Complexity of the Directed Robust b-matching Problem and its Variants on Different Graph Classes

Jenny Segschneider * Arie M.C.A. Koster †

September 24, 2024

Abstract

The b-matching problem is a well-known generalization of the classical matching problem with various applications in operations research and computer science. Given an undirected graph, each vertex v has a capacity b_v, indicating the maximum number of times it can be matched, while edges can also be used multiple times. The problem is solvable in polynomial time and has many real-world applications.

In some of them, a feasible matching must exactly satisfy the capacities b_v, leading to the so-called perfect b-matching problem. Typically, the capacities b_v are assumed to be fixed and known. However, in practice, these capacities often face uncertainties, such as worker availability or customer demand fluctuations.

This paper analyses a robust variant of both the b-matching and perfect b-matching problems, accounting for such capacity uncertainties, termed the Directed Robust b-Matching Problem. We study the computational complexity of this problem across different classes of graphs, providing insights into its tractability for potential applications.

Keywords

Directed matching, Computational complexity, Graph and network algorithms, Robust optimization

*RWTH Aachen University, Research Area Discrete Optimization
E-Mail: segschneider@math2.rwth-aachen.de
†RWTH Aachen University, Research Area Discrete Optimization
E-Mail: koster@math2.rwth-aachen.de

1 Introduction

Combinatorial optimization plays a crucial role in various fields, from logistics and network design to machine learning. One of the fundamental problems in this domain is the *matching problem*, where the goal is to find a subset of edges in a graph that do not share any common vertices. Among the different variants of matchings, b-matchings have garnered significant interest due to their flexibility and applicability in modeling real-world scenarios. In a b-matching, each vertex v in a graph is allowed to be incident to at most b_v edges, where b_v is a predefined capacity for that vertex. Additional constraints, such as upper bounds on the number of times each edge can be part of a b-matching as well as requiring a *perfect b-matching*, where the capacities are met exactly on each vertex, can be imposed depending on the use case.

While b-matchings have been extensively studied, their robustness against uncertainties in the capacities has received less attention. However, including these uncertainties is critical in many applications where uncertainty and variability are inherent. Robust optimization focuses on finding solutions that remain feasible and near-optimal under perturbations or uncertainties in the input data. In the case of perfect b-matchings, finding a feasible solution for different capacities b is impossible because a perfect b-matching is only perfect for one unique scenario of capacities b. Thus, we need to include some flexibility when defining robust b-matching. To address this, we explore a variant known as the Directed Robust Perfect b-Matching Problem, first introduced in the context of vaccination scheduling [11]. In this problem, only the sum over all outgoing arcs of each vertex, named a pre-matching, is fixed across all scenarios. This leads to a two-stage approach where the first stage sets the pre-matching on each vertex, while the second stage sets a perfect b-matching fitting the pre-matching for every capacity scenario b. Additionally, we introduce three new variants of the problem by relaxing the perfect matching constraint, resulting in the Directed Robust b-Matching Problem, and adding upper arc-bounds to both variants.

Due to the two-stage approach, this problem has multiple practical applications. In [11], the problem was introduced through an application related to healthcare: the scheduling of vaccinations requiring two doses. Due to limited and often unpredictable vaccination supply, the capacities are uncertain. In this application, the pre-matching would set appointments for a first dose for each patient, while the perfect matching in the second stage sets appointments for the second dose dynamically. Similarly, in supply chain management, the Directed Robust Perfect b-Matching Problem can be used to model problems where goods must be shipped to multiple locations with uncertain demand while ensuring that deliveries satisfy customer demand across all potential scenarios. This can be represented by a bipartite graph, where each vertex corresponds to either a manufacturer or a shop location with uncertain demand, and arcs represent possible shipments from manufacturers to shops. The pre-matching then represents the production of goods by manufacturers, and the final b-matching represents shipping these products to customers or shop locations in response to the uncertain demand.

Originally, robust matching problems have been examined for uncertainties in the objective value. Kouvelis and Gang [9] showed NP-hardness of the min-max assignment problem and the min-max regret assignment problem. For the same problems, Aissi et al. [1] showed strong NP-hardness if the number of scenarios is not bounded by a constant. Additionally, they extend this problem to interval data, where instead of discrete scenarios, upper and lower bounds for the objective coefficient are given, and show equivalence to the deterministic assignment problem for the min-max assignment problem and strong NP-hardness for the min-max regret assignment problem. Katriel et al. [7] examine both stochastic and robust variants of the two-stage recoverable matching problem, where edges can be purchased at a base cost in the first stage or at a higher cost in the second stage. They propose a randomized algorithm for the robust recoverable matching problem. Kasperski et al. [6] show NP-hardness of the rent-recoverable robust minimum assignment problem, which generalizes the robust recoverable matching problem. In this variant, a solution is rented in the first stage and can be adapted, not only expanded, in the second stage for additional implementation costs.

Recent literature considering the matching problem under uncertain capacity mostly focused on online variants of the problem. This is due to an application linked to online advertisements, the Ad Allocation Problem. We refer to Mehta et al. [10] for more details on the topic. To the best of our knowledge, the research on the robust matching or b-matching problem is very sparse. We are aware of only three publications considering variants of this problem. Housni et al. [5] consider a robust version of the ride-hailing problem where riders are matched to drivers. They propose a two stage model where the available drivers and a first batch of riders are known while the second batch of drivers are subject to uncertainty and only revealed in the second stage. Schmitz and Büsing [2] consider a version of the robust perfect b-matching problem under consistent selection constraints. They propose a two stage approach. In the first stage, the b-matching is only set on a given subset of all edges, while the b-matching on the remaining edges is set in the second stage when the scenario is known.

As already mentioned, in [11], we introduced the Directed Robust Perfect b-Matching Problem and showed its NP-hardness on instances with large numbers of scenarios or graphs with large bandwidth. However, we did not explore the influence of different graph classes on the complexity of the problem. Building on this foundation, this paper extends our earlier work by investigating the complexity of the four variants of the Directed Robust Perfect b-Matching Problem across different graph classes.

The contributions of this paper are many-fold. We formally define these four problem variants and examine their complexity on different graph classes. To this end, we focus on both directed and oriented paths, pearl graphs, trees, bipartite graphs, SP-graphs and cactus graphs. Our key results include proving NP-hardness of the Directed Robust b-Matching Problem even on oriented paths and directed trees and NP-hardness of the Directed Robust Perfect b-Matching Problem on oriented cactus graphs without upper arc-bounds and on directed SP-graphs when including upper arc-bounds. Furthermore, we present a polynomial-time algorithm for the Directed Robust Perfect b-Matching Problem on graphs without direction-alternating circuits and extend it to certain instances with upper arc-bounds.

This paper is organized as follows. In the next section, we introduce the problem variants and summarize relevant results from [11]. Over the next three sections, we examine the complexity of the problem variants across various graph classes split up by problem variant. First, we examine the not-perfect variants in Section 3 and show NP-hardness on all graph classes except directed paths and directed pearl graphs. In Section 4, we show NP-hardness of the Directed Robust Bounded Perfect b-Matching Problem with upper arc bounds on oriented pearl graphs and directed SP-graphs. Finally, in Section 5, we examine the Directed Robust Perfect b-Matching Problem without arc bounds by first showing NP-hardness for semi-directed cactus graphs. Then, we introduce a polynomial time algorithm on graphs without alternating cycles in Section 5.2. We conclude by showing that this algorithm can be used on directed SP-graphs and even for the Directed Robust Bounded Perfect b-Matching Problem on directed cactus graphs.

2 Terminology and Definitions

2.1 Terminology for Directed and Undirected Graphs

This section introduces the terminology and definitions used to describe subclasses of both undirected and directed graphs in this paper. Let $D = (V, A)$ be a directed graph, and let $G = (V, E)$ be an undirected graph. On directed graphs, for each vertex v, we denote the set of all outgoing arcs by $\delta^+(v) = \{(v, w) \in A\}$ and the set of all incoming arcs by $\delta^-(v) = \{(w, v) \in A\}$. The set of all adjacent arcs is called $\delta(v) = \delta^+(v) \cup \delta^-(v)$ and analogous $\delta(v) = \{e \in E : v \in e\}$ on undirected graphs.

A *walk* in G is defined as a sequence of edges $(e_1, \ldots, e_k)$ that connects a corresponding sequence of vertices $(v_0, \ldots, v_k)$. If all edges in this sequence are distinct, we refer to it as a *trail*. If, additionally, all vertices in the sequence are also distinct, the sequence is called a *path*. A *circuit* is a trail where the first and last vertices are identical. Similarly, a *cycle* is a path where the first and last vertices are the same. In other words, a cycle is a circuit in which all vertices, except for the first and last, are distinct.

Most graph classes discussed in this paper are defined on undirected graphs, as the deterministic b-matching problem is itself formulated on undirected graphs. Given a directed graph $D = (V, A)$, its *underlying undirected graph* is defined as $G = (V, E)$ where $E = \{\{v, w\} : (v, w) \in A \vee (w, v) \in A\}$. For a graph class $\mathcal{G}$, a given directed graph is termed an *oriented* $\mathcal{G}$-graph if its underlying undirected graph is a $\mathcal{G}$-graph. A directed graph is called a *directed* $\mathcal{G}$-graph if, in addition to being an oriented $\mathcal{G}$-graph, the direction of its arcs follows the arc orientation rules specified for $\mathcal{G}$. For instance, an *oriented path* is derived from an undirected path where all arcs are directed arbitrarily, whereas a *directed path* is one where all arcs point in the same direction, with each vertex having at most one incoming and one outgoing arc.

We define the following graph classes on undirected and directed graphs:

- An undirected graph G is a *path* if all of its vertices form a sequence as defined above.
- A *pearl graph* is an extension of a path graph that allows parallel edges between any two vertices. Formally, a pearl graph is a graph in which the vertex set can be arranged in a linear order such that every edge connects two consecutive vertices in this order. This structure allows for parallel edges between consecutive vertices but does not allow edges to skip any intermediate vertices. In a *directed pearl graph*, all arcs are oriented in the same direction, similar to a directed path.
- $G = (V, E)$ is *bipartite* if there exists a partition of its vertex set $V = V_1 \cup V_2$ such that $E \subseteq V_1 \times V_2$. In a directed bipartite graph, all arcs are directed from V_1 to V_2.
- A *series-parallel graph (SP-graph)* can be constructed from single edges by a sequence of two basic composition operations: serial composition and parallel composition. An SP-graph is defined with two distinct terminals, denoted s (source) and t (sink). The graph is either:
  1. A single edge $\{s, t\}$, or
  2. Constructed recursively by applying one of the following two operations to two smaller SP-graphs $G^1 = (V^1, E^1, s^1, t^1)$ and $G^2 = (V^2, E^2, s^2, t^2)$:

     Serial composition: Merge the terminal t^1 of G^1 with the terminal s^2 of G^2, creating a new SP-graph where $s = s^1$ and $t = t^2$.

     Parallel composition: Merge the sources s^1 and s^2 into a single source s, and merge the sinks t^1 and t^2 into a single sink t, constructing a new SP-graph with the same source and sink.

In a *directed SP-graph*, every arc is directed from the source s to the sink t.

- A *cactus graph* is a graph in which every edge belongs to at most one cycle. In a cactus graph, each block (maximal connected subgraph without cut-vertex) is either a cycle or a single edge. In a *directed cactus graph*, every cycle is oriented consistently in one direction.

2.2 The Directed Robust b-Matching Problem and its Variants

Given a graph $G = (V, E)$ with weights $c \in \mathbb{Z}^{|E|}$, the maximum-weight matching problem aims to find a set of edges $M \subseteq E$ with maximum weight $c(M) = \sum_{e \in M} c_e$ such that no two edges $e_1, e_2 \in M$ share a common vertex, i.e., $e_1 \cap e_2 = \emptyset$. For an edge $e = \{v, w\} \in M$, we say that vertex v is matched to vertex w via edge e. A matching M is called *perfect* if every vertex $v \in V$ is incident to exactly one edge $e = \{v, w\} \in M$. In general, the maximum-weight matching problem and the maximum-weight perfect matching problem are equivalent [8, Proposition 11.1], though this equivalence may not hold for specific graph classes. Therefore, we will distinguish between these two variants.

