Matching Drivers to Riders: A Two-stage Robust Approach

Omar El Housni
School of Operations Research and Information Engineering, Cornell Tech, New York, NY 10044, USA

Vineet Goyal
Industrial Engineering and Operations Research, Columbia University, New York, NY 10027, USA

Oussama Hanguir
Industrial Engineering and Operations Research, Columbia University, New York, NY 10027, USA

Clifford Stein
Industrial Engineering and Operations Research, Columbia University, New York, NY 10027, USA

Abstract

Matching demand (riders) to supply (drivers) efficiently is a fundamental problem for ride-hailing platforms who need to match the riders (almost) as soon as the request arrives with only partial knowledge about future ride requests. A myopic approach that computes an optimal matching for current requests ignoring future uncertainty can be highly sub-optimal. In this paper, we consider a two-stage robust optimization framework for this matching problem where future demand uncertainty is modeled using a set of demand scenarios (specified explicitly or implicitly). The goal is to match the current request to drivers (in the first stage) so that the cost of first stage matching and the worst-case cost over all scenarios for the second stage matching is minimized. We show that this two-stage robust matching is NP-hard under both explicit and implicit models of uncertainty. We present constant approximation algorithms for both models of uncertainty under different settings and show they improve significantly over standard greedy approaches.

2012 ACM Subject Classification Theory of computation → Approximation algorithms analysis

Keywords and phrases matching, robust optimization, approximation algorithms

Digital Object Identifier 10.4230/LIPIcs.APPROX/RANDOM.2021.12

Category APPROX

Related Version *Full Version*: `https://arxiv.org/abs/2011.03624`

Funding *Clifford Stein* : Research partly supported by NSF Grants CCF-1714818 and CCF-1822809.

1 Introduction

Matching demand (riders) with supply (drivers) is a fundamental problem for ride-hailing platforms such as Uber, Lyft and DiDi. These platforms need to continually make efficient matching decisions with only partial knowledge of future ride requests. A common approach in practice is batched matching: instead of matching each request sequentially as it arrives, aggregate the requests for a short amount of time (typically one to two minutes) and match the aggregated requests to available drivers in one batch [42, 33, 44]. However, computing this batch matching myopically without considering future requests can lead to a highly sub-optimal outcome for some subsequent drivers and riders.

Motivated by this shortcoming, and by the possibility of using historical data to hedge against future uncertainty, we study a two-stage framework for matching problems where the future demand uncertainty is modeled as a set of scenarios that are specified explicitly or implicitly. The goal is to compute a matching between the available drivers and the first batch of riders such that the total worst-case cost of first stage and second stage matching is minimized. More specifically, we consider an adversarial model of uncertainty where the adversary observes the first stage matching of our algorithms and presents a worst-case scenario from the list of specified scenarios in the second stage. We focus on the case where the first stage cost is the average weight of the first stage matching, and the second stage cost is the highest edge weight in the second stage matching. This is motivated by the goal of computing a low-cost first stage matching while also minimizing the worst case waiting time for any rider in any second stage. All the results of this paper hold when the first stage cost is the highest edge weight of the first stage matching. We also study several other metrics in the full version. We consider two common models to describe the uncertainty in the second stage: an *explicit* list of all possible scenarios and an *implicit* description of the scenarios using a cardinality constraint. Two-stage robust optimization is a popular model for hedging against uncertainty [8, 19]. Several combinatorial optimization problems have been studied in this model, including Set Cover, Capacity Planning [7, 11] and Facility Location [22]. While online matching is a classical problem in graph theory, two-stage matching problems with uncertainty, have not been studied extensively. We present related work in Section 1.2.

