

# Benders Decomposition \*

Z. Caner Taşkın

Department of Industrial Engineering

Boğaziçi University

34342 Bebek, İstanbul, Turkey

`caner.taskin@boun.edu.tr`

## Abstract

Benders decomposition is a solution method for solving certain large-scale optimization problems. Instead of considering all decision variables and constraints of a large-scale problem simultaneously, Benders decomposition partitions the problem into multiple smaller problems. Since computational difficulty of optimization problems increases significantly with the number of variables and constraints, solving these smaller problems iteratively can be more efficient than solving a single large problem. In this chapter, we first formally describe Benders decomposition. We then briefly describe some extensions and generalizations of Benders decomposition. We conclude our chapter by illustrating how the decomposition works on a problem encountered in Intensity Modulated Radiation Therapy (IMRT) treatment planning and giving a numerical example.

**Keywords:** Benders decomposition; large-scale optimization; linear programming; mixed-integer programming; matrix segmentation problem

---

\*This is the preprint of a book chapter published in Wiley Encyclopedia of Operations Research and Management Science. Published version is available at <http://dx.doi.org/10.1002/9780470400531.eorms0104>

An important concern regarding building and solving optimization problems is that the amount of memory and the computational effort needed to solve such problems grow significantly with the number of variables and constraints. The traditional approach, which involves making all decisions simultaneously by solving a monolithic optimization problem, quickly becomes intractable as the number of variables and constraints increases. Multi-stage optimization algorithms, such as Benders decomposition [1], have been developed as an alternative solution methodology to alleviate this difficulty. Unlike the traditional approach, these algorithms divide the decision-making process into several stages. In Benders decomposition a first-stage master problem is solved for a subset of variables, and the values of the remaining variables are determined by a second-stage subproblem *given* the values of the first-stage variables. If the subproblem determines that the proposed first-stage decisions are infeasible, then one or more constraints are generated and added to the master problem, which is then re-solved. In this manner, a series of small problems are solved instead of a single large problem, which can be justified by the increased computational resource requirements associated with solving larger problems.

The remainder of this chapter is organized as follows. We will first describe Benders decomposition formally in Section 1. We will then discuss some extensions of Benders decomposition in Section 2. Finally, we will illustrate the decomposition approach on a problem encountered in IMRT treatment planning in Section 3, and give a numerical example.

## 1 Formal Derivation

In this section, we describe Benders decomposition algorithm for linear programs. Consider the following problem:

$$\text{Minimize } c^T x + f^T y \tag{1a}$$

$$\text{subject to: } Ax + By = b \tag{1b}$$

$$x \geq 0, y \in Y \subseteq \mathbb{R}^q, \tag{1c}$$

where  $x$  and  $y$  are vectors of continuous variables having dimensions  $p$  and  $q$ , respectively,  $Y$  is a polyhedron,  $A, B$  are matrices, and  $b, c, f$  are vectors having appropriate dimensions. Suppose that  $y$ -variables are “complicating variables” in the sense that the problem becomes significantly easier to solve if  $y$ -variables are fixed, perhaps due to a special structure inherent in matrix  $A$ . Benders decomposition partitions problem (1) into two problems: (i) a master problem that contains the  $y$ -variables, and (ii) a subproblem that contains the  $x$ -variables. We first note that problem (1) can be written in terms of the  $y$ -variables as follows:

$$\text{Minimize } f^T y + q(y) \tag{2a}$$

$$\text{subject to: } y \in Y, \tag{2b}$$

where  $q(y)$  is defined to be the optimal objective function value of

$$\text{Minimize } c^T x \tag{3a}$$

$$\text{subject to: } Ax = b - By \tag{3b}$$

$$x \geq 0. \tag{3c}$$

Formulation (3) is a linear program for any given value of  $y \in Y$ . Note that if (3) is unbounded for some  $y \in Y$ , then (2) is also unbounded, which in turn implies unboundedness of the original problem (1). Assuming boundedness of (3), we can also calculate  $q(y)$  by solving its dual. Let us associate dual variables  $\alpha$  with constraints (3b). Then, the dual of (3) is