The b-matching problem generalizes the standard matching problem by allowing each vertex v to be incident to up to b_v edges for given capacities $b \in \mathbb{Z}_+^{|V|}$. In a b-matching, edges can appear multiple times. For $b \equiv 1$, the problem reduces to the standard matching problem. Formally, a b-matching is defined as a vector $m \in \mathbb{Z}_+^{|E|}$, where m_e denotes the number of times edge $e \in E$ is used. A b-matching is feasible if it satisfies the following, so-called matching constraints

$$\sum_{e \in \delta(v)} m_e \leq b_v \qquad \forall v \in V \tag{1}$$

where $\delta(v)$ denotes the set of edges incident to vertex v. A b-matching is called perfect if the matching constraints are satisfied with equality. Consequently, we call

$$\sum_{e \in \delta(v)} m_e = b_v \qquad \forall v \in V \tag{2}$$

the perfect matching constraints.

In some cases, each edge $e \in E$ also has an upper bound u_e such that $m_e \leq u_e$ is required. This is typically still referred to as a b-matching problem since the two variations are equivalent [4, Section 7.1.1]. However, this equivalence may not hold when restricted to specific graph classes. Thus, we denote this variant with upper bounds as the Bounded b-Matching Problem (BbM), and the perfect variant as Directed Robust Bounded Perfect b-Matching Problem (DRBPbM).

To account for uncertainties in the vertex capacities, we consider the Directed Robust b-Matching Problem (DRUbM) introduced in [11]. An instance of the DRUbM consists of a directed graph $D = (V, A)$ with arc weights $c \in \mathbb{Z}^{|A|}$ and a set of scenarios $\mathcal{B} \subseteq \mathbb{Z}_+^{|V|}$ for the uncertain capacities of each vertex. The problem consists of two stages: in the first stage, a pre-matching h_v is set for each vertex $v \in V$, representing the total matching over all outgoing arcs. In the second stage, after the realization of a scenario $b \in \mathcal{B}$, a maximum-weight b-matching that satisfies both the pre-matching and the matching constraints induced by the scenario is selected. This problem can be formulated as a three-stage integer linear program:

$$\max_{h\geq 0} \min_{b\in\mathcal{B}} \max_{m\geq 0} \sum_{a\in A} c_a m_a, \tag{3a}$$

$$\text{s.t.} \quad \sum_{a\in\delta^+(v)} m_a = h_v, \qquad \forall v \in V, \tag{3b}$$

$$\sum_{a\in\delta(v)} m_a \leq b_v, \qquad \forall v \in V, \tag{3c}$$

$$h_v \in \mathbb{Z}_+,\ m_a \in \mathbb{Z}_+, \qquad \forall a \in A,\ v \in V. \tag{3d}$$

The objective function (3a) maximizes the weight of the matching in the worst-case scenario. The constraints (3b) ensure that the matching is feasible for the pre-matching. We refer to these constraints as pre-matching constraints. Constraints (3c) represent the matching constraints for the worst-case scenario, as defined in (1). For the Directed Robust Perfect b-Matching Problem (DRUPbM), the second-stage matching must be a perfect b-matching, and constraints (3c) are replaced with the perfect matching constraints:

$$\sum_{a\in\delta(v)} m_a = b_v, \qquad \forall v \in V. \tag{3c'}$$

Analogously to the deterministic case, we can define a variant of the problem with upper bounds $u \in \mathbb{Z}_+^{|A|}$ on each arc. We call this problem the Directed Robust Bounded b-Matching Problem (DRBbM) or the Directed Robust Bounded Perfect b-Matching Problem (DRBPbM) for the perfect variant, respectively. This variant introduces the following upper-bound constraints to the three-stage formulation (3):

$$m_a \leq u_a, \qquad \forall a \in A. \tag{3e}$$

2.3 Adapting Known Results to new Variants

In this section, we begin by summarizing the relevant results from [11] concerning DRUPbM and then extend these findings to the other variants of the problem. Afterwards, we generalize the equivalence of the deterministic matching variants to the robust case.

The DRUPbM can be reformulated as the following Integer Linear Program (ILP) as shown in [11]:

$$\max_{h,m^b,z} \quad z \tag{4a}$$

$$\text{s.t.} \quad z \leq \sum_{a \in A} c_a m_a^b \qquad \forall b \in \mathcal{B} \tag{4b}$$

$$\sum_{a \in \delta^+(v)} m_a^b = h_v \qquad \forall v \in V,\ b \in \mathcal{B} \tag{4c}$$

$$\sum_{a \in \delta(v)} m_a^b = b_v \qquad \forall v \in V,\ b \in \mathcal{B} \tag{4d}$$

$$h_v \in \mathbb{Z}_+,\ m_a^b \in \mathbb{Z}_+ \qquad \forall a \in A,\ v \in V,\ b \in \mathcal{B} \tag{4e}$$

This formulation can be adapted to DRUbM by using matching constraints (3c) instead of those in (4d). Similarly, the upper bound constraints can be added for each matching m^b to get ILP formulations for DRBbM and DRBPbM.

In [11], we show that DRBPbM is strongly NP-hard on arbitrary graphs when $|\mathcal{B}| \in \mathcal{O}(n)$ and weakly NP-hard when $|\mathcal{B}| = 2$. The weak NP-hardness is shown through a reduction from the Partition Problem. The graph used in the reduction is an oriented bipartite graph that can also be extended to an oriented SP-graph. For a detailed proof, we refer to [11, Theorem 10]. Additionally, we show that the problem becomes strongly NP-hard if the directed bandwidth of the graph as defined in [3] is in $\mathcal{O}(n)$, even for a constant number of scenarios. However, we did not investigate the influence of different graph classes on the complexity of the problem. Thus, we investigate various graph classes to identify structures where the problems might be solvable in polynomial time.

All four variants of Directed Robust b-Matching Problem on arbitrary graphs are equivalent. The following reductions for including upper arc bounds are based on the work by Gerards [4, Section 7.1.1].

Theorem 1 The Directed Robust Bounded Perfect b-Matching Problem and the Directed Robust Perfect b-Matching Problem are equivalent.

Proof
For each instance $(D = (V, A), \mathcal{B}, c)$ of DRUPbM, an equivalent instance of DRBPbM can be obtained by setting large upper bounds $u_a \geq b_v$ for each $a \in A$, $v \in a$, and $b \in \mathcal{B}$.

Conversely, any instance of DRBPbM can be transformed into an equivalent instance of DRUPbM by replacing each arc $a = (v, w)$ with weight c_a and upper bound u_a with two new vertices v', w' and three new arcs $a^1 = (v, v')$, $a^2 = (w', v')$ and $a^3 = (w', w)$.

The vertices have capacities $b_{v'} = b_{w'} = u_a$ in each scenario $b \in \mathcal{B}$ and the arcs have weight $c_{a^1} = c_a$ and $c_{a^2} = c_{a^3} = 0$. Since v' has only incoming arcs and w' has only outgoing arcs, each feasible pre-matching h must satisfy $h_{v'} = 0$ and $h_{w'} = b_{w'} = u_a$. For each scenario $b \in \mathcal{B}$, and each perfect b-matching, the perfect matching constraints must hold and it holds $m_{a^1} = b_{v'} - m_{a^2} = b_{w'} - m_{a^2} = m_{a^3}$ and $m_{a^1} \leq b_{v'} = u_a$. Therefore, this matching m is equivalent to the corresponding matching m' for the DRBPbM instance with $m'_a = m_{a^1}$. □

We note that transforming an instance of DRUPbM into an instance of DRBPbM does not change the graph and thus, polynmial-time algorithms for DRBPbM on specific graph classes can also be used for DRUPbM on the same graph class. Similarly, if DRUPbM is NP-hard on a specific graph class, the same holds for DRBPbM. We can say that the arc-bounded variant is at least as hard as the arc-unbounded variant on each graph class.

For the perfect and not-perfect variants, the following result is based on Korte and Vygen [8, Proposition 11.1.].

Theorem 2 The Directed Robust b-Matching Problem and the Directed Robust Perfect b-Matching Problem are equivalent.

Proof

An instance of DRUPbM can be transformed into an instance of DRUbM by adding a large constant M to the weight of each arc. Then, each optimal solution of DRUbM will use as many arcs as possible (with $\sum_{a \in A} m_a = \frac{1}{2} \sum_{v \in V} b_v$) and will also be a solution to DRUPbM if such a solution exists .

An instance of the DRUbM $(D = (V, A), \mathcal{B}, c, u)$ can be reduced to the DRUPbM by adding a gadget V', A' with three new vertices $V' = \{w^1, w^2, w^3, d\}$ and arcs $A' = \{(w^1, v) \colon v \in V\} \cup \{(w^1, w^2), (w^1, w^3), (w^2, w^3), (w^1, d)\}$. We call the resulting graph $D' = (V \cup V', A \cup A')$. The gadget is depicted in Figure 1. The vertices have capacities $b_{w^i} = \max_{\tilde{b} \in \mathcal{B}} \sum_{v \in V} \tilde{b}_v =: M$ for each $i \in \{1, 2, 3\}$ and $b_d = M - \sum_{v \in V} b_v$ in each scenario $b \in \mathcal{B}$ and the weights of all arcs $a \in A'$ and $c_a = 0$. The 3-cycle w^1, w^2, w^3 is used to absorb unused capacities from the not-perfect matchings in D and the vertex d balances the different total capacities in each scenario. Clearly, each solution of the DRUPbM instance corresponds to a solution of the DRUbM instance by restricting the solution to D. Given a pre-matching h with b-matchings m^b in each scenario $b \in \mathcal{B}$ in graph D that are solutions to the DRUbM instance, the corresponding DRUPbM solution in D' is obtained by setting the pre-matching on the added gadget to $h'_{w^1} = M$, $h'_{w^2} = M - \sum_{v \in V} h_v$ and $h'_{w^3} = 0$. The perfect matchings on the gadget are set as indicated in blue by Figure 1. On the arcs connecting the gadget to D, the matchings are set to $m'^b_{(w^1,v)} = b_v - \sum_{a \in \delta(v)} m^b_a$. Finally, all matchings and pre-matchings on D

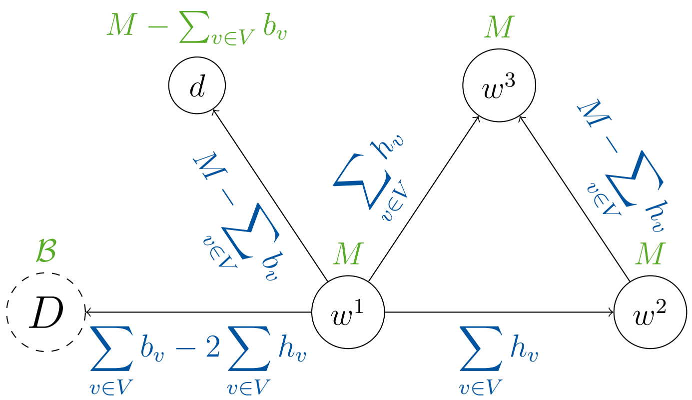

Figure 1: Reduction from the DRBbM to the DRBPbM

remain the same: $h'_v = h_v$ and ${m'}^b_a = m_a$ for all $v \in V$ and $a \in A$. On each vertex $v \in V$, the matchings $(m')^b$ are perfect due to the definition on the arcs (w^1, v). In each scenario, the sum over all these remaining capacities is given by

$$\sum_{v\in V} b_v - \sum_{a\in\delta(v)} m^b_a = \sum_{v\in V} b_v - 2\sum_{a\in A} m^b = \sum_{v\in V} b_v - 2\sum_{a\in A} h$$

as depicted in Figure 1. For all vertices in the gadget V', the perfect matching constraints and pre-matching constraints are also satisfied as shown in the illustration. Thus, $h', (m')^b$ is a feasible solution of the DRUPbM instance with the same weight as the corresponding solution of the DRUbM instance. □

Similar to the result for the arc-bounded variants, we note that the graph structure remains the same when transforming an instance of DRUPbM to an instance of DRUbM. Thus, on each graph class, DRUbM is at least as hard as DRUPbM.

We combine the techniques from the previous Theorems and obtain the following result.

Corollary 1 The Bounded b-Matching Problem, Bounded Perfect b-Matching Problem, Directed Robust Bounded b-Matching Problem and Directed Robust Bounded Perfect b-Matching Problem are equivalent.

As a consequence of this result, all complexity results for DRUPbM on arbitrary graphs presented in [11] can also be extended to the other three variants. Finally, we note that the decision problems of all four variants are in NP. Thus, in the following, NP-hardness directly implies NP-completeness.