1.1 Our Contributions

Problem definition. We consider the following *Two-stage Robust Matching Problem*. We are given a set of drivers D, a set of first stage riders R_1, a universe of potential second stage riders R_2 and a set of second stage scenarios $\mathcal{S} \subseteq \mathcal{P}(R_2)$[1]. We are given a metric distance d on $V = R_1 \cup R_2 \cup D$. The goal is to find a subset of drivers $D_1 \subseteq D$ ($|D_1| = |R_1|$) to match all the first stage riders R_1 such that the sum of cost of first stage matching and worst-case cost of second stage matching (between $D \setminus D_1$ and the riders in the second stage scenario) is minimized. More specifically,

$$\min_{D_1 \subset D} \Big\{ cost_1(D_1, R_1) + \max_{S \in \mathcal{S}} cost_2(D \setminus D_1, S) \Big\}.$$

The first-stage decision is denoted D_1 and its cost is $cost_1(D_1, R_1)$. Similarly, the second stage cost for scenario S is denoted $cost_2(D \setminus \ D_1, S)$, and $\max\{cost_2(D \setminus D_1, S) \mid S \in \mathcal{S}\}$ is the worst-case cost over all possible scenarios. Let $|R_1| = m$, $|R_2| = n$. We denote the objective function for a feasible solution D_1 by

$$f(D_1) = cost_1(D_1, R_1) + \max_{S \in \mathcal{S}} cost_2(D \setminus D_1, S).$$

We assume that there are sufficiently many drivers to satisfy both first and second stage demand. Given an optimal first-stage solution D_1^*, we denote

$$OPT_1 = cost_1(D_1^*, R_1), \qquad OPT_2 = \max\{cost_2(D \setminus D_1^*, S) \mid S \in \mathcal{S}\},$$
$$OPT = OPT_1 + OPT_2.$$

We consider the setting where the first stage cost is the average weight of the matching between D_1 and R_1, and the second stage cost is the bottleneck matching cost between $D \setminus D_1$ and S. The bottleneck matching is the matching that minimizes the longest edge in a maximum cardinality matching between $D \setminus D_1$ and S. We refer to this variant as the *Two-Stage Robust Matching Bottleneck Problem* (**TSRMB**). Formally, let M_1 be the

[1] $\mathcal{P}(R_2)$ is the power set of R_2, the set of all subsets of R_2.

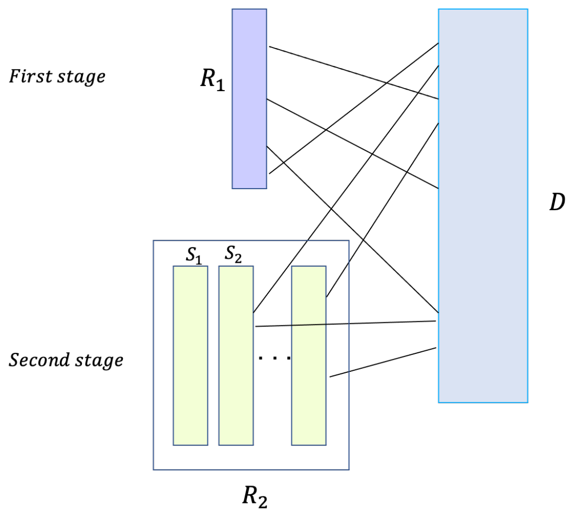

Figure 1 Bipartite graph of drivers and riders in our two-stage matching problem.

minimum weight perfect matching between R_1 and D_1, and given a scenario S, let M_2^S be the bottleneck matching between the scenario S and the available drivers $D \setminus D_1$, then the cost functions for the TSRMB are:

$$cost_1(D_1, R_1) = \frac{1}{m} \sum_{(i,j)\in M_1} d(i,j), \quad \text{and} \quad cost_2(D \setminus D_1, S) = \max_{(i,j)\in M_2^S} d(i,j).$$

The difference between the first and second stage metrics is motivated by the fact that the platform has access to the current requests and can exactly compute the cost of the matching. On the other hand, to ensure the robustness of the solution, we require all second stage assignments to have low waiting times by accounting for the maximum wait time in every scenario. We choose the first stage cost to be the average matching weight instead of the total weight for homogeneity reasons, so that first and second stage costs have comparable magnitudes. The bottleneck objective, i.e., finding a subgraph of a certain kind that minimizes the maximum edge cost in the subgraph, has been considered extensively in the literature [21, 16, 17]. While the main body of this paper will focus on studying TSRMB, we note that all our results hold when the first (resp. second) stage cost is equal to the highest edge weight in the first (resp. second) stage matching. In the full version, we study other variants of cost metrics, including a stochastic variant of TSRMB, and the case where both first and second stage costs are simply the total matching weights.

