2019/20

DP

Dieter Weninger and Laurence A. Wolsey

Benders' algorithm with (mixed)-integer subproblems

CORE
CENTER FOR OPERATIONS RESEARCH AND ECONOMETRICS

CORE
Voie du Roman Pays 34, L1.03.01
B-1348 Louvain-la-Neuve
Tel (32 10) 47 43 04
Email: immaq-library@uclouvain.be
https://uclouvain.be/en/research-institutes/lidam/core/discussion-papers.html

CORE DISCUSSION PAPER
2019/20

Benders' Algorithm with (Mixed)-Integer Subproblems

Dieter Weninger[1] and Laurence A. Wolsey[2]

November 2019

Abstract

We consider problems of the form $\min\{cx + hy : Ax + By \geq b, x \in \mathbb{Z}^n_+, y \in Y \subseteq \mathbb{R}^p_+\}$ that are often treated using Benders' algorithm, but in which some of the y-variables are required to be integer. We present two algorithms that hopefully add to and clarify some of the algorithms proposed since the year 2000. Both are branch-and-cut algorithms solving linear programs by maintaining a strict separation between a Master problem in (x, η)-variables and a subproblem in the y-variables. The first involves nothing but the solution of linear programs, but involves branching in (x, y)-space. It is demonstrated on a small capacitated facility location problem with single-sourcing. The second restricted to problems with $x \in \{0, 1\}^n$ only requires branching in the x-space, but uses cutting planes in the subproblem based on the integrality of the y-variables that are converted/lifted into valid inequalities for the original problem in (x, y)-variables. For the latter algorithm we show how the lifting can be carried out trivially for several classes of cutting planes. A 0-1 knapsack problem is provided as an example. To terminate we consider how the information generated in the course of the algorithms can be used to carry out certain post-optimality analysis.

Keywords: Benders' algorithm, Mixed-integer subproblems, Branch-and-cut, Value function.
AMS 2010 Mathematics Subject Classification: 90C10, 90C11, 49M27, 65K05.

[1]FAU Erlangen-Nürnberg, Discrete Optimization, 91058 Erlangen, Germany, dieter.weninger@fau.de.
[2]CORE, Université catholique de Louvain, laurence.wolsey@uclouvain.be.
This text presents research results of the Belgian Program on Interuniversity Poles of Attraction initiated by the Belgian State, Prime Minister's Office, Science Policy Programming. The scientific responsibility is assumed by the authors.

1. Introduction

In the last 20 years there has been a significant regain of interest in Benders' algorithm [6] designed to treat problems of the form $\min\{cx + hy : Ax + By \geq b, x \in \mathbb{Z}^n_+, y \in \mathbb{R}^p_+\}$ and also in tackling the more complicated problem in which some or all of the y-variables are also integer. The basic idea 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 of a linear programming "Subproblem" in the y-variables with x fixed, which feeds back information to the Master Problem so as to improve the estimate provided by η.

We now indicate some of the new ideas that have been proposed.

i) Since branch-and-cut algorithms have become standard, it is natural to solve the problem using branch-and-cut viewing the Benders' subproblem as a separation or cut-generation problem generating cuts to be added to the Master problem in the (x, η)-space. 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" cuts. These include pareto-optimal cuts (Magnanti and Wong [27], Papadakos [31]), the use of different normalizations to bound the feasible region of the dual (Fischetti et al. [15]), the in-out approach (Ben-Ameur and Neto [5], Fischetti et al. [14]), a partial re-optimization approach (Wentges [39]) and a facet-generating approach (Conforti and Wolsey [8]).

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. Also generating multiple cuts at each iteration has been recommended.

When some y-variables are integer, additional difficulties arise.

iv) First of all, 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 subproblem using Gomory fractional cuts (Gade et at. [16]), lift-and-project cuts (Sen and Higle [34]) or RLT extended formulations (Sherali and Fratelli [37]), the use of IP dual functions (Caroe and Tind [7]) and value functions (Hassanzadeh and Ralphs [23]) and branch-and-cut (Sen and Sherali [35]) among others. Another approach is to incorporate some of the y-variables in the Master problem, see [35], as well as Crainic et al. [11] in which some of the scenario subproblems are included in the Master problem. For the special case of two-level stochastic programs with integer recourse, numerous authors have approached the question of how to accelerate the repeated solution of the large number of related second stage recourse problems containing integer variables.

v) State-of-the-art commercial solvers such as CPLEX [24] and non-commerical solvers such as SCIP [18], see Achterberg [1] and Maher [28], provide interfaces for Benders' decomposition which save programming time for those familiar with the algorithm.

vi) Numerous successful applications have been reported including multicommodity distribution design (Geoffrion and Graves [17]), simultaneous aircraft routing and crew scheduling (Cordeau et al. [9]), fixed-charge network design problems (Costa [10]) and very recently capacitated facility location and covering problems (Fischetti et al. [13, 12]). In particular Rahmaniani et al. [32] provides an extensive literature review of Benders' decomposition, including references to many more applications.

Here we consider the problem in which some or all of the y-variables are integer. Our first suggestion is to take the following viewpoint:

Given a linear program $\min\{cx + hy : Ax + By \geq b, x \in \mathbb{R}^n_+, y \in \mathbb{R}^p_+\}$, *Benders' algorithm is a method to solve such a linear program in which one iterates between solving an LP in the* (x, η)*-space and an LP in the* y*-space.*

This viewpoint leads naturally to two branch-and-cut algorithms that we develop here. In the first algorithm, denoted BCxy (Branch-and-Cut in (x, y)-space), one can branch on both x and y-variables. At each node of the branch-and-cut tree, one uses Benders' approach to solve the linear program:

$$\begin{array}{rcccl} \min & cx & + & hy & \\ & Ax & + & By & \geq & b \\ & Cx & & & \geq & e \\ & \ell & \leq & y & \leq & k \\ & & x \in \mathbb{R}^n_+, y \in \mathbb{R}^p_+ & & & . \end{array}$$

Here the constraints $Cx \geq e$ contain original constraints just involving x-variables and the bounds from branching on x-variables and $k \in \mathbb{Z}^p_+$ and $\ell \in \mathbb{Z}^p_+$ are upper and lower bounds on the y-variables combining possibly original bounds and bounds from branching. Advantages are
i) throughout the algorithm only linear programming relaxations of the Master and Separation problems are solved,
ii) the cuts generated are based on problem structure as opposed to no-good cuts that just cut off the (x, η)-point just examined and
iii) the simplicity of a single pass branch-and-cut algorithm.
The disadvantage is that the enumeration involves both x and y-variables. We note that in [35] the same subproblem is solved to generate cuts, but then y-variables are introduced in the Master Problem and the subproblems solved are MIPs.

The second algorithm, denoted BxCy (Branching in x-space and Cutting in y-space), avoids the need to branch in the (x, y)-space. However the price to pay is that the subproblems must now be solved at least partially as (mixed)-integer programs. This leads to an algorithm in which the original formulation in the (x, y)-space and the subproblem in the y-space are updated by the addition of valid inequalities which are then used to generate new Benders' feasibility and optimality cuts. We show how a variety of cutting planes for the mixed-integer subproblems can be extended/lifted to provide these valid inequalities in the (x, y)-space. This generalizes the approach taken in [16] using Gomory fractional cutting planes.

In the next Section, we describe the algorithm BCxy and present a small instance of the capacitated facility problem with single-sourcing to demonstrate the algorithm. In Section 3 we describe the Algorithm BxCy for problems with $x \in \{0, 1\}^n$ and then indicate how Gomory mixed-integer cuts, 0-1 cover inequalities, Lift-and-Project cuts and flow cover inequalities provide examples of the cuts that can be used in the algorithm. In Section 4 we consider briefly the question of sensitivity analysis and re-optimization for small data changes after using the BCxy algorithm. Specifically the information provided by solving an instance with the BCxy algorithm is used to generate an underestimate of the value function $\phi(d,k,\ell) = \min\{hy : By \geq d, \ell \leq y \leq k, y \in \mathbb{Z}^p\}$. It is also shown how this information can be used to hot-start re-optimization when the data (c,b,A) changes. Finally in Section 5 we discuss a few of the many questions that remain.