Lemma 1 The decision variants of Bounded b-Matching Problem, Bounded Perfect b-Matching Problem, Directed Robust Bounded b-Matching Problem and Directed Robust Bounded Perfect b-Matching Problem are in NP.

Proof
For a given pre-matching h and lower bound on the objective value Z, computing maximum-weight (bounded, perfect) b-matchings m^b and testing whether $\sum_{a\in A} c_a m_a^b \geq Z$ for each scenario $b \in \mathcal{B}$ can be done in polynomial time. Thus, the decision variant of DRUbM on directed trees is in NP. □

3 Non-perfect Variants

Directed paths represent one of the simplest classes of digraphs. On such paths, all four variants of DRUbM become easy to solve. Each vertex possesses at most one outgoing and one incoming arc, leading to each pre-matching being feasible for exactly one b-matching, and vice versa. Consequently, the problem reduces to finding a b-matching that is feasible across all scenarios $b \in \mathcal{B}$.

The same holds for directed pearl graphs. The added multi-arcs have no influence on DRUbM and DRUPbM, because an optimal solution will only use the parallel arc with the highest weight. For the variants with upper bounds on the arcs, this is not necessarily the case. However, it still holds that for each feasible pre-matching, all feasible matchings for each scenario can only differ on parallel arcs. Assuming w.l.o.g. parallel arcs differ in their weight, each pre-matching already defines a unique, optimal b-matchings independent of the scenario Consequently, the problem can be solved in polynomial time by finding the optimal matching feasible in all scenarios. For DRBbM, this can be done by solving the deterministic BbM on $b^{\min}$ with $b_v^{\min} = \min_{b\in\mathcal{B}} b_v$.

These results no longer hold for directed trees or oriented paths, where each vertex can have outgoing arcs to multiple other vertices. Then, the (optimal) b-matchings in the second stage are no longer unique for each pre-matching and depend on the worst-case scenario. The following reduction illustrates the main technique used for NP-hardness proofs in this paper.

Theorem 3 The Directed Robust b-Matching Problem is weakly NP-complete on directed trees.

Proof
We show NP-hardness by a reduction from the well-known Partition problem. Given a set $S = \{s_1, \ldots, s_n\} \in \mathbb{Z}_+^n$, the Partition problem asks if there is a subset $T \subseteq \{1, \ldots, n\}$ such that $\sum_{i\in T} s_i = \sum_{i\notin T} s_i$. Hence, a set T is a solution to the instance of Partition if

$$\min\{\sum_{i\in T} s_i, \sum_{i\in S\setminus T} s_i\} = \frac{1}{2}\sum_{i=1}^{n} s_i.$$

Similar to DRUbM, Partition can be reformulated as a two stage problem:

$$\max_{T\subset\{1,\dots,n\}} \min_{D\in\{T,S\setminus T\}} \sum_{i\in D} s_i \geq \frac{1}{2}\sum_{i=1}^{n} s_i$$

Based on this idea, we construct an instance of DRUbM such that the pre-matching defines a set $T \subseteq \{1,\dots,n\}$ and one of two scenarios $b^T, b^{S\setminus T}$ being the worst-case scenarios represents either T or $S \setminus T$ being minimal.

Let S be an instance of Partition, let $P = \sum_{i=1}^{n} s_i$ and let $M > P$ be some large integer. We define a directed graph $D = (V, A)$ with $V = \{r\} \cup \bigcup_{i=1}^{n} V_i$, $A = A_r \cup \bigcup_{i=1}^{n} A_i$ and gadgets (V_i, A_i) for each element $s_i \in S$ with

$$\begin{aligned}
V_i &= \{d_i, v_i^{\text{in}}, v_i^{\text{out}}, v_i^{T,\text{in}}, v_i^{T,\text{out}}, v_i^{S\setminus T,\text{in}}, v_i^{S\setminus T,\text{out}}\}\\
A_i &= \{(d_i, v_i^{\text{in}}), (d_i, v_i^{\text{out}}), (v_i^{\text{in}}, v_i^{T,\text{in}}), (v_i^{\text{in}}, v_i^{S\setminus T,\text{in}}), (v_i^{\text{out}}, v_i^{T,\text{out}}), (v_i^{\text{out}}, v_i^{S\setminus T,\text{out}})\}\\
A_r &= \{(r, d_i) \colon i = 1,\dots,n\}
\end{aligned}$$

One gadget of the instance is depicted in Figure 2. D is a directed tree with dummy

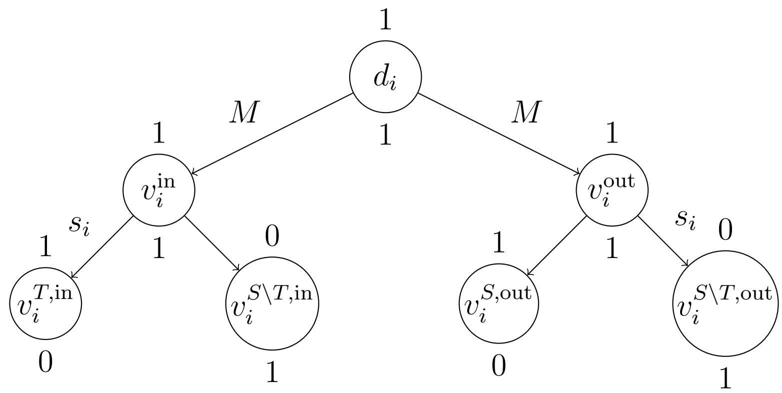

Figure 2: Gadget used to show NP-hardness of DRUbM on directed trees

root r that has no capacity and only ensures that D is a connected tree. Additionally, we define two scenarios b^T and $b^{S\setminus T}$ and objective value c as follows:

$$b_v^T = \begin{cases} 1 & \text{if } v \in \{d_i, v_i^{\text{in}}, v_i^{\text{out}} \colon i = 1, \dots, n\} \\ 1 & \text{if } v \in \{v_i^{T,\text{in}}, v_i^{T,\text{out}} \colon i = 1, \dots, n\} \\ 0 & \text{else} \end{cases},$$

$$b_v^{S\setminus T} = \begin{cases} 1 & \text{if } v \in \{d_i, v_i^{\text{in}}, v_i^{\text{out}} \colon i = 1, \dots, n\} \\ 1 & \text{if } v \in \{v_i^{S\setminus T,\text{in}}, v_i^{S\setminus T,\text{out}} \colon i = 1, \dots, n\} \\ 0 & \text{else} \end{cases},$$

$$c_a = \begin{cases} s_i & \text{if } a \in \{(v_i^{\text{in}}, v_i^{T,\text{in}}), (v_i^{\text{out}}, v_i^{S\setminus T,\text{out}})\} \\ M & \text{if } a \in \{(d_i, v_i^{\text{in}}), (d_i, v_i^{\text{out}}) \colon i = 1, \dots, n\} \\ 0 & \text{else} \end{cases}.$$

In Figure 2, the capacities in scenario b^T are written above each vertex and the capacities in $b^{S\setminus T}$ are written below. The non-zero weight of each arc is written on the arcs.

Each optimal pre-matching will set $h_{d_i} = 1$ to include one of its outgoing arcs with high weight M. Thus, d_i must be matched with either v_i^{in} or v_i^{out} and since both vertices have a capacity of one in both scenarios, only one of them can be matched through an outgoing arc and it has to hold $h_{v_i^{\text{in}}} + h_{v_i^{\text{out}}} \leq 1$. Because there are no negative weights and no other vertices with outgoing arcs, any optimal solution always sets either $h_{v_i^{\text{in}}} = 1$ or $h_{v_i^{\text{out}}} = 1$. All remaining vertices are leafs without outgoing arcs, thus every feasible pre-matching on these vertices is always zero. Thus, restricted to a gadget (V_i, A_i), there are two possible, optimal pre-matchings $h^{i,\text{in}}$ and $h^{i,\text{out}}$ defined by

$$h_v^{i,\text{in}} = \begin{cases} 1 & \text{if } v \in \{d_i, v_i^{\text{in}}\} \\ 0 & \text{else} \end{cases} \qquad \text{and} \qquad h_v^{i,\text{out}} = \begin{cases} 1 & \text{if } v \in \{d_i, v_i^{\text{out}}\} \\ 0 & \text{else.} \end{cases}$$

Using $h^{i,\text{in}}$ on gadget (V_i, A_i) then represents setting $i \in T$. On this basis, we can define a solution of the Partition instance from an optimal pre-matching h^* as $T^* = \{i \colon h_{v_i^{\text{in}}} = 1\}$, which is the set of all gadgets where $h^{i,\text{in}}$ is used. Then, T^* is a solution to the instance of Partition if and only if the objective value of h^* is $\geq n \cdot M + \frac{P}{2}$.

We can see in Figure 2 that for each pair of scenario and choice of pre-matching as described above, the b-matching in the second stage is unique. For example, restricted to gadget (V_i, A_i), the only b^T-matching fitting $h^{i,\text{in}}$ uses the arcs (d_i, v_i^{out}) and $(v_i^{\text{in}}, v_i^{T,\text{in}})$ once and no other arcs. This matching has objective value $M + s_i$. Similarly, for the combination $b^{S\setminus T}$, $h^{i,\text{out}}$, the unique $b^{S\setminus T}$-matching includes only the arcs (d_i, v_i^{in}) and $(v_i^{\text{in}}, v_i^{S\setminus T,\text{out}})$ once with objective value $M + s_i$. For the combinations $b^T, h^{i,\text{out}}$ and $b^{S\setminus T}, h^{i,\text{in}}$, the objective value is M with the same argumentation.

Let m^T and $m^{S\setminus T}$ be the max-weight b-matchings for h^* and the corresponding scenario $b \in \mathcal{B}$. It holds

$$\sum_{a\in A} c_a m_a^T = \sum_{i:\, h_{v_i^{\text{in}}}=1} M + s_i + \sum_{i:\, h_{v_i^{\text{out}}}=1} M = n \cdot M + \sum_{i\in T^*} s_i$$

$$\sum_{a\in A} c_a m_a^{S\setminus T} = \sum_{i:\, h_{v_i^{\text{in}}}=1} M + \sum_{i:\, h_{v_i^{\text{out}}}=1} M + s_i = n \cdot M + \sum_{i\in S\setminus T^*} s_i$$

Because we optimize the worst case scenario, it holds

$$\sum_{i\in T^*} s_i = \frac{P}{2} \quad \Leftrightarrow \quad \min\left\{\sum_{i\in T^*} s_i, \sum_{i\in S\setminus T^*} s_i\right\} \geq \frac{P}{2} \quad \Leftrightarrow \quad \min_{D\in\{T, S\setminus T\}} \sum_{a\in A} c_a m_a^D \geq n\cdot M + \frac{P}{2}$$

and T^* is a solution to the instance of Partition if and only if the objective value of h^* is $\geq n \cdot M + \frac{P}{2}$. With the above considerations, we also see that the objective value of the DRUbM instance cannot be larger than $n \cdot M + \frac{P}{2}$. Thus, there is a solution to the instance of Partition if and only if there is a solution to the instance of DRUbM with objective value $\geq n \cdot M + \frac{P}{2}$. This concludes the proof of NP-hardness. NP-completeness follows with Lemma 1. □

Using the reduction from Theorem 1, this also holds for DRBbM. Using a similar approach, NP-hardness on oriented paths can be shown.

Theorem 4 The Directed Robust b-Matching Problem is weakly NP-complete on oriented paths.

A proof of this Theorem is given in Appendix A.1.

In instances of the b-matching problem, we can augment the graph by adding vertices with capacity zero without affecting the solution as seen with the dummy vertices r in both reductions. The graph defined in the second reduction is an oriented tree, an oriented bipartite graph, and a directed (and oriented) cactus graph. The graph used in the proof of Theorem 3 is a directed tree and can be expanded into a directed SP-graph. Similar, simple reductions confirm that these problems remain NP-hard on directed bipartite graphs. Thus, it appears that both DRUbM and DRBbM are NP-hard across all graph classes examined in this paper that permit vertices to be matched with more than one vertex through outgoing arcs.