Hardness. We show that TSRMB is NP-hard even for two scenarios and NP-hard to approximate within a factor better than 2 for three scenarios. We also show that even when the scenarios are singletons, the problem is NP-hard to approximate within a factor better than 2. Given these hardness results, we focus on approximation algorithms for the TSRMB problem. A natural candidate is the greedy approach that minimizes only the first stage cost without considering the uncertainty in the second stage. However, we show that this myopic approach can be bad as $\Omega(m) \cdot OPT$ (See Figure 2.)

Approximations algorithms. We consider both explicit and implicit models of uncertainty. For the case of explicit model with two scenarios, we give a constant factor approximation algorithm for TSRMB (Theorem 4). We further generalize the ideas of this algorithm to a constant approximation for any fixed number of scenarios (Theorem 6). Our approximation

Uncertainty	Approx	Hardness
Explicit (2 scenarios)	5	NP-Hard
Explicit (p scenarios)	$O(p^{1.59})$	2
Implicit (surplus $\ell = 0$)	3	-
Implicit ($\ell < k$ and $k \le \sqrt{n/2}$)	17	2

Table 1 Summary of our results, where surplus $\ell = |D| - |R_1| - k$.

does not depend on the number of first stage riders or the size of scenarios but depends on the number of scenarios. The main idea is to reduce the problem with multiple scenarios to an instance with a single *representative scenario* while losing only a small factor. We then solve the single scenario instance (in polynomial time) to get an approximation for our original problem. The challenge in constructing the representative scenario is to find the right trade-off between capturing the demand of all second stage riders and keeping the cost of this scenario close to the optimal cost of the original instance.

For the implicit model of uncertainty, we consider the setting where we are given a universe of second stage riders R_2 and an integer k, and any subset of size less than k can be a scenario. Therefore, $\mathcal{S} = \{S \subset R_2 \text{ s.t. } |S| \le k\}$. The scenarios can be exponentially many in k, which makes even the evaluation of the cost of a feasible solution challenging and not necessarily achievable in polynomial time. Our analysis depends on the imbalance between supply and demand. In fact, when the number of drivers is very large compared to riders, the problem is less interesting in practice. However, it becomes interesting when the supply and demand are comparable. In this case, drivers might need to be shared between different scenarios. This leads us to define the notion of surplus $\ell = |D| - |R_1| - k$, which is the maximum number of drivers that we can afford not to use in a solution. As a warm-up, we first show that if the surplus is equal to zero (all the drivers are used), using any scenario as a representative scenario gives a 3-approximation. The problem becomes significantly more challenging even with a small surplus. We show that under a reasonable assumption on the size of scenarios, there is a constant approximation in the regime when the surplus ℓ is smaller than the demand k (Theorem 9). Our algorithm is based on finding a clustering of drivers and riders that yields a simplified instance of TSRMB which can be solved within a constant factor. We show that we can cluster the riders into a ball (riders close to each others) and a set of *outliers* (riders far from each others) and apply ideas from the explicit scenario analysis. Finally, since the number of scenarios can be exponential, we construct a set of a polynomial number of proxy scenarios on which we evaluate any feasible solution within a constant approximation. Table 1 summarizes our results. Due to space constraints, we defer some of the proofs to the appendix.

1.2 Related Work

Online bipartite matching. Finding a maximum cardinality bipartite matching has received a considerable amount of attention over the years. Online matching was first studied by Karp *et al.* [27] in the adversarial model. Since then, many online variants have been studied [37]. This includes AdWords [4, 5, 38], vertex-weighted [1, 6], edge-weighted [20, 31], stochastic matching [12, 35, 39, 13], random vertex arrival [18, 26, 34, 23], and batch arrivals [32, 14, 44]. In the *online bipartite metric matching* variant, servers and clients correspond to points from a metric space, and the objective is to find the minimum weight maximum cardinality matching. Khullet *et al.* [29] and Kalyanasundaram and Pruhs [24] provided deterministic algorithms in the adversarial model. In the random arrival model, Meyerson, *et al.* [40] and

Bansal *et al.* [2] provided poly-logarithmic competitive algorithms. Recently, Raghvendra [41] presented a $O(\log n)$-competitive algorithm.

