Effective Solution Algorithms for Bulk-Robust Optimization Problems*

Christoph Buchheim[1], Enrico Malaguti[2], Michele Monaci[2], Paolo Paronuzzi[2], and Pia Schreynemackers[1,3]

[1]Department of Mathematics, TU Dortmund University
[2]DEI, University of Bologna
[3]Fraunhofer Institute for Material Flow and Logistics IML

October 17, 2025

Abstract

Bulk-robust optimization is a recent paradigm for addressing problems in which the structure of a system is affected by uncertainty. It considers the case in which a finite and discrete set of possible failure scenarios is known in advance, and the decision maker aims to activate a subset of available resources of minimum cost so that a certain property is satisfied, regardless of which scenario occurs. Although theoretical properties of this paradigm have been studied and approximation algorithms have been proposed in the literature, we are not aware of any computational approach. Therefore, we devise a unified exact solution method, based on Benders' decomposition, and apply it to two core problems belonging to this class, the bulk-robust assignment problem and the bulk-robust connectivity problem. In our approach, the master problem determines which resources to activate, while the subproblems verify the feasibility of the master solution in each scenario. We propose combinatorial algorithms for the subproblems that significantly speed up their solution. Our extensive computational results confirm the effectiveness of our approach and the importance of the computational enhancements we propose.

1 Introduction

Traditional approaches to optimization problems assume that the input data are precisely known. Therefore, the quality and practical applicability of the resulting prescription strongly depend on the accuracy of the data. However, in many applications, part of the data may be subject to unpredictable perturbations, which prevents the use of deterministic approaches.

Different paradigms exist in the literature to deal with uncertainty, the most common ones being stochastic optimization and robust optimization. The former assumes that information about the probabilistic distribution of actual data is available and may result in very large optimization models [Birge and Louveaux, 2003]. Conversely, the latter

*Enrico Malaguti, Michele Monaci and Paolo Paronuzzi were funded by the Air Force Office of Scientific Research under award number FA8655-25-1-7013.

only requires to know an uncertainty set where parameters can realize and produces a solution that is feasible and worst-case optimal with respect to *any* realization of the data [Ben-Tal and Nemirovski, 1998, 2000, Bertsimas and Sim, 2004]. As a consequence, this risk-averse paradigm can result in over-conservative solutions when more detailed information on possible data realizations is available. In practice, decision-makers often face situations where the number of plausible perturbations is limited or where some uncertain parameter combinations (i.e., a limited number of scenarios) represent the majority of realistic outcomes. Bulk-robust optimization is a more targeted approach to uncertainty, which has been recently proposed for exploiting additional information when available. Bulk-robust optimization suggests that decision-makers focus their efforts on managing the uncertainty that truly matters, thus producing more efficient and effective solutions.

Bulk-robust optimization has been used for addressing graph problems where the availability of the resources – edges or nodes – is subject to uncertainty. The main difference from classical robust optimization approaches is that bulk-robust optimization is designed for situations where only a limited subset of the resources are subject to the risk of failure. More specifically, bulk-robust optimization defines an *interdiction set* as the set of resources that fail in a specific scenario, and assumes the number of interdiction sets to be limited.

Bulk-robust optimization was introduced in Adjiashvili et al. [2015], where the authors provide complexity results and approximation algorithms for bulk-robust counterparts of the minimum matroid basis problem and of the shortest path problem. In Adjiashvili et al. [2016], the authors study the computational complexity and approximability of the bulk-robust assignment problem and its variants. Additional complexity results addressing the bulk-robust counterpart of the bipartite matching problem, the k-edge disjoint s-t paths problem, and the minimum spanning tree problem are presented in Hommelsheim [2020] and later extended in Adjiashvili et al. [2022], Hommelsheim et al. [2021]. To the best of our knowledge, there are no other publications that specifically focus on bulk-robust optimization, and sound computational approaches are missing in the current literature.

Contributions. In this paper, we start to fill this gap by introducing a general and exact approach for bulk-robust optimization problems, which we illustrate and investigate for the case of the assignment problem and the spanning tree problem. More precisely, we propose a branch-and-cut approach in which valid inequalities are derived as a result of a Benders' decomposition. We discuss the meaning and derivation of the resulting cuts under different theoretical perspectives and computationally analyze the best way to include them in the formulation.

2 General problem

We first define the general bulk-robust optimization problem. For the sake of simplicity, we assume that the problem at hand is defined on an (undirected) graph, with scenarios given as subsets of unsafe edges, which are subject to failure. The decision maker is tasked with determining a subset of edges to be activated while ensuring that a specific property is maintained, regardless of the scenario that unfolds. This framework encapsulates the majority of bulk-robust optimization problems studied in the existing literature. In the following, for $p \in \mathbb{N}$, we write $[p]$ for $\{1, \dots, p\}$.

Definition 2.1 (Bulk-robust optimization problem)**.** Given an undirected graph $G = (V, E)$, a cost $c_e \in \mathbb{R}$ for each edge $e \in E$, and a collection of edge sets $\mathcal{F} = \{F^1, \dots, F^p\}$, i.e., $F^i \subset E$ for $i \in [p]$, the *bulk-robust optimization problem* asks to determine a set $L \subseteq E$ of minimum cost, such that the graph $G^i = (V, L \setminus F^i)$ has some desired property for each $i \in [p]$.

In the following, we will refer to an element of $\mathcal{F}$ as *(failure) scenario*. Edges in F^i (resp. $E \setminus F^i$) are called *unsafe* (resp. *safe*) for scenario i. Finally, we set $n = |V|$ and $m = |E|$, and we will characterize vectors by bold style. For a given subset of vertices $S \subseteq V$ we define $E(S)$ as the subset of edges with both endpoints in S.

2.1 Problem formulation

We first introduce an integer linear programming formulation for bulk-robust optimization problems, where binary *activation* variables z_e $(e \in E)$ determine the edges to be activated, and variables x_e^i $(e \in E, i \in [p])$ determine the feasibility of the required property for each scenario. The model is as follows:

$$\min \quad \sum_{e \in E} c_e z_e \tag{1a}$$

$$\text{s.t.} \quad \boldsymbol{x}^i \in \mathcal{X} \qquad \forall i \in [p] \tag{1b}$$

$$x_e^i = 0 \qquad \forall e \in F^i,\ i \in [p] \tag{1c}$$

$$x_e^i \le z_e \qquad \forall e \in E,\ i \in [p] \tag{1d}$$

$$z_e \in \{0, 1\} \qquad \forall e \in E\ . \tag{1e}$$

Here, $\mathcal{X}$ denotes the convex hull of the feasible set of the underlying problem, which depends on the property to be respected. Given $\boldsymbol{z} \in \mathbb{R}^{|E|}$ and $i \in [p]$, we also define

$$\mathcal{X}^i(\boldsymbol{z}) := \{\boldsymbol{x} \in \mathcal{X} : x_e \le z_e\ \forall e \in E,\ x_e = 0\ \forall e \in F^i\}\ ,$$

so that constraints (1b)–(1d) can be written as $\boldsymbol{x}^i \in \mathcal{X}^i(\boldsymbol{z})$, which only prevents $\mathcal{X}^i(\boldsymbol{z})$ to be empty.

2.2 Decomposition approach

Model (1) shows a block structure in which $\boldsymbol{z}$ variables are the complicating variables, while $\boldsymbol{x}^i$ variables appear only in the constraints describing each scenario $i \in [p]$. To exploit this structure, we design a decomposition approach in which the *master* problem contains only the objective function and the integrality constraints of the variables $\boldsymbol{z}$, and each subproblem is associated with a scenario i and contains the corresponding variables $\boldsymbol{x}^i$. We devise a Benders' decomposition approach (see the survey of Rahmaniani et al. [2017]) in which, at each iteration, the master problem is solved and the feasibility of each scenario is verified by checking whether $\mathcal{X}^i(\boldsymbol{z}) \neq \emptyset$ for the current value of the $\boldsymbol{z}$ variables. When the required graph property corresponds to the feasibility of a linear program (LP), this check can be performed efficiently. If at least one scenario turns out to be infeasible, we derive a Benders' feasibility cut in the space of $\boldsymbol{z}$ variables.