4 The Directed Robust Bounded Perfect b-Matching Problem

Next, we analyze the two perfect matching variants: DRUPbM and DRBPbM. A crucial distinction between these variants lies in their treatment of parallel arcs. In DRUPbM, parallel arcs can be disregarded since we always select the arc with greater weight. Conversely, in DRBPbM, such an arc might have an upper bound and we may have to use the parallel arc with lesser weight. As a consequence, DRUPbM can be solved in polynomial time on directed and oriented pearl graphs with the same reasoning as applied to paths. However, DRBPbM is NP-hard on oriented pearl graphs and directed SP-graphs, which both allow parallel arcs. We first show NP-hardness on oriented pearl graphs based on a reduction from [2, Corollary 6.1].

Theorem 5 The Directed Robust Bounded Perfect b-Matching Problem is weakly NP-complete on oriented pearl graphs.

Proof
We again show NP-hardness by reducing from Partition. Let $S = \{s_1, \ldots, s_n\}$ be an instance of the Partition Problem, $P = \sum_{i=1}^{n} s_i$ and $M > 2 \cdot P$ some large integer. We define a directed graph $D = (V, A)$ with gadgets $D = (V_i, A_1)$ for each $s_i \in S$ and an auxiliary gadget $D_0 = (V_0, A_0)$ as follows:

$$
\begin{aligned}
V_i &= \{v_i^1, v_i^2, v_i^3, r_i\} && \forall i = 0, \ldots, n\\
A_i &= \{(v_i^2, v_i^1), (v_i^2, v_i^3), (v_i^2, v_i^3), (r_i, v_i^3), (r_i, v_{i+1}^1)\} && \forall i = 1, \ldots, n\\
A_0 &= \{(v_0^2, v_0^1), (v_0^2, v_0^3), (r_0, v_0^3)\}\\
V &= \bigcup_{i=0}^{n} V_i \qquad A = \bigcup_{i=0}^{n} A_i.
\end{aligned}
$$

D is an oriented pearl graph. We define an instance of DRBPbM as shown in Figure 3. The capacities are written in green, with scenario b^T written above and scenario $b^{S\setminus T}$

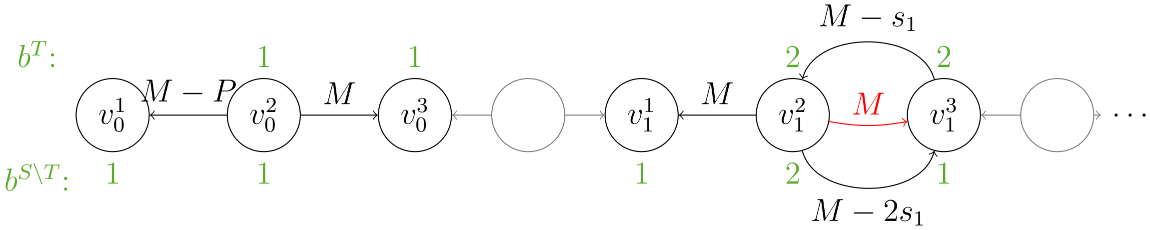

Figure 3: Gadget used to show NP-hardness of DRBPbM on oriented pearl graphs

written below the vertices. The costs of the arcs are written next to the arcs. The only arc with an upper bound is marked in red and has an upper bound of 1. The dummy vertices r_i are shown in gray since they are only necessary to connect the gadgets.

Using these definitions, there is a solution h^* with objective value $\geq (2n+1) \cdot M - \frac{3}{2}P$ if and only if $T^* = \{i\colon h^*_{v_i^3} = 1\}$ is a solution to the instance of Partition.

On the auxiliary gadget D_0 there is only one feasible solution with objective value M in scenario b^T and objective value $M - P$ in scenario $b^{S\setminus T}$.

In each gadget D_i, there are only two feasible pre-matchings: setting the pre-matching of v_i^2 to $h^1_{v_i^2} = 2$ and the others to zero or setting the pre-matching of both v_i^2 and v_i^3 to $h^2_{v_i^2} = h^2_{v_i^3} = 1$. Similar to the previous proof, there is a unique pre-matching for each combination of scenario and pre-matching. For the first pre-matching h^1 and scenario b^T, the upper bound on the red arc forces us to also use the parallel arc with worse weight $M - 2s_i$, resulting in an objective value of $2M - 2s_i$. In scenario $b^{S\setminus T}$, the objective value of the corresponding matching is $M + M = 2M$. For the second pre-matching h^2, the objective value of the matchings is the same in both scenarios: $M + (M - s_i) = 2M - s_i$. Each feasible pre-matching h has to be a combination of the pre-matchings h^1 and h^2 on each gadget D^i. On gadget D^0, the pre-matching is unique. Let h^* be part of the optimal solution and T^* as defined above. Additionally, let m^T be the optimal, perfect b^T-matching and $m^{S\setminus T}$ be the optimal, perfect $b^{S\setminus T}$-matching fitting h^*. Then, the objective values of the (unique) perfect b-matchings in both scenarios are given by

$$\begin{aligned}
\sum_{a\in A} c_a m_a^T &= M + \sum_{i\in T^*} 2M - 2s_i + \sum_{i\in S\setminus T^*} 2M - s_i = (2n+1)M - \sum_{i=1}^{n} s_i - \sum_{i\in T^*} s_i \\
&= (2n+1)M - P - \sum_{i\in T^*} s_i \\
\sum_{a\in A} c_a m_a^{S\setminus T} &= M - P + \sum_{i\in T^*} 2Ms_i + \sum_{i\in S\setminus T^*} 2M - s_i = (2n+1)M - P - \sum_{i\in S\setminus T^*} s_i
\end{aligned}$$

With this, it holds

$$\begin{aligned} & \min\{\sum_{a\in A} c_a m_a^T, \sum_{a\in A} c_a m_a^{S\setminus T}\} \geq (2n+1)\cdot M - \frac{3}{2}P \\ \Leftrightarrow\quad & \min\{(2n+1)M - P - \sum_{i\in T^*} s_i, (2n+1)M - P - \sum_{i\in S\setminus T^*} s_i\} \geq (2n+1)\cdot M - \frac{3}{2}P \\ \Leftrightarrow\quad & \min\{-\sum_{i\in T^*} s_i, -\sum_{i\in S\setminus T^*} s_i\} \geq -\frac{1}{2}P \\ \Leftrightarrow\quad & \sum_{i\in T^*} s_i \leq \frac{1}{2}P \wedge \sum_{i\in S\setminus T^*} s_i \leq \frac{1}{2}P \end{aligned}$$

Thus, T^* is a solution to the instance of Partition. The reverse follows directly using the same definitions. □

Next, we adjust this reduction to a directed SP-graph by "folding" the backward arc (v_i^3, v_i^2) to the right and adding two three-cycles, one to the left and one to the right. Additionally, we add dummy arcs that cannot be part of a feasible solution but ensure that the graph is a directed SP-graph.

Theorem 6 The Directed Robust Bounded Perfect b-Matching Problem is weakly NP-complete on directed SP-graphs.

Proof
We again show NP-hardness by reducing from Partition. Let $S = \{s_1, \ldots, s_n\}$ be an instance of the Partition Problem, $P = \sum_{i=1}^n s_i$ and $M > 2\cdot P$ some large integer. We define a directed graph $D = (V, A)$ with gadgets $D = (V_i, A_1)$ for each $s_i \in S$ and an auxiliary gadget $D_0 = (V_0, A_0)$ as follows:

$$\begin{aligned} V_i &= \{v_i^1, v_i^2, v_i^3, d_i^1, d_i^2, d_i^3, d_i^{-2}, d_i^{-3} r_i\} \\ A_i &= \{(d_i^{-3}, v_i^2), (d_i^{-3}, d_i^{-2}), (d_i^{-2}, v_i^2), (v_i^2, v_i^1), (v_i^2, v_i^3), (v_i^2, v_i^3), (v_i^3, d_i^1), (v_i^1, d_i^1), \\ &\quad (d_i^1, d_i^2), (d_i^1, d_i^3), (d_i^2, d_i^3), (d_i^3, r_i), (r_i, v_{i+1}^2)\} \\ A_0 &= \{(v_0^2, v_0^1), (v_0^2, v_0^3), (v_0^3, d_0), (v_0^1, r_0)\} \\ V &= \bigcup_{i=0}^n V_i \qquad\qquad A = \bigcup_{i=0}^n A_i. \end{aligned}$$

On this graph, we define an instance of DRBPbM with $\mathcal{B} = \{b^T, b^{S\setminus T}\}$ as shown in Figure 4.The capacities in b^T are written above the vertices and the capacities in $b^{S\setminus T}$ below the vertices. The only arc with a non-redundant upper bound is marked in red and has upper bound two. The illustration shows that D is a series parallel graph.

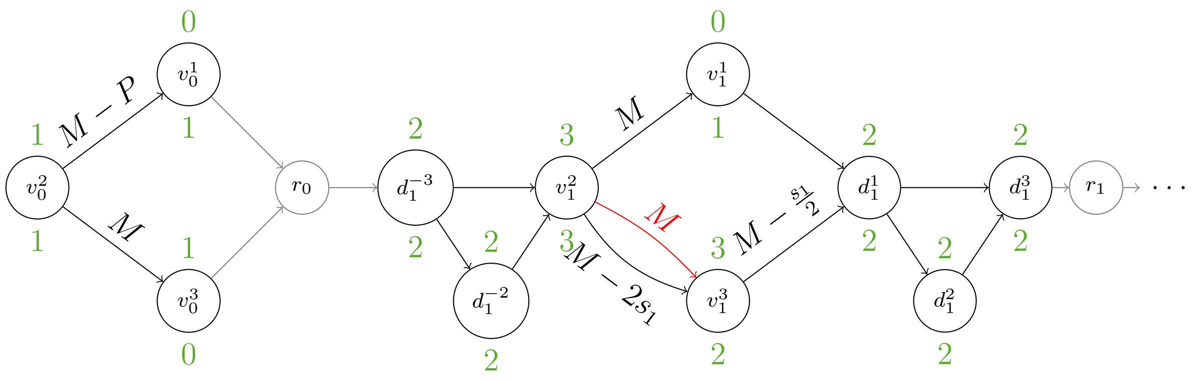

Figure 4: Gadget used to show NP-hardness of DRBPbM on oriented pearl graphs

The auxiliary gadget acts the same as in the previous proof. For each gadget D_i, there are still only two feasible pre-matchings: one pre-matching with $h_{v_i^2} = 3$ and one with $h_{v_i^2} = 1$. We note that there can be no feasible pre-matching with $h_{v_i^2} = 0$, otherwise the capacity in v_i^1 cannot be matched: it has only one incoming arc and cannot be matched through an outgoing arc because then the pre-matching would have to be one and the capacity in scenario b^T is zero. Similarly, if $h_{v_i^2} = 2$, the vertex v_i^2 would have to be matched through an incoming arc exactly once. This is impossible while also satisfying the perfect matching constraints in d_i^{-3} and d_i^{-2}.

Let h^1 with $h^1_{v_i^2} = 3$ be a feasible pre-matching. To satisfy the perfect matching constraints on d_i^{-3} and d_i^{-2}, it has to hold $h^1_{d_i^{-3}} = 2$ and $h_{d_i^{-2}} = 0$. When looking to the right of v_i^2, we see that the vertices v_i^3 and v_i^1 are matched with v_i^2 for their whole capacity in both scenarios. Thus, the only feasible matching for the three-cycle on the right is using each arc once. This results in a pre-matching with $h^1_{d_i^1} = 2$ and $h^1_{d_i^2} = 2$. Furthermore, this shows that h^1 has exactly one feasible, perfect b-matching in each scenario. In b^T, the objective value of this matching is $2 \cdot M + (M - s_i) = 3M - 2s_i$; in $b^{S\setminus T}$, it is $2 \cdot M + M = 3M$.

