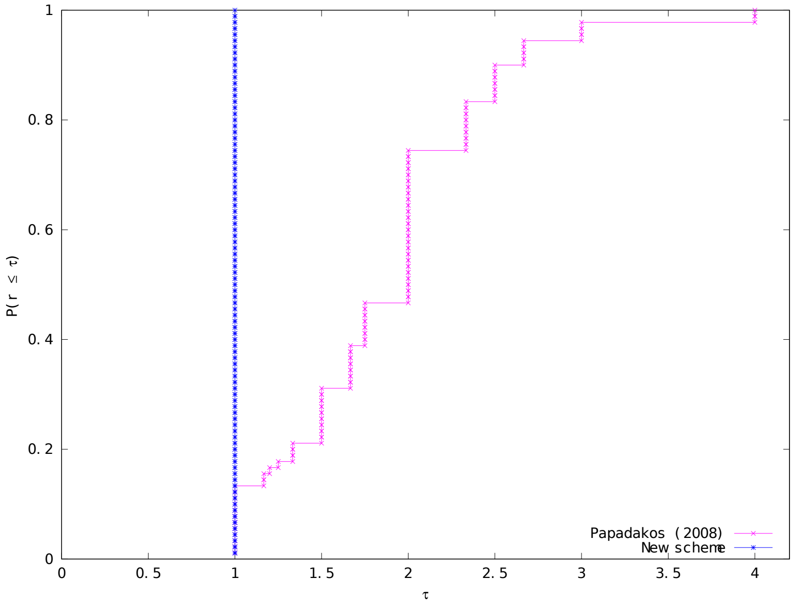


Fig. 6 Iteration count performance profile.

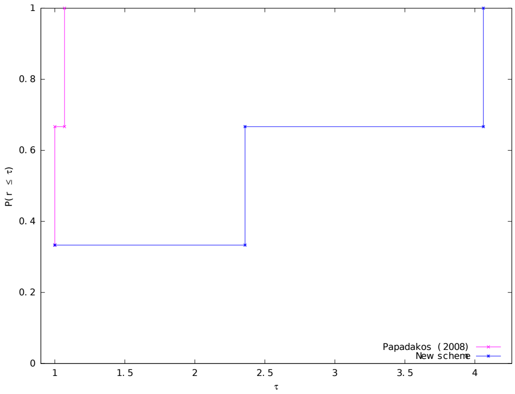
Since the new scheme lifts Papadakos BCs, the new scheme's performance gain is directly correlated with the number of iterations it can save with respect to the Papadakos Benders decomposition variant.

The performance profiles were rebuild considering only instances for which CPLEX takes more than 1000 seconds to solve. The new curves are displayed in Figure 7 for the uncapacitated facility location problem. Recalling that the integrality gap increases from class A to C, it is now possible to see a clear pattern: the harder the instances get, the more advantageous the new scheme becomes, highlighting its superior performance over difficult problems.

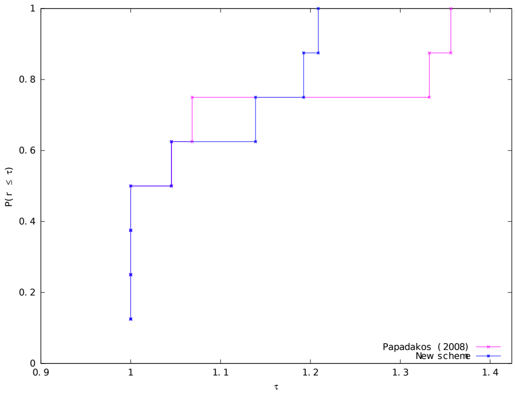
Finally, it is possible to confirm the second key observation by plotting the time difference between the Papadakos algorithm and the new scheme versus the number of iterations taken by the former to reach optimality for those instances in which CPLEX takes more than 1000 seconds. Figure 8 shows these differences. The new scheme out-performs the Papadakos algorithm regarding computing times whenever the Papadakos algorithm takes more iterations and therefore longer to reach optimality. It can also be inferred from Figure 8 that the new scheme is a better solution approach for those instances in which Papadakos algorithm takes more than six iterations to converge.