2. Algorithm BCxy: Branching in xy-space

For simplicity we assume that all the x and y-variables are integer variables. The problem to be solved is

$$\min\{cx + hy : Ax + By \geq b, x \in \mathbb{Z}^n_+, y \in \mathbb{Z}^p_+\}.$$

At a given node q of the branch-and-cut tree, the associated original problem (OP) is

$$\begin{array}{lccccc} \min & cx & + & hy & & \\ & Ax & + & By & \geq & b \\ & C^q x & & & \geq & e^q \\ & \ell^q & \leq & y & \leq & k^q \\ & & & \multicolumn{3}{l}{x \in \mathbb{Z}^n_+, y \in \mathbb{Z}^p.} \end{array}$$

To solve the linear programming relaxation of OP, one solves a relaxation of the Benders' Master linear program (BM)

$$\begin{array}{lrcll} \zeta = \min & cx + \eta & & & \\ & u_0^s(b - Ax) + u_1^s \ell^q - u_2^s k^q & \leq & \eta & s = 1, \ldots, S \\ & v_0^t(b - Ax) + v_1^t \ell^q - v_2^t k^q & \leq & 0 & t = 1, \ldots, T \\ & C^q x & \geq & e^q & \\ & \multicolumn{3}{c}{x \in \mathbb{R}^n_+, \eta \in \mathbb{R}^1,} & \end{array}$$

where (u_0^s, u_1^s, u_2^s) are some extreme points of

$$\Omega = \{(u_0, u_1, u_2) \in \mathbb{R}^m_+ \times \mathbb{R}^p_+ \times \mathbb{R}^p_+ : u_0 B + u_1 I_p - u_2 I_p = h\}$$

and (v_0^t, v_1^t, v_2^t) are some extreme rays of Ω. (x^*, η^*) denotes its optimal solution. The associated separation subproblem (SP) with x^* fixed is the linear program

$$\phi(x^*) = \min\{hy : By \geq b - Ax^*, y \geq \ell^q, -y \geq -k^q, y \in \mathbb{R}^p\}.$$

The resulting dual linear program (DSP) is

$$\max\{u_0(b - Ax^*) + u_1 \ell^q - u_2 k^q : (u_0, u_1, u_2) \in \Omega\}.$$

2.1 Outline of the BCxy Algorithm

We now discuss the different possibilities.

If DSP is unbounded, then SP is infeasible. Let (v_0^*, v_1^*, v_2^*) be the associated unbounded extreme ray with $v_0^*(b - Ax^*) + v_1^* \ell^q - v_2^* k^q > 0$.
i) If $v_0^* A \neq 0$, a feasibility cut $v_0^*(b - Ax) + v_1^* \ell^q - v_2^* k^q \leq 0$ cutting off x^* is added to BM.

ii) If $v_0^* A = 0$, the feasible set of SP is empty whatever the choice of x. So the node can be pruned.

If DSP has a finite optimal value $\phi(x^*)$, let y^* be the optimal primal solution and (u_0^*, u_1^*, u_2^*) the optimal dual solution.

i) If $\phi(x^*) > \eta^*$, the optimality cut $u_0^*(b - Ax) + u_1^* \ell^q - u_2^* k^q \leq \eta$ cutting off (x^*, η^*) is added to BM.

ii) If $\phi(x^*) = \eta^*$, OP is solved with optimal value $\zeta^* = cx^* + \eta^*$. There are now three cases.

a) If $(x^*, y^*) \in \mathbb{Z}_+^n \times \mathbb{Z}^p$, a new feasible solution has been found. The incumbent is updated. The node is pruned by optimality.

b) If $x^* \notin \mathbb{Z}^n$, one branches on a variable with $x_j^* \notin \mathbb{Z}^1$. Two new nodes are created with the new constraint sets $C^q x \geq e^q, x_j \leq \lfloor x_j^* \rfloor$ and $C^q x \geq e^q, x_j \geq \lceil x_j^* \rceil$ respectively in the branch-and-cut tree for the Benders' Master.

c) If $x^* \in \mathbb{Z}^n$ but $y^* \notin \mathbb{Z}_+^p$, one branches on a variable with $y_j^* \notin \mathbb{Z}^1$. Two new nodes are created with updated constraint sets, one in which $y_j \geq \ell_j^q$ is replaced by $y_j \geq \lceil y_j^* \rceil$ and the other in which $y_j \leq k_j^q$ is replaced by $y_j \leq \lfloor y_j^* \rfloor$.

Note that the set Ω never changes so only the objective function in DSP 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 in any node of the branch-and-cut tree. By treating varying bounds in the subproblem, the set Ω obtains larger sets of extreme points and extreme rays than the set $\Omega' = \{u_0 \in \mathbb{R}_+^m : u_0 B \leq h\}$. These provide additional cuts based on the problem structure for the x-space BM. Observe also that it is easy to incorporate many of the ideas for speeding up Benders' algorithm such as including valid inequalities for $\{x \in \mathbb{Z}_+^n : Cx \geq d\}$ in BM, and the selection of better dual solutions in the subproblems.

An Example: Capacitated Facility Location with Single-Sourcing

To demonstrate algorithm BCxy we consider a capacitated facility location problem (CFL) in which some clients require single-sourcing. The problem is as follows: There are n 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 a per unit transportation cost g_{ij} between client i and facility j. 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, one obtains the formulation

$$
\begin{array}{rcll}
\min \sum_{j=1}^{n} c_j x_j & + & \sum_{i=1}^{m} \sum_{j=1}^{n} h_{ij} y_{ij} & \\
\sum_{i=1}^{m} a_i y_{ij} & \leq & K_j x_j & j = 1, \ldots, n \\
\sum_{j=1}^{n} y_{ij} & = & 1 & i = 1, \ldots, m \\
y_{ij} & \leq & x_j & i = 1, \ldots, m, \quad j = 1, \ldots, n \\
\sum_{j=1}^{n} K_j x_j & \geq & \sum_{i=1}^{m} a_i & \\
\multicolumn{4}{l}{x \in \{0,1\}^n, y \in \mathbb{R}_+^{mn}, y_{ij} \in \{0,1\} \text{ for } i \in S}
\end{array}
$$

where $h_{ij} = a_i g_{ij}$ is the cost of shipping a_i units from facility j to client i.

Example 1 We present part of a possible branch-and-cut tree for the following instance:

```
ssy = [0,0,0,0,0,1,1,1,1,1]
c   = [50,64,37,49,65]
a   = [10,4,12,32,16,22,11,31,32,25]
K   = [60,80,80,70,60]
g   = [2,1,5,3,3,2,2,2,3,4,5,2,3,4,3,4,3,3,2,2,3,2,4,2,3,3,5,5,3,4,3,1,4,3,2,4,3,5,3,2,3,3,4,4,3,5,2,2,4,4]
```

Start with the initial Master LP (BM):

$$
\begin{aligned}
\zeta = \min cx + \eta & & \\
\sum_{j=1}^{n} K_j x_j & \geq & \sum_{i=1}^{m} a_i \\
\sum_{j=1}^{n} x_j & \geq & 3 \\
x \in [0,1]^n, \quad \eta \in \mathbb{R}^1 & &
\end{aligned}
$$

where the inequality $\sum_{j=1}^{n} x_j \geq 3$ is clearly valid as $\sum_{i=1}^{m} a_i = 195$ and $\max_{j\in[1,n]} K_j = 80$.

The initial separation problem (SP) takes the form:

$$
\begin{array}{rcll}
\min \sum_{i=1}^{m}\sum_{j=1}^{n} a_i g_{ij} y_{ij} & & & \\
\sum_{j=1}^{m} y_{ij} & = & 1 & i = 1,\ldots,m \\
-\sum_{i=1}^{m} a_i y_{ij} & \geq & -K_j x_j^* & j = 1,\ldots,n \\
y_{ij} & \geq & 0 & i = 1,\ldots,m, \quad j = 1,\ldots,n \\
-y_{ij} & \geq & -x_j^* & i = 1,\ldots,m, \quad j = 1,\ldots,n \\
y & \in & \mathbb{R}^{mn}. &
\end{array}
$$

Thus initially the bounds are $\ell_{ij} = 0$ and $k_{ij} = x_j^*$ for all i, j.

The steps of the algorithm are shown in Table 1 below including the value of the BM at each node, the corresponding x solution, as well as the number of cuts needed to solve the BM. In Figure 1 the complete branch-and-cut tree is shown. The algorithm terminates after 9 nodes with an optimal solution $x = (1,1,0,0,1)$, $y_{12} = 1$, $y_{21} = 0.25$, $y_{22} = 0.75$, $y_{32} = 1$, $y_{42} = 0.094$, $y_{45} = 0.906$, $y_{52} = 1$, $y_{61} = 1$, $y_{72} = 1$, $y_{85} = 1$, $y_{91} = 1$, $y_{10,2} = 1$ of total cost 605.

A list of the extreme points u_0^s and extreme rays v_0^t of Ω generated at each node is given in the appendix. The values of (u_1^s, u_2^s) and (v_1^t, v_2^t) follow from the equations defining Ω.