$$\text{Maximize } \alpha^T (b - By) \tag{4a}$$

$$\text{subject to: } A^T \alpha \leq c \tag{4b}$$

$$\alpha \text{ unrestricted.} \tag{4c}$$

A key observation is that feasible region of the dual formulation does not depend on the value of  $y$ , which only affects the objective function. Therefore, if the dual feasible region (4b)–(4c) is empty, then either the primal problem (3) is unbounded for some  $y \in Y$  (in which case the original problem (1) is unbounded), or the primal feasible region (3b)–(3c) is also empty for

all  $y \in Y$  (in which case (1) is also infeasible.) Assuming that the feasible region defined by (4b)–(4c) is not empty, we can enumerate all extreme points  $(\alpha_p^1, \dots, \alpha_p^I)$ , and extreme rays  $(\alpha_r^1, \dots, \alpha_r^J)$  of the feasible region, where  $I$  and  $J$  are the numbers of extreme points and extreme rays of (4b)–(4c), respectively. Then, for a given  $\hat{y}$ -vector, the dual problem can be solved by checking (i) whether  $(\alpha_r^j)^T(b - B\hat{y}) > 0$  for an extreme ray  $\alpha_r^j$ , in which case the dual formulation is unbounded and the primal formulation is infeasible, and (ii) finding an extreme point  $\alpha_p^i$  that maximizes the value of the objective function  $(\alpha_p^i)^T(b - B\hat{y})$ , in which case both primal and dual formulations have finite optimal solutions. Based on this idea, the dual problem (4) can be reformulated as follows:

$$\text{Minimize } q \tag{5a}$$

$$\text{subject to: } (\alpha_r^j)^T(b - By) \leq 0 \quad \forall j = 1, \dots, J \tag{5b}$$

$$(\alpha_p^i)^T(b - By) \leq q \quad \forall i = 1, \dots, I \tag{5c}$$

$$q \text{ unrestricted.} \tag{5d}$$

Note that (5) consists of a single variable  $q$  and, typically, a large number of constraints. Now we can replace  $q(y)$  in (2a) with (5) and obtain a reformulation of the original problem in terms of  $q$  and  $y$ -variables:

$$\text{Minimize } f^T y + q \tag{6a}$$

$$\text{subject to: } (\alpha_r^j)^T(b - By) \leq 0 \quad \forall j = 1, \dots, J \tag{6b}$$

$$(\alpha_p^i)^T(b - By) \leq q \quad \forall i = 1, \dots, I \tag{6c}$$

$$y \in Y, \quad q \text{ unrestricted.} \tag{6d}$$

Since there is typically an exponential number of extreme points and extreme rays of the dual formulation (4), generating all constraints of type (6b) and (6c) is not practical. Instead, Benders decomposition starts with a subset of these constraints, and solves a “relaxed master problem,” which yields a candidate optimal solution  $(y^*, q^*)$ . It then solves the dual subproblem (4) to calculate  $q(y^*)$ . If the subproblem has an optimal solution having  $q(y^*) = q^*$ , then the algorithm stops. Otherwise, if the dual subproblem is unbounded, then a constraint

of type (6b) is generated and added to the relaxed master problem, which is then re-solved. (Constraints of type (6b) are referred to as “Benders feasibility cuts” because they enforce necessary conditions for feasibility of the primal subproblem (3).) Similarly, if the subproblem has an optimal solution having  $q(y^*) > q^*$ , then a constraint of type (6c) is added to the relaxed master problem, and the relaxed master problem is re-solved. (Constraints of type (6c) are called “Benders optimality cuts” because they are based on optimality conditions of the subproblem.) Since  $I$  and  $J$  are finite, and new feasibility or optimality cuts are generated in each iteration, this method converges to an optimal solution in a finite number of iterations [1].