5 Final remarks

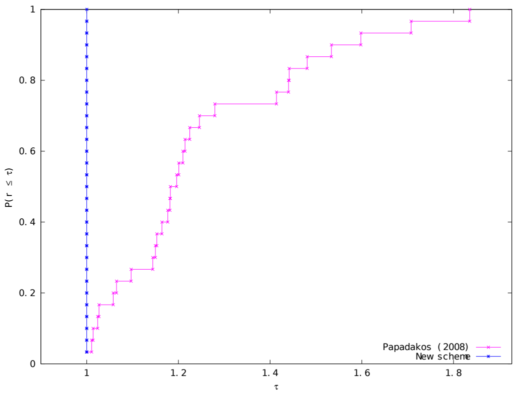
This article provides a new Benders cut selection scheme based on a non-linear program which produces high density cuts from Pareto optimal Benders cuts. A higher density implies that a larger number of the Benders master program variables are covered, i.e. have non-null coefficients. Well known results on the



(a) Running time profile - class A - Hard instances.



(b) Running time profile - class B - Hard instances.



(c) Running time profile - class C - Hard instances.

Fig. 7 Performance profile - Hard instances.

literature state that if the original inequality is facet defining, then the lifted one is a facet of augmented dimension, rendering in cutting planes with superior performance.

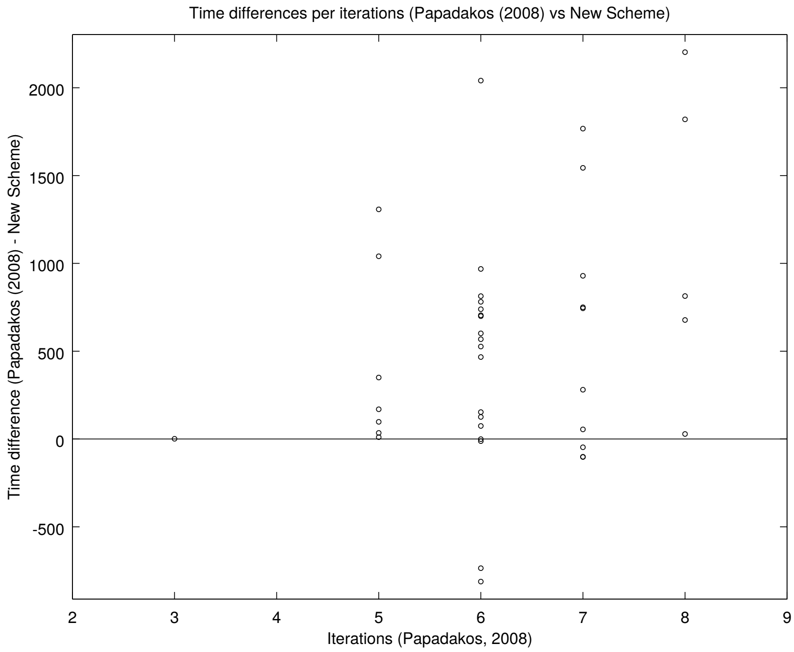


Fig. 8 Solution time differences versus Papadakos algorithm iterations number.

The new scheme was applied to two different location problems, the logistics facility location problem with capacity expansions Tang et al (2013), and the well known uncapacitated facility location problem. Data sets from Tang et al (2013) and Kochetov and Ivanenko (2005) with large integrality gaps were used to assess the performance of the new scheme. The computational results suggest that the improved Benders cuts are stronger than two Benders selection procedures of the literature: Pareto-Optimal cuts by Papadakos (2008), and the high density cuts by Tang et al (2013). For all instances, the new scheme required an equal number or much fewer iterations than the other methods, showing the strength of the cuts. For the harder instances, the new scheme outperformed the other methods, presenting great speedups; though not being outperformed by the other cut selection procedures for instances considered to be easy.

Though the auxiliary non-linear program works by increasing the density of Pareto optimal cuts, the perceived speedups are tied to the performance of its simpler version. In other words, for easier instances, it is better to use weaker, fast generated Benders cuts. However, for those instances considered to be the hardest ones, in which alternative Benders cuts selection techniques display slow convergence, the considerable computational effort to solve an extra subproblem, even though a linearized one, pays off given the attainable savings in the number iterations and in the total computational times.

