Robust Assignments via Ear Decompositions and Randomized Rounding

David Adjiashvili[1], Viktor Bindewald[*2], and Dennis Michaels[3]

1 **Institute for Operations Research, ETH Zürich, Zürich, Switzerland**
`addavid@ethz.ch`

2 **Department of Mathematics, TU Dortmund University, Dortmund, Germany**
`viktor.bindewald@math.tu-dortmund.de`

3 **Department of Mathematics, TU Dortmund University, Dortmund, Germany**
`dennis.michaels@math.tu-dortmund.de`

Abstract

Many real-life planning problems require making a priori decisions before all parameters of the problem have been revealed. An important special case of such problem arises in scheduling and transshipment problems, where a set of jobs needs to be assigned to the available set of machines or personnel (resources), in a way that all jobs have assigned resources, and no two jobs share the same resource. In its nominal form, the resulting computational problem becomes the *assignment problem*.

This paper deals with the *Robust Assignment Problem* (RAP) which models situations in which certain assignments are *vulnerable* and may become unavailable after the solution has been chosen. The goal is to choose a minimum-cost collection of assignments (edges in the corresponding bipartite graph) so that if any vulnerable edge becomes unavailable, the remaining part of the solution contains an assignment of all jobs.

We develop algorithms and hardness results for RAP and establish several connections to well-known concepts from matching theory, robust optimization, LP-based techniques and combinations thereof.

1998 ACM Subject Classification G.2.2 Graph algorithms

Keywords and phrases robust optimization, matching theory, ear decomposition, randomized rounding, approximation algorithm

Digital Object Identifier 10.4230/LIPIcs.ICALP.2016.71

1 Introduction

The need for incorporating system reliability into decision making has sprung wide-spread interest in optimization models which incorporate data uncertainty in the last decades. The latter trend has lead to the development of several new theories including the popular field of *Robust Optimization*. In robust optimization the nominal optimization problem is equipped with a set of *scenarios*, representing various possible states of nature that may occur after the solution to the problem is chosen. The goal is to determine a solution that will perform well (in terms of feasibility, or cost) in the *worst case* realization of the state of nature.

The *Assignment Problem* is one of the most fundamental optimization problems arising in many reliability-sensitive systems. In its nominal form, the input consists of a set of

* Financial support through RTG 1855 of the second author by the German Research Foundation (DFG) is gratefully appreciated.

43rd International Colloquium on Automata, Languages, and Programming (ICALP 2016).
Editors: Ioannis Chatzigiannakis, Michael Mitzenmacher, Yuval Rabani, and Davide Sangiorgi;
Article No. 71; pp. 71:1–71:14
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

n_T *tasks*, a set of n_R *resources* (with $n_T \leq n_R$), and *assignment costs* $c_{i,j}$ representing the cost associated with assigning resource i to task j. The set of allowed assignments can be represented by a bipartite graph $G := (R \dot\cup T, E)$ where each resource i corresponds to a node $r_i \in R$, each task j corresponds to a node $t_j \in T$, and the edge $\{r_i, t_j\}$ is present in E if the j-th task can be assigned to resource i. The goal is to find a *matching* $M \subseteq E$ of minimal cost that covers all nodes in T, i.e. a set of non-adjacent edges that is incident to every node in T. In the following, a subset M satisfying that property is called *an assignment*.

The Robust Assignment Problem (RAP) is the natural robust counterpart of the assignment problem and defined as follows[1]. An instance of RAP consists of an instance of the nominal assignment problem, i.e. of a bipartite graph $G = (R \dot\cup T, E)$ representing admissible assignments and a non-negative cost vector $c \in \mathbf{R}^E_{\geq 0}$, as well as a collection $F \subseteq E$ of *vulnerable edges*. Each $f \in F$ induces a failure scenario that leads to a deletion of f from G. The goal is to find a subset $X \subseteq E$ of minimal cost with the property that, for every vulnerable edge $f \in F$, the set $X \setminus \{f\}$ contains an assignment of G.

Intuitively, RAP asks to choose a redundant assignment, namely one that contains a feasible assignment, even when an arbitrary single vulnerable edge becomes unavailable. Therefore, the robustness paradigm considered in this paper falls into the topic of *redundancy-based robustness* – a well-motivated and widely studied approach (see e.g. [5, 23] for an overview of different robustness concepts). Some of the problems that fall into this category include the minimum k-edge connected spanning subgraph problem [12, 18] and the robust facility location problem [25, 31, 9]. More recently, Adjiashvili, Stiller and Zenklusen [2] introduced a robustness model called *bulk-robustness*, which combines the standard redundancy based robustness approach with a non-uniform failure model. In its general form, a bulk robust counterpart of a combinatorial optimization problem consists of an instance of the nominal problem, as well as a collection of scenarios, each comprising an arbitrary set of resources that may fail simultaneously. The goal is, as usual, to choose a minimum-cost set of resources that contains a feasible solution, even when the resources in any single failure scenario become unavailable. In the language of bulk-robustness, RAP is the bulk-robust assignment problem restricted to the case of where each scenario is composed of a single edge.

In the remainder of this section we provide a few motivating applications for RAP, establish some connections to related notions in matching theory and discuss results and technical contributions.

1.1 Motivation

The most natural applications of RAP, and redundancy-based robust optimization in general, emerge in situations where resources can not be easily made available on demand. In such applications, any resource that is intended for use at a certain point in time must be *reserved* at an earlier stage, and thus made available for potential deployment. Examples of such applications range from construction of robust power transmission networks [21] to supply chain management [32].