Two-stage stochastic combinatorial optimization. Within two-stage stochastic optimization, matching has been studied under various models. Kong and Schaefer [30] and Escoffier *et al.* [9] studied the stochastic two-stage maximum matching problem. Katriel *et al.* [28] studied the two-stage stochastic minimum weight maximum matching. Feng and Niazadeh [14] study K-stage variants of vertex weighted bipartite b-matching and AdWords problems, where online vertices arrive in K batches. More recently, Feng *et al.* [15] initiate the study and present online competitive algorithms for vertex-weighted two-stage stochastic matching as well as two-stage joint matching and pricing.

Two-stage robust combinatorial optimization. Within two-stage robust optimization, matchings have not been studied extensively. Matuschke *et al.* proposed a two-stage robust model for minimum weight matching with recourse [36]. Our model for TSRMB is different in three main aspects: i) We use a general class of uncertainty sets to describe the second stage scenarios while in [36] the only information given is the number of second stage vertices. ii) We do not allow any recourse and our first stage matching is irrevocable. iii) Our second stage cost is the bottleneck weight instead of the total weight.

2 Preliminaries

2.1 NP-hardness.

We show that TSRMB is NP-hard under both the implicit and explicit models. In the explicit model, it is NP-hard even for two scenarios and NP-hard to approximate within a factor better than 2 even for three scenarios.

In the explicit model with a polynomial number of scenarios, it is clear that the problem is in NP. However, in the implicit model, the problem can be described with a polynomial size input, but it is not clear that we can compute the total cost in polynomial time since there could be exponentially many scenarios. We show that it is NP-hard to approximate TSRMB in the implicit model within a factor better than 2 even when $k = 1$. The proof is presented in Appendix A.

▶ **Theorem 1.** *In the explicit model of uncertainty, TSRMB is NP-hard even with two scenarios. Furthermore, when the number of scenarios is ≥ 3, there is no $(2-\epsilon)$-approximation algorithm for any fixed $\epsilon > 0$, unless $P = NP$. In the implicit model of uncertainty, even when $k = 1$, there is no $(2-\epsilon)$-approximation algorithm for TSRMB for any fixed $\epsilon > 0$, unless $P = NP$.*

2.2 Greedy Approach.

A natural greedy approach is to choose the optimal matching for the first stage riders R_1 without considering the second stage uncertainty. It can lead to a solution with a total cost that scales linearly with m (cardinality of R_1) while OPT is a constant, even with one scenario. Consider the line example in Figure 2. We have m first stage riders and $m+1$ drivers alternating on a line with distances 1 and $1-\epsilon$. There is one second stage rider at the right endpoint of the line. The greedy matching minimizes the first stage cost and incurs a total cost of $(2-\epsilon)(m+1)$, while the optimal cost is equal to 2. Therefore any attempt to have a good approximation needs to consider the second stage riders.

▶ **Lemma 2.** *The cost of the Greedy algorithm can be $\Omega(m) \cdot OPT$.*

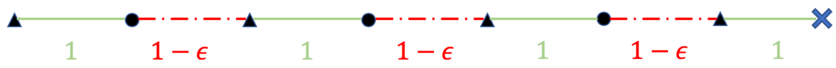

Figure 2 Riders in first stage are depicted as black dots and drivers as black triangles. The second stage rider is depicted as a blue cross.

2.3 Single Scenario.

The *deterministic* version of the TSRMB problem, i.e., when there is only a single scenario in the second stage, can be solved exactly in polynomial time. This is a simple preliminary result which we need for the general case. Denote S a single second stage scenario. The instance (R_1, S, D) of TSRMB is then simply given by

$$\min_{D_1 \subset D} \left\{ cost_1(D_1, R_1) + cost_2(D \setminus D_1, S) \right\} .$$

Since the second stage problem is a bottleneck problem [21], the value of the optimal second stage cost w is one of the edge weights between D and S. We iterate over all possible values of w (at most $|S| \cdot |D|$ values), delete all edges between R_2 and D with weights strictly higher than w and set the weight of the remaining edges between S and D to zero. This reduces the problem to finding a minimum weight maximum cardinality matching. We can also use binary search to iterate over the edge weights. We present the details of this algorithm below and refer to it as *TSRMB-1-Scenario* in the rest of this paper.