Future researches may address how to speed up the solution of the non-linear auxiliary subproblem, so that the computing time savings can be extended for a wider range of problem instances; or to better understand on how these new, denser Benders cuts interact with the branching strategies of a general solver like

CPLEX. Great speedups can be achieved if these high density cuts can be better handled or having the unfavorable interferences diminished.

References

Benders JF (1962) Partitioning procedures for solving mixed-integer programming problems. Numerisch Mathematik 4:238–252

Bertsekas DP (1982) Projected newton methods for optimization problems with simple constraints. SIAM Journal on Control and Optimization 20(2):221–246

Boggs PT, Tolle JW (2000) Sequential quadratic programming for large-scale nonlinear optimization. Journal of Computational and Applied Mathematics 124(1-2):123 – 137, numerical Analysis 2000. Vol. IV: Optimization and Nonlinear Equations

Contreras I, Cordeau JF, Laporte G (2011a) Benders decomposition for large-scale uncapacitated hub location. Operations Research 59:1477–1490

Contreras I, Cordeau JF, Laporte G (2011b) Exact solution of large-scale hub location problems with multiple capacity levels. Transportation Science To appear

Contreras I, Cordeau JF, Laporte G (2011c) Stochastic uncapacitated hub location. European Journal of Operational Research To appear

Cordeau JF, Soumis F, Desrosiers J (2001) Simultaneous assignment of locomotives and cars to passenger trains. Operations Research 49(4):531–548

Cornuéjols G, Nemhauser G, Wolsey L (1990) The uncapacitated facility location problem. In: Mirchandani P, RL F (eds) Discrete location theory, Wiley, New York, pp 119–71

Costa AM (2005) A survey on benders decomposition applied to fixed-charge network design problems. Comput Oper Res 32:1429–1450

Dolan ED, Moré JJ (2002) Benchmarking optimization software with performance profiles. Mathematical Programming A 91:201–213

Drezner Z (1995) Facility Location: A Survey of Applications and Methods. Springer Series in Operations Research and Financial Engineering, Springer New York

Drezner Z, Hamacher H (2004) Facility Location: Applications and Theory. Springer series in operations research, Springer

Duran M, Grossmann IE (1986a) An outer-approximation algorithm for a class of mixed integer nonlinear programms. Mathematical Programming 36:307–339

Duran MA, Grossmann I (1986b) An outer-approximation algorithm for a class of mixed-integer nonlinear programs. Mathematical Programming 36:307

Farahani R, Hekmatfar M (2009) Facility Location: Concepts, Models, Algorithms and Case Studies. Contributions to Management Science, Physica-Verlag HD

Fischetti M, Salvagnin D, Zanette A (2010) A note on the selection of benders' cuts. Mathematical Programming B 124:175–182

Fletcher R, Leyfer S (1994) Solving mixed integer nonlinear programs by outer approximation. Mathematical Programming 66:327

França P, Luna H (1982) Solving stochastic transportation-location problem by generalized benders decomposition. Transportation Science 16(2):113–126

Frank M, Wolfe P (1956) An algorithm for quadratic programming. Naval Research Logistics Quarterly 3:95–110

Geoffrion AM (1972) Generalized benders decomposition. Journal of Optimization Theory And Applications 10:237–260

Geoffrion AM, Graves GW (1974) Multicomodity distribution system design by Benders decomposition. Management Science 20:822–844

Kochetov Y, Ivanenko D (2005) Computationally difficult instances for the uncapacitated facility location problem. In: Ibaraki T, Nonobe K, Yagiura M (eds) Metaheuristics: Progress as Real Problem Solvers, Operations Research/Computer Science Interfaces Series, vol 32, Springer US, pp 351–367

Lasdon LS (1972) Optimization Theory for Large Systems. The Macmillan Company, New York

Magnanti T, Wong R (1981) Accelerating benders decomposition: Algorithmic enhancement and model selection criteria. Operations Research 29(3):464–483

Magnanti TL, Wong RT (1984) Network design and transportation planning: Models and algorithms. Transportation Science 18:1–55

Magnanti TL, Mirchandani P, Wong RT (1986) Tailoring Benders decomposition for uncapacitated network design. Mathematical Programming Study 26:112–154

Martin RK (1999) Large Scale Linear and Integer Optimization; A Unified Approach, Kluwer, chap Large integer programs: projection and inverse projection, pp 613–617