## 2 Extensions

Benders decomposition is closely related to other decomposition methods for linear programming (see Section **CROSS-REF 1.1.2.5** for relationships among Benders, Dantzig-Wolfe, and Lagrangian optimization). Furthermore, Benders decomposition can be applied to a broader class of problems, some of which we will describe in this section. We first observe that only linear constraints are added to the master problem throughout the iterations of Benders decomposition. Therefore, the master problem does not have to be a linear program; but can take the form of an integer (e.g. [1] and Section 3), a nonlinear (e.g. [2]) or a constraint programming problem (e.g. [3]). Also note that the subproblem is only used to obtain dual information in order to generate Benders cuts. Therefore, the subproblem does not have to be a linear program; but can also be a convex program since dual multipliers satisfying strong duality conditions can be calculated for such problems [4]. (See Section **CROSS-REF 1.2.3.10** for detailed information about convex optimization.) The extension of Benders decomposition that allows for nonlinear convex programs to be used as subproblems is referred to as “generalized Benders decomposition” [5]. Similarly, “logic-based Benders decomposition” generalizes the use of linear programming duality in the subproblem to “inference duality,” which allows the use of logic-based methods for solving the subproblem and generating Benders cuts. (See Section **CROSS-REF 1.4.3.4** for more information

about constraint programming and its relationships with mathematical programming.) In some applications, subproblems can be solved efficiently by specialized algorithms instead of the explicit solution of linear programs (e.g. [6, 7]). If the subproblem is a linear feasibility problem (i.e. a linear programming problem having no objective function), cuts based on irreducible infeasible subsets of constraints can be generated using a technique known as “combinatorial Benders decomposition” [8].

It is often the case that decisions for several groups of second-stage variables can be made independently given the first-stage decisions. In such cases, multiple subproblems can be defined and solved separately. For instance, in stochastic programming models, some action needs to be taken in a first stage, which is followed by the occurrence of a random event (typically modeled by a number of scenarios) that affects the outcome of the first-stage decision. A recourse decision can then be made in a second stage after the uncertainty is resolved. (See Sections **CROSS-REF 1.5.1** and **CROSS-REF 1.5.2** for more information about two-stage stochastic programming models.) In such problems, second-stage recourse problems can be solved independently given the first-stage decisions, and hence are amenable to parallel implementations [9]. (See also Section **CROSS-REF 1.5.2.2**.)

## 3 Illustrative Example

### 3.1 Problem Definition

In this section, we consider a matrix segmentation problem arising in Intensity Modulated Radiation Therapy (IMRT) treatment planning, which is described in detail in [10]. (See also Section **CROSS-REF 4.3.1.1** for an introduction to optimization models in cancer treatment.) The problem input is a matrix of intensity values that are to be delivered to a patient from some given angle, under the condition that the IMRT device can only deliver radiation through rectangular apertures. An aperture is represented as a binary matrix whose ones appear consecutively in each row and column, and hence form a rectangular shape. A feasible segmentation is one in which the original desired intensity matrix is equal

to the weighted sum of a number of feasible binary matrices, where the weight of each binary matrix is the amount of intensity to be delivered through the corresponding aperture. We seek a matrix segmentation that uses the smallest number of aperture matrices to segment the given intensity matrix. This goal corresponds to minimizing setup time in the IMRT context [10]. The example below shows an intensity matrix and a feasible segmentation using three rectangular apertures:

$$\begin{bmatrix} 2 & 7 & 0 \\ 2 & 10 & 3 \\ 0 & 8 & 3 \end{bmatrix} = 2 \times \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} + 3 \times \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix} + 5 \times \begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix}.$$

We will denote the intensity matrix to be delivered by an  $m \times n$  matrix  $B$ , where the element at row  $i$  and column  $j$ ,  $(i, j)$  requires  $b_{ij} \in \mathbb{Z}^+$  units of intensity. Let  $R$  be the set of all  $O(m^2n^2)$  possible rectangular apertures (i.e., binary matrices of size  $m \times n$  having contiguous rows and columns) that can be used in a segmentation of  $B$ . For each rectangle  $r \in R$  we define a continuous variable  $x_r$  that represents the intensity assigned to rectangle  $r$ , and a binary variable  $y_r$  that equals 1 if rectangle  $r$  is used in decomposing  $B$  (i.e., if  $x_r > 0$ ), and equals 0 otherwise. We say that element  $(i, j)$  is “covered” by rectangle  $r$  if the  $(i, j)$  element of  $r$  is 1. Let  $C(r)$  be the set of matrix elements that is covered by rectangle  $r$ . We define  $M_r = \min_{(i,j) \in C(r)} \{b_{ij}\}$  to be the minimum intensity requirement among the elements of  $B$  that are covered by rectangle  $r$ . Furthermore, we denote the set of rectangles that cover element  $(i, j)$  by  $R(i, j)$ . Given these definitions, we can formulate the problem as follows:

$$\text{Minimize } \sum_{r \in R} y_r \tag{7a}$$

$$\text{subject to: } \sum_{r \in R(i,j)} x_r = b_{ij} \quad \forall i = 1, \dots, m, \quad j = 1, \dots, n \tag{7b}$$

$$x_r \leq M_r y_r \quad \forall r \in R \tag{7c}$$

$$x_r \geq 0, \quad y_r \in \{0, 1\} \quad \forall r \in R. \tag{7d}$$

The objective function (7a) minimizes the number of rectangular apertures used in the segmentation. Constraints (7b) guarantee that each matrix element receives exactly the

required dose. Constraints (7c) enforce the condition that  $x_r$  cannot be positive unless  $y_r = 1$ . Finally, (7d) states bounds and logical restrictions on the variables. Note that the objective (7a) guarantees that  $y_r = 0$  when  $x_r = 0$  in any optimal solution of (7).

### 3.2 Decomposition Approach

Formulation (7) contains two variables and a constraint for each rectangle, resulting in a large-scale mixed-integer program for problem instances of clinically relevant sizes. Furthermore, the  $M_r$ -terms in constraints (7c) lead to a weak linear programming relaxation due to the “big-M” structure. These difficulties can be alleviated by employing a Benders decomposition approach. Our decomposition approach will first select a subset of the rectangles in a master problem, and then check whether the input matrix can be segmented using only the selected rectangles in a subproblem. Let us first reformulate the problem in terms of the  $y$ -variables.

$$\text{Minimize } \sum_{r \in R} y_r \tag{8a}$$

$$\text{subject to: } y \text{ corresponds to a feasible segmentation} \tag{8b}$$

$$y_r \in \{0, 1\} \quad \forall r \in R, \tag{8c}$$

where we will address the form of (8b) next. Given a vector  $\hat{y}$  that represents a selected subset of rectangles, we can check whether constraint (8b) is satisfied by solving the following subproblem:

$$\text{SP}(\hat{y}): \text{Minimize } 0 \tag{9a}$$

$$\text{subject to: } \sum_{r \in R(i,j)} x_r = b_{ij} \quad \forall i = 1, \dots, m, j = 1, \dots, n \tag{9b}$$

$$x_r \leq M_r \hat{y}_r \quad \forall r \in R \tag{9c}$$

$$x_r \geq 0 \quad \forall r \in R. \tag{9d}$$

If  $\hat{y}$  corresponds to a feasible segmentation then  $\text{SP}(\hat{y})$  is feasible, otherwise it is infeasible. Note that formulation (8) is a pure integer programming problem (since it only contains



the  $y$ -variables), and  $\text{SP}(\hat{y})$  is a linear programming problem (since it only contains the  $x$ -variables). Furthermore, constraints (9c) reduce to simple upper bounds on  $x$ -variables for a given  $\hat{y}$ , which avoids the “big-M” issue associated with constraints (7c). Given a  $\hat{y}$ -vector, if  $\text{SP}(\hat{y})$  has a feasible solution  $\hat{x}$ , then  $(\hat{x}, \hat{y})$  constitutes a feasible solution of the original problem (7). On the other hand, if  $\text{SP}(\hat{y})$  does not yield a feasible solution, then we need to ensure that  $\hat{y}$  is eliminated from the feasible region of (8). Benders decomposition uses the theory of linear programming duality to achieve this goal.