From a more practical perspective, general-purpose LP solvers offer the option of querying the values of dual solutions or dual rays. Thus, in case the LP model has been proved to be infeasible, a textbook implementation of Benders' decomposition would simply exploit this dual information to derive feasibility cuts. Instead, our objective is to develop problem-specific combinatorial algorithms to address subproblems, a scheme that has proven to be effective in reducing the computational effort of the solution process, especially when the subproblems have a well-studied combinatorial structure [Contreras et al., 2011, Fischetti et al., 2017, Randazzo et al., 2001]. While this approach is very general in principle, we illustrate it for the assignment problem in Section 3 and for the connectivity problem in Section 4. Finally, the decomposition scheme can be embedded into a branch-and-cut framework, thus allowing to solve the problem to integer optimality.

2.3 Additional valid inequalities

It is well known that a straightforward application of the classical Benders' decomposition may have a slow convergence and/or present undesirable zigzagging behavior of the primal solutions. This is usually due to the weakness of the initial master problem formulation. Several studies have shown the effectiveness of strengthening the master problem formulation by adding sets of valid inequalities [Naoum-Sawaya and Elhedhli, 2010, Saharidis et al., 2011, Tang et al., 2013]. One straightforward way to produce such inequalities is to consider any inequality $a^\top \boldsymbol{x}^i \geq b$ valid for $\mathcal{X}^i$ that satisfies $a \geq 0$. In this case, using $\boldsymbol{x}^i \leq \boldsymbol{z}$, we immediately derive $a^\top \boldsymbol{z} \geq b$. Examples of this idea will be given in Sections 3.4 and 4.3.

3 Bulk-robust assignment problem

We first consider the case where the required property in Definition 2.1 is the existence of an assignment, i.e., a perfect matching, in a given bipartite graph. The bulk-robust assignment problem is not only known to be NP-hard, but even hard to approximate [Adjiashvili et al., 2016]. Formally, this problem is defined as follows:

Definition 3.1 (Bulk-robust assignment problem, BAP)**.** Given an undirected bipartite graph $G = (U \cup W, E)$, a cost $c_{uw} \in \mathbb{R}$ for each $(u, w) \in E$, and a collection of edge sets $\mathcal{F} = \{F^1, \ldots, F^p\}$, the *bulk-robust assignment problem* asks to determine a set $L \subseteq E$ of minimum cost, such that the graph $G^i = (U \cup W, L \setminus F^i)$ contains a perfect matching for each $i \in [p]$.

In the literature, also the *augmenting* version of this problem has been considered, in which some edges have already been fixed in set L [Hommelsheim, 2020]. This can be modeled here by setting $c_{uw} = 0$ for such edges.

3.1 Problem formulation

We model BAP according to the framework of Section 2.1 by defining variables z_{uw} and x^i_{uw}. For a given solution $\bar{z}$ of the master problem and for each scenario $i \in [p]$, condition $\mathcal{X}^i(\bar{z}) \neq \emptyset$ can be modeled as follows:

$$\sum_{w \in W:\ (u,w) \in E^i} x^i_{uw} \leq 1 \qquad \forall u \in U, \tag{2a}$$

$$\sum_{u \in U:\ (u,w) \in E^i} x^i_{uw} \leq 1 \qquad \forall w \in W, \tag{2b}$$

$$\sum_{(u,w) \in E^i} x^i_{uw} \geq N, \tag{2c}$$

$$x^i_{uw} \leq \bar{z}_{uw} \qquad \forall (u,w) \in E^i, \tag{2d}$$

$$x^i_{uw} \geq 0 \qquad \forall (u,w) \in E^i \tag{2e}$$

Here, we assume $N = |U| = |W| = \frac{n}{2}$ and $E^i = E \setminus F^i$ denotes the set of safe edges for scenario i. We emphasize that the $\boldsymbol{x}$ variables are allowed to be continuous in this model. Indeed, since G is bipartite, whenever (2) is feasible and $\bar{z}$ is integer, there exists an optimal integer solution of (2).

3.2 Decomposition approach

In order to derive feasibility cuts for the master problem, we first reformulate the feasibility problem (2) as a maximization problem by moving the constraint (2c) to the objective function. For the sake of notation simplicity, we omit the index i from variables x^i_{uw} in the following:

$$\max \quad \sum_{(u,w) \in E^i} x_{uw} \tag{3a}$$

$$\text{s.t.} \quad \sum_{w \in W:\ (u,w) \in E^i} x_{uw} \leq 1 \qquad \forall u \in U, \tag{3b}$$

$$\sum_{u \in U:\ (u,w) \in E^i} x_{uw} \leq 1 \qquad \forall w \in W, \tag{3c}$$

$$x_{uw} \leq \bar{z}_{uw} \qquad \forall (u,w) \in E^i, \tag{3d}$$

$$x_{uw} \geq 0 \qquad \forall (u,w) \in E^i. \tag{3e}$$

When disregarding constraints (3d), the resulting model is a maximum cardinality bipartite matching problem. The latter can be solved as a maximum flow problem on a auxiliary directed graph defined as follows:

- there exist a dummy source s and a dummy terminal t;
- the source is connected to each vertex $u \in U$ with an arc (s,u) having capacity equal to 1;
- each vertex $w \in W$ is connected to the terminal with an arc (w,t) having capacity equal to 1;
- each arc (u,w) has capacity equal to 1.

In our case, additional constraints (3d) result in arc capacities equal to the master solution $\bar{z}$. Though these values may be fractional, we can still apply efficient maximum flow algorithms in order to check the feasibility of the problem in polynomial time. In case the resulting flow value is strictly less than N, the current solution $\bar{z}$ is infeasible, and a feasibility cut can be produced by imposing the solution to the dual problem to be at least N. The dual of (3) reads

$$\min \quad \sum_{u \in U} \pi_u + \sum_{w \in W} \pi_w + \sum_{(u,w) \in E^i} \bar{z}_{uw} \lambda_{uw} \tag{4a}$$

$$\text{s.t.} \quad \pi_u + \pi_w + \lambda_{uw} \geq 1 \qquad \forall (u,w) \in E^i \tag{4b}$$

$$\pi_u \geq 0 \qquad \forall u \in U \tag{4c}$$

$$\pi_w \geq 0 \qquad \forall w \in W \tag{4d}$$

$$\lambda_{uw} \geq 0 \qquad \forall (u,w) \in E^i. \tag{4e}$$

Since the primal problem is bounded and always admits the zero solution, the dual is always feasible and the optimal solution values of the two problems coincide. If one solves problem (3) as an LP, an optimal solution of (4) is immediately available. Otherwise, one can use the following proposition to derive an optimal solution of (4) in a purely combinatorial way, starting from an optimal flow.

Proposition 3.2. *Given an optimal solution of* (3)*, any corresponding minimum s-t cut* $(S^*, \bar{S}^*)$ *yields an optimal solution of* (4) *as follows:*

$$\pi_u^* = \begin{cases} 1 & \text{if } u \in \bar{S}^* \\ 0 & \text{otherwise} \end{cases} \qquad \forall u \in U \tag{5}$$

$$\pi_w^* = \begin{cases} 1 & \text{if } w \in S^* \\ 0 & \text{otherwise} \end{cases} \qquad \forall w \in W \tag{6}$$

$$\lambda_{uw}^* = \begin{cases} 1 & \text{if } u \in S^* \text{ and } w \in \bar{S}^* \\ 0 & \text{otherwise} \end{cases} \qquad \forall (u,w) \in E^i \tag{7}$$

Proof. Proof. First, let us show that solution $(\pi_u^*, \pi_w^*, \lambda_{uw}^*)$ is feasible, i.e., it satisfies constraints (4b). Given any minimum s-t cut $(S^*, \bar{S}^*)$ and any edge (u, w), four outcomes are possible: (i) $u \in S^*$ and $w \in \bar{S}^*$; (ii) both u and w are in S^*; (iii) both u and w are in $\bar{S}^*$; (iv) $u \in \bar{S}^*$ and $w \in S^*$; it is easy to see that the solution we provide satisfies constraints (4b) in all these cases. Secondly, we observe that in formulation (4), each arc of the auxiliary graph corresponds to a dual variable (π_u for arcs (s, u) and π_w for arcs (w, t), λ_{uw} for arcs (u, w)), and we assign a value of 1 to each variable when the corresponding arc is part of the minimum s-t cut. Since all such variables appear in the objective function with a coefficient that equals their capacity (1 for arcs (s, u) and (w, t)), the value of solution $(\pi_u^*, \pi_w^*, \lambda_{uw}^*)$ agrees with the capacity of the given minimum s-t cut and, thus, this solution is optimal. □

We recall that we obtain the feasibility cut by imposing that the objective function (4a) be at least N. After moving the z_{uw} variables to the left-hand side, the resulting cut reads

$$\sum_{(u,w) \in E^i} \lambda_{uw}^* z_{uw} \geq N - \sum_{u \in U} \pi_u^* - \sum_{w \in W} \pi_w^*. \tag{8}$$

We note that the same maximum flow may be associated with different minimum s-t cuts, resulting in different feasibility cuts.