McDaniel D, Devine M (1977) A modified Benders partitioning algorithm for mixed integer programming. Management Science 24(3):312–319

Mercier A (2008) A theoretical comparison of feasibility cuts for the integrated aircraft-routing and crew-pairing problem. Transportation Science 42(1):87–104

Mercier A, Cordeau JF, Soumis F (2005) A computational study of benders decomposition for the integrated aircraft routing and crew scheduling problem. Computers and Operations Research 32(6):1451–1476

Papadakos N (2008) Practical enhancements to the MagnantiWong method. Operations Research Letters 36:444–449

Papadakos N (2009) Integrated airline scheduling. Computers and Operations Research 36:176–195

de Sá EM, Camargo RS, Miranda G (2013) An improven Benders decomposition algorithm for the tree of hubs location problem. European Journal of Operational Research 226(4):185–202

Tang L, Jiang W, Saharidis GK (2013) An improved benders decomposition algorithm for the logistics facility location problem with capacity expansions. Annals of Operations Research 210(1):165–190

Appendix A

Computational results for the logistics facility location problem with capacity expansions.

Table 4 Result comparison of the three Benders decomposition schemes and CPLEX.

Class	# integer vars	Instance #	CPLEX Time [s]	Papadakos (2008) Time [s]	Papadakos (2008) Iters.	Tang et al (2013) Time [s]	Tang et al (2013) Iters.	New scheme Time [s]	New scheme Iters.
1	807	1	0.87	1.91	6	0.09	5	0.17	2
		2	0.74	0.76	4	1.89	8	0.15	2
		3	0.90	0.16	3	0.74	8	0.36	2
		4	0.43	0.30	5	0.06	6	0.18	2
		5	0.15	0.39	4	1.27	9	0.92	2
		6	0.97	0.06	3	0.93	4	0.48	2
		7	0.81	0.42	3	1.11	7	0.33	2
		8	0.36	0.05	3	0.46	7	0.42	3
		9	0.28	0.30	2	0.33	10	0.95	2
		10	0.44	0.07	3	0.05	7	0.40	3
		11	0.12	1.78	8	1.66	12	1.54	7
		12	0.89	0.99	3	0.77	6	0.32	2
		13	0.57	0.18	3	1.85	7	1.10	3
		14	0.64	1.68	6	1.47	8	0.62	5
		15	0.32	1.05	3	1.12	3	1.45	3
		16	0.96	1.44	5	1.61	9	1.21	5
		17	0.66	1.28	3	1.53	6	1.99	2
		18	0.80	0.43	5	1.81	10	0.02	5
		19	1.62	1.29	5	1.55	10	1.51	5
		20	0.17	0.54	3	1.95	9	0.14	3
2	6,251	1	0.64	0.49	4	0.22	8	0.55	3
		2	0.18	0.63	6	0.48	7	0.34	4
		3	1.16	1.89	6	1.58	8	1.92	3
		4	0.21	0.26	5	1.34	8	0.08	4
		5	0.34	1.10	5	1.04	6	1.22	3
		6	0.58	1.08	4	1.31	6	0.62	2
		7	4.85	1.83	4	1.20	5	2.12	3
		8	0.95	0.52	4	1.64	4	0.29	2
		9	1.18	1.81	7	1.66	8	1.70	4
		10	1.11	1.78	5	1.68	5	0.85	2
		11	0.74	0.96	6	0.77	5	0.91	3
		12	2.00	0.09	4	2.59	8	1.94	2
		13	0.96	0.43	5	1.28	7	0.56	2
		14	0.51	1.29	5	1.17	4	0.54	2
		15	0.06	0.17	6	0.23	6	0.32	2
		16	0.27	0.28	6	1.37	7	0.50	3
		17	0.24	1.24	5	0.87	6	0.77	3
		18	1.89	0.75	4	1.30	5	0.51	2
		19	1.00	0.99	4	1.60	6	1.02	2
		20	3.55	6.52	9	5.92	11	4.89	6
3	13,014	1	2.16	1.70	5	1.70	7	0.51	3
		2	7.82	3.51	4	5.50	6	7.22	2
		3	2.65	0.77	4	2.95	7	0.19	2
		4	1.58	2.33	5	3.99	8	1.27	2
		5	38.88	8.50	5	18.87	8	8.55	2
		6	3.95	2.85	5	3.50	8	1.04	2
		7	3.42	1.82	5	2.93	5	1.81	3
		8	3.66	2.34	5	2.34	7	2.13	2
		9	32.61	6.76	6	11.51	9	12.95	3
		10	2.69	1.61	4	2.57	6	2.02	3
		11	1.37	1.08	4	1.42	5	1.97	2
		12	4.54	1.10	5	3.31	8	2.42	2
		13	12.54	13.89	6	4.66	7	3.10	2
		14	3.92	1.36	3	1.19	5	2.66	2
		15	4.77	2.37	5	4.95	9	1.69	3
		16	3.34	5.24	6	1.34	6	2.68	3
		17	14.76	7.05	3	5.39	6	6.24	3
		18	88.67	13.28	6	20.88	6	14.35	3
		19	5.38	1.68	6	1.57	6	1.67	2
		20	10.41	40.03	6	14.75	6	4.08	3