With the same argumentation, we can show that there is exactly one feasible pre-matching h^2 with $h^2_{v_i^2} = 1$. Again, there is exactly one feasible, perfect b-matching in each scenario. The objective value of this matching is $M + 2 \cdot (M - \frac{s_i}{2}) = 3M - s_i$ in both scenarios. Except for an additional weight of M in each solution, this is the same situation as in the previous reduction. Thus, it holds that an optimal pre-matching h^* has objective value $\geq (3n+1)M - \frac{3}{2}P$ if and only if $T^* = \{i \colon h^*_{v_i^2} = 3\}$ is a solution to the instance of Partition. Let m^T be the optimal, perfect b^T-matching and $m^{S\setminus T}$ be the optimal, perfect $b^{S\setminus T}$-matching fitting h^*. Then, it again holds

$$c^T m^T = M + \sum_{i \in T^*} 3M - 2s_i + \sum_{i \in S\setminus T^*} 3M - s_i = (3n+1)M - P - \sum_{i \in T^*} s_i$$
$$c^T m^{S\setminus T} = M - P + \sum_{i \in T^*} 3M + \sum_{i \in S\setminus T^*} 3M - s_i = (3n+1)M - P - \sum_{i \in S\setminus T^*} s_i$$

Thus, the worst-case objective value is $\geq (3n+1)M - \frac{3}{2}P$ if and only if

$$\sum_{i \in T^*} s_i = \frac{P}{2} = \sum_{i \in S\setminus T^*} s_i$$

and T^* is a solution to the instance of Partition. □

We note that both of the presented reduction graphs are no longer part of their respective graph class after using the reduction from Theorem 1 to reduce the instance to an instance of DRUPbM without upper bounds. Thus, this does not show NP-hardness of DRUPbM on oriented pearl graphs or directed SP-graphs.

5 The Directed Robust Perfect b-Matching Problem

5.1 NP-hardness

In [11], it was shown that DRUPbM is weakly NP-hard on oriented SP-graphs and oriented bipartite graphs. Using the three-cycle gadgets introduced in the previous proof of Theorem 6 and the general approach used in Theorem 3, we expand on these results and show weak NP-hardness on oriented cactus graphs for both DRUPbM and DRBPbM.

Theorem 7 The Directed Robust Perfect b-Matching Problem is weakly NP-complete on oriented cactus graphs.

A proof of this Theorem is given in Appendix A.2.

To summarize, we prove the NP-hardness of DRUPbM and DRBPbM on oriented bipartite graphs, oriented SP-graphs and oriented cactus graphs. For DRBPbM, we also show NP-hardness on oriented pearl graphs and directed SP-graphs. In the next section, we show that the problem can be solved in polynomial time on the directed version of these graph classes as well as oriented paths and oriented trees. The graph classes, where a polynomial time algorithm exists (excluding directed bipartite graphs), share a common property: they do not allow alternating circuits. We use this property and give a polynomial time algorithm on all graphs without alternating circuits.

5.2 Directed Graphs without Alternating Circuits

In this section, we characterize a class of graphs on which the Directed Robust Perfect b-Matching Problem can be solved in polynomial time.

Definition 1 Let $D = (V, A)$ a directed graph, and let $C = (a_1, \ldots, a_n)$ be a circuit in D. We call C an **alternating circuit** if the directions of the arcs in C alternate. More formally, C is an alternating circuit if for any two consecutive arcs a_i, a_{i+1} incident to a common vertex $v \in V$, both arcs are either directed towards v or away from it.

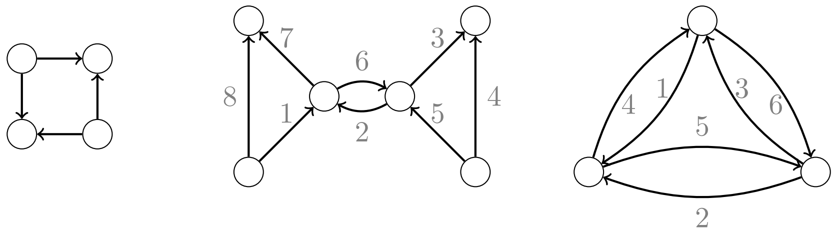

Figure 5: Examples for alternating circuits of length 4, 8 and 6 respectively.

Figure 5 illustrates three examples of alternating circuits. The first graph is an alternating cycle of length 4. In the second and third graph, arcs are numbered to highlight the alternating circuits. The second graph also appears as part of the reduction used in Theorem 7.

We show that the Directed Robust Perfect b-Matching Problem can be solved in polynomial time on graphs without alternating circuit. First, we establish an upper bound on the number of arcs that a graph without alternating circuit can have. This is based on the similar fact that undirected graphs with $|V| \leq |E|$ must have a cycle.

Lemma 2 Let $D = (V, A)$ be a directed graph with $|A| \geq 2|V|$. Then, D has an alternating circuit.

Proof
For $D = (V, A)$, we define an undirected graph $G = (\tilde{V}, E)$ with

$$\tilde{V} = \{v^{\text{in}}, v^{\text{out}} \colon v \in V\} \qquad \text{and} \qquad E = \{\{v^{\text{out}}, w^{\text{in}}\} \colon (v, w) \in A\}.$$

For each vertex $v \in V$, there are two vertices v^{in} and v^{out} in $\tilde{V}$. For each arc $(v, w) \in A$, there is one edge between v^{out} and v^{in} in E. Then, G is a bipartite graph with partition $V^{\text{in}} = \{v^{\text{in}} \colon v \in V\}$ and $V^{\text{out}} = \{v^{\text{out}} \colon v \in V\}$. This way, each path in G alternates between vertices from V^{in} and V^{out} and thus, in the corresponding path in D, the orientation of the arcs alternates. Then, each path in G corresponds to an alternating trail in D and vice versa.

Because cycles (and circuits) are paths (and trails) where the first and last vertex are identical, it also holds that each circuit in G corresponds to an alternating circuit in D and vice versa. Thus, D contains an alternating circuit if and only if G contains a cycle. It holds $|A| = |E|$ and $|\tilde{V}| = 2|V|$. If $|A| \geq 2|V|$, then it is $|E| \geq |\tilde{V}|$ and there is a cycle in G. Thus, there is also an alternating circuit in D if $|A| \geq 2|V|$. □

Next, we examine the influence of alternating circuits on the perfect b-matchings chosen in the second stage for a fixed pre-matching and scenario. It turns out that these perfect matchings are unique. Furthermore, for each pair of scenario and pre-matching, there is at most one not necessarily integer or non-negative matching satisfying both the perfect matching constraints (3c') and the pre-matching constraint (3b).

Lemma 3 Let $D = (V, A), \mathcal{B}, c$ be an instance of the Directed Robust Perfect b-Matching Problem. Then, D has no alternating circuits if and only if for each pair of scenario $b \in \mathcal{B}$ and (not necessarily integer) pre-matching $h \in \mathbb{R}^{|V|}$, there is at most one $\tilde{m} \in \mathbb{R}^{|A|}$ satisfying both the pre-matching constraint (3b) and the perfect matching constraint (3c').

Proof
Let $C = (a_1, a_2, \dots, a_k)$ be an alternating circuit in D and $h, m^1, \dots, m^{|\mathcal{B}|}$ a feasible solution. Then, we define a matching m^C restricted to the circuit as

$$m^C_a = \begin{cases} 0 & a \notin C \\ 1 & \text{if } a = a_i \in C \text{ and } i \text{ is even} \\ -1 & \text{if } a = a_i \in C \text{ and } i \text{ is odd} \end{cases}$$

Then, it holds

$$\sum_{a \in \delta(v)} m^C_a = 0 \qquad \text{and} \qquad \sum_{a \in \delta^+(v)} m^C_a = 0$$

for all vertices $v \in V$. Thus, for each perfect b^i-matching m^i fitting h, $m^i + m^C$ is also a perfect b^i-matching fitting h and m^i is not unique.

Let $D = (V, A)$ be a digraph and $V = \{v_1, \dots, v_{|V|}\}$, $A = \{a_1, \dots, a_{|A|}\}$ enumerations of the arcs and vertices. A matching $m \in \mathbb{R}^{|A|}$ is a perfect b-matching fitting h if and only if it holds

$$\sum_{a \in \delta^-(v)} m_a = b_v - h_v \qquad \forall v \in V$$

$$\sum_{a \in \delta^+(v)} m_a = h_v \qquad \forall v \in V$$

This is equivalent to $Cm = \binom{b-h}{h}$ with $C \in \{0,1\}^{2|V| \times |A|}$ and

$$C_{i,j} = \begin{cases} 1 & \text{if } i \leq |V| \text{ and } a_j \in \delta^-(v_i), \\ 0 & \text{if } i \leq |V| \text{ and } a_j \notin \delta^-(v_i), \\ 1 & \text{if } i > |V| \text{ and } a_j \in \delta^+(v_i), \\ 0 & \text{if } i > |V| \text{ and } a_j \notin \delta^+(v_i). \end{cases}$$

It is well-known that there is at most one solution to the equality $Cm = \binom{b-h}{h}$ if the matrix C has full column rank. We show that C has linear dependent columns if and only if there is an alternating circuit in D. If $2|V| < |A|$, C has more columns than rows and cannot have full column rank. With Lemma 2, there always exists an alternating circuit in D in this case. Thus, the assumption holds in this case and we now examine the case $2|V| \geq |A|$. We denote the i-th column of C by $C_{-,i}$. This column corresponds to the arc $a_i = (v, w) \in A$. There are exactly two 1 entries in the column, one in the upper half corresponding to the start vertex v of the arc and one in the lower half corresponding to the end vertex w.

The columns of C are linear dependent if and only if a column of C, w.l.o.g. the first column $C_{-,1}$ is a linear combination of the other columns. Let $\lambda_2, \ldots, \lambda_{|A|}$ such that

$$C_{-,1} = \sum_{i=2}^{|A|} \lambda_i C_{-,i}$$

Let $a_1 = (v_j, v_k)$. Then, there must be a set $M_j \subseteq \delta^+(v_j)$ of outgoing arcs from v_j with $1 = \sum_{a_i \in M} \lambda_i$. With the same reasoning, there has to be a set $N_k \subseteq \delta^-(v_k)$ of incoming arcs from v_k with $1 = \sum_{a_i \in N} \lambda_i$. Now, for each arc $a_{j'} = (v_j, v_{j'}) \in M_j$, there has to be a set $N_{j'} \subseteq \delta^-(v_{j'})$ with $-\lambda_{j'} = \sum_{a_{j''} \in N_{j'}} \lambda_{j''}$. By repeating this argument, we form alternating paths in the digraph D. Since D is a finite graph and the argumentation can be repeated infinitely many times, these paths have to form alternating circuits. Thus, if the rows of C are linear dependent, there exists an alternating circuit in D. Reversely, it is easy to see that the columns corresponding to an alternating circuit in D are linearly dependent. Thus, the assumption holds and there is at most one solution to $Cm = \binom{b-h}{h}$ if and only if there is no alternating circuit in D. □

Building on this, we examine the influence of scenario changes on the unique, perfect b-matching. It turns out that the differences between these b-matchings remain constant, independent of the shared pre-matching. Therefore, the perfect matchings for each scenario can be computed by solving the problem for one scenario and adding the fixed differences. Additionally, the difference in objective values between the scenarios is also fixed and independent of the pre-matching. Thus, in each instance, there is a fixed scenario which is the worst-case scenario for each pre-matching.

Lemma 4 Let $D = (V, A), \mathcal{B}, c$ be an instance of the Directed Robust Perfect b-Matching Problem and for fixed $b \in \mathcal{B}$ and $h \in \mathbb{R}^{|V|}$, let $m^{b,h} \in \mathbb{R}^{|A|}$ be the unique matching satisfying both the pre-matching constraint for h and the matching constraint for b. Then, D has no alternating circuit if and only if for each pair of scenario $b^1, b^2 \in \mathcal{B}$ and (not necessarily integer) pre-matching $h^1, h^2 \in \mathbb{R}^{|V|}$, it holds $m_a^{b^1,h^1} - m_a^{b^2,h^1} = m_a^{b^1,h^2} - m_a^{b^2,h^2} =: \Delta_a$ for all $a \in A$.