Using similar arguments, one can consider an arbitrary s-t cut $(S, \bar{S})$ in Proposition 3.2 and define the corresponding feasibility cut with the associated coefficients

$\pi^{(S,\bar{S})}$ and $\lambda^{(S,\bar{S})}$. The violation of the resulting inequality is precisely the difference between N and the capacity of cut $(S, \bar{S})$. This yields the following family of valid inequalities

$$\sum_{(u,w)\in E^i} \lambda_{uw}^{(S,\bar{S})} z_{uw} \geq N - \sum_{u\in U} \pi_u^{(S,\bar{S})} - \sum_{w\in W} \pi_w^{(S,\bar{S})} \quad \text{for all } s\text{-}t \text{ cuts } (S, \bar{S}). \tag{9}$$

3.3 An alternative viewpoint

Let us consider two subsets $U' \subseteq U$ and $W' \subseteq W$ such that $|U'| \leq |W'|$. Feasibility for a scenario i requires that at least $|W'|$ edges in E^i with an endpoint $w \in W'$ be activated. At most $|U'| < |W'|$ of such edges can have an endpoint $u \in U'$. So, at least $|W'| - |U'|$ of the activated edges must have an endpoint $u \in U \setminus U'$. This natural observation leads to the following family of valid inequalities

$$\sum_{e\in E^i(U\setminus U', W')} z_e \geq |W'| - |U'| \quad \forall U' \subseteq U, W' \subseteq W, |U'| \leq |W'| \tag{10}$$

which are a special case of the inequalities characterizing the up-hull of the perfect matching polytope [Fulkerson, 1970]. Similar inequalities are obtained by swapping the role of U' and W'. It turns out that these inequalities correspond precisely to the inequalities (9) derived above.

Proposition 3.3. *The inequalities* (9) *correspond to the inequalities* (10).

Proof. Proof. First, we show that any inequality (9), induced by an s-t cut $(S, \bar{S})$, corresponds to an inequality (10). Indeed, setting $U' = U \cap \bar{S}$ and $W' = W \cap \bar{S}$, inequality (9) can be rewritten as

$$\sum_{e\in E^i(U\setminus U', W')} z_e \geq N - |U'| - (N - |W'|) = |W'| - |U'|.$$

Since the capacity of the arc entering a vertex in U (resp. leaving a vertex in W) is one, flow conservation implies $|U'| \leq |W'|$, i.e., the cut belongs to family (10).

Conversely, given any pair $W' \subseteq W$, $U' \subseteq U$, such that $|W'| \geq |U'|$, we define the s-t cut such that $S = \{s\} \cup (U \setminus U') \cup (W \setminus W')$ and $\bar{S} = \{t\} \cup U' \cup W'$. Defining the associated coefficients $\pi^{(S,\bar{S})}$ and $\lambda^{(S,\bar{S})}$ as in Proposition 3.2, we precisely obtain the inequality (9). □

Figure 1 illustrates the proof of Proposition (3.3).

We conclude this section by emphasizing that the method we proposed for generating valid inequalities can be applied to any infeasible master solution, regardless of whether it is integer or fractional. As a result, the separation problem for fractional and integer nodes of the branching tree can be addressed in the same manner, with time complexity determined by the algorithm used to solve the maximum flow problem.

3.4 Additional valid inequalities for BAP

For each scenario $i \in [p]$, at least one variable x_{uw}^i associated with a safe edge must be selected for each vertex $u \in U$ (resp. for each vertex $w \in W$). Therefore, by

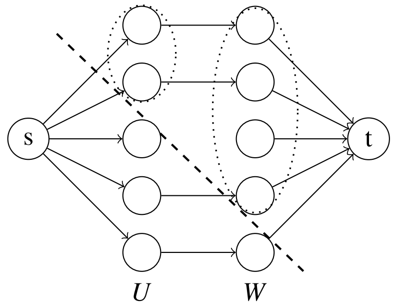

Figure 1: Example of a minimum s-t cut identifying sets U' and W' (circled).

the reasoning given at the end of Section 2.3, for each $i \in [p]$, the following valid inequalities can be added to the master problem:

$$\sum_{w \in W:\ (u,w) \in E^i} z_{uw} \geq 1 \qquad \forall u \in U$$
$$\sum_{u \in U:\ (u,w) \in E^i} z_{uw} \geq 1 \qquad \forall w \in W.$$

Note that these inequalities are special cases of (10), arising when the smaller subset is empty and the larger subset contains a single vertex.

4 Bulk-robust connectivity problem

In the second problem that we consider, the property required in Definition 2.1 is connectivity of the underlying graph. The resulting bulk-robust connectivity problem is formally defined as follows:

Definition 4.1 (Bulk-robust connectivity problem, BCP)**.** Given an undirected graph $G = (V, E)$, a cost $c_e \in \mathbb{R}$ for each edge $e \in E$, and a collection $\mathcal{F}$ of p subsets of edges, i. e., $\mathcal{F} = \{F^1, \dots, F^p\}$ where $F^i \subset E$ for $i \in [p]$, the *bulk-robust connectivity problem* asks to determine a set $L \subseteq E$ of minimum cost, such that the graph $G^i = (V, L \setminus F^i)$ is connected for each $i \in [p]$.

In other words, the problem requires that the set of activated edges contains a spanning tree for each scenario. In Adjiashvili et al. [2015], this problem is introduced as a special case of the *bulk-robust minimum matroid basis* problem and is denoted as the *bulk-robust minimum spanning tree* problem. When the scenario set $\mathcal{F}$ consists of all k-tuples of edges, the problem is denoted as *k–flexible graph connectivity* problem [Adjiashvili et al., 2022].

4.1 Spanning Tree formulation

We model BCP according to the framework of Section 2.1 by means of variables z_e and x^i_e. For a given solution $\bar{z}$ of the master problem and for each scenario $i \in [p]$ (where $E^i = E \setminus F^i$ denotes the set of safe edges), condition $\mathcal{X}^i(\bar{z}) \neq \emptyset$ can be modeled by imposing that solution $\bar{z}$ defines a graph that contains a spanning tree, as follows:

$$\sum_{e \in E^i} x_e^i = n - 1 \tag{11a}$$

$$\sum_{e \in E^i(S)} x_e^i \leq |S| - 1 \qquad \forall S \subseteq V, |S| > 2 \tag{11b}$$

$$x_e^i \leq \bar{z}_e \qquad \forall e \in E^i \tag{11c}$$

$$x_e^i \geq 0 \qquad \forall e \in E^i. \tag{11d}$$

Also in this case, the $\boldsymbol{x}$ variables are allowed to be continuous, because there exists a solution of (11) where $\boldsymbol{x}$ variables assume integer values whenever (11) is feasible and $\bar{z}$ is integer.