Table 5 (Continued from Table 4) Result comparison of the three Benders decomposition schemes and CPLEX.

Class	# integer vars	Instance #	CPLEX Time [s]	Papadakos (2008) Time [s]	Papadakos (2008) Iters.	Tang et al (2013) Time [s]	Tang et al (2013) Iters.	New scheme Time [s]	New scheme Iters.
4	25,514	1	213.30	10.39	6	7.12	8	4.21	2
		2	72.93	17.57	5	23.50	6	35.41	2
		3	763.39	69.69	4	114.99	10	533.46	4
		4	6.83	4.84	5	2.99	6	4.11	4
		5	5.73	4.26	6	3.69	6	1.38	2
		6	36.29	7.41	4	38.20	7	15.67	3
		7	210.62	31.21	5	88.02	8	99.72	4
		8	505.81	147.75	6	252.81	6	421.29	6
		9	1291.70	299.17	6	235.87	8	51.34	2
		10	3.67	1.81	4	2.76	5	0.78	2
		11	568.09	267.43	7	134.34	9	488.30	4
		12	47.25	3.49	4	4.10	6	11.24	2
		13	3.41	2.42	6	3.40	8	5.54	5
		14	10.27	1.85	4	3.82	7	1.13	2
		15	67.96	34.83	6	29.81	9	9.95	2
		16	3139.50	820.80	7	752.39	5	1439.62	3
		17	7.94	2.81	4	4.95	5	2.97	3
		18	142.51	53.94	7	19.20	8	9.69	4
		19	10.51	1.19	3	8.59	9	3.07	3
		20	42.42	5.83	5	5.67	5	17.27	3
5	22,166	1	1.44	0.56	5	0.53	8	1.56	4
		2	0.89	1.84	4	1.29	11	1.00	3
		3	0.50	0.38	4	2.94	12	1.47	4
		4	0.25	0.11	6	1.15	7	1.55	3
		5	16.63	4.92	6	17.44	11	3.34	2
		6	0.63	0.62	4	2.48	10	0.22	4
		7	0.86	2.25	6	2.99	8	1.66	2
		8	1.62	0.31	3	2.95	10	1.64	3
		9	1.62	1.66	4	1.02	7	1.55	3
		10	1.88	2.80	6	4.12	11	2.88	4
		11	0.98	2.66	6	3.23	12	1.58	3
		12	1.88	1.66	6	9.34	15	1.98	3
		13	1.48	1.20	5	4.05	11	1.35	2
		14	1.20	3.60	6	3.09	9	1.71	2
		15	12.71	5.79	7	7.74	10	6.21	5
		16	12.98	4.34	4	5.62	7	4.07	4
		17	6.85	1.28	4	1.73	7	2.44	3
		18	3.76	2.49	3	2.64	5	1.14	2
		19	6.81	6.27	8	5.43	11	2.45	2
		20	1.11	1.29	8	3.51	10	1.83	3
6	17,653	1	212.69	23.30	6	64.00	9	67.92	4
		2	23.49	150.67	5	5.29	6	4.21	3
		3	20.24	4.24	4	8.53	6	4.03	2
		4	377.99	33.63	6	92.04	8	39.37	3
		5	32.44	4.65	5	16.50	5	4.62	2
		6	149.69	29.74	4	65.87	7	50.85	3
		7	8.92	2.14	5	4.59	7	1.21	3
		8	65.13	7.74	5	14.33	6	10.98	3
		9	519.56	55.10	6	156.91	9	150.62	4
		10	22.60	7.59	6	6.73	7	10.50	3
		11	13.83	4.77	5	11.81	7	8.21	3
		12	15.04	3.74	5	3.50	6	4.84	4
		13	1228.32	260.38	8	149.75	8	308.55	6
		14	647.60	29.87	6	37.84	7	34.41	3
		15	69.89	112.44	4	16.33	9	45.14	4
		16	3.70	1.66	5	2.30	7	1.98	2
		17	13.02	8.56	9	84.08	14	8.91	5
		18	5296.75	1095.81	5	207.84	9	1713.93	4
		19	47.92	9.79	7	298.38	10	11.26	4
		20	31.60	1.29	4	3.89	7	2.97	3
7	36,746	1	387.73	82.51	8	48.70	11	76.05	4
		2	140.15	29.34	6	39.12	8	143.51	5
		3	31.53	7.08	6	7.40	8	7.03	4
		4	401.75	96.47	6	153.04	9	155.02	5
		5	64.82	6.72	6	82.73	9	18.36	4
		6	14.38	18.07	7	19.17	8	7.86	4
		7	608.78	37.51	6	57.81	10	23.69	3
		8	31.72	10.61	9	23.60	10	9.70	4
		9	20.11	13.91	7	9.50	7	11.81	4
		10	13.77	51.80	9	83.80	8	9.10	4
		11	30.63	8.35	7	6.02	6	4.40	2
		12	41.01	5.22	5	7.03	7	5.58	2
		13	304.27	18.87	7	25.38	10	42.14	5
		14	203.67	79.07	6	32.43	7	185.45	5
		15	memory	memory	> 7	1467.65	9	4880.91	3
		16	23.43	14.78	4	7.51	6	8.76	3
		17	391.76	99.66	6	742.94	10	33.94	3
		18	160.82	16.76	7	10.15	8	7.83	3
		19	9.61	6.44	7	4.21	9	4.48	3
		20	165.98	20.26	6	8.04	10	15.81	3