Let us associate variables  $\alpha_{ij}$  with (9b), and  $\beta_r$  with (9c). Then, the dual formulation of  $\text{SP}(\hat{y})$  can be given as:

$$\text{DSP}(\hat{y}): \text{Maximize } \sum_{i=1}^m \sum_{j=1}^n b_{ij} \alpha_{ij} + \sum_{r \in R} M_r \hat{y}_r \beta_r \quad (10a)$$

$$\text{subject to: } \sum_{(i,j) \in \mathcal{C}(r)} \alpha_{ij} + \beta_r \leq 0 \quad \forall r \in R \quad (10b)$$

$$\alpha_{ij} \text{ unrestricted} \quad \forall i = 1, \dots, m, \quad j = 1, \dots, n \quad (10c)$$

$$\beta_r \leq 0 \quad \forall r \in R. \quad (10d)$$

Our Benders decomposition strategy first relaxes constraints (8b) and solves (8) to optimality, which yields  $\hat{y}$ . If  $\text{SP}(\hat{y})$  has a feasible solution  $\hat{x}$ , then  $(\hat{x}, \hat{y})$  corresponds to an optimal matrix segmentation. On the other hand, if  $\text{SP}(\hat{y})$  is infeasible, then the dual formulation  $\text{DSP}(\hat{y})$  is unbounded (since the all-zero solution is always a feasible solution of  $\text{DSP}(\hat{y})$ ). Let  $(\hat{\alpha}, \hat{\beta})$  be an extreme ray of  $\text{DSP}(\hat{y})$  such that  $\sum_{i=1}^m \sum_{j=1}^n b_{ij} \hat{\alpha}_{ij} + \sum_{r \in R} M_r \hat{y}_r \hat{\beta}_r > 0$ . Then, all  $y$ -vectors that are feasible with respect to (8b) must satisfy

$$\sum_{i=1}^m \sum_{j=1}^n b_{ij} \hat{\alpha}_{ij} + \sum_{r \in R} (M_r \hat{\beta}_r) y_r \leq 0. \quad (11)$$

We add (11) to (8), and re-solve it in the next iteration to obtain a new candidate optimal solution.

Now let us consider a slight variation of the matrix segmentation problem, where the goal is to minimize a weighted combination of the number of matrices used in the segmentation (corresponding to setup time) and the sum of the matrix coefficients (corresponding

to “beam-on-time”). In IMRT treatment planning context, this objective corresponds to minimizing total treatment time [10]. In order to incorporate this change in our model, we simply replace the objective function (7a) with

$$\text{Minimize } w \sum_{r \in R} y_r + \sum_{r \in R} x_r, \quad (12)$$

where  $w$  is a parameter that represents the average setup time per aperture relative to the time required to deliver a unit of intensity.

The Benders decomposition procedure discussed above needs to be adjusted accordingly. We first add a continuous variable  $t$  to (8), which “predicts” the minimum beam-on-time that can be obtained by the set of rectangles chosen. The updated formulation can be written as follows:

$$\text{Minimize } w \sum_{r \in R} y_r + t \quad (13a)$$

$$\text{subject to: } y \text{ corresponds to a feasible segmentation} \quad (13b)$$

$$t \geq \text{minimum beam-on-time corresponding to } y \quad (13c)$$

$$t \geq 0, y_r \in \{0, 1\} \quad \forall r \in R. \quad (13d)$$

Given a vector  $\hat{y}$ , we can find the minimum beam-on-time for the corresponding segmentation, if one exists, by solving:

$$\text{SPTT}(\hat{y}): \text{Minimize } \sum_{r \in R} x_r \quad (14a)$$

$$\text{subject to: } \sum_{r \in R(i,j)} x_r = b_{ij} \quad \forall i = 1, \dots, m, j = 1, \dots, n \quad (14b)$$

$$x_r \leq M_r \hat{y}_r \quad \forall r \in R \quad (14c)$$

$$x_r \geq 0 \quad \forall r \in R. \quad (14d)$$

Let  $\alpha_{ij}$  and  $\beta_r$  be optimal dual multipliers associated with constraints (14b) and (14c),

respectively. Then, the dual of  $\text{SPTT}(\hat{y})$  is:

$$\text{DSPTT}(\hat{y}): \text{Maximize } \sum_{i=1}^m \sum_{j=1}^n b_{ij} \alpha_{ij} + \sum_{r \in R} M_r \hat{y}_r \beta_r \quad (15a)$$

$$\text{subject to: } \sum_{(i,j) \in C(r)} \alpha_{ij} + \beta_r \leq 1 \quad \forall r \in R \quad (15b)$$