4.1.1 Decomposition approach

Rather than formulating problem (11) as a feasibility problem, we move constraint (11a) into the objective function as follows. Again, index i is omitted from the variables x_e^i.

$$\max \quad \sum_{e \in E^i} x_e \tag{12a}$$

$$\text{s.t.} \quad \sum_{e \in E^i(S)} x_e \leq |S| - 1 \qquad \forall S \subseteq V, |S| > 2 \tag{12b}$$

$$x_e \leq \bar{z}_e \qquad \forall e \in E^i \tag{12c}$$

$$x_e \geq 0 \qquad \forall e \in E^i. \tag{12d}$$

The current master solution $\bar{z}$ is infeasible if and only if the optimal value of (12) is strictly less than $n - 1$, and in that case a feasibility cut can be produced by imposing the solution to the dual problem to be at least $n - 1$. The dual of (12) reads:

$$\min \quad \sum_{S \subseteq V, |S| > 2} (|S| - 1)\pi_S + \sum_{e \in E^i} \bar{z}_e \lambda_e \tag{13a}$$

$$\text{s.t.} \quad \sum_{S \subseteq V : e \in E^i(S)} \pi_S + \lambda_e \geq 1 \qquad \forall e \in E^i \tag{13b}$$

$$\pi_S \geq 0 \qquad \forall S \subset V, |S| > 2 \tag{13c}$$

$$\lambda_e \geq 0 \qquad \forall e \in E^i. \tag{13d}$$

The primal problem (12) is bounded and always admits the zero solution. Therefore, the dual is always feasible and the optimal values of (12) and (13) coincide. If one solves problem (12) as an LP, an optimal solution of (13) is immediately available. In the following, we show that an optimal solution of (13) can be computed by means of a combinatorial algorithm.

4.1.2 Optimal primal-dual solution

We propose a procedure that, given $\bar{z}_e \in [0, 1]$ for all $e \in E^i$, computes a feasible solution of (12) which is used to define a feasible solution of (13). By complementary slackness we show optimality of this pair.

The procedure, outlined in Algorithm 1, considers edges one at a time, according to an arbitrary order. At any given iteration, let $\bar{x}_e$ denote the value of variable x_e for $e \in E^i$ (initialized as $\bar{x}_e = 0, \forall e \in E^i$), and let f represent the edge considered in the current iteration. The objective is to determine the maximum feasible value of variable x_f. More precisely, we seek the value α of x_f such that no constraint (12b) is violated. If setting x_f to the resulting value of α satisfies constraint (12c), then we set $\bar{x}_f = \alpha$; otherwise, we resort to $\bar{x}_f = \bar{z}_f$. Therefore, our aim is to identify the largest α such that

$$\alpha \leq -1 + |S| - \sum_{e \in E^i(S)} \bar{x}_e \qquad \forall S \subseteq V, f \in E^i(S) \,, \tag{14}$$

and the algorithm can be summarized as follows:

Algorithm 1 Feasible solution to (12)

```
1: Initialize x̄_e ← 0 for all e ∈ E^i
2: for each edge f ∈ E^i do
3:     Compute the largest α w.r.t. (14)
4:     Set x̄_f ← min(α, z̄_f)
5: end for
```

For Step 3 of this algorithm, we can alternatively express α as

$$\alpha = -1 + \min_{S \subseteq V, f \in E^i(S)} \Big(|S| - \sum_{e \in E^i(S)} \bar{x}_e \Big) \,.$$

We thus need to solve a minimization problem over subsets $S \subseteq V$ containing both endpoints of f. By following an idea of Padberg and Wolsey [1983], we exploit a reduction to a network flow problem as follows. For each vertex $v \in V$, we define $d_v = 2 - \sum_{e \in \delta^i(v)} \bar{x}_e$ where $\delta^i(v)$ is the set of edges incident to v that are safe for scenario $i \in [p]$. Similarly, for $S \subset V$, we define $\delta^i(S)$ as the set of safe edges with exactly one endpoint in the set. We first claim that

$$\alpha = -1 + \frac{1}{2}\Big(\sum_{\substack{v \in V \\ d_v < 0}} d_v + \min_{\substack{S \subseteq V \\ f \in E^i(S)}} \big(\sum_{e \in \delta^i(S)} \bar{x}_e + \sum_{\substack{v \in S \\ d_v > 0}} d_v - \sum_{\substack{v \in V \setminus S \\ d_v < 0}} d_v \big) \Big). \tag{15}$$

The proof of this result is given in Appendix A.

Now, we build an auxiliary directed and capacitated graph from G^i, consisting of two opposite arcs for each edge of G^i and two additional vertices s and t. The existing arcs and their respective capacities $\boldsymbol{c}$ are

- $c_{(u,w)} = c_{(w,u)} = \bar{x}_e$ for all $e = (u, w) \in E^i \setminus f$;
- $c_f = \infty$ for both arcs corresponding to edge f;
- $c_{(s,u)} = d_u$ for all vertices such that $d_u > 0$;
- $c_{(u,t)} = -d_u$ for all vertices such that $d_u < 0$ and u is not an endpoint of f;
- $c_{(u,t)} = \infty$ for one arbitrary endpoint u of f.

By computing a minimum s-t cut (or, equivalently, a maximum s-t flow) on this graph, we obtain a solution in which both endpoints of f belong to the t-side of the cut, and whose objective value coincides with

$$\min_{\substack{S\subseteq V\\ f\in E^i(S)}} \Big(\sum_{e\in\delta^i(S)} \bar{x}_e + \sum_{v\in S:d_v>0} d_v - \sum_{v\in V\setminus S:d_v<0} d_v \Big) \,, \tag{16}$$

thus yielding the desired value of α for the current iteration. Let S be the subset of vertices of the cut including t (and the endpoints of f). By setting $\bar{x}_f = \min(\alpha, \bar{z}_f)$, the constraint (12b) associated with S is tight when $\alpha \le \bar{z}_f$.

Without loss of generality, we can assume disjointness of subsets whose associated constraint (12b) is tight. Indeed, the following result shows that two intersecting sets for which (12b) is tight can always be replaced by their union.

Lemma 4.2. *Let $\bar{x}$ be a feasible solution of* (12) *such that* (12b) *is tight for two sets $S_1, S_2 \subseteq V$ with $S_1 \cap S_2 \neq \emptyset$. Then* (12b) *is also tight for $S_1 \cup S_2$.*

Proof. Proof. We have that

$$\begin{aligned}\sum_{e\in E^i(S_1\cup S_2)} \bar{x}_e &\ge \underbrace{\sum_{e\in E^i(S_1)} \bar{x}_e}_{=|S_1|-1} + \underbrace{\sum_{e\in E^i(S_2)} \bar{x}_e}_{=|S_2|-1} - \sum_{e\in E^i(S_1\cap S_2)} \bar{x}_e \\ &\ge |S_1| - 1 + |S_2| - 1 - |S_1 \cap S_2| + 1 = |S_1 \cup S_2| - 1 \,,\end{aligned}$$

where the second inequality follows from (12b) if $|S_1 \cap S_2| > 2$, from (12c) together with $\bar{z}_e \le 1$ if $|S_1 \cap S_2| = 2$, and is straightforward if $|S_1 \cap S_2| = 1$. □

By this result, we can now produce a dual solution and show optimality of the primal-dual pair.