We define the bottleneck graph of w to be $BOTTLENECKG(w) = (R_1 \cup S \cup D, E_1 \cup E_2)$ where $E_2 = \{(i,j) \in D \times S,\ d(i,j) \leq w\}$ and $E_1 = \{(i,j) \in D \times R_1\}$. Furthermore, we assume that there are q edges $\{e_1, \dots, e_q\}$ between S and D with weights $w_1 \leq w_2 \leq \dots \leq w_q$.

Algorithm 1 TSRMB-1-Scenario(R_1, S, D)

Input: First stage riders R_1, scenario S and drivers D.
Output: First stage decision D_1.

```
for i ∈ {1, ..., q} do
    G_i := BOTTLENECKG(w_i).
    Set all weights between D and S in G_i to be 0.
    M_i := minimum weight maximum cardinality matching on G_i.
    if R_1 ∪ S is not completely matched in M_i then
        output certificate of failure.
    else
        D_1^i := first stage drivers in M_i.
    end if
end for
return D_1 = argmin_{D_1^i : 1 ≤ i ≤ q} { cost_1(D_1^i, R_1) + cost_2(D \ D_1^i, S) }.
```

Note that the arg min in the last step of Algorithm 1 is only taken over values of i for which there was no certificate of failure.

▶ **Lemma 3.** *TSRMB-1-Scenario gives an optimal solution for the single scenario case.*

Proof of Lemma 3. Let OPT_1 and OPT_2 be the first and second stage cost of an optimal solution, and $i \in \{1, \dots, q\}$ such that $w_i = OPT_2$. In this case, G_i contains all the edges of this optimal solution. By setting all the edges in E_2 to 0, we are able to compute a minimum weight maximum cardinality matching between $R_1 \cup S$ and D that matches both R_1 and S and minimizes the weight of the edges matching R_1. The first stage cost of this matching is less than OPT_1, the second stage cost is clearly less than OPT_2 because we only allowed edges with weight less than OPT_2 in G_i. ◀

We also observe that we can use binary search in Algorithm 1 to iterate over the edge weights. For an iteration i, a failure to find a minimum weight maximum cardinality matching on G_i that matches both R_1 and S implies that we need to try an edge weight higher than w_i. On the other hand, if M_i matches R_1 and S such that D_1^i gives a smaller total cost, then the optimal bottleneck value is lower than w_i.

3 Explicit Scenarios

3.1 Two scenarios

Our main contribution in this section is a constant approximation algorithm for TSRMB with two scenarios. Our analysis shows that we can reduce the problem to an instance with a single representative scenario by losing a small factor. We then use TSRMB-1-Scenario to solve the single representative scenario case.

Consider two scenarios $\mathcal{S} = \{S_1, S_2\}$. First, we can assume without loss of generality that we know the exact value of OPT_2 which corresponds to one of the edges connecting second stage riders R_2 to drivers D (we can iterate over all the weights of second stage edges). We construct a representative scenario that serves as a proxy for S_1 and S_2 as follows. In the second stage, if a pair of riders $i \in S_1$ and $j \in S_2$ is served by the same driver in the optimal solution, then they should be close to each other. Therefore, we can consider a single representative rider for each such pair. While it is not easy to guess all such pairs, we can approximately compute the representative riders by solving a maximum matching on $S_1 \cup S_2$ with edges less than $2OPT_2$. More formally, let G_I be the induced bipartite subgraph of G on $S_1 \cup S_2$ containing only edges between S_1 and S_2 with weight less than or equal to $2OPT_2$. We compute a maximum cardinality matching M between S_1 and S_2 in G_I, and construct a representative scenario containing S_1 as well as the unmatched riders of S_2. We solve the single scenario problem on this representative scenario and return its optimal first stage solution. We show in Theorem 4 that this solution leads to a 5-approximation.