3. Algorithm BxCy: Branching in x-space, Cutting in y-space

Here we present a branch-and-cut algorithm based on Benders' solution of the linear programs in which the branching in the Master problem only involves the x-variables. We require that $x \in \{0,1\}^n$ and we assume for simplicity of presentation that all the y-variables are integer.

node q	ζ^q	x^*	Comments
1	136	(1,0,1,1,0)	Initial solution of BM.
			9 optimality cuts added.
	601	(0,1,0,1,1)	$y_{92} = 0.09375, y_{95} = 0.90625$, else $y_{ij} \in \{0,1\}$
			Branch on y_{95}
2			Branch $y_{95} = 0$. Add $y_{95} = 0$ in SP. 3 optimality cuts added
	601.7	(0.7,1,0,0.3,1)	Branch on x_1
3			Branch: $y_{95} = 0$, $x_1 = 1$. Add $x_1 = 1$ in BM
			3 optimality cuts added.
	605	**(1,1,0,0,1)**	OP feasible. Update incumbent $\bar{z} = 605$
			Prune and Backtrack.
4			Branch $y_{95} = 0$, $x_1 = 0$. Set $x_1 = 0$ in BM.
			Keep all 15 cuts for $y_{95} = 0$ in BM.
	602.4	(0,1,0.48,1,0.52)	Branch on x_5.
5			Branch $y_{95} = 0, x_1 = 0, x_5 = 1$. No cuts added.
	605*	(0,1,0.4,0.6,1)	Prune by bound. Backtrack
6			Branch $y_{95} = 0, x_1 = 0, x_5 = 0$. No cuts added.
	620*	(0,1,1,1,0)	Prune by bound. Backtrack.
7			Branch $y_{95} = 1$. Set $y_{95} = 1$ in SP.
			2 Optimality cuts added
	604	(0,1,0,1,1)	$y_{85} = 0.903, y_{82} = 0.097$, else $y_{ij} \in \{0,1\}$
			Branch on y_{85}
8			$y_{95} = 1$, $y_{85} = 1$ Set $y_{95} = y_{85} = 1$ in SP.
			1 feasibility cut added
	$+\infty$		BM infeasible. Prune and backtrack.
9			$y_{95} = 1$, $y_{85} = 0$ Set $y_{95} = 1$, $y_{85} = 0$ in SP
			1 optimality cut added
	614.8*	(0.0.36,0.64,1,1)	Prune by bound.
			Search completed.

Table 1: CFL with Single-Sourcing. * indicates lower bound on ζ^q.

The main observation is:

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

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.

Thus we work with the original problem to which a set of valid inequalities $\Pi x + \Theta y \geq \Pi_0$ have been added, denoted (OP):

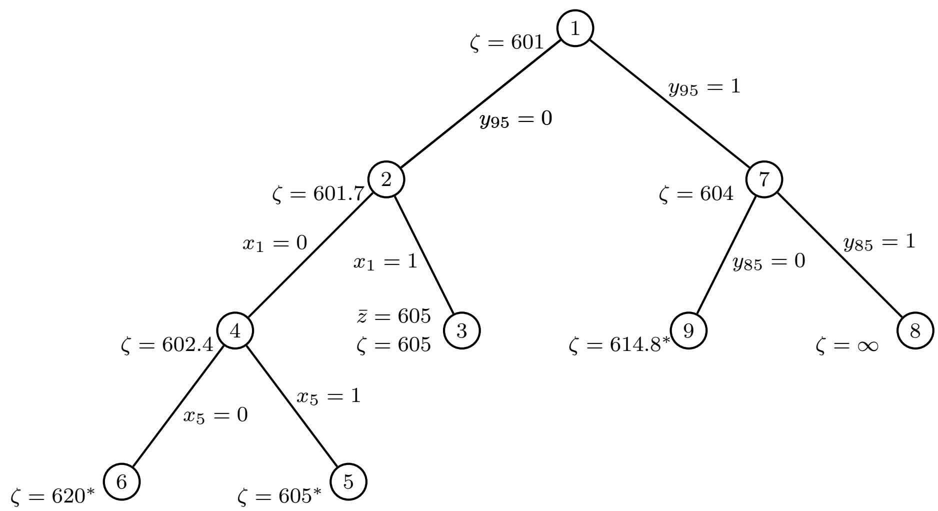

Figure 1: Branch-and-cut tree for Benders' algorithm

$$
\begin{array}{rlcl}
z = \min & cx + hy & & \\
& Ax + By & \geq & b \\
& \Pi x + \Theta y & \geq & \Pi_0 \\
& Cx & \geq & e \\
& x \in \{0,1\}^n, y \in \mathbb{Z}_+^p. & &
\end{array}
$$

This leads us to the Benders' subproblem $\mathrm{SP}^I(x^*)$

$$
\begin{array}{rlcl}
\phi^I(x^*) = \min & hy & & \\
& By & \geq & b - Ax^* \\
& \Theta y & \geq & \Pi_0 - \Pi x^* \\
& y & \in & \mathbb{Z}_+^p
\end{array}
$$

with its linear programming relaxation $\mathrm{SP}(x^*)$ of value $\phi(x^*)$.

Finally we work with the linear programming relaxation (BM) of the Benders' Master Problem

$$
\begin{array}{rcll}
\zeta_{LP} & = \min cx + \eta & & \\
u^s(b - Ax) + \tilde{u}^s(d - \Pi x) & \leq & \eta & s = 1, \ldots, S \\
v^t(b - Ax) + \tilde{v}^t(d - \Pi x) & \leq & 0 & t = 1, \ldots, T \\
Cx & \geq & e & \\
& x \in [0,1]^n, \eta \in \mathbb{R}^1, & &
\end{array}
$$

where $\{(u^s, \tilde{u}^s)\}_{s=1}^S$ and $\{(v^t, \tilde{v}^t)\}_{t=1}^T$ are subsets of the extreme points and extreme rays of the dual region $\Omega = \{(u, \tilde{u}) : uB + \tilde{u}\Theta \leq h, (u, \tilde{u}) \geq 0\}$.

3.1 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 $x^* \in \{0,1\}^n$. At a given node, we define an iteration of the Benders' Master (BM) problem.

Solve the linear program BM. Solution (x^*, η^*). $\zeta_{LP} = cx^* + \eta^*$.
Solve SP(x^*) with optimal solution y^*, dual solution $(u^*, \tilde{u}^*)$, $\phi(x^*) = hy^* = u^*(b - Ax^*) + \tilde{u}^*(\Pi_0 - \Pi x^*)$. There are six cases.

1. $x^* \notin \mathbb{Z}^n_+$. $\eta^* = \phi(x^*)$.
   The linear programming relaxation of OP is solved. Branch on a variable x_j with $x_j^* \notin \mathbb{Z}^1_+$. Create two new nodes.

2. $x^* \notin \mathbb{Z}^n_+$. $\eta^* < \phi(x^*)$.
   Add a standard Benders' infeasibility cut $v^*(b - Ax) + \tilde{v}^*(\Pi_0 - \Pi x) \leq 0$ or optimality cut $u^*(b - Ax) + \tilde{u}^*(\Pi_0 - \Pi x) \leq \eta$ to BM and return to BM.

3. $x^* \in \mathbb{Z}^n_+$. $y^* \in \mathbb{Z}^p_+$. $\eta^* < \phi(x^*) = \phi^I(x^*)$.
   Feasible solution found. If best so far, update of the incumbent. Add an optimality cut $u^*(b - Ax) + \tilde{u}^*(\Pi_0 - \Pi x) \leq \eta$ to BM and return to BM.

4. $x^* \in \mathbb{Z}^n_+$. $y^* \in \mathbb{Z}^p_+$. $\eta^* = \phi(x^*) = \phi^I(x^*)$.
   Prune the node of BM by optimality.

5. $x^* \in \mathbb{Z}^n_+$. $y^* \notin \mathbb{Z}^p_+$. $\eta^* = \phi(x^*) \leq \phi^I(x^*)$.
   Generate an (x, y)-space cut $\theta y \geq \pi_0 - \pi x$ cutting off the point (x^*, y^*).
   Add the cut to OP and SP(x^*). Return to SP(x^*).

6. $x^* \in \mathbb{Z}^n_+$. $y^* \notin \mathbb{Z}^p_+$. $\eta^* < \phi(x^*)$
   Here there are two options:
   Option 1. Add an infeasibility or optimality cut to BM as in case 2. Return to BM.
   Option 2. Generate an (x, y)-space cut as in case 5. Return to SP(x^*).

Here we consider how to generate an (x, y)-space cut from the subproblem SP$^I(x^*)$ in cases 5 and 6.

Let $\bar{x}_j = 1 - x_j$ for $j = \{1, \ldots, n\}$, $N_0 = \{j : x_j^* = 0\}$, $N_1 = \{j : x_j^* = 1\}$, $\tilde{b} = b - \sum_{j \in N_1} a_j$ and $\tilde{\Pi}_0 = \Pi_0 - \sum_{j \in N_1} \Pi_j$. Note that $-\sum_{j \in N_0} a_j x_j + \sum_{j \in N_1} a_j \bar{x}_j = 0$ and $-\sum_{j \in N_0} \Pi_j x_j + \sum_{j \in N_1} \Pi_j \bar{x}_j = 0$.
SP(x^*) can now be rewritten as:

$$
\begin{aligned}
\phi(x^*) = \min hy & \\
By &\geq \tilde{b} - \sum_{j \in N_0} a_j x_j + \sum_{j \in N_1} a_j \bar{x}_j \\
\Theta y &\geq \tilde{\Pi}_0 - \sum_{j \in N_0} \tilde{\Pi}_j x_j + \sum_{j \in N_1} \tilde{\Pi}_j \bar{x}_j \\
x &= x^* \\
y &\in \mathbb{R}^p_+.
\end{aligned}
$$

Generate a valid inequality $\theta y \geq \pi_0$ for SP$^I(x^*)$ cutting off y^* and extend it to a valid inequality $\theta y \geq \tilde{\pi}_0 - \sum_{j \in N_0} \pi_j x_j + \sum_{j \in N_1} \pi_j \bar{x}_j$ for OP cutting off (x^*, y^*).