$$\alpha_{ij} \text{ unrestricted} \quad \forall i = 1, \dots, m, j = 1, \dots, n \quad (15c)$$

$$\beta_r \leq 0 \quad \forall r \in R. \quad (15d)$$

Note that  $\text{SPTT}(\hat{y})$  is obtained by simply changing the objective function of  $\text{SP}(\hat{y})$ , and  $\text{DSPTT}(\hat{y})$  is obtained by changing the right hand side of (10b) in  $\text{DSP}(\hat{y})$ . If  $\text{DSPTT}(\hat{y})$  is unbounded, then we add a Benders feasibility cut of type (11) as before, and re-solve (13). Otherwise, let the value of  $t$  in (13) be  $\hat{t}$ , and the optimal objective function value of  $\text{DSPTT}(\hat{y})$  be  $t^*$ . If  $\hat{t} = t^*$ , then  $(\hat{y}, \hat{t})$  is an optimal solution of (13), which minimizes the total treatment time. However, if  $\hat{t} < t^*$ , then we need to add a constraint that satisfies the following properties: (i) the optimal value of  $t = \hat{t}$  if  $\hat{y}$  is generated by (13) in a future iteration, and (ii) the optimal value of  $t \leq \hat{t}$  for all  $y$ . Benders decomposition, once again, uses linear programming duality theory to generate such a constraint. Let  $\hat{\alpha}_{ij}$  and  $\hat{\beta}_r$  be optimal dual multipliers. It can be seen that the following constraint satisfies both requirements.

$$t \geq \sum_{i=1}^m \sum_{j=1}^n b_{ij} \hat{\alpha}_{ij} + \sum_{r \in R} (M_r \hat{\beta}_r) y_r. \quad (16)$$

### 3.3 Numerical Example

In this section, we give a simple numerical example illustrating the steps of Benders decomposition approach on our matrix segmentation problem. Consider the input matrix

$B = \begin{bmatrix} 8 & 3 \\ 5 & 0 \end{bmatrix}$ . The set of rectangular apertures that can be used to segment  $B$  is:

$$R = \left\{ \left[ \begin{array}{cc} 1 & 0 \\ 0 & 0 \end{array} \right], \left[ \begin{array}{cc} 0 & 1 \\ 0 & 0 \end{array} \right], \left[ \begin{array}{cc} 0 & 0 \\ 1 & 0 \end{array} \right], \left[ \begin{array}{cc} 1 & 0 \\ 1 & 0 \end{array} \right], \left[ \begin{array}{cc} 1 & 1 \\ 0 & 0 \end{array} \right] \right\}$$

Let the average setup time per aperture relative to the time required to deliver a unit of intensity be  $w = 7$ . Defining an  $x_r$  and a  $y_r$ -variable for each rectangle  $r = 1, \dots, 5$ , the

problem of minimizing total treatment time can be expressed as the following mixed-integer program:

$$\text{Minimize } 7 \times (y_1 + y_2 + y_3 + y_4 + y_5) + x_1 + x_2 + x_3 + x_4 + x_5 \quad (17a)$$

$$\text{subject to: } x_1 + x_4 + x_5 = 8 \quad (17b)$$

$$x_2 + x_5 = 3 \quad (17c)$$

$$x_3 + x_4 = 5 \quad (17d)$$

$$x_1 \leq 8y_1, x_2 \leq 3y_2, x_3 \leq 5y_3, x_4 \leq 5y_4, x_5 \leq 3y_5 \quad (17e)$$

$$x_r \geq 0, y_r \in \{0, 1\} \quad \forall r = 1, \dots, 5. \quad (17f)$$

For a given  $\hat{y}$ -vector, the primal subproblem  $\text{SPTT}(\hat{y})$  can be given as

$$\text{SPTT}(\hat{y}): \text{Minimize } x_1 + x_2 + x_3 + x_4 + x_5 \quad (18a)$$

$$\text{subject to: } x_1 + x_4 + x_5 = 8 \quad (18b)$$

$$x_2 + x_5 = 3 \quad (18c)$$

$$x_3 + x_4 = 5 \quad (18d)$$

$$x_1 \leq 8\hat{y}_1, x_2 \leq 3\hat{y}_2, x_3 \leq 5\hat{y}_3, x_4 \leq 5\hat{y}_4, x_5 \leq 3\hat{y}_5 \quad (18e)$$

$$x_r \geq 0 \quad \forall r = 1, \dots, 5. \quad (18f)$$

Associating dual variables  $\alpha_{11}$  with (18b),  $\alpha_{12}$  with (18c),  $\alpha_{21}$  with (18d), and  $\beta_1, \dots, \beta_5$  with (18e), we get the dual subproblem DSPTT( $\hat{y}$ ).