In a nutshell, redundancy-based models deal with the problem of choosing the optimal set of (potentially unreliable) resources to reserve, in order to guarantee that the available set of resources at the time of solution implementation, i.e. the reserved resources that did not fail, contains a feasible solution in every scenario. While we believe that RAP can be a

---

[1] Several other robust counterparts of the assignment problem have been considered in the literature under the same, or similar names. We review these models in Section 2.

useful model to incorporate robustness in any assignment model with up-front decisions of the latter type, we bring hereafter a few concrete applications.

Staff Training. Large companies often employ intensive training programs for their employees, designed to adapt the available pool of skills to their dynamic needs. For example, large software firms starting a new project involving new technologies, might need to train some employees to use these technologies. It is natural to incorporate the incurred training costs into the task assignment problem. The cost of assigning an employee to perform a given task in the project corresponds to the cost to train the employee to perform this task.

In a more realistic scenario, some employee to task assignments might become unavailable *even if the employee were trained to perform the task*. This type of vulnerability is very common, and can be caused, e.g. by employee dissatisfaction from his task assignment, or by unexpected inability to perform the task (due to injury or unavailability of equipment, etc.). RAP is a suitable model for deciding on robust training programs for the project, where skill sets of the employees allow for reassignments even if any single employee to task assignment becomes unavailable.

Continuity of Service. In industries such as health care and consulting it is often desirable to maintain very stable client to service provider relationships [7]. It is hence natural for a service provider to model the resulting resource allocation problem as an assignment problem, where an available pool of trained employees (nurses, consultants etc.) is matched to the set of customers. In the nominal variant, the company might want to minimize the cost incurred by the assignment, where the cost is computed as the total cost incurred by establishing all relationships in the assignment (establishing such relationships incurs significant costs).

It is however common that certain established relationships go off track in the course of a long interaction (e.g. due to customer or employee dissatisfaction). These relationships correspond to vulnerabilities of individual assignments. With RAP it is possible to account for such vulnerabilities by establishing a cost-effective set of relationships that, even if any nominal interaction becomes unavailable, the organization can quickly adjust the assignment to satisfy all clients.

1.2 Overview of results and techniques

This paper addresses the computational complexity of RAP. In particular, we present approximation algorithms and hardness of approximation results. We justify the study of approximation algorithms by showing that RAP is NP-hard even in very restricted variants. Due to space constraints we omit technical proofs as well as details on the complexity results.

The assignment problem has a well-known natural interpretation as a bipartite matching problem in the graph $G = (R \dot\cup T, E)$. It is hence also natural to view RAP as a robust version of the bipartite matching problem: find a minimum-cost set of edges in $M \subseteq E$ such that for every $f \in F$, the set $M \setminus \{f\}$ contains a matching incident to all nodes in T. Furthermore, if $|R| = |T|$, the problem becomes a robust variant of the *perfect* matching problem. We henceforth adopt this point of view, as it facilitates a clearer exposition of our results, and highlights an inherent connection between RAP and *matching-covered graphs*, a notion that we repeatedly use in our approximation algorithms.

The next statement shows that it suffices to consider RAP on balanced bipartite graphs, implying that we can state the feasibility condition for RAP using perfect matchings.

▶ **Proposition 1.** *Any* RAP *instance can be efficiently transformed to an equivalent weighted* RAP *instance with a balanced bipartite graph such that any α-approximation for the new instance can be used to efficiently construct an α-approximation of the original instance for all $\alpha \geq 1$.*

It is important to note that the transformation in Proposition 1 leads to an instance on a balanced bipartite graph that is equipped with a weighted cost function. Interpreting the resulting instance as an unweighted one may destroy the preservation of the transformation's approximation quality, thus Proposition 1 can not be used as a black box reduction for the unweighted case.

In the following, RAP refers to general instances of the robust assignment problem on balanced bipartite graphs with weighted cost function, while card-RAP is used for the unweighted version of RAP. We denote $n = n_T + n_R$ and $m = |E|$.

We remark that feasibility of a given RAP instance can be efficiently verified as one only needs to check, for each $f \in F$, whether the graph contains a perfect matching not using f. The latter can be done using any polynomial algorithms for finding maximum matchings in bipartite graphs.

1.2.1 Matching-Covered Graphs and Ear Decompositions

Our algorithmic results rely on a tight connection between RAP and *matching-covered graphs*, a well-known notion in matching theory. A graph $G = (V, E)$ is *matching-covered* if every edge $e \in E$ appears in some perfect matching of G.[2]

It turns out that inclusion-wise minimal solutions of any RAP instance are matching-covered as the following proposition states.

▶ **Proposition 2.** *A set $X \subseteq E$ is an inclusion-wise minimal feasible solution to* RAP *if and only if $(R \dot\cup T, X)$ is an inclusion-wise minimal non-empty graph with the properties of being matching-covered and that every isolated edge $e \in X$ is not vulnerable, i.e. $e \notin F$.*

Proposition 2 provides a very useful characterization of minimal solutions of RAP, as it allows us to use various results on matching-covered graphs in our algorithms for RAP. In particular, it allows us to identify feasible subgraphs and augment them to feasible solutions for the entire instance by adding structures that maintain the property of being matching-covered. One particularly useful tool is an *ear decomposition* of a bipartite matching-covered graph, i.e. a certain decomposition of G into edge-disjoint paths of odd length. In the following, we denote by $V[G]$ and $E[G]$ the set of nodes and edges of a graph G, respectively. For two graphs G and H we denote by $G + H$ their union $(V[G] \cup V[H], E[G] \cup E[H])$.

▶ **Definition 3** (Ear Decomposition of a Bipartite Graph). Let H be a bipartite graph, and let H' be a subgraph of H. An *odd ear of H with respect to H'* is a path P in H with an odd number of edges and such that P and H' have exactly two nodes in common. Those two nodes form the end points of P, and belong to different parts of the bipartition.