Proof
Let $D = (V, A), \mathcal{B}, c$ be an instance of DRUPbM. Let $b^1, b^2 \in \mathcal{B}$ be arbitrary scenarios, $h^1, h^2 \in \mathbb{Q}^{|V|}$ rational pre-matchings and $m^{b^1,h^1}, m^{b^1,h^2}, m^{b^2,h^1}, m^{b^2,h^2} \in \mathbb{Q}^{|A|}$ the unique, perfect b-matchings fitting for the corresponding pre-matching and scenario as shown in Lemma 3. Additionally, we define the differences

$$\begin{aligned}\Delta_a^1 &:= m_a^{b^2,h^1} - m_a^{b^1,h^1}\\ \Delta_a^2 &:= m_a^{b^2,h^2} - m_a^{b^1,h^2}.\end{aligned}$$

Because the matchings are fitting for their corresponding pre-matching, we can use the pre-matching constraints (3b) to show for $i \in \{1, 2\}$ and all $v \in V$:

$$h_v^i \overset{(3b)}{=} \sum_{a\in\delta^+(v)} m^{b^1,h^i} \qquad \text{and} \qquad h_v^i \overset{(3b)}{=} \sum_{a\in\delta^+(v)} m^{b^2,h^i}$$

$$\Rightarrow h_v^2 - h_v^1 = \sum_{a\in\delta^+(v)} m^{b^1,h^2} - m^{b^1,h^1} \qquad \text{and} \tag{5}$$

$$h_v^2 - h_v^1 = \sum_{a\in\delta^+(v)} m^{b^2,h^2} - m^{b^2,h^1} \overset{\text{def.}}{=} \sum_{a\in\delta^+(v)} m^{b^1,h^2} + \Delta_a^2 - m^{b^1,h^1} - \Delta_a^1$$

$$\Rightarrow \sum_{a\in\delta^+(v)} m^{b^1,h^2} - m^{b^1,h^1} = \sum_{a\in\delta^+(v)} m^{b^1,h^2} + \Delta_a^2 - m^{b^1,h^1} - \Delta_a^1$$

$$\Rightarrow \sum_{a\in\delta^+(v)} \Delta_a^1 = \sum_{a\in\delta^+(v)} \Delta_a^2 \tag{6}$$

Additionally, with the perfect matching constraints (3c'), it holds for $i \in \{1, 2\}$ and all $v \in V$:

$$b_v^1 \overset{(3c')}{=} \sum_{a\in\delta(v)} m_a^{b^1,h^i} \overset{\text{def.}}{=} \sum_{a\in\delta(v)} m_a^{b^2,h^i} - \Delta_a^i \qquad \text{and}$$

$$b_v^2 \overset{(3c')}{=} \sum_{a\in\delta(v)} m_a^{b^2,h^i}$$

$$\Rightarrow b_v^2 - b_v^1 = \sum_{a\in\delta(v)} \Delta_a^i \tag{7}$$

Now, let

$$\tilde{m}_a = m_a^{b^1,h^1} + \Delta_a^2$$

Then, $\tilde{m}$ is a (rational) perfect b^2-matching fitting h^1:

$$\sum_{a\in\delta^+(v)} \tilde{m}_a = \sum_{a\in\delta^+(v)} \left(m_a^{b^1,h^1} + \Delta_a^2\right) \overset{(6)}{=} \sum_{a\in\delta^+(v)} \left(m_a^{b^1,h^1} + \Delta_a^1\right) \overset{\text{def.}}{=} \sum_{a\in\delta^+(v)} m_a^{b^2,h^1} \overset{\text{(3c')}}{=} h_v^1$$

$$\sum_{a\in\delta(v)} \tilde{m}_a = \sum_{a\in\delta(v)} \left(m_a^{b^1,h^1} + \Delta_a^2\right) \overset{(7)}{=} b_v^2 - b_v^1 + \sum_{a\in\delta(v)} m_a^{b^1,h^1} \overset{\text{(3b)}}{=} b_v^2 - b_v^1 + b_v^1 = b_v^2$$

With Lemma 3, m^{b^2,h^1} is the unique perfect b^2-matching fitting h^1. Thus, the two matchings have to be the same, which implies

$$\tilde{m} = m^{b^2,h^1}$$
$$\Rightarrow m_a^{b^1,h^1} + \Delta_a^2 = m_a^{b^1,h^1} + \Delta_a^1$$
$$\Rightarrow \Delta_a^2 = \Delta_a^1$$

Thus, the difference between two matchings fitting the same pre-matching is always the same, independent of the matchings and pre-matchings chosen. □

These results lead to a polynomial-time algorithm. We compute the differences Δ^i relative to a fixed scenario b^1 by solving the relaxed ILP formulation (4). Then,we determine an optimal solution m^* for b^1 under the additional constraint that the unique solutions for the other scenarios, $m^* + \Delta^i$, are feasible. The algorithm is detailed in Algorithm 1.

Algorithm 1 polynomial-time algorithm for the Directed Robust Perfect b-Matching Problem without alternating circuits

Input: $D = (V, A)$ without alternating cycles, scenarios $\mathcal{B} = \{b^1, \dots, b^k\} \subset \mathbb{Z}_+^{|V|}$, weights $c \in \mathbb{R}^{|A|}$

Output: $h, m^1, \dots, m^k$ maximizing worst-case value $\min_{i=1,\dots,k} c^T m^i$

$z^{\text{LP}}, h^{\text{LP}}, m^{1,\text{LP}}, \dots, m^{k,\text{LP}} \leftarrow$ optimal solution of the relaxed problem (4a)-(4d)
$\Delta_a^i \leftarrow m_a^{i,\text{LP}} - m_a^{1,\text{LP}}$ for all $b^i \in \mathcal{B} \setminus \{b^1\}$ and $a \in A$
$\text{lb}_a \leftarrow \max_{i\in\{2,\dots,k\}} -\Delta_a^i$
$m^1 \leftarrow$ optimal b^1-matching with lower bounds lb on each arc
$h \leftarrow$ fitting pre-matching for m^1
Return $h, m^1, m^1 + \Delta^2, \dots, m^1 + \Delta^k$

Theorem 8 Algorithm 1 computes an optimal solution to the Directed Robust Perfect b-Matching Problem on graphs without alternating circuits in polynomial time.

Proof
We begin by showing feasibility of the solution $h, m^1, m^1 + \Delta^2, \ldots, m^1 + \Delta^k$ returned by Algorithm 1. With Lemmas 3 and 4, we know that $m^1 + \Delta^2, \ldots, m^1 + \Delta^k$ are the unique matchings satisfying the perfect matching constraint (4d) for b^i and the pre-matching constraints (4c) for the same pre-matching as m^1. Thus, it only remains to be shown that the matchings are non-negative and integer if and only if $m^1_a \geq \text{lb}_a$. Since $m^1 \in \mathbb{Z}_+^{|V|}$ per definition, the matchings are integer if and only if Δ^i is integer. However, with Lemma 4, we know that the values of Δ^i are fixed for the instance. Thus, if Δ^i is non-integer, the instance is infeasible. It holds $m^1_a + \Delta^i_a \geq 0 \Leftrightarrow m^1_a \geq -\Delta^i_a$. Thus, all matchings are non-negative if and only if $m^1_a \geq \max_{i \in \{2,\ldots,k\}} -\Delta^i_a$. With this, we have shown that the matchings $m^1 + \Delta^1, \ldots, m^1 + \Delta^k$ are perfect b^i-matchings fitting for h. Now, it is easy to see that the computed solution is also optimal because m^1 is an optimal b^1-matching under the additional condition that the other matchings are feasible.

Finally, it only remains to be shown that the algorithm can be computed in polynomial time. Linear Programs can be solved in polynomial time as well as perfect b-matchings with lower bounds on the arcs. The remaining steps of the algorithm are simple, arithmetic computations which can be done in polynomial time. This concludes the proof. □

This algorithm can also handle certain instances of DRBPbM by using the reduction from Theorem 1.

Corollary 2 The Directed Robust Bounded Perfect b-Matching Problem can be solved in polynomial time on graphs without alternating ciruits and without parallel arcs.

Proof
We can use the reduction from Theorem 1 to transform a given instance of DRBPbM into an instance of DRUPbM. In this reduction, each arc with an upper bound is replaced by an alternating path of length three. Thus, there is an alternating circuit in the instance of DRUPbM if and only if there is an alternating circuit or parallel arcs in the instance of DRBPbM. Using Algorithm 1, this instance of DRUPbM can be solved in polynomial time. □

This implies that DRBPbM can be solved in polynomial time on directed cactus graphs. We note that both reduction graphs used in Theorems 5 and 6 use parallel arcs with upper bounds and the algorithm cannot be used on these instances.

5.3 Graph Classes Without Alternating Circuits

Next, we explore the graph classes for which this algorithm is applicable. Clearly, graphs without cycles, such as oriented paths, and oriented trees, do not contain alternating circuits. As mentioned before, the same holds for directed cactus graphs. For directed SP-graphs, it is less obvious that they do not allow alternating circuits.

Theorem 9 Let $D = (V, A, s, t)$ be a directed SP-graph without parallel arcs. The, D does not contain an alternating cycle. Furthermore, there is no alternating path between s and t of length ≥ 2. Thus, the only possible alternating path between s and t consist of the arc (s, t) only.

Proof
We proof the statement by induction over the graph structure. The statement holds for single arcs. Let D be a directed SP-graph gained through composition of $D_1 = (V_1, A_1, s_1, t_1)$ and $D_2 = (V_2, A_2, s_2, t_2)$. Then, the statement holds per induction on D_1 and D_2.

Case 1: D is a series composition. We begin by showing that there is no alternating path of length ≥ 2 in D. Each (s, t)-path p between $s_1 = s$ and $t_2 = t$ would have to pass through vertex $t_1 = s_2$. However, when crossing from V_1 to V_2 through $t_1 = s_2$, the path would necessarily use two arcs oriented in the same direction, since $t_1 = s_2$ has only incoming arcs from V_1 and only outgoing arcs to V_2. Thus, p cannot be an alternating path. The same argumentation holds for any pair of vertices $v_1 \in V_1$ and $v_2 \in V_2$. Thus, an alternating cycle in D would have to lie completely in either D_1 or D_2, which is a contradiction to the induction hypothesis.

Case 2: D is a parallel composition. With the induction hypothesis, there is no alternating cycle or alternating path length ≥ 2 between s and t in D_1 and D_2. Each path in D using vertices from $V_1 \setminus \{s, t\}$ and $V_2 \setminus \{s, t\}$ has to pass through s or t. Thus, each alternating path in D from s to t either lies completely in D_1 or D_2 or includes an alternating cycle in one of the two subgraphs. This is a contradiction to the induction hypothesis.

Similar, an alternating cycle in D would have to use arcs from both A_1 and A_2 by induction hypothesis. Since these arcs only share the two vertices s and t, such a cycle would have to include both vertices and consist of two alternating s, t-paths without cycles, one in D_1 and one in D_2. Since we do not allow multi-arcs, at least one of these alternating s, t paths does not only consist of the arc (s, t) and thus has length ≥ 2. This is a contradiction to the induction hypothesis. □

Corollary 3 The Directed Robust Perfect b-Matching Problem can be solved in polynomial time on directed SP-graphs.

5.4 Polynomial Solvable Graph Classes with Alternating Circuits

The presence of alternating circuits in a graph class does not necessarily imply the NP-hardness of the Directed Robust Perfect b-Matching Problem problem. One example are directed bipartite graphs. In a directed bipartite graph $D = (V_1 \cup V_2, A)$, each arc is directed from V_1 to V_2 and thus each circuit in D is an alternating circuit. However, each vertex $w \in V_2$ has only incoming arcs and thus $h_w = 0$ and each vertex $v \in V_1$ has only outgoing vertices and thus either $h_v = b_v^1 = \cdots = b_v^k$ or the instance is infeasible. Thus, there is only one feasible pre-matching and the problem can be easily solved in polynomial time by solving k deterministic b-matching problems on a bipartite graph. A similar argumentation works for graphs where each alternating cycle (not circuit) is a maximum connected subgraph and thus not connected to any vertex outside this alternating cycle. We call these cycles *isolated*.

Lemma 5 Let $D = (V, A)$ be a directed graph where each alternating circuit is also an alternating cycle and isolated. Then, the Directed Robust Perfect b-Matching Problem can be solved in polynomial time on D.

Proof
Let $\tilde{D} = (\tilde{V}, \tilde{A})$ be the maximum subgraph of D without alternating cycles and $D - \tilde{D} = (V \setminus \tilde{D}, A \setminus \tilde{A})$ the subgraph consisting of all alternating cycles in D. We note that there are no arcs connecting $\tilde{V}$ and $V \setminus \tilde{V}$ in A because all alternating cycles are isolated.

First, we show that there is a unique, feasible pre-matching on $D - \tilde{D}$. Since D consists of isolated, alternating cycles, each vertex in an alternating cycle in D has either only incoming or only outgoing arcs. Thus, similar to directed bipartite graphs, there is at most one feasible pre-matching for each vertex on an alternating cycle: for $v \in V \setminus \tilde{D}$ with $\delta^+(v) = \emptyset$, we set $h_v = 0$ and for vertices $v \in V \setminus \tilde{D}$ with $\delta^+(v) = \delta(v)$, it has to hold $h_v = b_v^1 = \cdots = b_v^{|\mathcal{B}|}$ or the instance is infeasible.