$$\begin{aligned} \text{DSPTT}(\hat{y}): \text{ Maximize } & 8\alpha_{11} + 3\alpha_{12} + 5\alpha_{21} + \\ & 8\hat{y}_1\beta_1 + 3\hat{y}_2\beta_2 + 5\hat{y}_3\beta_3 + 5\hat{y}_4\beta_4 + 3\hat{y}_5\beta_5 & (19a) \\ \text{subject to: } & \alpha_{11} + \beta_1 \leq 1 & (19b) \\ & \alpha_{12} + \beta_2 \leq 1 & (19c) \\ & \alpha_{21} + \beta_3 \leq 1 & (19d) \\ & \alpha_{11} + \alpha_{21} + \beta_4 \leq 1 & (19e) \\ & \alpha_{11} + \alpha_{12} + \beta_5 \leq 1 & (19f) \\ & \alpha_{11}, \alpha_{12}, \alpha_{21} \text{ unrestricted} & (19g) \\ & \beta_r \leq 0 \quad \forall r = 1, \dots, 5. & (19h) \end{aligned}$$

**Iteration 1:** We first relax all Benders cuts in the master problem, and solve

$$\text{Minimize } 7 \times (y_1 + y_2 + y_3 + y_4 + y_5) + t \quad (20a)$$

$$\text{subject to: } t \geq 0, y_r \in \{0, 1\} \quad \forall r = 1, \dots, 5. \quad (20b)$$

The optimal solution of (20) is  $\hat{y} = [0, 0, 0, 0, 0]$ ,  $\hat{t} = 0$ . In order to solve the subproblem corresponding to  $\hat{y}$ , we set the objective function (19a) to Maximize  $8\alpha_{11} + 3\alpha_{12} + 5\alpha_{21}$ , and solve DSPTT( $\hat{y}$ ). DSPTT( $\hat{y}$ ) is unbounded having an extreme ray  $\alpha_{11} = 2, \alpha_{12} = -1, \alpha_{21} = -1, \beta_1 = -2, \beta_2 = 0, \beta_3 = 0, \beta_4 = -1, \beta_5 = -1$ , which yields the Benders feasibility cut  $8 - 16y_1 - 5y_4 - 3y_5 \leq 0$ .

**Iteration 2:** We add the generated Benders feasibility cut to our relaxed master problem, and solve

$$\text{Minimize } 7 \times (y_1 + y_2 + y_3 + y_4 + y_5) + t \quad (21a)$$

$$\text{subject to: } 8 - 16y_1 - 5y_4 - 3y_5 \leq 0 \quad (21b)$$

$$t \geq 0, y_r \in \{0, 1\} \quad \forall r = 1, \dots, 5. \quad (21c)$$

An optimal solution of (21) is  $\hat{y} = [1, 0, 0, 0, 0]$ ,  $\hat{t} = 0$ . We set the objective function (19a) to Maximize  $8\alpha_{11} + 3\alpha_{12} + 5\alpha_{21} + 8\beta_1$ , and solve DSPTT( $\hat{y}$ ). DSPTT( $\hat{y}$ ) is, again, unbounded. An extreme ray is  $\alpha_{11} = 0, \alpha_{12} = 0, \alpha_{21} = 1, \beta_1 = 0, \beta_2 = 0, \beta_3 = -1, \beta_4 = -1, \beta_5 = 0$ , which yields the following Benders feasibility cut:  $5 - 5y_3 - 5y_4 \leq 0$ .