Algorithm 2 Two explicit scenarios.

Input: First stage riders R_1, two scenarios S_1 and S_2, drivers D and value of OPT_2.
Output: First stage decision D_1.
1: Let G_I be the induced subgraph of G on $S_1 \cup S_2$ with only the edges between S_1 and S_2 of weights less than $2OPT_2$.
2: Set $M :=$ maximum cardinality matching between S_1 and S_2 in G_I.
3: Set $S_2^{Match} := \{r \in S_2 \mid \exists\, s \in S_1 \text{ s.t } (s, r) \in M\}$ and $S_2^{Unmatch} = S_2 \setminus S_2^{Match}$.
4: **return** $D_1 :=$ TSRMB-1-Scenario$(R_1, S_1 \cup S_2^{Unmatch}, D)$.

▶ **Theorem 4.** *Algorithm 2 yields a solution with total cost less than $OPT_1 + 5OPT_2$ for TSRMB with 2 scenarios.*

The proof of Theorem 4 relies on the following structural lemma where we show that the set D_1 returned by Algorithm 2 yields a total cost at most $(OPT_1 + 3OPT_2)$ when evaluated only on the single representative scenario $S_1 \cup S_2^{Unmatch}$.

▶ **Lemma 5.** *Let D_1 be the set of first stage drivers returned by Algorithm 2. Then $cost_1(D_1, R_1) + cost_2(D \setminus D_1, S_1 \cup S_2^{Unmatch}) \leq OPT_1 + 3OPT_2$.*

Proof. It is sufficient to show the existence of a matching M_a between $R_1 \cup S_1 \cup S_2^{Unmatch}$ and D with a total cost less than $OPT_1 + 3OPT_2$. This would imply that the optimal solution D_1 of TSRMB-1-Scenario($R_1, S_1 \cup S_2^{Unmatch}, D$) has a total cost less than $OPT_1 + 3OPT_2$ and concludes the proof. We show the existence of M_a by construction.

Step 1. We first match R_1 with their mates in the optimal solution of TSRMB. Hence, the first stage cost of our constructed matching M_a is OPT_1.

Step 2. Now, we focus on $S_2^{Unmatch}$. Let $S_2^{Unmatch} = S_{12} \cup S_{22}$ be a partition of $S_2^{Unmatch}$ where S_{12} contains riders with a distance less than $2OPT_2$ from S_1 and S_{22} contains riders with a distance strictly bigger than $2OPT_2$ from S_1, where the distance from a set is the minimum distance to any element of the set. A rider in S_{22} cannot share any driver with a rider from S_1 in the optimal solution of TSRMB, because otherwise, the distance between these riders will be less than $2OPT_2$ by using the triangle inequality. Therefore we can match S_{22} to their mates in the optimal solution and add them to M_a, without using the optimal drivers of S_1. We pay less than OPT_2 for matching S_{22}.

Step 3. We still need to simultaneously match riders in S_1 and S_{12} to finish the construction of M_a. Notice that some riders in S_{12} might share their optimal drivers with riders in S_1. We can assume without loss of generality that all riders in S_{12} share their optimal drivers with S_1 (otherwise we can match them to their optimal drivers without affecting S_1). Denote $S_{12} = \{r_1, \dots, r_q\}$ and $S_1 = \{s_1, \dots, s_k\}$. For each $i \in [q]$ let's say $s_i \in S_1$ is the rider that shares its optimal driver with r_i. We show that $q \leq |M|$. In fact, every rider in S_{12} shares its optimal driver with a different rider in S_1, and is therefore within a distance $2OPT_2$ from S_1 by the triangle inequality. But since S_{12} is not covered by the maximum cardinality matching M, this implies by the maximality of M that there are q other riders from S_2^{Match} that are covered by M. Hence $q \leq |M|$. Finally, let $\{t_1, \dots, t_q\} \subset S_2^{Match}$ be the mates of $\{s_1, \dots, s_q\}$ in M, i.e., $(s_i, t_i) \in M$ for all $i \in [q]$. Recall that $d(s_i, t_i) \leq 2OPT_2$ for all $i \in [q]$. In what follows, we describe how to match S_{12} and S_1:

(i) For $i \in [q]$, we match r_i to its optimal driver and s_i to the optimal driver of t_i. This is possible because the optimal driver of t_i cannot be the same as the optimal driver of r_i since both r_i and t_i are part of the same scenario S_2. Therefore, we pay a cost OPT_2 for the riders r_i and a cost $3OPT_2$ (follows from the triangle inequality) for the riders s_i where $i \in [q]$.