The DRUPbM can be solved in polynomial time on $\tilde{D}$ using Algorithm 1. Let $\tilde{h}$ be the pre-matching computed by the algorithm. Since the differences between scenarios Δ are fixed by Lemma 4, this pre-matching is optimal in each scenario on $\tilde{D}$. Thus, the pre-matching h^* defined by

$$h_v^* = \begin{cases} \tilde{h}_v & \text{if } v \in \tilde{V} \\ 0 & \text{if } \delta^+(v) = \emptyset \\ b_v^1 & \text{else} \end{cases}$$

is the optimal pre-matching for the instance on D. It can be computed in polynomial time because $\tilde{h}$ can be computed in polynomial time as shown in Theorem 8. The worst-case scenario and perfect matchings can be computed in polynomial time by solving the perfect b-matching problem on each scenario. □

However, this does not hold for isolated, alternating circuits.

Theorem 10 The Directed Robust Perfect b-Matching Problem is NP-complete on graphs where all alternating circuits are isolated.

A proof of this Theorem is given in Appendix A.3.

These results seem to imply that DRUPbM can be solved in polynomial time on graphs where each alternating circuit is also an alternating cycle. However, the reduction graph used in [11] to show weak NP-hardness of DRUPbM includes only alternating cycles. Thus, DRUPbM remains NP-hard on graphs where each alternating circuit is also an alternating cycle. However, it may be possible to find a pseudo-polynomial algorithm that first isolates all alternating cycles and then applies Lemma 5.

6 Conclusions

In this paper, we have analyzed the complexity of four variants of the Directed Robust b-Matching Problem on different graph classes. A summary of the complexity results is given in Table 6. The results for each graph class are split between directed (dir.) and

	path		tree		pearl		bipartite		SP-graph		cactus		no alt.
	dir.	or.	dir.	or.	dir.	or.	dir.	or.	dir.	or.	dir.	or.	circuits
DRUbM	P	NP-c.	NP-c.	NP-c.	P	NP-c.	NP-c.*	NP-c.	NP-c.	NP-c.	NP-c.	NP-c.	NP-c.
DRBbM	P	NP-c.	NP-c.	NP-c.	P	NP-c.	NP-c.*	NP-c.	NP-c.	NP-c.	NP-c.	NP-c.	NP-c.
DRUPbM	P	P	P	P	P	P	P	NP-c.*	P	NP-c.	P	NP-c.*	P
DRBPbM	P	P	P	P	P	NP-c.	P	NP-c.*	NP-c.	NP-c.	P	NP-c.*	NP-c.

Figure 6: Summary Complexity Results

oriented (or.) graph class. We use P to represent the existence of a polynomial time algorithm and NP-c. for NP completeness on the corresponding graph class. Results marked with a star $*$ are not proven in this paper.

The perfect variants of the problem are easier to solve than the other two variants. For both DRUbM and DRBbM, we could only find a polynomial time algorithm on directed paths and directed pearl graphs. On all other graph classes, the problem variants are NP-complete. For DRUPbM, we found a polynomial time algorithm for all graphs without alternating circuits. When further examining the influence of alternating circuits on DRUPbM, we found that graphs with isolated alternating cycles could still be solved in polynomial time, but graphs with isolated alternating circuits are already NP-complete. For DRBPbM, we could find polynomial-time algorithms for the directed variants of most graph classes we examined here. However, it is NP-complete on oriented pearl graphs and directed SP-graphs, which were both easy to solve for DRUPbM.

Further research might contain finding approximation algorithms or heuristics for the NP-hard cases or showing approximation hardness. Additionally, given that we have only proven weak NP-hardness in this paper, it might be interesting to either show strong NP-hardness or find a pseudo-polynomial algorithm for these cases. Specifically, it may be possible to find a pseudo-polynomial algorithm for DRUPbM by isolating all alternating cycles as proposed in the previous section. Alternatively, a polynomial time algorithm could be found on graphs with a constant number of alternating circuits. Finally, the problem could also be further extended by using interval data instead of discrete scenarios.

References

[1] Hassene Aissi, Cristina Bazgan, and Daniel Vanderpooten. Complexity of the min–max and min–max regret assignment problems. *Operations Research Letters*, 33(6):634–640, 2005.

[2] Christina Büsing and Sabrina Schmitz. Robust two-stage combinatorial optimization problems under discrete demand uncertainties and consistent selection constraints. *Discrete Applied Mathematics*, 347:187–213, 2024.

[3] Michael R Garey, Ronald L Graham, David S Johnson, and Donald Ervin Knuth. Complexity results for bandwidth minimization. *SIAM Journal on Applied Mathematics*, 34(3):477–495, 1978.

[4] A.M.H. Gerards. Chapter 3 matching. In *Network Models*, volume 7 of *Handbooks in Operations Research and Management Science*, pages 135–224. Elsevier, 1995.

[5] Omar El Housni, Vineet Goyal, Oussama Hanguir, and Clifford Stein. Matching drivers to riders: A two-stage robust approach. preprint on webpage at `https://arxiv.org/abs/2011.03624`, 2021.

[6] A. Kasperski, A. Kurpisz, and P. Zieliński. Recoverable robust combinatorial optimization problems. In *Operations Research Proceedings 2012*, pages 147–153. Springer International Publishing, 2014.

[7] Irit Katriel, Claire Kenyon-Mathieu, and Eli Upfal. Commitment under uncertainty: Two-stage stochastic matching problems. *Theoretical Computer Science*, 408:213–223, 2008.

[8] Bernhard H Korte, Jens Vygen, B Korte, and J Vygen. *Combinatorial optimization*, volume 1. Springer, 2011.

[9] Panos Kouvelis and Gang Yu. *Robust discrete optimization and its applications*, volume 14. Springer Science & Business Media, 2013.

[10] Aranyak Mehta et al. Online matching and ad allocation. *Foundations and Trends in Theoretical Computer Science*, 8(4):265–368, 2013.

[11] Jenny Segschneider and Arie M. C. A. Koster. Robust two-dose vaccination schemes and the directed b-matching problem. *Discrete Applied Mathematics*, 356:369–392, 2024.

Backmatter

Declaration of Generative AI and AI-assisted technologies in the writing process

During the preparation of this work, the authors used ChatGPT to rewrite the original draft with the goal of improving readability and correct spelling/grammatical mistakes. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

Financial disclosure

None reported.

Conflict of interest

The authors declare no potential conflict of interests.

A Additional NP-hardness proofs

A.1 DRUbM on oriented paths

Theorem (4)
The Directed Robust b-Matching Problem is weakly NP-complete on oriented paths. □

Proof

We show NP-hardness by reduction from the Partition Problem. Let S be an instance of Partition, $P = \sum_{i=1}^{n} s_i$ and $M > P$ some large integer. Analogous to the proof of Theorem 3, we define a directed graph $D = (V, A)$ with $V = \bigcup_{i=1}^{n} V_i$, $A = \bigcup_{i=1}^{n} A_i$ and gadgets

$$\begin{aligned} V_i &= \{r_i, d_i, v_i^{\text{in}}, v_i^{\text{out}}, v_i^{T,\text{in}}, v_i^{T,\text{out}}, v_i^{S\setminus T,\text{in}}, v_i^{S\setminus T,\text{out}}\} \\ A_i &= \{(v_i^{\text{in}}, v_i^{T,\text{in}}), (v_i^{\text{in}}, v_i^{S\setminus T,\text{in}}), (v_i^{S\setminus T,\text{in}}, d_i), (v_i^{T,\text{out}}, d_i), (v_i^{\text{out}}, v_i^{T,\text{out}}), (v_i^{\text{out}}, v_i^{S\setminus T,\text{out}}), \\ &\qquad (r_i, v_i^{S\setminus T,\text{out}}), (r_i, v_{i+1}^{1})\} \end{aligned}$$

D is an oriented path. Additionally, we define the two scenarios and objective value as follows:

$$b_v^T = \begin{cases} 0 & \text{if } v \in \{v_i^{S\setminus T,\text{out}}, r_i\} \\ 1 & \text{else} \end{cases} \qquad b_v^{S\setminus T} = \begin{cases} 0 & \text{if } v \in \{v_i^{T,\text{in}}, r_i\} \\ 1 & \text{else} \end{cases}$$

$$c_a = \begin{cases} s_i & \text{if } a \in \{(v_i^{\text{in}}, v_i^{T,\text{in}}), (v_i^{\text{out}}, v_i^{S\setminus T,\text{out}}) \colon i = 1, \dots, n\} \\ M & \text{if } a \in \delta(d_i),\ i = 1, \dots, n\} \\ 0 & \text{else} \end{cases}$$

One gadget of the instance is depicted in Figure 7. The capacities in scenario b^T are

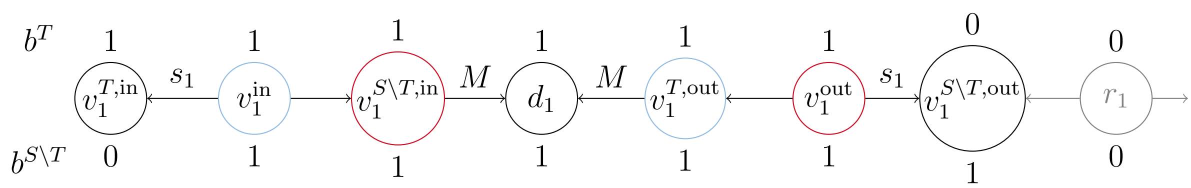

Figure 7: Gadget used to show NP-hardness of DRUbM on oriented paths

written above the vertices and the capacities in $b^{S\setminus T}$ below the vertices. Non-zero weights are written on the arcs.

The graph consists of n gadgets connected through the dummy vertices r_i with capacity zero in both scenarios. We now examine solutions for only one gadget V_i for some fixed $i \in \{1, \dots, n\}$. Due to the high weights of the arcs adjacent to d_i and $b_{d_i}^T = b_{d_i}^{S\setminus T} = 1$, the matching corresponding to h^* has to include one of the two adjacent arcs exactly one time. Because all weights are non-negative, an optimal solution will then always set either $h_{v_i^{\text{in}}} = 1$ or $h_{v_i^{\text{out}}} = 1$. There are only two pre-matchings (restricted to V_i) satisfying this: the pre-matching marked in light blue with $h_{v_i^2}^1 = h_{v_i^5}^1 = 1$ and $h_{v_i^j}^1 = 0$ else and the pre-matching in red with $h_{v_i^3}^2 = h_{v_i^6}^2 = 1$ and $h_{v_i^j}^2 = 0$ else. For each combination of pre-matching and scenarios, there is only one feasible matching. For pre-matching h^1 on the gadget, the corresponding objective value on gadget V_i is s_i+M for scenario b^T and M for scenario $b^{S\setminus T}$. For pre-matching h^2, this is reversed: the objective value in scenario b^T is M and the objective value for scenario $b^{S\setminus T}$ is $s_i + M$.