A *bipartite ear decomposition* is a sequence $P_0, P_1, \ldots, P_q$ of paths in H, such that: (i) $P_0 = (\{v_1, v_2\}, \{\{v_1, v_2\}\})$ is a graph composed of a single edge; (ii) $H = P_0 + \cdots + P_q$; and (iii)µfor every $j = 1, \ldots, q$, the path P_j is an odd ear with respect to $H_{j-1} := P_0 + \cdots + P_{j-1}$.

[2] The notion of matching-covered graphs is originally introduced for connected graphs. In this paper we use this term also for disconnected graphs. Furthermore, note that some authors use synonymously the notion 1-*extendable* or in the bipartite case *elementary* (cf. [28]).

We exploit the following well-known connection between matching-covered bipartite graphs and bipartite ear decompositions.

▶ **Theorem 4** ([28, Thm. 4.1.6])**.** *A bipartite graph is matching-covered if and only if it has a bipartite ear decomposition.*

1.2.2 Results for card-RAP

Theorem 4 allows us to prove the following results for card-RAP. An instance of RAP is called *uniform* if every edge is vulnerable, i.e. if $F = E$.

▶ **Theorem 5.** card-RAP *admits a polynomial* 1.5*-approximation algorithm in the uniform case, and a* 3*-approximation algorithm in the general case.*

Our algorithm starts by producing an ear decomposition of the input graph. Then, it iteratively selects a certain subset of the edges to be part of the solution, by processing the ears in the decomposition in the order given by the decomposition, and omitting the edges corresponding to ears of length one.

We complement the latter algorithmic result by showing that card-RAP is NP-hard to approximate within some constant $\delta > 1$ even in the restricted case of a uniform scenario set, as stated in the following theorem.

▶ **Theorem 6.** *There exists a constant $\delta > 1$, such that there is no polynomial δ-approximation algorithm for the uniform* card-RAP*, unless* P=NP*.*

Theorems 5 and 6 imply that the true approximability thresholds for uniform card-RAP and card-RAP lie in the intervals $[\delta, 1.5]$ and $[\delta, 3]$, respectively.

To complete the complexity landscape of card-RAP we also consider the case of only two vulnerable edges. This special case comprises the simplest variant of card-RAP that is not equivalent to a standard matching problem[3]. In the following theorem we prove that already this special case is NP-hard, thus drawing a sharp threshold for polynomial solvability of card-RAP.

▶ **Theorem 7.** card-RAP *is* NP*-hard even when restricted to instances with two vulnerable edges, i.e. with $F = \{f_1, f_2\}$.*

To the best of our knowledge, this is the first example of an NP-hard robust counterpart of a polynomial optimization problem, with a constant number of vulnerable resources.[4] To prove Theorem 7 we first show NP-hardness of a problem of partitioning a graph into a cycle containing a given node and a matching whose union covers all nodes, so as to minimize the length of the cycle, a problem that might be interesting in its own right.

1.2.3 Results for RAP

Our main algorithmic result for RAP is a randomized $O(\log n)$-approximation for the general case, as stated hereafter.

[3] Observe that the case of a single vulnerable edge $F = \{f\}$ is solvable by reporting any minimum-cost perfect matching in the graph $(R \dot\cup T, E \setminus \{f\})$ as a solution.

[4] There are many examples of optimization problems that become NP-hard when the robust counterpart is allowed to contain a constant number of scenarios (see e.g. [26]). In all such examples, however, each scenario affects a *non-constant* number of resources.

▶ **Theorem 8.** *RAP admits a randomized polynomial $O(\log n)$-approximation algorithm.*

Our approximation algorithm for RAP builds upon our simple approximation algorithm for card-RAP in that it also iteratively constructs a solution maintaining the invariant that at any point in the algorithm, the edges selected so far form a matching-covered graph. It is however unclear how to arrive at the desired approximation for RAP relying only on properties of matching-covered graphs. We therefore combine the latter techniques with additional tools from linear programming (LP) theory and randomized rounding. Concretely, we start by solving an LP relaxation of RAP, derived from a natural integer linear programming (ILP) formulation of the problem. The obtained fractional solution is used to guide an iterative randomized procedure. In each iteration a fractional bipartite matching corresponding to part of the fractional solution is selected. A decomposition of this fractional matching into a convex combination of integral matchings is then used to randomly pick one matching, and a carefully selected subset of this matching is added to the current solution. To bound the quality it does not suffice to bound the number of iterations, or the expected number of times an edge is part of a candidate matching. Instead we use a discharging argument that assigns costs to nodes depending on the graph selected so far.

We complement our algorithmic results for RAP with hardness of approximation result with the same asymptotic bound, as stated hereafter.

▶ **Theorem 9.** *Provided that* NP $\not\subseteq$ DTIME$(n^{\log\log n})$*, the uniform* RAP *admits no $c\log n$-approximation for any $c < 1$. RAP unless P=NP.*

2 Related work

Redundancy-based robustness is a paradigm that motivates many well studied problems, including the minimum k-connected subgraph problem [18, 12, 30], survivable and robust network design problems [24, 10, 2, 1], robust clustering problems [25, 31], robust spanner problems [8, 15] and many more. All of the latter models bare a close resemblance to RAP: they assume resources to be vulnerable and ask to find a minimum-cost set of resources that contains a desired structure even in case any vulnerable resource, or set of resources, fails.

A relatively new approach to redundancy based robustness is the incorporation of non-uniform uncertainty sets [2, 1]. RAP is seen as a robust model of this type, as we allow both vulnerable and invulnerable edges in the same instance.