Table 6 (Continued from Table 4) Result comparison of the three Benders decomposition schemes and CPLEX.

Class	# integer vars	Instance #	CPLEX Time [s]	Papadakos (2008) Time [s]	Papadakos (2008) Iters.	Tang et al (2013) Time [s]	Tang et al (2013) Iters.	New scheme Time [s]	New scheme Iters.
8	60,621	1	57.63	6.57	6	12.52	8	23.51	3
		2	1851.73	722.65	15	707.10	13	92.29	3
		3	2058.09	166.81	5	57.95	7	748.19	4
		4	46.72	15.75	8	15.23	7	17.97	4
		5	364.51	61.20	12	32.70	10	26.44	3
		6	1918.64	111.01	6	69.97	10	64.47	3
		7	192.78	36.94	9	108.06	12	44.51	5
		8	86.43	15.91	7	175.16	14	20.18	4
		9	187.59	16.92	7	16.75	9	34.16	6
		10	514.70	30.47	8	213.51	10	78.36	6
		11	3631.50	147.69	12	53.81	9	156.43	5
		12	252.15	31.58	9	46.23	10	47.68	6
		13	4114.10	78.30	8	84.39	9	694.44	7
		14	47.26	20.16	9	17.56	8	13.29	5
		15	207.40	19.69	8	20.58	10	47.58	5
		16	384.24	31.78	7	231.03	10	127.48	7
		17	452.89	214.73	10	481.90	10	646.39	7
		18	530.64	483.12	10	202.40	12	517.92	6
		19	628.07	167.70	11	124.97	9	263.36	6
		20	167.67	9.69	6	20.98	8	28.30	6
9	226,285	1	2189.17	1235.44	10	404.88	10	289.33	4
		2	1838.97	1226.76	9	960.56	11	87.51	3
		3	467.02	42.29	5	74.88	9	53.98	3
		4	78.70	115.91	8	252.18	11	20.41	2
		5	820.33	22.09	6	25.63	8	28.21	3
		6	133.61	35.59	7	422.41	11	78.56	6
		7	1157.98	1895.11	8	88.95	10	453.91	4
		8	530.78	36.00	6	84.03	9	59.84	4
		9	390.44	33.45	6	28.38	7	41.81	4
		10	memory	1456.29	8	252.23	8	4883.35	3
		11	150.73	53.42	5	378.28	13	97.19	3
		12	memory	memory	> 5	2410.68	7	2021.70	3
		13	1350.18	72.26	6	99.06	8	37.23	2
		14	memory	3536.06	10	4007.88	10	2400.34	3
		15	968.78	194.01	6	302.24	9	75.91	2
		16	547.40	memory	> 7	40.79	8	233.00	4
		17	250.43	97.04	10	38.86	8	50.07	3
		18	596.16	25.59	5	41.16	10	78.41	5
		19	300.76	56.00	9	10836.58	10	48.28	3
		20	memory	560.89	7	1475.31	10	2500.30	4
10	360,615	1	2191.40	2046.41	12	51.01	10	41.55	2
		2	4022.27	199.95	7	212.11	7	156.66	2
		3	memory	595.59	10	14741.99	9	140.02	3
		4	memory	280.37	12	2675.72	8	361.99	3
		5	3490.82	306.76	9	>36000	> 7	10476.13	4
		6	258.86	9875.64	8	70.63	10	68.37	5
		7	memory	172.08	7	577.75	11	150.65	2
		8	memory	128.39	6	923.08	8	1179.87	3
		9	memory	121.15	9	memory	> 6	345.99	4
		10	3165.86	122.31	8	137.49	9	279.23	6
		11	1507.77	33.21	6	57.36	9	72.00	4
		12	767.10	539.45	6	171.90	10	197.60	3
		13	1671.48	>36000	> 6	112.34	8	286.14	5
		14	memory	83.59	7	108.49	9	152.40	4
		15	21091.28	591.97	10	276.04	8	13593.89	3
		16	349.77	2215.43	9	243.30	9	62.08	3
		17	memory	339.42	12	313.08	7	277.49	2
		18	memory	487.45	7	>36000	> 7	308.79	3
		19	memory	69.05	6	101.65	9	123.68	3
		20	memory	93.93	7	88.17	8	112.34	3

Appendix B

Computational results for the uncapacitated facility location problem.

Table 7 Result comparison of the three Benders decomposition schemes and CPLEX.