If no suitable cuts can be found for SP$^I(x^*)$ and no suitable optimality cut can be returned, solve SP$^I(x^*)$ to optimality by some algorithm , and return a no-good feasibility $\sum_{j \in N_0} x_j +$

$\sum_{j\in N_1}(1-x_j) \geq 1$ or no-good optimality cut $\eta \geq \phi^I(x^*)-(\phi^I(x^*)-L)(\sum_{j\in N_0} x_j+\sum_{j\in N_1}(1-x_j))$ to BM where L is a lower bound on $\min\{\phi^I(x) : x \in \{0,1\}^n\}$.

Below we indicate some of the various pure integer and mixed integer inequalities that can be used to generate (x,y)-space cuts in Algorithm BxCy. For ease of description, we suppose that the cut is the first cut generated for $\mathrm{SP}^I(x^*)$.

3.2 Cuts for All-Integer Subproblems

Gomory Fractional Cuts (Gomory [19])
Solving the linear programming relaxation of $\mathrm{SP}^I(x^*)$, with LP solution $y^* \notin \mathbb{Z}^p$, rows of the optimal LP tableau take the form:

$$\sum_{j=1}^{p} \bar{b}_{ij} y_j + \sum_{i=1}^{m} \bar{\sigma}_i s_i = \bar{b}_i - \sum_{j\in N_0} \bar{a}_{ij} x_j + \sum_{j\in N_1} \bar{a}_{ij}\bar{x}_j, x = x^*$$

where $s = Ax + By - b \geq 0$ are the slack variables. If the basic variable in the row takes value $\bar{b}_i \notin \mathbb{Z}^1$, generate a fractional Gomory cut

$$\sum_{j=1}^{p} \lfloor \bar{b}_{ij} \rfloor y_j + \sum_{i=1}^{m} \lfloor \bar{\sigma}_i \rfloor s_i \leq \lfloor \bar{b}_i \rfloor - \sum_{j\in N_0} \lfloor \bar{a}_{ij} \rfloor x_j - \sum_{j\in N_1} \lfloor -\bar{a}_{ij} \rfloor \bar{x}_j$$

cutting off y^* when $x = x^*$.

0-1 Extended Cover Inequalities (Balas [3], Hammer et al. [22], Wolsey [40])
Suppose that some row of $\mathrm{SP}^I(x^*)$ is a 0-1 knapsack constraint of the form:

$$\sum_{j\in P} b_j y_j \leq d - \sum_{j\in N_0} a_j x_j + \sum_{j\in N_1} a_j \bar{x}_j, x = x^*, y \in \{0,1\}^n$$

and that the extended minimal cover inequality with cover $C \subset P$ is violated by y^*, namely the inequality

$$\sum_{j\in E(C)} y_j \leq |C| - 1$$

where $E(C) = C \cup \{i \notin C : b_i \geq \max_{j\in C} b_j\}$. Thus $\sum_{j\in C} b_j = d + \lambda$ with $\lambda > 0$ and $\lambda < b_j$ for $j \in C$. Let $\bar{b} = \max_{j\in C} b_j$. Adding $y_j \leq 1$ with weight $\bar{b} - b_j$ for $j \in C$ to the knapsack constraint and dividing by $\bar{b}$ gives

$$\sum_{j\in C} y_j + \sum_{j\in P\setminus C} \frac{b_j}{\bar{b}} y_j \leq |C| - \frac{\lambda}{\bar{b}} - \sum_{j\in N_0} \frac{a_j}{\bar{b}} x_j + \sum_{j\in N_1} \frac{a_j}{\bar{b}} \bar{x}_j.$$

Now taking a fractional Gomory cut gives the inequality

$$\sum_{j\in C} y_j + \sum_{j\in P\setminus C} \lfloor \frac{b_j}{\bar{b}} \rfloor y_j \leq |C| - 1 - \sum_{j\in N_0} \lfloor \frac{a_j}{\bar{b}} \rfloor x_j - \sum_{j\in N_1} \lfloor \frac{-a_j}{\bar{b}} \rfloor \bar{x}_j$$

that is at least as strong as the extended cover inequality and cuts off y^* when $x = x^*$.

Example 2 We demonstrate Algorithm BxCy on a small 0-1 knapsack problem:

$$\begin{aligned} \max 1x_1 + 4x_2 + 6x_3 + 8y_1 + 9y_2 + 11y_3 + 13y_4 & \\ 1x_1 + 2x_2 + 3x_3 + 6y_1 + 7y_2 + 8y_3 + 11y_4 &\leq 16 \\ x \in \{0,1\}^3, y \in \{0,1\}^4. & \end{aligned}$$

We initialize BM with an upper bound $\eta \leq \sum_{j=1}^{4} h_j = 41$.
Node 1. Iteration 1.
BM: $\zeta_{LP} = 52, x^* = (1,1,1), \eta^* = 41$.
SP(x^*): Solving the LP relaxation gives $\phi(x^*) = 13\frac{2}{3} < \eta^*$. The dual variables are $\frac{4}{3}$ on the knapsack constraint is and $\frac{1}{3}$ on the bound constraint $y_3 \leq 1$.
Case 2. The optimality cut

$$\eta \leq \frac{4}{3}(16 - x_1 - 2x_2 - 3x_3) + \frac{1}{3}$$

is added to BM.
Node 1. Iteration 2.
BM: $\zeta_{LP} = 25, x^* = (0,1,1), \eta^* = 15$.
SP: $\phi(x^*) = 15$. $y^* = (0.5, 0, 1, 0)$.
BM is solved, $x^* = (0,1,1)$, but $y^* \notin \{0,1\}^4$.
Case 5. The subproblem SP$^I(x^*)$ now takes the form:

$$\begin{aligned} \phi^I(x^*) = \max 8y_1 + 9y_2 + 11y_3 + 13y_4 & \\ 6y_1 + 7y_2 + 8y_3 + 11y_4 &\leq 11 - x_1 + 2\bar{x}_2 + 3\bar{x}_3 \\ x = x^*, y \in \{0,1\}^4. & \end{aligned}$$

Adding a Gomory Fractional Cut
Consider the optimal LP tableau of SP(x^*) after adding s as a slack variable. Let $\bar{y}_3 = 1 - y_3$. The row in which y_1 is basic is:

$$y_1 + \frac{7}{6}y_2 - \frac{8}{6}y_3 + \frac{11}{6}y_4 + \frac{1}{6}s = \frac{3}{6} - \frac{1}{6}x_1 + \frac{2}{6}\bar{x}_2 + \frac{3}{6}\bar{x}_3.$$

The Gomory fractional cut on this row is

$$y_1 + y_2 - 2\bar{y}_3 + y_4 + 0s \leq 0 + \bar{x}_2 + \bar{x}_3.$$

Add the Gomory fractional cut

$$y_1 + y_2 + 2y_3 + y_4 + x_2 + x_3 \leq 4. \tag{c1}$$

to OM and SP(x^*).
Solving SP with (c1) gives $\phi(x^*) = 14\frac{2}{3}$, $y^* = (1, \frac{1}{3}, \frac{1}{3}, 0) \notin \mathbb{Z}_+^4$.
Case 6. Choose Option 2 to generate an (x,y)-cut.
Adding a Cover Inequality
Now the extended knapsack cover inequality with minimal cover $C = \{1,2\}$ is violated. With $\lambda = 2$ and $\bar{a} = 7$, this leads to the inequality

$$y_1 + y_2 + \frac{8}{7}y_3 + \frac{11}{7}y_4 \leq |C| - \frac{2}{7} - \frac{1}{7}x_1 + \frac{2}{7}\bar{x}_2 + \frac{3}{7}\bar{x}_3.$$

The extended cover inequality

$$y_1 + y_2 + y_3 + y_4 + x_2 + x_3 \le 3 \tag{c2}$$

is added to OP and SP(x^*). Solving SP with (c1) and (c2) gives $\phi(x^*) = 13$, $y^* = (0,0,0,1)$. As $y^* \in \{0,1\}^4$, $x^* = (0,1,1), y^* = (0,0,0,1)$ is a new incumbent solution of value $\underline{\zeta} = 23$. The dual variable on the cover cut is 13. All other dual variables are 0.

Case 3. The optimality cut

$$\eta \le 13(3 - x_2 - x_3)$$

is now added to BM.

Node 1. Iteration 3

BM: $\zeta_{LP} = 24.741935, x^* = (0, 0.806452, 1)$ $\eta^* = 15.5161$,

SP(x^*) with (c1) and (c2): $\phi'(x^*) = 14.354836 < \eta^*$ with dual variables $\frac{2}{3}$ on the knapsack constraint and $\frac{17}{3}$ on the cover inequality (c2).

Case 2. The optimality cut

$$\eta \le \frac{2}{3}(16 - x_1 - 2x_2 - 3x_3) + \frac{17}{3}(3 - x_2 - x_3)$$

is added to BM.

Node 1. Iteration 4

BM: $\zeta^* = 24.384615$, $x^* = (0, 0.538462, 1), \eta^* = 16.2308$

SP(x^*): $\phi(x^*) = 15.8461524$ with dual variables 1 on the knapsack constraint, 2 on the cover inequality (c2) and 1 on the bound $y_3 \le 1$.