The study of robustness with respect to cost uncertainty was initially studied by Kouvelis and Yu [26], and Yu and Yang [33]. For a survey we refer to Aissi, Bazgan and Vanderpooten [4]. A closely related class of multi-budgeted problems has received considerable attention (see e.g. [19] and references therein). The latter works include variants of the related multi-objective matching problem.

Various variants of robust matching problems have been considered in the literature. Hassin and Rubinstein [22], and Fujita, Kobayashi and Makino [17] study the following notion of an α-robust matching. A perfect matching M in a weighted graph is α-robust (for $\alpha \in (0, 1]$), if for every $p \leq |M|$, the p heaviest edges of the matching have total weight at least α times the weight of a maximum weight matching of size p. Deineko and Woeginger [14] showed that the min-max-robust assignment problem with a fixed number of scenarios is equivalent to the *exact perfect matching problem*, a famous problem with unknown complexity status. In the case of a variable number of scenarios the min-max-robust problem is NP-hard, as was proved by Aissi, Bazgan and Vanderpooten [3]. Additional work on robust variants of the matching problem include models with recovery [16], models with node failures [27], and the closely related matching interdiction problem [34]. Plesník [29] provided conditions under which an r-regular graph remains perfectly matchable after removing $r-1$ arbitrarily chosen edges. Brigham, Harary, Violin and Yellen [6] and Cheng, Lesniak, Lipman and Liptak [11] studied the minimum number of edges to be removed from a graph to arrive at a graph without a perfect matching.

3 Approximation Algorithms for RAP

3.1 $O(1)$-Approximation for card-RAP

Our algorithm relies on an ear decomposition of the underlying bipartite graph. Similar ideas using ear decompositions were successfully used to approximate various combinatorial optimization problems including, among others, the minimum edge connected subgraph problem and the path traveling salesman problem [12, 30]. Recall, that a balanced bipartite graph admits an ear decomposition (which is by no means unique) if and only if it is matching-covered.

As an initial pre-processing step, a given card-RAP instance consisting of $G = (R \dot\cup T, E)$ and F is transformed into a balanced instance. For this, we introduce a set D of $n_R - n_T$ dummy task nodes, and connect each such node to all nodes from R. Let E_D denote the set of newly introduced edges, and let $G_b = (R \dot\cup (T \cup D), E \cup E_D)$. In a second step we remove from G_b all *dispensable edges*, i.e. all edges not appearing in any perfect matching of G_b. This way, we obtain a graph, that is, by definition, matching-covered. Note that the second step can be implemented in polynomial time using any efficient algorithm for finding bipartite matchings. Moreover, we remark that omitting dispensable edges clearly does not change the underlying card-RAP instance, since such edges can be removed from any feasible solution without breaking feasibility. In the following, we allow some abuse of notation and call the new graph G_b as well. Next, we assume that G (and equivalently G_b) is feasible, i.e. there do not exist any isolated edges. If an isolated edge exists the algorithm terminates and reports that the instance is infeasible. Now let $G_b = P_0 + \cdots + P_q$ be any ear decomposition of G_b with the initial edge P_0 not covering a dummy node from D. We call an ear P_j *trivial* if it is not P_0 and if it consists of a single edge. The next lemma shows that a feasible solution to card-RAP can be obtained from the ear decomposition of G_b by skipping trivial ears.

▶ **Lemma 10.** *Let $J = \{j \in [q] \mid P_j \text{ is a trivial ear}\}$. Define $G'_b = P_0 + \sum_{i \in [q] \setminus J} P_i$, and $X := E[G'_b] \setminus E_D$. Then, the set X is a feasible solution to the* card-RAP *instance. Furthermore, $|X| \leq 3n_T$.*

Lemma 10 allows us to arrive at an approximation algorithm, summarized as Algorithm 1.

Algorithm 1 : $O(1)$-Approximation for card-RAP

Require: $G = (R \dot\cup T, E)$ and $F \subseteq E$.
Ensure: a feasible solution X to card-RAP on G and F
1: $X \leftarrow \emptyset$
2: Transform G into a balanced graph G_b and remove all dispensable edges
3: Compute an ear decomposition $G_b = P_0 + \ldots + P_q$
4: $X \leftarrow P_0 \cup \bigcup\{E[P_j] \mid P_j \text{ is not trivial}, \ j = 1, \ldots, q\}$
5: **return** X

Proof of Theorem 5. According to [13], an ear decomposition of a matching-covered graph can be computed in polynomial time. Furthermore, all other computations can also be implemented efficiently, such that the running time of Algorithm 1 is polynomial. From Lemma 10, it follows that the set X returned by Algorithm 1 is feasible. For $F = E$, any feasible solution must have at least two edges incident to any node from the set T. Hence, $\text{OPT} \geq 2n_T$. Since $|X| \leq 3n_T$, the approximation guarantee is indeed 1.5. If $F \subsetneq E$, then G can contain a perfect matching not including any edge from F. Thus, we can use $\text{OPT} \geq n_T$ yielding an approximation factor of 3. ◀

3.2 $O(\log n)$-Approximation for RAP

In this section we provide a polynomial $O(\log n)$-approximation algorithm for RAP, thus proving Theorem 8. Again, we assume that the RAP instance is feasible. For a clean presentation, we first describe an algorithm for the uniform case $F = E$ and then explain how it can be extended to the non-uniform case.