(ii) We still need to match $\{s_{q+1}, \dots, s_k\}$. Consider a rider s_j with $j \in \{q+1, \dots, k\}$. If the optimal driver of s_j is not shared with any $t_i \in \{t_1, \dots, t_q\}$, then this optimal driver is still available and can be matched to s_j with a cost less than OPT_2. If the optimal driver of s_j is shared with some $t_i \in \{t_1, \dots t_q\}$, then s_j is also covered by M. Otherwise M can be augmented by deleting (s_i, t_i) and adding (r_i, s_i) and (s_j, t_i). Therefore s_j is covered by M and has a mate $\tilde{t}_j \in S_2^{Match} \setminus \{t_1, \dots, t_q\}$. Furthermore, the driver assigned to $\tilde{t}_j$ is still available. We can then match s_j to the optimal driver of $\tilde{t}_j$. Similarly if the optimal driver of some $s_{j'} \in \{s_{q+1}, \dots, s_k\} \setminus \{s_j\}$ is shared with $\tilde{t}_j$, then $s_{j'}$ is covered by M. Otherwise $(r_i, s_i, t_i, s_j, \tilde{t}_j, s_{j'})$ is an augmenting path in M. Therefore $s_{j'}$ has a mate in M and we can match $s_{j'}$ to the optimal driver of its mate. We keep extending these augmenting paths until all the riders in $\{s_{q+1}, \dots, s_k\}$ are matched. Furthermore, the augmenting paths $(r_i, s_i, t_i, s_j, \tilde{t}_j, s_{j'} \dots)$ starting from two different riders $r_i \in S_{12}$ are vertex disjoint. This ensures that every driver is used at most once. Again, by the triangle inequality, the edges that match $\{s_{q+1}, \dots, s_k\}$ in our solution have weights less then $3OPT_2$.

Putting it all together, we have constructed a matching M_a where the first stage cost is exactly OPT_1 and the second-stage cost is less than $3OPT_2$ since the edges used for matching

$S_1 \cup S_2^{Unmatch}$ in M_a have a weight less than $3OPT_2$. Therefore, the total cost of M_a is less than $OPT_1 + 3OPT_2$. ◀

Proof of Theorem 4. Let D_1 be the drivers returned by Algorithm 2. Lemma 5 implies

$$cost_1(D_1, R_1) + cost_2(D \setminus D_1, S_1) \leq \; OPT_1 + 3OPT_2 \tag{1}$$

and

$$cost_1(D_1, R_1) + cost_2(D \setminus D_1, S_2^{Unmatch}) \leq \; OPT_1 + 3OPT_2.$$

We have $S_2 = S_2^{Match} \cup S_2^{Unmatch}$. If the scenario S_2 is realized, we use the drivers that were assigned to S_1 in the matching constructed in Lemma 5 to match S_2^{Match}. This is possible with edges of weights less than $cost_2(D \setminus D_1, S_1) + 2OPT_2$ because S_2^{Match} is matched to S_1 with edges of weight less than $2OPT_2$. Hence,

$$cost_2(D \setminus D_1, S_2) \leq \max \left\{ cost_2(D \setminus D_1, S_2^{Unmatch}),\; cost_2(D \setminus D_1, S_1) + 2OPT_2 \right\},$$

and therefore

$$cost_1(D_1, R_1) + cost_2(D \setminus D_1, S_2) \leq OPT_1 + 5OPT_2. \tag{2}$$

From (1) and (2), $cost_1(D_1, R_1) + \max\limits_{S \in \{S_1, S_2\}} cost_2(D \setminus D_1, S) \leq OPT_1 + 5OPT_2$. ◀

Algorithm 3 p explicit scenarios.