Class	Instance #	Linear gap [%]	CPLEX Time [s]	Papadakos (2008) Time [s]	Papadakos (2008) Iters.	New scheme Time [s]	New scheme Iters.
A	332	25.43	203.87	37.91	5	122.09	4
	432	24.71	213.17	259.74	8	83.76	2
	532	27.68	188.89	185.15	10	205.90	5
	632	24.40	117.16	58.74	5	50.36	2
	732	21.73	65.43	7.30	3	11.06	2
	832	26.22	109.18	39.15	6	56.39	3
	932	26.55	166.27	172.92	7	86.97	3
	1032	26.16	181.06	108.93	7	97.48	3
	1132	25.66	164.81	65.42	6	67.11	3
	1232	29.69	1370.33	240.36	6	976.05	4
	1332	24.50	101.46	107.42	8	31.28	2
	1432	27.19	113.30	76.33	7	99.95	4
	1532	23.56	309.44	63.07	4	48.05	2
	1632	26.33	218.40	54.12	4	88.78	3
	1732	24.27	111.66	85.54	6	39.89	2
	1832	25.79	127.99	63.77	6	73.32	3
	1932	24.34	108.57	98.18	8	55.85	3
	2032	25.62	110.10	72.64	6	77.68	3
	2132	23.70	224.47	94.62	7	75.32	3
	2232	23.51	188.05	44.12	4	41.45	2
	2332	24.37	411.96	74.44	5	131.61	3
	2432	27.37	149.21	27.66	4	80.28	3
	2532	25.09	100.53	59.99	6	53.80	3
	2632	23.48	141.43	126.81	8	75.02	3
	2732	28.58	2156.62	844.29	7	789.55	3
	2832	25.21	117.51	92.17	7	73.54	3
	2932	24.50	134.95	102.14	6	45.64	2
	3032	27.02	132.49	32.22	4	86.55	3
	3132	30.48	1611.18	597.63	6	1409.48	4
	3232	25.82	103.34	70.16	5	27.89	2
B	331	24.26	1953.58	371.92	5	274.21	2
	431	22.52	773.26	194.34	6	312.08	6
	531	19.25	210.34	44.10	5	53.04	3
	631	19.22	237.22	67.58	6	75.08	5
	731	24.01	1136.31	270.62	6	282.85	3
	831	22.86	419.83	123.20	6	161.12	6
	931	20.69	177.95	13.52	2	186.64	2
	1031	16.91	132.29	5.18	2	15.81	2
	1131	17.74	76.66	9.70	3	11.11	2
	1231	19.72	580.78	121.68	6	172.85	6
	1331	18.05	51.74	14.96	5	9.77	2
	1431	18.65	92.91	8.00	2	10.09	2
	1531	17.72	73.59	6.94	2	18.96	2
	1631	21.89	413.43	107.28	6	142.56	6
	1731	19.22	304.51	27.29	3	48.17	2
	1831	22.72	1225.54	339.06	7	386.17	4
	1931	22.80	474.23	155.32	7	209.27	6
	2031	22.55	359.28	122.37	6	134.50	4
	2131	24.14	1582.24	533.24	7	635.87	4