Our algorithm is based on an LP-rounding procedure that works with a relaxation of the integer linear formulation of RAP, that is defined as follows. Let $G = (R \dot\cup T, E)$ be a balanced, bipartite graph and let $c \in \mathbf{R}^E_{\geq 0}$ be a non-negative cost vector. Moreover, let $P_G \subseteq \mathbf{R}^E$ denote the perfect matching polytope associated with G (i. e. P_G is the convex hull of all incidence vectors of perfect matchings in G). A standard ILP formulation of RAP contains the following variables: (i) $x^{-f} \in \{0,1\}^E$ representing a perfect matching in $G - f := (R \dot\cup T, E \setminus \{f\})$, for all $f \in F$, and (ii) $y \in \{0,1\}^E$ encoding a feasible solution to RAP. Then, RAP can be modeled as an ILP as follows.

$$\begin{array}{lrcll} \min & c^\top y & & & \\ \text{s.t.} & x^{-f} & \in & P_G \cap \{x \in \mathbf{R}^E \mid x_f = 0\}, & \text{for each } f \in F, \\ & y & \geq & x^{-f}, & \text{for each } f \in F, \\ & x^{-f} & \in & \{0,1\}^E, & \text{for each } f \in F, \\ & y & \in & \{0,1\}^E & \end{array} \tag{ILP}$$

The LP-relaxation (LP) is obtained by relaxing all integrality constraints in (ILP). To keep notation short, we let $x \in \left(\mathbf{R}^E\right)^E$ be the vector with parts x^{-f}, $f \in F$. It is straightforward to verify that integer solutions to (ILP) coincide with feasible solutions to the RAP instance. In the following, we will denote, by $\chi^S \in \{0,1\}^E$, the incidence vector of a subset $S \subseteq E$.

Now, let (x, y) be a fractional solution to (LP). We describe hereafter a rounding procedure that yields an approximation for RAP. Consider some edge $f \in F$. Since x^{-f} is contained in $P_G \cap \{x \in \mathbf{R}^E \mid x_f = 0\}$, there exist positive scalars $\lambda_1^{-f}, \cdots, \lambda_k^{-f}$ with $\sum_{i \in [k]} \lambda_i^{-f} = 1$, and perfect matchings $M_1^{-f}, \cdots, M_k^{-f}$ in $G - f$ such that $x^{-f} = \sum_{i \in [k]} \lambda_i^{-f} \chi^{M_i^{-f}}$. By Caratheodory's theorem, there is a decomposition of the latter type with k bounded by $O(m) = O(n^2)$. Furthermore given x^{-f}, such a decomposition can be computed in polynomial time using polyhedral techniques [20, Thm. 6.5.11].

Our algorithm performs several iterations of randomized rounding based on the latter decomposition of fractional matchings. More precisely, at each iteration, an infeasible set $X \subseteq E$ of edges, that was chosen so far, is augmented with an additional set M of edges chosen randomly as follows. First, an arbitrary edge f is chosen from E among all edges not yet covered by X, i.e. among all $e' \in E$ such that the edge set X selected so far contain no perfect matching that does not include e'. Next, a decomposition of the vector x^{-f} as a convex combination of perfect matchings is computed, as above. This decomposition is then used to select a single perfect matching $\bar{M}$ from $\{M_1^{-f}, \cdots, M_k^{-f}\}$ *randomly*, where M_i^{-f} is chosen with probability λ_i^{-f} for all $i \in [k]$. Finally, the augmenting set $M \subseteq \bar{M}$ is chosen to contain all edges of $\bar{M}$ connecting *distinct connected components* of X. The edges of M

are added to X and the algorithm proceeds to the next iteration. The algorithm terminates when X is a feasible solution. A summary of the algorithm is presented as Algorithm 2.

To prove the correctness of the algorithm we resort again to properties of matching-covered graphs. Concretely, as a main ingredient of the proof of Lemma 12, which states a useful structural property of intermediate solutions in the algorithm, we use the following classic result.

Algorithm 2 : Randomized $O(\log n)$-Approximation for RAP

Require: $G = (R \dot\cup T, E)$ with $|R| = |T|$, and $c \in \mathbf{R}^E_{\geq 0}$.
Ensure: A feasible solution X to RAP on G with $F = E$ and cost vector c.

1: Solve (LP) to obtain an optimal solution (x, y)
2: $X \leftarrow \emptyset$
3: **while** X is infeasible **do**
4: Select an edge $f \in F$ such that $X \setminus \{f\}$ contains no perfect matching
5: Compute a decomposition of x^{-f} as $x^{-f} = \sum_{i=1}^{k} \lambda_i^{-f} \chi^{M_i^{-f}}$ and select one matching $\bar{M} \in \{M_i^{-f} \mid i \in [k]\}$ with $\Pr\left[\bar{M} = M_i^{-f}\right] = \lambda_i^{-f}$ for all $i \in [k]$
6: Add to X all edges from $\bar{M}$ that connect distinct connected components in $(R \dot\cup T, X)$
7: **end while**
8: **return** X

▶ **Theorem 11** ([28, Thm. 4.1.1., p. 122]). *A connected bipartite graph $H = (U \dot\cup W, E)$ with $|U \dot\cup W| \geq 4$ is matching-covered if and only if for any $u \in U$ and $w \in W$ the graph $H - \{u\} - \{w\}$ has a perfect matching.*

▶ **Lemma 12.** *Let X be a non-empty set of edges already selected in an arbitrary iteration of Algorithm 2. Then, the graph $G[X] := (R \dot\cup T, X)$ is matching-covered.*

Proof. As X is assumed to be non-empty, X contains at least one perfect matching of G. Thus, $G[X]$ does not have isolated nodes.