Proposition 4.3. *The solution produced by Algorithm 1 is optimal for* (12) *and, given the corresponding disjoint sets $S_1^*, \dots, S_k^*$, an optimal solution of* (13) *is as follows:*

$$\begin{aligned}\pi_S^* &= \begin{cases} 1 & \text{if } S \in \{S_1^*, \dots, S_k^*\} \\ 0 & \text{otherwise} \end{cases} \qquad (S \subseteq V, |S| > 2) \\ \lambda_e^* &= \begin{cases} 1 & \text{if } e \notin E^i(S_1^*) \cup \dots \cup E^i(S_k^*) \\ 0 & \text{otherwise} \end{cases} \qquad (e \in E^i)\end{aligned} \tag{17}$$

Proof. Proof. This solution is feasible because, by construction, the left-hand side of each constraint (13b) associated with an edge $e \in E^i$ either includes exactly one variable $\pi_S^* = 1$ (if $e \in E^i(S)$) or has $\lambda_e^* = 1$. In particular, all constraints (13b) are satisfied with equality, so in order to show that both the primal and the dual solution are optimal, it suffices to verify one direction of the complementarity slackness conditions. For this, first observe that $\pi_S^* = 1$ means that (12b) is tight by definition of π^*. Moreover, if $\lambda_e^* = 1$ for some $e \in E$, we know that e does not belong to any set $E(S)$ such that (12b) is tight for S. This means that x_e was set to $\bar{z}_e$ in the corresponding iteration of Algorithm 1, thus (12c) is tight for e. □

We conclude this section by observing that in the special case $\bar{z}_e \in \{0, 1\}$, for all $e \in E^i$, the problem can be solved in a more efficient way, as follows. Denoting by $E^i(\bar{z}) \subseteq E^i$ the edges in E^i that are activated by $\bar{z}$, the primal problem (12) asks for a maximum cardinality spanning forest F in $(V, E^i(\bar{z}))$. This problem is solvable in $O(|E^i(\bar{z})|)$ time by iteratively applying a breadth-first search on graph $(V, E^i(\bar{z}))$. An optimal solution with value $n-k$ induces k disjoint vertex sets $S_1^*, \ldots, S_k^*$ corresponding to the connected components of F. Also in this case, Proposition 4.3 shows the dual optimality of the solution defined as in (17).

4.1.3 The feasibility cut

If the optimal value of (12) is strictly less than $n-1$, the current solution $\bar{z}$ is infeasible and a feasibility cut can be derived imposing the dual objective function (13a) to be at least $n-1$. After re-arranging the terms, the resulting cut reads

$$\sum_{e \in E^i} \lambda_e^* z_e \geq n - 1 - \sum_{S \subseteq V, |S|>2} (|S| - 1)\pi_S^*.$$

Indeed, as shown in Proposition 4.3, for any collection of non-empty disjoints vertex sets $P = \{S_1, \ldots, S_\ell\}$, the dual solution as defined in (17), which we denote by π^P and λ^P, turns out to be dual feasible. This observation yields the following family of valid inequalities

$$\sum_{e \in E^i} \lambda_e^P z_e \geq n - 1 - \sum_{S \subseteq V, |S|>2} (|S| - 1)\pi_S^P \quad \forall P = \{S_1, \ldots, S_\ell\}.$$

By replacing the values of the dual variables as specified in (17), we obtain

$$\sum_{e \in E^i \setminus \bigcup_{j=1}^{\ell} E(S_j)} z_e \geq n - 1 - \sum_{j=1}^{\ell} (|S_j| - 1) \geq n - 1 - (n - \ell) = \ell - 1.$$

In summary, we have

$$\sum_{e \in E^i \setminus \bigcup_{j=1}^{\ell} E(S_j)} z_e \geq \ell - 1 \quad \forall P = \{S_1, \ldots, S_\ell\}. \tag{18}$$

These inequalities impose that, for disjoint sets $S_1, \ldots, S_\ell$, at least $\ell - 1$ of the edges in $E^i \setminus \bigcup_{j=1}^{\ell} E(S_j)$ are activated. The same observation can also be derived from the Nash-Williams theorem [Nash-Williams, 1961] together with the reasoning of Section 2.3, where $S_1, \ldots, S_\ell$ can be extended to a partition of V by adding singletons containing the vertices in $V \setminus \cup_{j=1}^{\ell} S_i$. However, unlike our approach, the latter does not directly yield a separation algorithm.

4.2 Flow formulation

In this section, we propose an alternative way of modeling the feasibility problem for fixed scenario $i \in [p]$ and solution $\bar{z}$ of the master problem, resulting in a different family of valid inequalities. The two approaches may even cut off different fractional solutions $\bar{z}$ of the master problem, while necessarily defining the same set of feasible *binary* solutions. In other words, the two approaches are equivalent as long as $\bar{z}$ is integer but otherwise define a different feasible set.

More specifically, we model connectivity by imposing that one can send one unit of flow from an arbitrary source $s \in V$ to all other vertices. For each $e = (u, v) \in E^i$, we define two arcs (u, v) and (v, u) both with capacity $\bar{z}_e$, and let A be the resulting set of arcs. For each $t \in V \setminus \{s\}$, we introduce variables f_a^t modeling the flow on arc $a = (u, v)$ when $t \in V \setminus \{s\}$ is the terminal. The condition $\mathcal{X}^i(\bar{z}) \neq \emptyset$ can then be modeled by imposing, for each $t \in V \setminus \{s\}$, the following constraints:

$$
\begin{aligned}
\sum_{a \in \delta^+(s)} f_a^t &\geq 1 && \\
\sum_{a \in \delta^+(u)} f_a^t - \sum_{a \in \delta^-(u)} f_a^t &= 0 && u \in V \setminus \{s, t\} \\
f_a^t &\leq \bar{z}_{uv} && a = (u, v) \in A \\
f_a^t &\geq 0 && a \in A .
\end{aligned}
$$

We again turn this into a maximum flow problem by moving the first constraint to the objective function, and obtain the dual of the resulting problem as

$$
\begin{aligned}
\min \quad & \sum_{a=(u,v) \in A} \bar{z}_{uv} \lambda_a^t && && \text{(20a)} \\
\text{s.t.} \quad & \lambda_a^t - \pi_u^t \geq 1 && a = (s, u) \in \delta^+(s) && \text{(20b)} \\
& \lambda_a^t + \pi_u^t \geq 0 && a = (u, t) \in \delta^-(t) && \text{(20c)} \\
& \lambda_a^t + \pi_u^t - \pi_v^t \geq 0 && a = (u, v) \in A \setminus (\delta^+(s) \cup \delta^-(t)) && \text{(20d)} \\
& \lambda_a^t \geq 0 && a \in A . && \text{(20e)}
\end{aligned}
$$

For each terminal t, an optimal dual solution of the maximum flow problem can be obtained by computing a minimum s-t cut in G^i, say $(S, \bar{S})$, and one can show that there exists a dual solution in which $\lambda_a^t = 1$ for all arcs $a \in \delta^+(S)$, and 0 for all remaining arcs.

In case the maximum flow is strictly less than 1 for some terminal t, the current solution $\bar{z}$ is infeasible for the scenario considered, and a feasibility cut can be produced by imposing the dual objective function (20a) be at least 1. By observing that only arcs in $\delta^+(S)$ give a non-zero contribution in the objective function, and that each such arc corresponds to one edge in $\delta(S)$, the resulting inequality reads

$$
\sum_{e \in \delta(S)} z_e \geq 1 . \tag{21}
$$

which is obviously valid for all non-empty vertex sets $S \subsetneq V$, i.e., for all non-trivial s-t cuts. This already follows from the reasoning of Section 2.3, since (21) holds for every spanning tree. We observe that each of these cuts corresponds to an inequality of type (18) for $\ell = 2$, i.e., the family of inequalities defined in this section is included in the one defined by (18). Indeed, it is well known that the constraints of type (21) do not yield a complete description of the spanning tree polytope. Although this implies that the bounds resulting by adding (21) to the master are in general weaker than those resulting by adding (18), we show by experimental analysis that faster separation of the latter can be beneficial from a computational point of view.

4.3 Additional valid inequalities for BCP

For each scenario $i \in [p]$, any feasible solution must select at least $n - 1$ variables x_e^i associated with a safe edge. Therefore, for each $i \in [p]$, the following valid inequality can be added to the master problem:

$$\sum_{e \in E^i} z_e \geq n - 1 \ . \tag{22}$$

In addition, for each vertex $v \in V$, any feasible solution must select at least one variable x_e^i associated with a safe incident edge of v. This yields the following valid inequalities:

$$\sum_{e \in \delta^i(v)} z_e \geq 1 \qquad v \in V \ . \tag{23}$$

Although both (22) and (23) can be seen as special cases of (18), we include them in our initial master problem formulation.

5 Computational Experiments

In this section, we present computational results obtained by applying our algorithms to various instances of BAP and BCP. For both problems, we first examine the performance achieved when solving the continuous relaxation and then extend our analysis to the integer problem. This allows us to evaluate the tightness of the formulations and the overall effort needed to obtain optimal integer solutions.