Case 2. The optimality cut

$$\eta \le (16 - x_1 - 2x_2 - 3x_3) + 2(3 - x_2 - x_3) + 1$$

is added to BM.

Node 1. Iteration 5

BM: $\zeta_{LP} = 24$, $x^* = (0, \frac{2}{3}, 1), \eta^* = 15\frac{1}{3}$

SP(x^*): $\phi(x^*) = 15\frac{1}{3}$.

The linear programming relaxation of BM is solved. $x^* \notin \{0,1\}^3$.

Case 1. Need to branch on x_2.

Node 2. Branch $x_2 = 0$. Add constraint $x_2 = 0$ to BM.

$\zeta_{LP} = 23\frac{2}{3}$. Prune by bound as objective function is integer valued.

Node 3. Branch $x_2 = 1$. Add constraint $x_2 = 1$ to BM.

$\zeta_{LP} = 23\frac{3}{4}$. Prune by bound as objective function is integer valued.

All nodes are pruned. So the incumbent $x^* = (0,1,1), y^* = (0,0,0,1)$ of value 23 is optimal.

3.3 Cuts for Mixed-Integer Subproblems

Here we indicate some valid inequalities that can be used for mixed-integer subproblems.

Gomory Mixed Integer Cuts (Gomory [20])

As for the Gomory fractional cut above, Gomory mixed integer cuts can be generated off a row of the optimal LP tableau and the x-variable coefficients are obtained automatically.

Mixed Integer Rounding Inequalities (Nemhauser and Wolsey [29])
Suppose that the subproblem $\mathrm{SP}^I(x^*)$ involves both continuous P^C and integer P^I variables with either an original constraint or tableau constraint:

$$\sum_{j\in P} b_j y_j \leq b - \sum_{j\in N_0} a_j x_j + \sum_{j\in N_1} a_j \bar{x}_j.$$

Using the function $F_\alpha : \mathbb{R}^1 \to \mathbb{R}^1$ satisfying $F_\alpha(d) = \lfloor d \rfloor + \frac{(f_d-\alpha)^+}{(1-\alpha)}$, $\overline{F}_\alpha(d) = \min(0, \frac{d}{1-\alpha})$, the MIR inequality is:

$$\sum_{j\in P^I} F_\alpha(b_j) y_j + \sum_{j\in P^C} \overline{F}_\alpha(b_j) y_j \leq F_\alpha(b) - \sum_{j\in N_0} F_\alpha(a_j) x_j - \sum_{j\in N_1} F_\alpha(-a_j)\bar{x}_j$$

where $\alpha = b - \lfloor b \rfloor$ the fractional part of b and $(f_d - \alpha)^+ = \max(f_d - \alpha, 0)$. It is well known that this corresponds to the Gomory mixed integer cut when when applied to a row of the LP tableau.

Lift-and-Project Cuts (Balas et al. [4])
Suppose that $\mathrm{SP}(x^*)$ has constraints: $By \geq \tilde{b} - \sum_{j\in N_0} a_j x_j + \sum_{j\in N_1} a_j \bar{x}_j$ with solution $y^* \notin \mathbb{Z}^p$. Select y_j for which $y_j^* \notin \mathbb{Z}^1$. Solve the lift-and-project LP

$$\begin{array}{lll} \min & \gamma y^* - \gamma_0 & \\ & \gamma - uB + \tilde{u} e_j & \geq \; 0 \\ & \gamma - vB - \tilde{v} e_j & \geq \; 0 \\ & \gamma_0 - u\tilde{b} + \tilde{u}\lfloor y_j^* \rfloor & \leq \; 0 \\ & \gamma_0 - v\tilde{b} - \tilde{v}(\lfloor y_j^* \rfloor + 1) & \leq \; 0 \\ & \sum_{i=1}^m u_i + \tilde{u} + \sum_{i=1}^m v_i + \tilde{v} & = \; 1 \\ & u, v \in \mathbb{R}^m_+, \tilde{u}, \tilde{v} \in \mathbb{R}^1_+, & \end{array}$$

where e_j denotes the j-th unit vector. Let $(\gamma^*, \gamma_0^*, u^*, \tilde{u}^*, v^*, \tilde{u}^*)$ be an optimal solution. The part $(u^*, \tilde{u}^*)$ is associated with the disjunction $-y_j \geq -\lfloor y_j^* \rfloor$ and $(v^*, \tilde{v}^*)$ is associated with the disjunction $y_j \geq \lfloor y_j^* \rfloor + 1$. This gives the inequality

$$\gamma^* y \geq \gamma_0^* + u^*(-\sum_{j\in N_0} a_j x + \sum_{j\in N_1} a_j \bar{x}_j)$$

for $y_j \leq \lfloor y_j^* \rfloor$ and

$$\gamma^* y \geq \gamma_0^* + v^*(-\sum_{j\in N_0} a_j x + \sum_{j\in N_1} a_j \bar{x}_j)$$

for $y_j \geq \lfloor y_j^* \rfloor + 1$ and thus the resulting globally valid inequality that cuts off y^* when $x = x^*$ is

$$\gamma^* y \geq \gamma_0^* + \sum_{j\in N_0} \min[-u^* a_j, -v^* a_j] x_j + \sum_{j\in N_1} \min[u^* a_j, v^* a_j] \bar{x}_j.$$

Flow-cover Inequalities (Padberg et al. [30])
Suppose that the subproblem $\mathrm{SP}^I(x^*)$ contains constraints of the form

$$\begin{array}{rcl} \sum_{j\in P_+} y_j - \sum_{j\in P^-} y_j & \leq & b - \sum_{j\in N_0} a_j x_j + \sum_{j\in N_1} a_j \bar{x}_j \\ y_j & \leq & m_j z_j \quad j \in P \\ x & = & x^* \\ y & \in & \mathbb{R}^p_+, z \in \{0,1\}^p. \end{array} \tag{1}$$

Different flow cover inequalities have been proposed for such sets. We present one such inequality from Van Roy and Wolsey [38]. The lifting follows from the validity of the inequality. To describe the inequality, let $C_+ \subseteq P_+, C_- \subseteq P_-$, $\lambda = \sum_{j\in C_+} m_j - \sum_{j\in C_-} m_j - b > 0$, $L \subseteq P_- \setminus C_-$. Then the flow cover inequality

$$\sum_{j\in C_+} y_j + \sum_{j\in C_+} (m_j - \lambda)^+(1 - z_j) \leq b + \sum_{j\in C_-} m_j + \lambda \sum_{j\in L} z_j + \sum_{j\in P_-\setminus(C_-\cup L)} y_j$$

is a valid inequality for $\mathrm{SP}^I(x^*)$.

Rewriting (1) so that all the $x_j, \bar{x}_j$ terms have a positive coefficient, we obtain.

$$\sum_{j\in P_+} y_j - \sum_{j\in P^-} y_j + \sum_{j\in N_0:a_j>0} a_j x_j + \sum_{j\in N_1:a_j<0} (-a_j)\bar{x}_j \leq b + \sum_{j\in N_0:a_j<0} (-a_j)x_j + \sum_{j\in N_1:a_j>0} a_j\bar{x}_j$$

To extend it to a valid inequality including the x-variables, we observe that for y_j for $j \in P_+\setminus C_+$, the coefficient in the inequality is 0. On the other hand for $j \in P_- \setminus C_-$, the term obtained is either y_j or λz_j. Taking $y_j = |a_j|x_j$ or $y_j = |a_j|\bar{x}_j$, the resulting inequality is

$$\begin{aligned} \sum_{j\in C_+} y_j + \sum_{j\in C_+} (m_j - \lambda)^+(1 - z_j) \leq b + \sum_{j\in C_-} m_j + \lambda \sum_{j\in L} z_j + \sum_{j\in P_-\setminus(C_-\cup L)} y_j \\ + \sum_{j\in N_0:a_j<0} \min[\lambda, (-a_j)]x_j + \sum_{j\in N_1:a_j>0} \min[a_j, \lambda]\bar{x}_j \end{aligned}$$

Example 3 Consider the instance

$$\begin{aligned} \zeta = \max x_1 + x_2 + 2y_1 + 10y_2 + 2y_3 - 4y_4 - 5z_1 - 2z_2 - 2z_3 - z_4 \\ 11x_1 + 3x_2 + y_1 + y_2 - y_3 - y_4 = 15 \\ y_1 \leq z_1, y_2 \leq 2z_2, y_3 \leq 5z_3, y_4 \leq 4z_4 \\ x \in \{0,1\}^2, y \in \mathbb{R}^4_+, z \in \{0,1\}^4 \end{aligned}$$

in which we separate the x and (y, z) variables and apply Algorithm BxCy. Here the subproblem is mixed-integer.

Node 1 The initial relaxed Master problem BM is:

$$\begin{aligned} \zeta_{LP} = \max x_1 + x_2 + \eta \\ x \in [0,1]^2, \eta \leq 30 \end{aligned}$$

with optimal solution $x^* = (1,1), \eta^* = 30$.