	2231	19.75	714.77	25.89	2	51.75	2
	2331	21.82	590.30	104.51	5	93.00	3
	2431	24.28	1146.99	446.60	8	418.02	4
	2531	26.26	1606.55	503.52	6	377.92	3
	2631	23.83	634.89	186.16	6	238.70	6
	2731	16.96	299.70	8.51	2	16.82	2
	2831	17.56	129.51	12.50	2	17.22	2
	2931	24.55	1650.65	484.77	7	585.95	6
	3031	20.88	701.19	187.58	7	216.33	4
	3131	21.68	225.99	32.27	4	27.88	2
	3231	23.58	1151.50	236.87	5	226.55	3

Table 8 (Continued from Table 7) Result comparison of the three Benders decomposition schemes and CPLEX.

Class	Instance #	Linear gap [%]	CPLEX Time [s]	Papadakos (2008) Time [s]	Papadakos (2008) Iters.	New scheme Time [s]	New scheme Iters.
C	333	28.39	9925.95	3307.85	6	2339.50	3
	433	28.42	10117.04	3596.42	6	2994.82	3
	533	22.99	8412.93	107.81	3	106.66	2
	633	28.42	9526.34	4290.34	7	3546.13	4
	733	28.42	11012.61	2770.76	5	2601.51	3
	833	28.41	9856.95	3986.69	6	3987.61	4
	933	28.39	7328.57	2505.02	5	2470.37	3
	1033	28.41	8947.08	4122.42	6	3308.97	3
	1133	28.41	11783.54	5316.10	8	3113.31	3
	1233	28.41	9524.42	5379.54	8	4701.89	4
	1333	28.41	8214.66	3697.10	6	3128.92	3
	1433	28.42	9307.36	5131.19	7	4850.66	4
	1533	28.40	7499.77	3811.37	6	3111.95	3
	1633	28.42	12548.69	5230.34	8	3410.24	3
	1733	28.41	7230.34	5415.68	8	4601.76	4
	1833	28.42	15481.05	5855.39	6	5702.24	4
	1933	28.41	8898.67	5443.28	7	3675.88	3
	2033	28.40	10485.35	3581.38	6	3114.68	3
	2133	28.38	7409.57	3170.67	6	3096.41	3
	2233	28.39	6516.03	3494.75	5	2187.08	2
	2333	28.40	8750.30	4424.39	6	3643.12	3
	2433	28.41	9001.21	4254.93	7	3325.57	3
	2533	28.41	11402.50	5051.41	7	3507.39	3
	2633	28.40	7122.69	3396.99	5	2356.46	2
	2733	28.39	8475.83	4483.84	6	2442.99	2
	2833	28.42	6128.11	3734.73	6	3208.26	3
	2933	28.40	7452.69	3964.38	5	3614.29	3
	3033	28.42	11024.41	4565.20	6	3860.67	3
	3133	28.42	9873.62	5654.21	7	4904.02	4
	3233	28.42	7240.72	4505.81	6	3766.75	3