Computational setting. Our algorithms are implemented in C++ and use the Gurobi 10.0 solver to address mathematical formulations and manage branch-and-cut schemes. All experiments are run in single-thread mode, with a time limit of 1 hour and all other Gurobi parameters left at their default values. The experiments are carried out on a computer featuring an AMD Ryzen 3960X processor clocked at 3.8 GHz, 128 GB RAM, running Linux. For both BAP and BCP, all maximum flow subproblems are solved by means of the push-relabel algorithm proposed by Goldberg and Tarjan [1988], which exhibits a time complexity of $O(nm \log(n^2/m))$, where m and n denote the number of nodes and arcs in the corresponding network, respectively.

5.1 Computational Experiments for BAP

Benchmark instances. We generated a benchmark of instances defined by complete bipartite graphs with $N \in \{20, 40, 60, 80, 100, 120\}$, and edge costs randomly generated as integers between 1 and 100. Instances have $p = N$ scenarios, and $|F^i| \in \{0.2 \cdot N, 0.3 \cdot N, 0.4 \cdot N, 0.5 \cdot N\}$ for all $i \in [p]$. To define the set of scenarios, for each instance we solve a minimum cost assignment problem without considering failures and define one scenario for each edge in the optimal assignment. Then, we add to each scenario other $|F^i| - 1$ randomly selected edges. We generated 10 different instances for each parameter combination. Thus, the benchmark contains 240 instances in total.

Feasibility cut selection. It is well known that the performance of Benders' decomposition strongly depends on cuts selection policies when multiple choices are possible [Seo et al., 2022]. In our setting, the generated feasibility cut depends on the minimum *s*-*t* cut, which is not necessarily unique for a given maximum flow. While all candidate cuts exhibit the same violation with respect to the current master solution, they typically have different impact at subsequent iterations, when the master is re-optimized. Our preliminary computational experiments confirmed the effectiveness of a strategy which prioritizes inequalities that (a) do not include z variables associated with edges that have been activated in the current master solution, and (b) are sparse.

To achieve these two goals, given a maximum flow, we build the minimum s-t cut in which S is the connected component identified by the vertices which can be reached from s either by the forward use of unsaturated arcs (i.e., arcs where the flow is smaller than the capacity) or by the backward use of arcs that have positive flow. The generated s-t cut does not contain any arc (u, w) with $\bar{z}_{uw} = 1$ and hence yields an inequality that satisfies (a). An alternative minimum s-t cut that does not contain any arc (u, w) with $\bar{z}_{uw} = 1$ can be obtained by a similar approach in which set $\bar{S}$ is defined starting from node t. In order to pursue objective (b), we generate both inequalities and select the one that contains the smaller number of variables.

Algorithms under consideration. We tested the following algorithms for BAP:

- COMP: the compact model (1) solved by Gurobi;
- B-*LP*: Benders' decomposition where feasibility cuts are separated by solving problems (2) as linear programs with Gurobi;
- B-*flow*: Benders' decomposition where feasibility cuts are separated by an ad hoc maximum flow algorithm as explained in Section 3.2. The first minimum s-t cut returned by this algorithm is used to derive the feasibility cut;
- B-*flow**: Benders' decomposition where feasibility cuts are separated by an ad hoc maximum flow algorithm as explained in Section 3.2. Cut selection is performed as described above.

For all algorithms based on Benders' decomposition, a single feasibility cut is added for each infeasible solution of the master problem. Scenarios are considered in a random order, and the master is re-optimized as soon as a violated inequality is detected. In addition, the valid inequalities introduced in Section 3.4 are included in the master problem.

5.1.1 Results on continuous relaxations

Table 1 evaluates the continuous relaxation of BAP in terms of computing time of solution algorithms and tightness of the formulation. Each cell shows an average value computed over 10 instances. The first and second columns display the number of vertices and size of scenarios, with the latter expressed as a fraction of the former. Then, for each algorithm, the table reports the average computing time in seconds. Finally, the table reports the average percentage gap, computed as $(z^* - LB)/z^* \times 100$, where z^* is the optimal integer solution and LB is the dual bound, which is the same for all algorithms. This table reports the results for instances with up to $N = 80$, as for larger values of N, some methods cannot solve the continuous relaxation within the time limit. However, the selected instances highlight the key differences between the computational performance of the algorithms.

As expected, the computing time for COMP is very large for large values of N. Concerning methods based on Benders' decomposition, a textbook implementation, such as B-*LP*, reduces the computing time by almost an order of magnitude. The same performance is obtained by B-*flow*, although the combinatorial procedure used to check feasibility is extremely fast in practice. The lack of improvement is due to the naive

N	$\frac{\lvert F^i \rvert}{N}$	COMP	B-*LP*	B-*flow*	B-*flow**	Gap
20	0.2	0.11	0.08	0.02	0.02	0.11
	0.3	0.18	0.15	0.07	0.02	0.47
	0.4	0.26	0.16	0.08	0.04	1.02
	0.5	0.27	0.19	0.09	0.04	2.20
40	0.2	14.65	5.23	3.68	0.61	0.37
	0.3	23.27	5.74	2.89	0.82	1.32
	0.4	16.99	5.39	2.81	0.77	2.18
	0.5	17.29	5.28	2.72	0.70	2.63
60	0.2	147.47	37.12	39.67	5.41	0.58
	0.3	150.69	45.02	36.15	7.37	1.35
	0.4	126.03	45.06	31.19	7.15	2.55
	0.5	122.37	39.85	21.89	6.44	3.45
80	0.2	1060.61	213.21	491.22	28.16	0.21
	0.3	896.90	235.27	342.67	36.39	1.36
	0.4	695.71	193.37	178.54	25.77	2.29
	0.5	568.62	166.76	100.86	23.21	3.42
		240.09	62.37	78.41	8.93	1.59

Table 1: Continuous relaxation of BAP: average computing time and percentage gap.

selection of a minimum s-t cut used to derive the feasibility cut. A more sophisticated strategy for cut selection, as in B-*flow**, reduces the computing time by a further order of magnitude, thus confirming that cut selection is of paramount importance for this kind of approach. As to the average percentage gap, it increases with the fraction of failing edges in the scenarios, while still being rather small for all instance sizes (at most 3.45%).

5.1.2 Results on the integer problem

When integer solutions are required, Benders' decomposition has to be embedded into an enumerative scheme. To this end, we use Gurobi's branch-and-cut framework. Based on the results presented in the previous section, we have chosen to focus our analysis on B-*flow** and to compare it with the reference approach COMP. For B-*flow**, as suggested by preliminary experiments, we separate fractional solutions at the root node only.

Table 2 reports, for the two solution methods, the average computing time, computed on all instances, the number of instances solved to optimality, and the average optimality gap at the time limit. For some instances, this figure is not available as no lower or upper bound can be computed. For this reason, we report in parentheses the number of instances for which the figure is evaluated.

The table shows that COMP is able to consistently solve instances with up to 40 nodes and fails for 2 instances with 60 nodes. It can solve 21 out of 40 instances with $N = 80$ and only one instance for both $N = 100$ and $N = 120$. Conversely, B-*flow** solves all instances with up to 80 nodes to optimality with a computing time that is an order of magnitude smaller than that of COMP. It is unable to solve 6 instances out of 40 for $N = 100$, and 10 instances with $N = 120$. In general, both methods require more time for instances with scenarios having more unsafe edges.