Now the linear programming relaxation of $\mathrm{SP}^I(x^*)$ is the problem

$$\begin{aligned} \phi(x^*) = \max 2y_1 + 10y_2 + 2y_3 - 4y_4 - 5z_1 - 2z_2 - 2z_3 - z_4 & \\ y_1 + y_2 - y_3 - y_4 = 1 + 11\bar{x}_1^* + 3\bar{x}_2^* & \quad (2) \\ y_1 \leq z_1, y_2 \leq 2z_2, y_3 \leq 5z_3, y_4 \leq 4z_4 & \quad (3) \\ y \in \mathbb{R}^4_+, z \in [0,1]^4 & \end{aligned}$$

with solution $\phi'(x^*) = 19.6$, $y^* = (0,2,1,0)$, $z = (0,1,0.2,0) \notin \{0,1\}^4$.

Case 6. Choose the option of generating an (x,y)-inequality.
The flow cover inequality with $C_+ = \{2\}, C_- = \emptyset$, $\lambda = 1$, $L = \{3\}$ giving

$$y_2 + 1(1 - z_2) \leq 1 + 1z_3 + y_4 + 1\bar{x}_1 + 1\bar{x}_2 \tag{4}$$

is violated by 0.8. The inequality is added to OM and SP(x^*).
Solving SP(x^*) gives $\phi(x^*) = 18, y = (0,2,1,0), z = (0,1,1,0)$. This is a feasible solution giving an incumbent value of $\underline{\zeta} = 20$. The dual solution is: -2 on constraint (2), (5,5.75,0,0.25) on the constraints (3), (0,15.75,4.25,0) on the $z_j \leq 1$ constraints and 6.25 on the flow cover inequality (4).
Case 3. The optimality cut

$$\eta \leq 2.5 + 15.75x_1 - 0.25x_2$$

is added to BM.
The new optimal solution of BM is: $\zeta_{LP} = 20$, $x^* = (1,1)$, $\eta^* = 18$.
As lower bound $\underline{\zeta}$ and upper bound ζ_{LP} are equal, the incumbent solution is optimal.

Other stronger variants of flow cover inequalities can be found in Gu et al. [21] and Louveaux and Wolsey [26]. These can be lifted by inspection in the same way as above. Atamtürk [2] also develops inequalities for mixed integer knapsack sets. Other cuts that can be used include lift-and-project cuts, see Balas et al. [4], Sen and Higle [34], or other cuts obtained from the projection of an extended formulation, such as RLT [36].

The choice made above to restrict Algorithm BxCy to instances with $x \in \{0,1\}^n$ follows from the fact that it is always possible to lift a valid inequality $\theta y \geq \pi_0$ into a valid inequality $\theta y \leq \tilde{\pi}_0 - \pi x$. For obvious reasons of ease of computation we have chosen to use inequalities for which the lifting is essentially by inspection. With other inequalities, this lifting may involve solving MIP problems to obtain the coefficients of the x-variables. On the other hand lifting of an inequality may no longer be possible when x are more general integer variables. For instance, if $x \in [0,k]^n$ and x_j^* is an integer strictly between the bounds, Wolsey [41] shows how to calculate the potential lifting coefficients. However even with $n = p = 1$, it is easy to produce an inequality in the y_1-space that has no valid lifting coefficient for x_1.

4. The estimated value function and sensitivity analysis

Here we show how the branch-and-cut tree from algorithm BCxy, represented by a rooted digraph $D = (N, A)$, can be used both to estimate the value function $\phi(d,k,\ell) = \min\{hy : By \geq d, \ell \leq y \leq k, y \in \mathbb{Z}^p\}$ with $\ell, k \in \mathbb{Z}^p_+$ and to provide a hot start if one then wishes to optimize a modified instance in which the data (c,b,A) is changed.

4.1 Underestimates of the Value Function ϕ

Let $\phi_t(d, k^t, \ell^t) = \min\{hy : By \geq d, \ell^t \leq y \leq k^t, y \in \mathbb{R}^p\}$ where the bounds ℓ^t and k^t come from initial bounds on y and the branching constraints leading to node $t \in N$. Thus $\phi = \phi_1$ where node 1 is the root of the tree. Let U_t, V_t denote the indices of the sets of extreme points and extreme rays generated at node t and $U^* = \cup_{t \in N} U_t$ and $V^* = \cup_{t \in N} V_t$ be all the extreme points and extreme rays of Ω generated in the course of the algorithm. Given an extreme ray

$(v_0^i, v_1^i, v_2^i) \in \Omega$ with $i \in V_t$ generated at node t, the feasibility constraint associated to this ray is $v_0^i d + \beta \leq 0$ where $\beta = v_1^i \ell^t - v_2^i k^t$ and the value associated to an extreme point $(u_0^j, u_1^j, u_2^j) \in \Omega$ with $j \in U_t$ is $u_0^j d + \alpha$ where $\alpha = u_1^j \ell^t - u_2^j k^t$.

We define the function

$$\psi_t(d) = \begin{cases} +\infty, & \text{if} \quad v_0^i d + v_1^i \ell^t - v_2^i k^t > 0 \quad \text{for some} \quad i \in V^*, \\ \max\limits_{i \in U^*} (u_0^i d + u_1^i \ell^t - u_2^i k^t), & \text{otherwise.} \end{cases}$$

This function has the following property.

Lemma 1 $\psi_t(d) \leq \phi_t(d, k^t, \ell^t)$ *for all* $t \in N$.

Proof. If there exists some extreme ray $(v_0^i, v_1^i, v_2^i) \in \Omega$, $i \in V^*$ with $v_0^i d + v_1^i \ell^t - v_2^i k^t > 0$, then node t is infeasible for the corresponding bounds ℓ^t and k^t on the y variables. Otherwise, each extreme point $(u_0^j, u_1^j, u_2^j) \in \Omega$, $j \in U^*$ provides a lower bound $\psi_t(d)$ which is a lower bound $u_0^j d + u_1^j \ell^t - u_2^j k^t$ on $\phi_t(d, k^t, \ell^t)$. □

Next we can use the information obtained by branching on the y-variables to strengthen the underestimating functions higher up in the tree.

Lemma 2 *Let* L_t *be the leaves of the subtree of the branch-and-cut tree rooted at* t*, then we have* $\psi_t^*(d) = \min_{u \in L_t} \psi_u(d) \leq \phi_t(d, k^t, \ell^t)$ *for all* $t \in N$.

Proof. As every leaf of the subtree has bounds (ℓ^*, k^*) with $\ell^t \leq \ell^* \leq k^* \leq k^t$ and $u_1^j, u_2^j \geq 0$, $u_0^j d + u_1^j \ell^t - u_2^j k^t \leq u_0^j d + u_1^j \ell^* - u_2^j k^*$ and so the leaf provides a bound that is at least as strong as that at node t for every $(u_0^j, u_1^j, u_2^j) \in \Omega$. □

Finally we note that for a given d, several nodes have the same value of $\psi_t^*(d)$ and so we can limit our attention to a subset of the nodes. The branch-and-cut tree consists of y-variable branches and rooted subtrees consisting of only x-variable branches. Let $\mathcal{R}$ be the roots of the corresponding subtrees and, for $t \in \mathcal{R}$, let N_t be the set of nodes in the subtree. All nodes $q \in N_t$ have the same y-variable bounds k^t, ℓ^t and thus $\psi_q^*(d) = \psi_t^*(d)$.

So, as we only need to consider the nodes in $\mathcal{R}$, we construct a reduced tree (see Figure 2) in which, given two nodes $p, q \in \mathcal{R}$, there is an edge $(p, q) \in A$ if and only if there is a directed path from p to q in the branch-and-cut tree containing no other node of $\mathcal{R}$.

Theorem 3 *For* $t \in \mathcal{R}$*, let* L'_t *be the leaves of the subtree rooted at* t *in a reduced tree. Then for all* $t \in \mathcal{R}$*,* $\psi_t^*(d) = \min_{u \in L'_t} \psi_u(d) \leq \phi_t(d, k^t, \ell^t)$ *and, for all* $q \in N_t \setminus \{t\}$ *we have* $\psi_q^*(d) = \psi_t^*(d)$.