Using the same argumentation as in the proof of Theorem 3, it now holds that an optimal pre-matching h^* has objective value $\geq \left(n + \frac{1}{2}\right) \cdot P$ if and only if $T = \{i\colon h^*_{v_i^2} = 1\}$ is a feasible solution for the Partition problem. Thus, there is a solution to the instance of Partition if and only if there is a solution of the instance of DRUbM with objective value $\geq n \cdot M + \frac{P}{2}$. □

A.2 DRUPbM on oriented cactus graphs

Theorem (7)
The Directed Robust Perfect b-Matching Problem is weakly NP-complete on oriented cactus graphs. □

Proof
We show NP-hardness by reduction from the Partition problem. Let $S = \{s_1, \ldots, s_n\} \subset \mathbb{Z}_+$ be an instance of the Partition problem. Similar to the proof of Theorem 3, we use the two-stage structure of the Partition problem and define an instance of the Directed Robust Perfect b-Matching Problem where setting the pre-matching defines a subset $T \subset \{1, \ldots, n\}$ and picking one of the two pre-matchings $b^T, b^{S\setminus T}$ picks either $\sum_{i \in T} s_i$ or $\sum_{i \notin T} s_i \in S$ with

$$
\begin{aligned}
V_i &= \{v_i^{-5}, \ldots, v_i^{-1}, v_i^1, \ldots, v_i^5\} \\
A_i &= \{(v_i^1, v_i^{-1}), (v_i^1, v_i^2), (v_i^3, v_i^2), (v_i^3, v_i^4), (v_i^3, v_i^5), (v_i^4, v_i^5)\} \\
&\quad \cup \{(v_i^{-1}, v_i^1), (v_i^{-1}, v_i^{-2}), (v_i^{-3}, v_i^{-2}), (v_i^{-3}, v_i^{-4}), (v_i^{-3}, v_i^{-5}), (v_i^{-4}, v_i^{-5})\}
\end{aligned}
$$

$$
b^T_{v_i^j} = \begin{cases} 0 & \text{if } j = 2 \\ 2 & \text{else} \end{cases} \qquad b^{S\setminus T}_{v_i^j} = \begin{cases} 0 & \text{if } j = -2 \\ 2 & \text{else} \end{cases} \qquad c_a = \left\{ \frac{s_i}{2} \quad \text{if } a \in \{(v_i^1, v_i^{-1}), (v_i^{-1}, v_i^1)\} \right.
$$

The gadget graph is depicted in Figure 8. The green numbers next to the vertices represent the capacities in both scenarios. The black number on the arcs are the non-negative weights and the blue numbers on the arcs are exemplary perfect matchings for both scenarios fitting the same pre-matching. We note that the gadgets are symmetrical as indicated by the naming of the vertices. We set $D = (\bigcup_{i=1}^n V_i, \bigcup_{i=1}^n A_i)$. Similar to the previous proof, there is a solution to the instance of Partition if and only if there exists a solution to this instance of the Directed Robust Perfect b-Matching Problem with objective value $\geq \frac{1}{2} \sum_{i=1}^n s_i$.

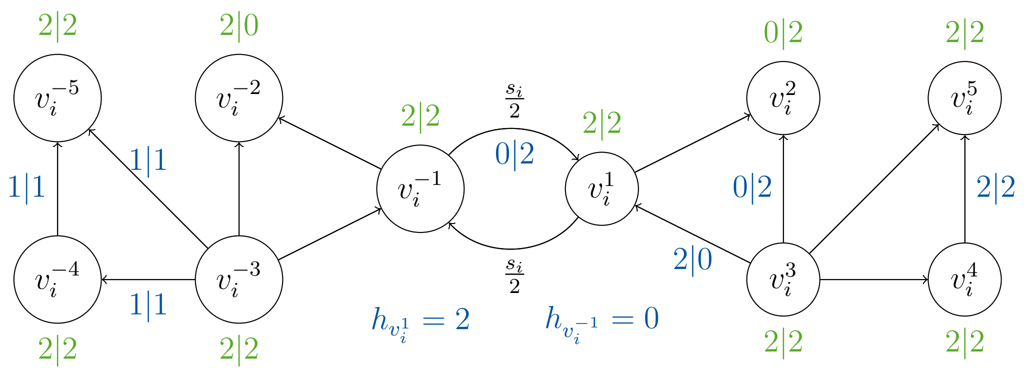

Figure 8: Gadget D_i used to construct the choice of setting $1 \in M$ for the Partition Problem

We now show that there are only two feasible pre-matchings on each gadget D_i. We begin by noting that the four vertices $v_i^{\pm 2}$ and $v_i^{\pm 5}$ have no outgoing arcs and thus pre-matching 0. Additionally, vertices $v_i^{\pm 3}$ have only outgoing arcs and thus pre-matching $h_{v_i^{\pm 3}} = b^T_{v_i^{\pm 3}} = b^{S \setminus T}_{v_i^{\pm 3}} = 2$. Thus, the pre-matching is only flexible on $v_i^{\pm 1}$ and $v_i^{\pm 4}$. For the three-cycles $v_i^{\pm 3}, v_i^{\pm 4}, v_i^{\pm 5}$ on the outside, the two only feasible matchings are shown in Figure 8. Either each arc is used once and $v_i^{\pm 3}$ cannot be matched with any vertex outside the cycle (left cycle in the Figure) or the arc $(v_i^{\pm 4}, v_i^{\pm 5})$ is used twice and $v_i^{\pm 3}$ has to be matched twice with vertices outside the cycle (right cycle in the Figure). In the first case, $h_{v_i^{\pm 4}} = 1$ and in the second case, $h_{v_i^{\pm 4}} = 2$. There is no feasible solution for $h_{v_i^{\pm 4}} = 0$.

We examine the first case with $h^{i,\text{in}}_{v_i^{-4}} = 1$ depicted in the Figure where v_i^{-3} is matched with v_i^{-4} and v_i^{-5}. Since vertex v_i^{-2} can now only be matched with v_i^{-1}, we have to set $h^{i,\text{in}}_{v_i^{-1}} = 2$. However, since the capacity of v_i^{-2} is 0 in the second scenario $b^{S \setminus T}_{v_i^{-2}} = 0$, the vertex v_i^-1 must be matched with v_i^1 via (v_i^{-1}, v_i^1) in the second scenario. This also implies $h^{i,\text{in}}_{v_i^1} = 0$, as the capacity of vertex v_i^1 in the second scenario is fully satisfied through an incoming arc. This, again, implies that v_i^1 in the first scenario and v_i^2 in the second scenario must be matched with v_i^3. Then, v_i^4 and v_i^5 have to be matched to each other and it holds $h^{i,\text{in}}_{v_i^4} = 2$ resulting in the matching depicted. Thus, when setting

$h^{i,\text{in}}_{v_i^{-3}} = 2$, the only feasible pre-matching is given by

$$h^{i,\text{in}}_{v_i^j} = \begin{cases} 2 & \text{if } j \in \{-3,-1,3,4\} \\ 1 & \text{if } j = -4 \\ 0 & \text{if } j \in \{-5,-2,1,2,5\} \end{cases}$$

$$m^{\text{in},T}_{(v_i^j, v_j^k)} = \begin{cases} 2 & \text{if } (j,k) \in \{(-1,-2),(3,1),(4,5)\} \\ 1 & \text{if } (j,k) \in \{(-3,-5),(-3,-4),(-4,-5)\} \\ 0 & \text{else} \end{cases}$$

$$m^{\text{in},S\setminus T}_{(v_i^j, v_j^k)} = \begin{cases} 2 & \text{if } (j,k) \in \{(-1,1),(3,2),(4,5)\} \\ 1 & \text{if } (j,k) \in \{(-3,-5),(-3,-4),(-4,-5)\} \\ 0 & \text{else} \end{cases}$$

with matchings as depicted in Figure 8. Using the symmetries of the graph and the same argumentation in reverse, we get the only feasible pre-matching for the second case $h^{i,\text{out}}_{v_i^{-4}} = 1$ as $h^{i,\text{out}}_{v_i^j} = h^{i,\text{in}}_{v_i^{-1\cdot j}}$ with matchings $m^{\text{out},T}_{(v_i^j, v_j^k)} = m^{\text{in},T}_{(v_i^{-1\cdot j}, v_j^{-1\cdot k})}$ and $m^{\text{out},S\setminus T}_{(v_i^j, v_j^k)} = m^{\text{in},S\setminus T}_{(v_i^{-1\cdot j}, v_j^{-1\cdot k})}$.

Since only two arcs in the gadget have non-zero weight, the objective values are easy to compute:

$$\sum_{a\in A} c_a m_a^{\text{in},T} = 0 \qquad \sum_{a\in A} c_a m_a^{\text{out},T} = s_i$$
$$\sum_{a\in A} c_a m_a^{\text{in},S\setminus T} = s_i \qquad \sum_{a\in A} c_a m_a^{\text{out},S\setminus T} = 0$$

Since these are the only feasible solutions for the gadget, each feasible solution for the complete instance can be defined by the choice between these two solutions on every gadget. Using $h^{i,\text{in}}$ on a gadget (V_i, A_i) would then represent picking $i \in T$. Then, the pre-matching h^* has objective value $\geq \frac{1}{2}\sum_{i=1}^n s_i$ if and only if $T^* = \{i \colon h^*_{v_i^j} = h^{i,\text{in}}_{v_i^j} \forall j \in \{-6,\dots,-1,1,\dots,6\}\}$ is a solution to the instance of Partition. Let $m^{*,T}, m^{*,S\setminus T}$ be the unique, fitting, perfect b-matching, then

$$\sum_{a\in A} c_a m_a^{*,T} = \sum_{i\in T}\sum_{a\in A} c_a m_a^{\text{out},T} + \sum_{i\in S\setminus T}\sum_{a\in A} c_a m_a^{\text{in},T} = \sum_{i\in T} s_i$$
$$\sum_{a\in A} c_a m_a^{*,S\setminus T} = \sum_{i\in T}\sum_{a\in A} c_a m_a^{\text{out},S\setminus T} + \sum_{i\in S\setminus T}\sum_{a\in A} c_a m_a^{\text{in},S\setminus T} = \sum_{i\in S\setminus T} s_i$$

Thus, if there is a solution with objective value $\geq \frac{1}{2}\sum_{i=1}^{n} s_i$, then

$$\sum_{i\in T} s_i = \frac{1}{2}\sum_{i=1}^{n} s_i = \sum_{i\in S\setminus T} s_i$$

and T is a solution to the instance of Partition. The reverse follows similar by constructing the corresponding pre-matching from the solution of Partition T. $\square$

A.3 DRUPbM on graphs with isolated, alternating ciruits

Theorem (10)
The Directed Robust Perfect b-Matching Problem is NP-complete on graphs where all alternating circuits are isolated. $\square$

Proof
We show NP-hardness by reduction from the Partition Problem, similar to Theorem 3. Let $S = \{s_1, \ldots, s_n\}$ be an instance of the Partition problem. For each $s_i \in T$, we define a gadget consisting of an alternating circuit with 3 vertices and 6 arcs as follows:

$$V^i = \{v_1^i, v_2^i, v_3^i\} \quad A^i = V^i \times V^i \setminus \{(v_1^i, v_1^i), (v_2^i, v_2^i), (v_3^i, v_3^i)\}$$

The objective value and capacities are set as

$$c_a = \begin{cases} -s_i & a = (v_1^i, v_2^i) \\ s_i & a = (v_3^i, v_2^i) \\ 0 & \text{else} \end{cases}, \qquad b^1_{v_j^i} = \begin{cases} 2 & j = 3 \\ 1 & \text{else} \end{cases}, \qquad b^2 = \begin{cases} 2 & j = 2 \\ 1 & \text{else} \end{cases}.$$

The gadget for $s_i \in S$ is depicted in Figure 9. Next, we examine the solution on one

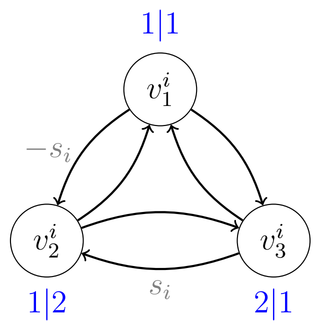

Figure 9: Alternating circuit gadget implementing the decision of picking $s_i \in T$

gadget only. In each alternating circuits, the sum of capacities in both scenarios is 4.

Thus, two arcs must be part of the matching and the sum over all pre-matchings for the three vertices must be two. Since none of the vertices has capacity ≥ 2 in both scenarios, the pre-matching on each vertex can be at most one. Thus, the only feasible pre-matchings are setting two of the three vertices to one and the remaining to zero. This results in three distinct pre-matchings: for $j \in \{1, 2, 3\}$, let

$$h^k_{v^i_j} = \begin{cases} 0 & j = k, \\ 1 & \text{else.} \end{cases}$$

For each pair of scenario and pre-matching, there is exactly one feasible matching. For h^1, the corresponding objective values are 0 in scenario b^1 and s_i in scenario b^2, for h^2 they are s_i and 0 and for h^3, the corresponding objective values are 0 and $-a_i$. The third pre-matching returns the worst solution in both scenarios. Thus, an optimal solution will use either h^1 or h^2 with objective value of s_i in either the first or the second scenarios.

Analogous to the proof of Theorem 3, it holds that an optimal solution of the instance of DRUPbM has an optimal pre-matching h^* objective value $\geq \frac{1}{2}\sum_{\imath \in S} s_i$ if and only if $T := \{i \colon h^*_{v^i_j} = h^1_{v^i_j} \text{ for } j \in \{1, 2, 3\}\}$ is a solution to the instance of Partition. $\square$