N	$\frac{\|F^i\|}{N}$	COMP Time	COMP Opt	COMP Gap		B-*flow** Time	B-*flow** Opt	B-*flow** Gap	
20	0.2	0.15	10	0.00		0.01	10	0.00	
	0.3	0.37	10	0.00		0.03	10	0.00	
	0.4	0.84	10	0.00		0.05	10	0.00	
	0.5	1.00	10	0.00		0.08	10	0.00	
40	0.2	19.28	10	0.00		0.75	10	0.00	
	0.3	35.46	10	0.00		2.52	10	0.00	
	0.4	67.15	10	0.00		6.86	10	0.00	
	0.5	69.24	10	0.00		6.29	10	0.00	
60	0.2	268.56	10	0.00		20.92	10	0.00	
	0.3	497.01	10	0.00		29.87	10	0.00	
	0.4	898.23	9	0.06		59.49	10	0.00	
	0.5	1548.70	9	0.03		111.28	10	0.00	
80	0.2	1374.59	10	0.00		16.52	10	0.00	
	0.3	2859.14	7	0.52		107.85	10	0.00	
	0.4	2876.11	3	1.20		286.15	10	0.00	
	0.5	3401.91	1	2.30		666.54	10	0.00	
100	0.2	3294.78	1	1.67	(2)	219.16	10	0.00	
	0.3	3599.92	0	-		635.11	9	0.00	(9)
	0.4	3599.92	0	52.75	(2)	975.97	9	0.00	(9)
	0.5	3599.95	0	28.67	(5)	2599.94	6	0.97	(7)
120	0.2	3577.61	1	0.00	(1)	221.55	10	0.00	
	0.3	3599.87	0	-		907.35	10	0.00	
	0.4	3599.90	0	-		2334.45	9	0.00	(9)
	0.5	3599.84	0	-		3519.09	1	0.00	(1)
		1766.23	141	1.73		530.33	224	0.03	

Table 2: Problem BAP: average computing time, optimal solutions and optimality gap.

5.2 Computational Experiments for BCP

Benchmark instances. We generated a benchmark of instances defined by complete graphs with $n \in \{40, 60, 80, 100, 120, 140, 160\}$, and edge costs randomly generated as integers between 1 and 100. Instances have $p = n - 1$ scenarios, and $|F^i| \in \{0.2 \cdot n, 0.3 \cdot n, 0.4 \cdot n, 0.5 \cdot n\}$ for all scenarios $i \in [p]$. To define the set of scenarios, for each instance we solve a minimum spanning tree problem without considering failures, and, for each edge in the optimal tree, we initialize a scenario where that edge fails. Then, each scenario i is completed by randomly selecting other $|F^i| - 1$ unsafe edges. We generated 10 different instances for each combination of the parameters considered. Thus, the benchmark contains 280 instances in total.

Algorithms under consideration. We tested the following algorithms for BCP:

- B-*span*: Benders' decomposition where feasibility cuts are separated by means of the combinatorial procedure described in Section 4.1;
- B-*flow*: Benders' decomposition where feasibility cuts are separated as explained in Section 4.2. The source vertex s in this approach is immaterial; we always

choose the first vertex as source, while terminals $t \in \{2, \ldots, n\}$ are considered in random order.

For both algorithms, a single cut is added for each infeasible solution of the master problem. Scenarios are considered in a random order, and the master is re-optimized as soon as a violated inequality is detected. In addition, the valid inequalities introduced in Section 4.3 are included in the master problem.

We did not consider the direct solution of model (1) using Gurobi. Regarding the spanning tree formulation, addressing (11) would have required the separation of subtour elimination constraints, which is out of the scope of our experiments and whose outcome would be highly sensitive to implementation. Concerning the compact flow-based formulation of the spanning tree problem, this has an unmanageable number of variables and constraints.

5.2.1 Results on continuous relaxations

In this section, we compare the continuous relaxation of the two formulations. Table 3 reports, for both B-*span* and B-*flow*, the average computing time and the average percentage gap, computed with respect to the best known solution. For some instances, B-*span* hits the time limit before all feasibility cuts are added, and the percentage gap cannot be computed. In these cases, we report in parentheses the number of instances for which the gap is computed.

These results reflect the fact that B-*flow* considers a proper subset of constraints included in formulation B-*span*, and therefore provides a weaker dual bound. However, the tighter relaxation obtained by B-*span* requires a considerably larger computational effort, which is on average two orders of magnitude larger than that required by B-*flow*.

5.2.2 Results on the integer problem

The previous section showed that dual bounds resulting from B-*span* are tighter, but those for B-*flow* can be computed in much shorter time. It is thus an interesting question which method is preferable for solving BCP to optimality, i.e., with integrality constraints on the variables. Table 4 reports the results obtained by algorithms B-*flow* and B-*span* for this setting. Also for BCP we separate fractional solutions at the root node only.

It turns out that both methods consistently find optimal solutions for instances with up to 100 vertices, with B-*span* being slightly faster than B-*flow*. For $n = 120$, B-*span* hits the time limit in two cases, while B-*flow* still solves all instances to optimality. For larger instances both methods struggle in solving the instances to optimality, and, in some cases, they hit the time limit without a lower or upper bound, thus preventing to compute the optimality gap. Overall, the two methods perform similarly: B-*flow* finds three more optimal solutions although, on average, B-*span* is marginally faster and returns smaller gaps.

n	$\frac{\vert F^i\vert}{n}$	B-*span* Time	B-*span* Gap		B-*flow* Time	B-*flow* Gap
40	0.2	4.83	2.41		0.14	4.99
	0.3	5.33	2.18		0.14	5.27
	0.4	6.00	2.55		0.14	5.34
	0.5	5.48	3.26		0.15	6.19
60	0.2	38.22	2.86		0.90	5.72
	0.3	36.50	2.91		0.84	5.56
	0.4	33.24	3.40		0.85	5.89
	0.5	39.08	3.37		0.98	6.09
80	0.2	157.77	2.73		3.39	5.72
	0.3	438.92	2.71		3.10	5.71
	0.4	186.08	2.95		3.13	5.54
	0.5	308.52	3.37		2.84	5.80
100	0.2	1324.36	2.58	(7)	7.91	5.17
	0.3	1161.35	3.24	(8)	7.76	5.29
	0.4	1415.28	3.72	(8)	8.13	5.97
	0.5	987.22	3.93	(9)	6.54	5.85
120	0.2	1903.98	2.06	(7)	19.58	4.50
	0.3	1951.71	2.55	(7)	15.10	4.68
	0.4	1599.63	3.41	(8)	15.37	5.17
	0.5	2058.15	3.40	(8)	14.13	5.16
140	0.2	2679.40	1.38	(5)	31.07	4.29
	0.3	2477.66	2.34	(6)	28.26	4.39
	0.4	1895.82	2.48	(8)	24.20	4.02
	0.5	2939.22	2.76	(3)	27.96	3.98
160	0.2	3310.61	1.40	(3)	57.79	3.54
	0.3	2925.09	7.21	(5)	61.61	9.16
	0.4	3383.04	2.52	(3)	54.79	4.01
	0.5	2933.49	42.59	(4)	49.88	20.41
		1293.07	3.70		15.95	5.84

Table 3: Continuous relaxation of BCP: average computing time and percentage gap.

n	$\frac{\lvert F^i \rvert}{n}$	B-*span* Time	Opt	Gap		B-*flow* Time	Opt	Gap	
40	0.2	1.51	10	0.00		1.98	10	0.00	
	0.3	1.23	10	0.00		1.97	10	0.00	
	0.4	1.52	10	0.00		1.51	10	0.00	
	0.5	1.81	10	0.00		2.62	10	0.00	
60	0.2	31.54	10	0.00		32.49	10	0.00	
	0.3	22.45	10	0.00		14.76	10	0.00	
	0.4	20.02	10	0.00		17.35	10	0.00	
	0.5	11.63	10	0.00		14.28	10	0.00	
80	0.2	66.82	10	0.00		97.23	10	0.00	
	0.3	68.89	10	0.00		86.89	10	0.00	
	0.4	45.78	10	0.00		76.69	10	0.00	
	0.5	57.88	10	0.00		62.33	10	0.00	
100	0.2	255.53	10	0.00		432.32	10	0.00	
	0.3	283.75	10	0.00		344.08	10	0.00	
	0.4	416.59	10	0.00		469.86	10	0.00	
	0.5	149.43	10	0.00		215.25	10	0.00	
120	0.2	1161.60	8	7.86		784.28	10	0.00	
	0.3	1071.37	10	0.00		700.09	10	0.00	
	0.4	308.15	10	0.00		706.61	10	0.00	
	0.5	531.58	10	0.00		702.64	10	0.00	
140	0.2	2766.30	4	0.19	(6)	2497.62	6	6.47	(9)
	0.3	2734.41	5	1.77	(9)	2459.42	7	2.05	
	0.4	2029.96	7	0.31		2234.45	8	0.12	
	0.5	1701.49	9	5.32		1725.54	8	0.11	
160	0.2	3029.79	5	0.22	(8)	3402.03	2	9.37	(8)
	0.3	3365.43	2	5.93	(8)	3320.83	2	5.29	(9)
	0.4	3127.99	3	0.44	(9)	3112.62	5	0.73	
	0.5	2748.04	5	7.15		2942.32	3	8.04	
		830.10	238	1.02		906.53	241	1.06	

Table 4: Problem BCP: average computing time, optimal solutions and optimality gap.