Proof. All nodes $q \in N_t$ have the same y-variable bounds k^t and ℓ^t. The rest follows from Lemma 2. □

Example 4 (Example 1 continued) *For the branch-and-cut tree of Example 1, we have:*
$U_1 = [1, 9], U_2 = [10, 12], U_3 = [13, 15], U_7 = [16, 17], V_8 = [18], U_9 = [19]$.
Thus $U^* = [1, 19] \setminus [18]$ *and* $V^* = [18]$*. Now suppose that we wish to estimate the minimum cost* y^* *such that* (x^*, y^*) *is feasible with* $x^* = (0, 0, 1, 1, 1)$*. Thus we need to calculate* $\psi_t^*(d)$ *for* $t = 1$ *where* $d = b - Ax^*$*. In Figure 1* $\mathcal{R} = \{1, 2, 7, 8, 9\}$ *are roots and the subtree rooted at node 2 contains the nodes* $N_2 = \{2, 3, 4, 5, 6\}$ *while* $N_t = \{t\}$ *for each of the nodes* $t \in \{1, 7, 8, 9\}$*. The restricted tree is shown in Figure 2 and the calculation of* $\psi_t^*(d)$ *in Table 2.*
Thus a lower bound on the cost is $\psi_1^*(d) = 470$*. The actual value of* ϕ^* *is 498.*

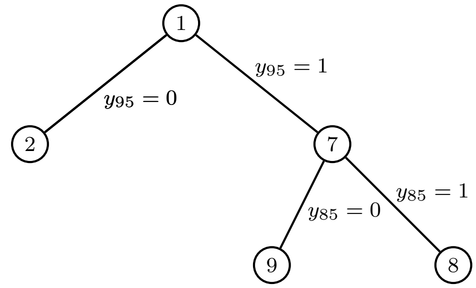

Figure 2: Restricted branch-and-cut tree

node t	1	2	7	8	9
$\psi_t(d)$	466	488	470	∞	470
$\psi_t^*(d)$	470	488	470	∞	470

Table 2: Evaluation of $\psi^*(d)$

Note that, because Example 4 is so simple, the edges of the restricted tree are also edges of the original branch-and-cut tree. In general this will not be the case.

4.2 Sensitivity Analysis

Suppose that (c, b, A) is changed to (c', b', A'). To re-optimize, one can choose to keep part or all of the branch-and-cut tree. At each node q of the tree, having the x-variable branches $C^q x \geq e^q$, one can hot start with the Master LP

$$\begin{array}{rrcll} \zeta = \min & cx + \eta & & & \\ & u_0^s(b - Ax) + u_1^s \ell^q - u_2^s k^q & \leq & \eta & s \in U^* \\ & v_0^t(b - Ax) + v_1^t \ell^q - v_2^t k^q & \leq & 0 & t \in V^* \\ & C^q x & \geq & e^q & \\ & & & x \in \mathbb{R}_+^n, \eta \in \mathbb{R}^1. & \end{array}$$

Example 5 (Example 1 continued) *Suppose that c_5 is changed from 65 to $c_5' = 100$. We decide to start from the top node of the branch-and-cut tree using arbitrarily just $U^* = [1, 9]$ and $V^* = \emptyset$. The calculations are shown in Table 3 and Figure 3. A list of the new extreme points 20 and 21 generated is given in the appendix. Now the set of extreme points of Ω has increased, so a strengthened underestimate of ϕ using $U^* = [1, 17] \cup [19, 21]$ and $V^* = [18]$ can be used if further changes are to be evaluated.*

Note that if Algorithm BxCy is used, changes in the profit vector c are easily dealt with. On the other hand, if b or A change, the added inequalities in the (x, y)-variables have to be updated. This requires further work, but is straightforward for cuts described by functions such as the MIR function F_α or its generalizations.

5. Further remarks

One question that has not been discussed is the convergence of the two Benders' algorithms. Algorithm BCxy is a standard branch-and-cut algorithm. The number of possible cuts is limited

node	ζ	x^*	Comments
1	610.4	(0.88,0.81,0.3,1,0)	Initial solution of BM.
			1 optimality cut added.
	611.2	(0.55,1,0.45,1,0)	
			Branch on x_1
2			Branch $x_1 = 1$. 1 optimality cut added
	617	(1,1,0,1,0)	$y_{82}, y_{84}, y_{91}, y_{92} \notin \mathbb{Z}$
3			Branch: $y_{84} = 1$, $x_1 = 1$. Add $y_{84} = 1$ in SP
	617	**(1,1,0,1,0)**	$y_{91}, y_{92} \notin \mathbb{Z}$
4			Branch $y_{91} = 1$ $y_{84} = 1$, $x_1 = 1$. Add $y_{91} = 1$ in SP.
			New Incumbent.
			Prune and Backtrack
5	617*		Branch $y_{91} = 0$, $y_{84} = 1$, $x_1 = 1$. Prune by bound and Backtrack
6	617*		Branch $y_{84} = 0$, $x_1 = 1$. Prune by bound and Backtrack
7	620*		Branch $x_1 = 0$
			Search Completed

Table 3: CFL with Single-Sourcing. * indicates lower bound on ζ.

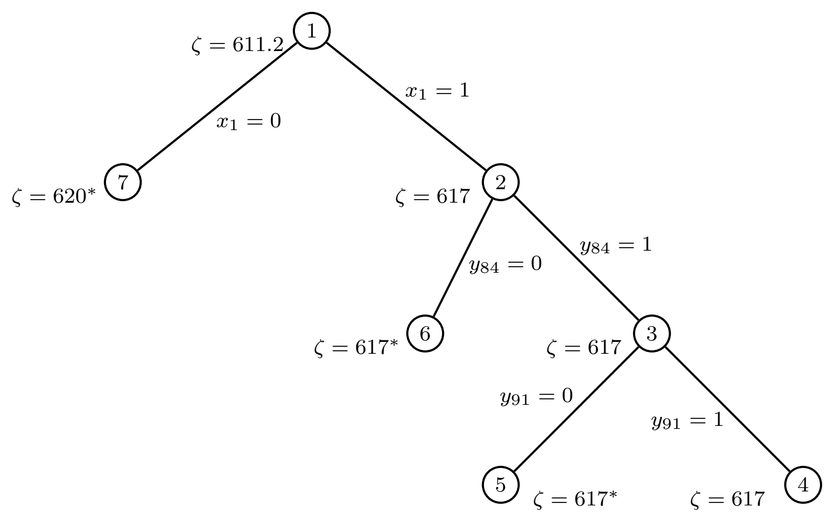

Figure 3: Branch-and-cut tree for re-optimization

to the number of extreme points and rays of the dual region Ω, so the only concern is that arising in branch-and-bound if the x-space is unbounded. For algorithm BxCy there is theoretically the problem of the non-convergence of certain cutting plane algorithms, but in practice this is avoided by stopping cut-generation as in cases 5 and 6 and using no-good feasibility or optimality cuts. As the x-variables are 0-1, there is no problem if the subproblems are solved to optimality.

The version BCxy of Benders' algorithm proposed here appears to be suitable for problems with a relatively small number of integer y-variables, or problems such as CFL with single-sourcing in which one expects to have a small number of y-variables taking fractional values. Even though the algorithm proposed is a straightforward branch-and-cut algorithm, its implementation requires several choices. Should one always branch on x-variables in preference to

y-variables? After solving the initial LP relaxation, should one immediately impose integrality on the y-variables, or should one first develop a tree of x-variable branches? Which of the procedures for generating more effective dual solutions should one select? As there is an over-abundance of inequalities which can slow down the solution of the relaxed master LPs, which should be invoked at a given node and which should be discarded?

Algorithm BxCy appears to be more suitable for problems with many integer variables and/or multiple subproblems, in particular, as in [16], to two-stage stochastic programs with integer recourse of the form

$$\min\{cx + \sum_{q=1}^{Q} p^q hy : Ax + B^q y^q \geq b^q, q \in [1, Q], x \in \mathbb{R}^n_+/\mathbb{Z}^n_+, y^q \in \mathbb{Z}^p_+ \; q \in [1, Q]\},$$

where $q \in [1, Q]$ denote the scenarios each of probability p^q. If only the requirements vectors b^q are random, the subproblem is the same for each possible scenario, so each extreme point or ray can generate a cut in the MLP for each of the Q scenarios. Here there are further choices: When there are multiple related subproblems, which should one choose to treat first so as to generate good information for the other subproblems? How much information should be kept, etc.?

The approximation of the value function for the subproblem developed in Section 5 is clearly related to the development of a value function from a branch-and-bound tree, for use in sensitivity analysis, see Wolsey [42], Schrage and Wolsey [33]. Apart from its importance for post-optimality analysis of mixed-integer programs, calculating value functions is of interest in the study of cooperative games. Value functions can also appear in bilevel optimization problems that are reformulated as single-level optimization problems, however the resulting constraints are highly non-convex.

References

[1] T. Achterberg. SCIP: solving constraint integer programs. *Mathematical Programming Computation*, 1(1):1–41, 2009.

[2] A. Atamtürk. On the facets of the mixed-integer knapsack polyhedron. *Math. Program., Ser. B*, 98:145–175, 2003.

[3] E. Balas. Facets of the knapsack polytope. *Mathematical Programming*, 8(1):146–164, 1975.

[4] E. Balas, S. Ceria, and G. Cornuéjols. A lift-and-project cutting plane algorithm for mixed 0–1 programs. *Mathematical Programming*, 58(1):295–324, 1993.

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

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

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

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

[11] T. G. Crainic, M. Hewitt, and W. Rei. Partial decomposition strategies for two-stage stochastic integer programs. *Publication CIRRELT-2014-13, Centre interuniversitaire de recherche sur les réseaux d'entreprise, la logistique et le transport, Université de Montréal, Montréal, QC, Canada*, 2014.

[12] 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.

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

[14] 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.