Now, let $S \subseteq R \dot\cup T$ be the nodes of some connected component of $G[X]$. It suffices to prove that the graph $(S, X[S])$, with $X[S] := \{e \in X \mid e = \{s_1, s_2\},$ for some $s_1, s_2 \in S\}$, is matching-covered. For $|S| = 2$, $X[S]$ contains exactly one edge that belongs to a perfect matching in X. Thus, the claim is proved.

Next, assume that $|S| > 2$. To prove that $(S, X[S])$ is matching-covered, we proceed by induction on the number of iterations in Algorithm 2. In the first iteration, a perfect matching is added to X in Step 6, thus the claim holds in that case.

Now, let $X' \subseteq X$ be the set of edges selected until the beginning of the iteration preceding the current iteration. By the induction hypothesis, we can assume that every connected component of $(R \dot\cup T, X')$ is matching-covered.

To prove the claim we need to show that every edge $e \in X[S]$ is contained in some perfect matching of S. If $e \in X'$, this claim holds by the inductive hypothesis, and due to $X' \subseteq X$. In case that $e \notin X'$, we have that $e \in \bar{M}$, where $\bar{M}$ is selected in Step 5 in the current iteration. This means that e connects nodes from two distinct connected components of $(R \dot\cup T, X')$.

Now, pick any cycle $C \subseteq X$ in $G[X]$ containing e with a minimum number of edges from $\bar{M}$. Let $D_1, \cdots, D_l \subseteq R \dot\cup T$ be the components in $(R \dot\cup T, X')$ that have edges in C. From minimality of $|C \cap \bar{M}|$ it follows that C is a simple cycle (i.e. each node is contained in at most two of its edges) and that each component D_j, $j = 1, \cdots, l$ contains exactly two nodes incident to the cycle. This cycle can now be used to demonstrate the existence of the desired perfect matching M' as follows. First, include in M' all edges in $C \cap \bar{M}$. Then, in every component D_j for $j = 1, \cdots, l$ pick a matching covering all nodes, except the two nodes incident to the cycle C. Due to Theorem 11, such a matching exists since each component D_j is matching-covered. The matching chosen so far covers exactly the nodes in $D_1 \cup \cdots \cup D_l$. Finally, pick any matching covering all other components of $(R \dot\cup T, X')$ that are not incident to C. This matching exists, since again, $(R \dot\cup T, X')$ is matching-covered. The result is a perfect matching in $G[X]$ containing e, which completes the proof. ◀

Lemma 12 guarantees that at every stage in the algorithm, the only edges not yet covered by the current solution X are the isolated edges of $G[X]$. Now, since at an iteration where an uncovered edge f is chosen in Step 4, the set M must contain two edges distinct from f, that are incident to the endpoints of f, the edge f is guaranteed to be covered in the end of this iteration. This immediately implies that the algorithm terminates with a feasible solution after at most $|E|$ iterations. It hence remains to bound the expected cost of the solution returned by Algorithm 2.

▶ **Theorem 13.** *The expected cost of the solution returned by Algorithm 2 is $O(\log n) \cdot \mathrm{OPT}$, where* OPT *is the optimal solution value for the* RAP *instance.*

Proof. The feasibility of the obtained solution and the bound on the running time are guaranteed by Lemma 12.

For a set $Q \subseteq E$ of edges we denote by $c_{\mathrm{LP}}(Q)$ the contribution of the edges in Q to the LP cost, i.e. $c_{\mathrm{LP}}(Q) = \sum_{e \in Q} c_e y_e$. For a node $v \in R \dot\cup T$, we denote by $\delta(v) \subseteq E$ the set of edges incident to v. To bound the approximation guarantee we bound the expectation of the ratio $c(X)/c_{\mathrm{LP}}(E)$. Since the LP is a relaxation of the problem we have $c_{\mathrm{LP}}(E) \leq \mathrm{OPT}$. Thus, this ratio is a valid bound on the approximation guarantee.

To obtain the bound we design a scheme that charges every selected edge in any stage of the algorithm to one of its endpoints. We then show that the expected cost charged to any node $v \in V$ is bounded by $O(\log n)$ times the *fractional cost* $c_{\mathrm{LP}}(\delta(v))$ associated with the node. This then implies that the expected cost of all edges added by the algorithm is at most

$$O(\log n) \cdot \sum_{v \in R \dot\cup T} c_{\mathrm{LP}}(\delta(v)) \leq O(\log n) \cdot \mathrm{OPT},$$

where the last inequality follows from linearity of expectation, $c_{\mathrm{LP}}(E) \leq \mathrm{OPT}$ and $c_{\mathrm{LP}}(E) = \frac{1}{2} \sum_{v \in R \dot\cup T} c_{\mathrm{LP}}(\delta(v))$.

We describe next how the costs of the selected edges are charged to the nodes of the graph. Let $\bar{X} \subseteq E$ be the set of edges returned by the algorithm. Formally, with each node $v \in R \dot\cup T$ we associate a collection of edges $R_v \subseteq \bar{X}$ such that $\bigcup_{v \in R \dot\cup T} R_v = \bar{X}$ and such that $c(R_v)$ is bounded by $O(\log n)$ times the fractional load at v in expectation.

The sets R_v are constructed as follows. In the beginning $R_v = \emptyset$ for all $v \in R \dot\cup T$. Let X be the set of edges selected so far by the algorithm and let $M \subseteq E \setminus X$ be the set of edges selected to be added to X in Step 6 of the current iteration. At this stage, the sets R_v might already contain some edges. We describe how these sets change as a result of the selection of M. Consider an edge $f = \{r, t\} \in M$. Recall that the algorithm only includes edges in the solution if they connect different connected components in $(R \dot\cup T, X)$. Thus, r and t lie in different connected components. Let D_r and D_t be the node sets of the connected components to which r and t belong, respectively, and assume without loss of generality that $|D_r| \leq |D_t|$. Then, f is charged to r, i.e. f is included in R_r. In other words, an edge added by the algorithm in any iteration is charged to the node contained in the *smaller connected component*, with ties broken arbitrarily.