**Iteration 3:** We update our relaxed master problem by adding the generated Benders feasibility cut:

$$\text{Minimize } 7 \times (y_1 + y_2 + y_3 + y_4 + y_5) + t \quad (22a)$$

$$\text{subject to: } 8 - 16y_1 - 5y_4 - 3y_5 \leq 0 \quad (22b)$$

$$5 - 5y_3 - 5y_4 \leq 0 \quad (22c)$$

$$t \geq 0, y_r \in \{0, 1\} \quad \forall r = 1, \dots, 5. \quad (22d)$$

An optimal solution of (22) is  $\hat{y} = [0, 0, 0, 1, 1]$ ,  $\hat{t} = 0$ . We set the objective function (19a) to Maximize  $8\alpha_{11} + 3\alpha_{12} + 5\alpha_{21} + 5\beta_4 + 3\beta_5$ , and solve DSPTT( $\hat{y}$ ). This time DSPTT( $\hat{y}$ ) has an optimal solution  $\alpha_{11} = 1, \alpha_{12} = 1, \alpha_{21} = 1, \beta_1 = 0, \beta_2 = 0, \beta_3 = 0, \beta_4 = -1, \beta_5 = -1$ , and the corresponding objective function value is  $t^* = 8$ . Since  $t^* > \hat{t}$ , we generate the following Benders optimality cut:  $16 - 5y_4 - 3y_5 \leq t$ .

**Iteration 4:** The updated relaxed Benders master problem is:

$$\text{Minimize } 7 \times (y_1 + y_2 + y_3 + y_4 + y_5) + t \quad (23a)$$

$$\text{subject to: } 8 - 16y_1 - 5y_4 - 3y_5 \leq 0 \quad (23b)$$

$$5 - 5y_3 - 5y_4 \leq 0 \quad (23c)$$

$$16 - 5y_4 - 3y_5 \leq t \quad (23d)$$

$$t \geq 0, y_r \in \{0, 1\} \quad \forall r = 1, \dots, 5. \quad (23e)$$

An optimal solution of (23) is  $\hat{y} = [0, 0, 0, 1, 1]$ ,  $\hat{t} = 8$ . Note that  $\hat{y}$  is equal to the solution generated in the previous iteration, and therefore  $t^* = 8$ . Since  $t^* = \hat{t}$ , optimality has been reached and we stop.

## References

- [1] J. F. Benders. Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik*, 4(1):238–252, 1962.
- [2] X. Cai, D. C. McKinney, L. S. Lasdon, and Jr. D. W. Watkins. Solving large non-convex water resources management models using generalized Benders decomposition. *Operations Research*, 49(2):235–245, 2001.
- [3] A. Eremin and M. Wallace. Hybrid Benders decomposition algorithms in constraint logic programming. In T. Walsh, editor, *Principles and Practice of Constraint Programming - CP 2001*, volume 2239 of *Lecture Notes in Computer Science*, pages 1–15. Springer, 2001.
- [4] M. S. Bazaraa, H. D. Sherali, and C. C. Shetty. *Nonlinear Programming: Theory and Algorithms*. John Wiley & Sons, Inc., Hoboken, NJ, 2nd edition, 1993.
- [5] A. M. Geoffrion. Generalized Benders decomposition. *Journal of Optimization Theory and Applications*, 10(4):237–260, 1972.
- [6] A. M. Costa. A survey on Benders decomposition applied to fixed-charge network design problems. *Computers & Operations Research*, 32(6):1429–1450, 2005.
- [7] A. K. Andreas and J. C. Smith. Decomposition algorithms for the design of a nonsimultaneous capacitated evacuation tree network. *Networks*, 53(2):91–103, 2009.
- [8] G. Codato and M. Fischetti. Combinatorial Benders’ cuts for mixed-integer linear programming. *Operations Research*, 54(4):756–766, 2006.
- [9] S. S. Nielsen and S. A. Zenios. Scalable parallel Benders decomposition for stochastic linear programming. *Parallel Computing*, 23(8):1069–1088, 1997.
- [10] Z. C. Taşkın, J. C. Smith, and H. E. Romeijn. Mixed-integer programming techniques for decomposing IMRT fluence maps using rectangular apertures. Technical report,

Department of Industrial and Systems Engineering, University of Florida, Gainesville,  
FL, 2009.