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

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

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

[18] A. Gleixner, L. Eifler, T. Gally, G. Gamrath, P. Gemander, R. L. Gottwald, G. Hendel, C. Hojny, T. Koch, M. Miltenberger, B. Müller, M. E. Pfetsch, C. Puchert, D. Rehfeldt, F. Schlösser, F. Serrano, Y. Shinano, J. M. Viernickel, S. Vigerske, D. Weninger, J. T. Witt, and J. Witzig. The SCIP Optimization Suite 5.0. Technical report, Optimization Online, December 2017.

[19] R. E. Gomory. Outline of an algorithm for integer solutions to linear programs. *Bulletin of the American Mathematical Society*, 64(5):275–278, 1958.

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

[21] Z. Gu, G. L. Nemhauser, and M. W. P. Savelsbergh. Lifted flow cover inequalities for mixed 0-1 integer programs. *Mathematical Programming*, 85(3):439–467, 1999.

[22] P. L. Hammer, E. L. Johnson, and U. N. Peled. Facet of regular 0–1 polytopes. *Mathematical Programming*, 8(1):179–206, 1975.

[23] A. Hassanzadeh and T. Ralphs. A Generalization of Benders' Algorithm for Two-Stage Stochastic Optimization Problems with Mixed Integer Recourse. Technical report, COR@L Laboratory Report 14T-005, Lehigh University, 2014.

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

[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] Q. Louveaux and L. A. Wolsey. Lifting, superadditivity, mixed integer rounding and single node flow sets revisited. *Annals of Operations Research*, 153(1):47–77, 2007.

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

[28] S. J. Maher. Implementing the branch-and-cut approach for a general purpose Benders' decomposition framework. Technical report, University of Exeter, 2019.

[29] 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.

[30] M. W. Padberg, T. J. Van Roy, and L. A. Wolsey. Valid linear inequalities for fixed charge problems. *Oper. Res.*, 33(4):842–861, 1985.

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

[32] 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.

[33] L. Schrage and L. A. Wolsey. Sensitivity analysis for branch and bound integer programming. *Operations Research*, 33(5):1008–1023, 1985.

[34] 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.

[35] 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.

[36] H. D. Sherali and W. P. Adams. A hierarchy of relaxations between the continuous and convex hull representations for zero-one programming problems. *SIAM Journal on Discrete Mathematics*, 3:411–430, 1990.

[37] H. D. Sherali and B. M. P. Fraticelli. A modification of Benders' decomposition algorithm for discrete subproblems: An approach for stochastic programs with integer recourse. *Journal of Global Optimization*, 22(1):319–342, 2002.

[38] T. J. Van Roy and L. A. Wolsey. Valid inequalities for mixed 0-1 programs. *Discrete Applied Mathematics*, 14(2):199 – 213, 1986.

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

[40] L. A. Wolsey. Faces for a linear inequality in 0–1 variables. *Mathematical Programming*, 8(1):165–178, 1975.

[41] L. A. Wolsey. Facets and strong valid inequalities for integer programs. *Operations Research*, 24:367–372, 1976.

[42] L. A. Wolsey. Integer programming duality: price functions and sensitivity analysis. *Mathematical Programming*, 20:173–195, 1981.

6. Annexe

The dual extreme points and rays generated in Example 1 are shown below, where the unit vector associated to client i and location j is denoted by e_{ij}.

Node 1:

$u_0^1 = (30, 8, 36, 96, 48, 88, 44, 124, 128, 50), (1, 0, 0, 1, 0),$
$(0, 20, 0, 0, 0, 0, 0, 0, 0, 0, 0, 12, 0, 0, 0, 0, 0, 0, 0, 32, 0, 16, 0, 0, 0, 0, 0, 0, 0, 0, 0, 33, 0, 0, 22, 0, 31, 0, 0, 62, 0, 32, 0, 0, 32, 0, 0, 0, 0, 0)$

$u_0^2 = (10, 8, 24, 64, 32, 66, 11, 62, 96, 50), (0, 0, 0, 0, 0),$
$(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)$

$u_0^3 = (20, 12, 36, 96, 48, 88, 22, 124, 128, 75), (0, 1, 0, 1, 1),$
$(0, 0, 0, 0, 0, 4, 0, 4, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 22, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 31, 32, 0, 0, 0, 0, 0, 0, 25, 0, 0)$

$u_0^4 = (20, 8, 36, 64, 32, 66, 22, 62, 96, 100), (0, 0, 0, 0, 0),$
$(0, 10, 0, 0, 0, 0, 0, 0, 0, 0, 0, 12, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 11, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 50, 50, 0, 0)$

$u_0^5 = (30, 8, 36, 64, 32, 66, 22, 93, 96, 50), (0, 0, 0, 0, 0),$
$(10, 20, 0, 0, 0, 0, 0, 0, 0, 0, 0, 12, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 11, 0, 0, 0, 0, 0, 0, 0, 31, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)$

$u_0^6 = (10, 8, 24, 64, 32, 88, 11, 62, 96, 50), (0, 0, 0, 0, 0),$
$(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 22, 0, 0, 22, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)$

$u_0^7 = (10, 8, 24, 64, 32, 66, 11, 93, 96, 50), (0, 0, 0, 0, 0),$
$(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 31, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)$

$u_0^8 = (20, 8, 36, 64, 32, 88, 22, 93, 96, 50), (0, 0, 0, 0, 0),$
$(0, 10, 0, 0, 0, 0, 0, 0, 0, 0, 0, 12, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 22, 0, 0, 22, 0, 0, 11, 0, 0, 0, 0, 0, 0, 0, 31, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)$

$u_0^9 = (20, 8, 36, 64, 32, 66, 22, 93, 96, 100), (0, 0, 0, 0, 0),$
$(0, 10, 0, 0, 0, 0, 0, 0, 0, 0, 0, 12, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 11, 0, 0, 0, 0, 0, 0, 0, 31, 0, 0, 0, 0, 0, 0, 50, 50, 0, 0)$

Node 2:

$u_0^{10} = (20, 12, 36, 64, 32, 66, 22, 62, 128, 75), (0, 1, 0, 0, 0),$
$(0, 0, 0, 0, 0, 4, 0, 4, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 32, 0, 0, 0, 0, 0, 0, 25, 0, 0)$

$u_2^{10} = 32 e_{95}$

$u_0^{11} = (20, 12, 36, 64, 48, 66, 22, 93, 128, 75), (0, 1, 0, 0, 0),$
$(0, 0, 0, 0, 0, 4, 0, 4, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 16, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 31, 32, 0, 0, 0, 0, 0, 0, 25, 0, 0)$

$u_2^{11} = 32 e_{95}$

$u_0^{12} = (20, 8, 36, 64, 32, 66, 22, 93, 96, 50), (0, 0, 0, 0, 0),$
$(0, 10, 0, 0, 0, 0, 0, 0, 0, 0, 0, 12, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 11, 0, 0, 0, 0, 0, 0, 0, 31, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)$

$u_2^{12} = 0 e_{95}$

Node 3:

$u_0^{13} = (10, 8, 24, 96, 32, 66, 11, 93, 96, 50), (0, 0, 0, 0, 1),$
$(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 32, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)$

$u_2^{13} = 0 e_{95}$

$u_0^{14} = (20, 8, 48, 96, 48, 66, 33, 93, 96, 50), (0, 0, 0, 0, 1),$
$(0, 10, 0, 0, 0, 0, 0, 0, 0, 0, 24, 12, 0, 0, 0, 0, 0, 32, 0, 0, 16, 0, 16, 0, 0, 0, 0, 0, 0, 0, 22, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)$

$u_2^{14} = 0 e_{95}$

$u_0^{15} = (20, 8, 48, 96, 32, 66, 33, 93, 96, 100), (0, 0, 0, 0, 1),$
$(0, 10, 0, 0, 0, 0, 0, 0, 0, 0, 24, 12, 0, 0, 0, 0, 0, 32, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 22, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 50, 50, 0, 0)$

$u_2^{15} = 0 e_{95}$

Node 7:

$u_0^{16} = (10, 8, 24, 64, 32, 66, 11, 93, 0, 50), (0, 0, 0, 0, 1),$
$(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)$

$u_1^{16} = 128 e_{95}$

$u_0^{17} = (20, 8, 48, 96, 32, 66, 33, 93, 0, 50), (0, 0, 0, 0, 1),$
$(0, 10, 0, 0, 0, 0, 0, 0, 0, 0, 24, 12, 0, 0, 0, 0, 0, 32, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 22, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)$

$u_1^{17} = 128 e_{95}$

Node 8:

v_0^{18}=(0, 0, 0, 0, 0, 0, 0, 0, 0, 0), (0, 0, 0, 0, 1),
(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
$v_1^{18} = 31e_{95} + 32e_{85}$

Node 9:

u_0^{19}=(10, 8, 24, 64, 32, 66, 11, 93, 0, 50), (0, 0, 0, 0, 0),
(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
$u_1^{19} = 96e_{95}$
$u_2^{19} = 31e_{85}$

Node 1 of Example 5 when re-optimizing:

u_0^{20}=(10, 8, 24, 96, 32, 66, 11, 93, 96, 50), (0, 0, 0, 0, 0),
(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 32, 32, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 31, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
u_0^{21}=(10, 8, 24, 64, 32, 66, 11, 0, 96, 50), (0, 0, 0, 0, 0),
(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)