It is obvious that the latter scheme charges all edges in $\bar{X}$ to some nodes, such that $\bigcup_{v\in V} R_v = \bar{X}$ holds in the end of the last iteration. It remains to analyze the quantity $c(R_v)$ for a single node $v \in R\dot{\cup}T$. The bound on $c(\bar{X})$ will then follow from linearity of expectation and the previous discussion. To arrive at the desired bound it suffices to make the following two observations.

First, at any time, if an edge is charged to v, its expected cost is at most $c_{\mathrm{LP}}(\delta(v))$. Indeed, recall that the edges in M come from a perfect matching chosen at random from the decomposition of some fractional perfect matching x^{-f} in the graph (this x^{-f} corresponds to the edge f chosen in Step 4 in the current iteration). Let this decomposition be

$$x^{-f} = \sum_{i\in[k]} \lambda_i^{-f} \chi_{M_i^{-f}}.$$

The distribution over the integral matchings defining x^{-f} induces a distribution over the edges incident to v: each edge $e \in \delta(v)$ is contained in the perfect matching with a probability $p_e \in [0, 1]$ given by

$$p_e = \sum_{i\in[k]\,:\,e\in M_i^{-f}} \lambda_i^{-f} = x_e^{-f}.$$

Since $x_e^{-f} \le y_e$ for all $e \in E$ we have that the expected cost of the edge charged to v is at most $\sum_{e\in\delta(v)} c_e x_e^{-f} \le c_{\mathrm{LP}}(\delta(v))$, proving the first property.

The second observation concerns the number of times the node v is charged in the course of the algorithm. Consider any iteration in which some edge was charged to v, and let D_v be the nodes in the component of v in the beginning of the iteration. Since we always charge an edge to the node in the smaller component, and since charged edges always merge connected components, the size of the connected component containing v in the end of the iteration is *at least* $2|D_v|$. Since the graph only contains n nodes, this doubling can only happen at most $\log n$ times.

We conclude that $c(R_v)$ is, in expectation, indeed at most $O(\log n)c_{LP}(\delta(v))$, which concludes the proof of the theorem. ◀

Lemma 12 and Theorem 13 imply the correctness of Theorem 8 for the uniform case. The generalization to the non-uniform case is explained in the proof of Theorem 8, which we bring next.

Proof of Theorem 8. It remains to show how to treat the case $F \neq E$. For this, we provide a transformation to reduce such an instance to a uniform instance by losing only a factor of 2 in the approximation guarantee.

The transformation first adds to the graph one parallel edge $\bar{e}$ for every $e \notin F$. Let G' be the obtained graph. The new set of vulnerable edges is set to $F' = E[G']$. Solutions for the two RAP instances are in obvious correspondence: A solution $X \subseteq E[G]$ to the original instance can be transformed to a solution for the new instance of at most double the cost by taking $X' = X \cup \{\bar{e} \mid e \in X \setminus F\}$. Conversely, a solution X' for the new instance can be transformed to a solution for the new instance with the same, or better cost, by choosing $X = X' \cap E[G]$.

Let OPT' denote the optimal solution value of the transformed uniform instance. Our $O(\log n)$-approximation algorithm for the general case proceeds by first transforming the instance to a uniform instance of RAP, as above, then invoking Algorithm 2 to obtain the set X', having expected cost at most $O(\log n)\mathrm{OPT}' = O(\log n)\mathrm{OPT}$ and then returning $X = X' \cap E[G]$. ◀

We conclude this section by arguing why simpler randomized rounding techniques, for instance, ones that lead to logarithmic approximation to many covering problems, are unlikely to lead to a similar result for RAP. The reason for this is that there does not seem to be a simple way to obtain a compact set cover-type representation of RAP without losing a super-logarithmic factor in the approximation guarantee. One natural attempt could be to consider every vulnerable edge $f \in F$ as an *element* that needs to be covered, and every possible perfect matching $M \subseteq E$ that does not contain f, a *covering set* that covers the edge f (and all other edges in $F \setminus M$). The cost of the covering set is simply the sum of the costs of edges in the corresponding perfect matching. Unfortunately, it is easy to come up with examples in which the optimal solution value in the latter set covering model has cost $\Omega(n)\mathrm{OPT}$. Such instances can be constructed, for example by choosing an instance, such that any feasible solution must have some nodes with very high degree, while an optimal solution has cost $O(n)$.

4 Conclusion and Future Work

This paper studies a novel practically relevant robust variant of the assignment problem (RAP). We showed tight connections between RAP and classical notions in matching theory, including matching-covered graphs and ear decompositions, and used these connections to obtain asymptotically tight approximation results for RAP. In our approximation algorithm for the general variant of RAP we combined classical results for matching-covered graphs with LP randomized rounding techniques.

Some ongoing and future work includes the following lines of research. Study a version of RAP with node failures, or with a combination of node and edge failures. This problem has many potential applications beyond the ones listed here. Study the variant of RAP where each scenario consists of at most k edges, for some input parameter $k > 1$. This paper treats the case $k = 1$. Besides, it is interesting to study the complexity of RAP in general graphs.

Acknowledgement. The work of the second author and the third author is part of the Research Training Group *"Discrete Optimization of Technical Systems under Uncertainty"* (RTG 1855).