6 Conclusion and future directions

In this paper, we have explored the bulk-robust paradigm for optimization problems under uncertainty from a computational perspective. We introduced a general solution approach based on Benders' decomposition and tested this approach on two classical bulk-robust problems, namely the bulk-robust assignment problem and the bulk-robust connectivity problem.

For the former, we formulated the feasibility check as a maximum flow problem and noticed that the resulting Benders' cuts have a natural interpretation as general cuts for the matching polytope. For the latter, we characterized scenario feasibility using two distinct approaches: (i) enforcing the existence of a spanning tree, and (ii) ensuring the possibility of sending a unit of flow from one vertex to all others. The second approach leads to weaker bounds which, however, can be computed much more quickly in practice. This shows, in particular, that different formulations of the Benders' subproblems may lead to very different results, even though they are equivalent for integer values of the master variables. Computationally, we demonstrated the effectiveness of our algorithms on a benchmark of randomly generated instances.

References

D Adjiashvili, S Stiller, and R Zenklusen. Bulk-robust combinatorial optimization. *Mathematical Programming*, 149:361–390, 2015.

D Adjiashvili, V Bindewald, and D Michaels. Robust assignments via ear decompositions and randomized rounding. In Ioannis Chatzigiannakis, Michael Mitzenmacher, Yuval Rabani, and Davide Sangiorgi, editors, *43rd International Colloquium on Automata, Languages, and Programming (ICALP 2016)*, volume 55 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 71:1–71:14, Dagstuhl, Germany, 2016. Schloss Dagstuhl–Leibniz-Zentrum für Informatik.

D Adjiashvili, F Hommelsheim, and M Mühlenthaler. Flexible graph connectivity: Approximating network design problems between 1-and 2-connectivity. *Mathematical Programming*, 192(1):409–441, 2022.

A Ben-Tal and A Nemirovski. Robust convex optimization. *Mathematics of Operations Research*, 23(4):769–805, 1998.

A Ben-Tal and A Nemirovski. Robust solutions of linear programming problems contaminated with uncertain data. *Mathematical Programming*, 88:411–424, 2000.

D Bertsimas and M Sim. The price of robustness. *Operations Research*, 52(1):35–53, 2004.

JR Birge and F Louveaux. *Introduction to Stochastic Programming*. Springer, New York, 2003.

I Contreras, J-F Cordeau, and G Laporte. Benders decomposition for large-scale uncapacitated hub location. *Operations Research*, 59(6):1477–1490, 2011.

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

DR Fulkerson. Blocking polyhedra. In B. Harris, editor, *Graph Theory and its Applications*, pages 93–112. Academic Press, New York, 1970.

AV Goldberg and RE Tarjan. A new approach to the maximum-flow problem. *Journal of the ACM*, 35(4):921–940, 1988.

F Hommelsheim. *Complexity of bulk-robust combinatorial optimization problems*. PhD thesis, TU Dortmund University, 2020.

F Hommelsheim, M Mühlenthaler, and O Schaudt. How to secure matchings against edge failures. *SIAM Journal on Discrete Mathematics*, 35(3):2265–2292, 2021.

J Naoum-Sawaya and S Elhedhli. A nested Benders decomposition approach for telecommunication network planning. *Naval Research Logistics*, 57(6):519–539, 2010.

CSJ Nash-Williams. Edge-disjoint spanning trees of finite graphs. *Journal of the London Mathematical Society*, 1(1):445–450, 1961.

M. W. Padberg and L. A. Wolsey. Trees and cuts. In *Combinatorial Mathematics: Proceedings of the International Colloquium on Graph Theory and Combinatorics*, pages 511–517. Elsevier, Amsterdam, 1983.

R Rahmaniani, TG Crainic, M Gendreau, and W Rei. The Benders decomposition algorithm: A literature review. *European Journal of Operational Research*, 259(3): 801–817, 2017.

CD Randazzo, HPL Luna, and P Mahey. Benders decomposition for local access network design with two technologies. *Discrete Mathematics & Theoretical Computer Science*, 4, 2001.

GKD Saharidis, M Boile, and S Theofanis. Initialization of the Benders master problem using valid inequalities applied to fixed-charge network problems. *Expert Systems with Applications*, 38(6):6627–6636, 2011.

K Seo, S Joung, C Lee, and S Park. A closest Benders cut selection scheme for accelerating the Benders decomposition algorithm. *INFORMS Journal on Computing*, 34 (5):2804–2827, 2022.

L Tang, W Jiang, and GKD Saharidis. An improved Benders decomposition algorithm for the logistics facility location problem with capacity expansions. *Annals of Operations Research*, 210:165–190, 2013.

A Proof of (15)

For a given S, by using the definition of d_v, we have

$$-2\sum_{e\in E^i(S)}\bar{x}_e = \sum_{e\in\delta^i(S)}\bar{x}_e - \sum_{e\in\delta^i(S)}\bar{x}_e - 2\sum_{e\in E^i(S)}\bar{x}_e = \sum_{e\in\delta^i(S)}\bar{x}_e - \sum_{v\in S}\sum_{e\in\delta^i(v)}\bar{x}_e$$
$$= \sum_{e\in\delta^i(S)}\bar{x}_e - \sum_{v\in S}(2-d_v) = \sum_{e\in\delta^i(S)}\bar{x}_e - 2|S| + \sum_{v\in S} d_v \ ,$$

which can be equivalently written as

$$|S| - \sum_{e\in E^i(S)}\bar{x}_e = \frac{1}{2}\Big(\sum_{e\in\delta^i(S)}\bar{x}_e + \sum_{v\in S} d_v\Big)$$
$$= \frac{1}{2}\Big(\sum_{e\in\delta^i(S)}\bar{x}_e + \sum_{v\in S: d_v>0} d_v + \sum_{v\in S: d_v<0} d_v\Big) \tag{24}$$
$$= \frac{1}{2}\Big(\sum_{e\in\delta^i(S)}\bar{x}_e + \sum_{\substack{v\in S\\ d_v>0}} d_v - \sum_{\substack{v\in V\setminus S\\ d_v<0}} d_v + \sum_{\substack{v\in V\\ d_v<0}} d_v\Big) \tag{25}$$

In (24) we partition the sum by positive and negative values of d_v. In (25) we add negative d_v values of all vertices in V and re-subtract the additional ones that are not in S. Now we can rewrite the term for α as

$$\alpha = -1 + \min_{\substack{S\subseteq V\\ f\in E^i(S)}} \Big(|S| - \sum_{e\in E^i(S)}\bar{x}_e\Big)$$
$$= -1 + \frac{1}{2}\Big(\min_{\substack{S\subseteq V\\ f\in E^i(S)}} \Big(\sum_{e\in\delta^i(S)}\bar{x}_e + \sum_{\substack{v\in S\\ d_v>0}} d_v - \sum_{\substack{v\in V\setminus S\\ d_v<0}} d_v + \sum_{\substack{v\in V\\ d_v<0}} d_v\Big)\Big)$$
$$= -1 + \frac{1}{2}\Big(\sum_{\substack{v\in V\\ d_v<0}} d_v + \min_{\substack{S\subseteq V\\ f\in E^i(S)}} \Big(\sum_{e\in\delta^i(S)}\bar{x}_e + \sum_{\substack{v\in S\\ d_v>0}} d_v - \sum_{\substack{v\in V\setminus S\\ d_v<0}} d_v\Big)\Big)$$

where we exploit that the last sum in (25) is independent on S. The result follows.