References

1 D. Adjiashvili. Non-uniform robust network design in planar graphs. In *Proc. of APPROX*, pages 61–77, 2015.

2 D. Adjiashvili, S. Stiller, and R. Zenklusen. Bulk-robust combinatorial optimization. *Math Program*, 149(1-2):361–390, 2014.

3 H. Aissi, C. Bazgan, and D. Vanderpooten. Complexity of the min–max and min–max regret assignment problems. *Oper Res Lett*, 33(6):634–640, 2005.

4 H. Aissi, C. Bazgan, and D. Vanderpooten. Min–max and min–max regret versions of combinatorial optimization problems: A survey. *Eur J Oper Res*, 197(2):427–438, 2009.

5 D. Bertsimas, D.B. Brown, and C. Caramanis. Theory and applications of robust optimization. *SIAM review*, 53:464–501, 2011.

6 R.C. Brigham, F. Harary, E.C. Violin, and J. Yellen. Perfect-matching preclusion. *Congressus Numerantium*, 174:185–192, 2005. 00042.

7 G. Carello and E. Lanzarone. A cardinality-constrained robust model for the assignment problem in Home Care services. *Eur J Oper Res*, 236(2):748–762, 2014.

8 S. Chechik, M. Langberg, D. Peleg, and L. Roditty. Fault-tolerant spanners for general graphs. In *Proc. of STOC*, pages 435–444, 2009.

9 S. Chechik and D. Peleg. Robust fault tolerant uncapacitated facility location. In *Proc. of STACS*, pages 191–202, 2010.

10 C. Chekuri. Routing and network design with robustness to changing or uncertain traffic demands. *ACM SIGACT News*, 38(3):106–129, 2007.

11 E. Cheng, L. Lesniak, M.J. Lipman, and L. Liptak. Conditional matching preclusion sets. *Information Sciences*, 179(8):1092–1101, 2009. 00039.

12 J. Cheriyan, A. Sebo, and Z. Szigeti. Improving on the 1.5-Approximation of a Smallest 2-Edge Connected Spanning Subgraph. *SIAM J Discrete Math*, 14(2):170–180, 2001.

13 M.H. de Carvalho and J. Cheriyan. An O(VE) algorithm for ear decompositions of matching-covered graphs. In *Proc. of SODA*, pages 415–423, 2005.

14 V.G. Deineko and G.J. Woeginger. On the robust assignment problem under a fixed number of cost scenarios. *Oper Res Lett*, 34(2):175–179, 2006.

15 M. Dinitz and R. Krauthgamer. Fault-tolerant spanners: better and simpler. In *Proc. of PODC*, pages 169–178. ACM, 2011.

16 M.C. Dourado, D. Meierling, L.D. Penso, D. Rautenbach, F. Protti, and A.R. de Almeida. Robust recoverable perfect matchings. *Networks*, 66(3):210–213, 2015.

17 R. Fujita, Y. Kobayashi, and K. Makino. Robust matchings and matroid intersections. In *Proc. of ESA*, pages 123–134. Springer, 2010.

18 H. N. Gabow, M. X. Goemans, É. Tardos, and D. P. Williamson. Approximating the smallest k-edge connected spanning subgraph by LP-rounding. In *Proc. of SODA*, pages 562–571, 2005.

19 F. Grandoni, R. Ravi, M. Singh, and R. Zenklusen. New approaches to multi-objective optimization. *Math Program*, 146(1-2):525–554, 2014.

20 M. Grötschel, L. Lovasz, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization*. Algorithms and Combinatorics. Springer Berlin Heidelberg, 1993.

21 M. Hajiaghayi, N. Immorlica, and V.S. Mirrokni. Power optimization in fault-tolerant topology control algorithms for wireless multi-hop networks. In *Proc. of MobiCom*, pages 300–312. ACM, 2003.

22 R. Hassin and S. Rubinstein. Robust matchings. *SIAM J Discrete Math*, 15(4):530–537, 2002.

23 W. Herroelen and R. Leus. Project scheduling under uncertainty: Survey and research potentials. *Eur J Oper Res*, 165(2):289–306, 2005.

24 K. Jain. A factor 2 approximation algorithm for the generalized steiner network problem. *Combinatorica*, 21:39–60, 2001.

25 K. Jain and V.V. Vazirani. An approximation algorithm for the fault tolerant metric facility location problem. In *Proc. of APPROX*, pages 177–183, 2000.

26 P. Kouvelis and G. Yu. *Robust discrete optimization and its applications*, volume 14. Springer Science & Business Media, 1997.

27 P. Laroche, F. Marchetti, S. Martin, and Z. Roka. Bipartite complete matching vertex interdiction problem: Application to robust nurse assignment. In *Proc. of CoDIT*, pages 182–187, 2014.

28 L. Lovász and M.D. Plummer. *Matching theory*. North-Holland, Amsterdam, 1986.

29 J. Plesník. Connectivity of regular graphs and the existence of 1-factors. *Matematický časopis*, 22(4):310–318, 1972.

30 A. Sebő and J. Vygen. Shorter tours by nicer ears: 7/5-approximation for the graph-tsp, 3/2 for the path version, and 4/3 for two-edge-connected subgraphs. *Combinatorica*, 5(34):597–629, 2014.

31 C. Swamy and D. B. Shmoys. Fault-tolerant facility location. In *Proc. of SODA*, pages 735–736, 2003.

32 C.S. Tang. Robust strategies for mitigating supply chain disruptions. *International Journal of Logistics: Research and Applications*, 9(1):33–45, 2006.

33 G. Yu and J. Yang. On the robust shortest path problem. *Computers & Operations Research*, 25(6):457–468, 1998.

34 R. Zenklusen. Matching interdiction. *Discrete Appl Math*, 158(15):1676–1690, 2010.