Input: First-stage riders R_1, scenarios $\{S_1, S_2, \dots, S_p\}$, drivers D and value of OPT_2.
Output: First stage decision D_1.

Initialize $\hat{S}_j := S_j$ for $j = 1, \dots,\ p$.
for $i = 1, \dots,\ \log_2 p$ **do**
  for $j = 1, 2, \dots,\ \frac{p}{2^i}$ **do**
    $\sigma(j) = j + \frac{p}{2^i}$
    $M_j :=$ maximum cardinality matching between $\hat{S}_j$ and $\hat{S}_{\sigma(j)}$ with edges of weight less than $2 \cdot 3^{i-1} \cdot OPT_2$.
    $\hat{S}_{\sigma(j)}^{Match} := \{ r \in \hat{S}_{\sigma(j)} \mid \exists\, s \in \hat{S}_j \;$ s.t $(s, r) \in M_j \}$.
    $\hat{S}_{\sigma(j)}^{Unmatch} := \hat{S}_{\sigma(j)} \setminus \hat{S}_{\sigma(j)}^{Match}$
    $\hat{S}_j = \hat{S}_j \cup \hat{S}_{\sigma(j)}^{Unmatch}$.
  end for
end for
return $D_1 :=$ TSRMB-1-Scenario$(R_1, \hat{S}_1, D)$.

3.2 Constant number of scenarios

We now consider the case of explicit list of p scenarios, i.e., $\mathcal{S} = \{S_1, S_2, \dots, S_p\}$. Building upon the ideas from Algorithm 2, we present a $O(p^{1.59})$-approximation in this case. The idea is to construct the representative scenario recursively by processing pairs of "scenarios" at each step. Hence, we need $O(\log_2 p)$ iterations to reduce the problem to an instance of a single scenario. At each iteration, we show that we only lose a multiplicative factor of 3 so that the final approximation ratio is $O(3^{\log_2 p}) = O(p^{1.59})$. We present details in Algorithm 3.

The approximation guarantee of our algorithm grows sub-quadratically with p and it is an interesting question if there exists an approximation that does not depend on the number of scenarios.

▶ **Theorem 6.** *Algorithm 3 yields a solution with total cost of $O(p^{1.59}) \cdot OPT$ for TSRMB with an explicit list of p scenarios.*

Proof of Theorem 6. The algorithm reduces the number of considered "scenarios" by half in every iteration, until only one scenario remains. In iteration i, we have $\frac{p}{2^{i-1}}$ scenarios that we aggregate in $\frac{p}{2^i}$ pairs, namely $(\hat{S}_j\ ,\ \hat{S}_{\sigma(j)})$ for $j \in \{1, 2, \dots, \frac{p}{2^i}\}$. For each pair, we construct a single representative scenario which plays the role of the new $\hat{S}_j$ at the start of the next iteration $i+1$.

Claim. There exists a first stage decision D_1^*, such that at every iteration $i \in \{1, \dots, \log_2 p\}$, we have for all $j \in \{1, 2, \dots, \frac{p}{2^i}\}$:

(i) R_1 can be matched to D_1^* with a first stage cost of OPT_1.

(ii) $\hat{S}_j \cup \hat{S}_{\sigma(j)}^{Unmatch}$ can be matched to $D \setminus D_1^*$ with a second stage cost less than $3^i \cdot OPT_2$.

(iii) There exists a matching between $\hat{S}_{\sigma(j)}^{Match}$ and $\hat{S}_j$ with edge weights less than $2 \cdot 3^{i-1} \cdot OPT_2$.

Proof of the claim. Statement (iii) follows from the definition of $\hat{S}_{\sigma(j)}^{Match}$ in Algorithm 3. Let's show (i) and (ii) by induction over i.

- **Initialization:** for $i = 1$, let's take any two scenarios $\hat{S}_j = S_j$ and $\hat{S}_{\sigma(j)} = S_{\sigma(j)}$. We know that these two scenarios can be matched to drivers of the optimal solution in the original problem with a cost less than OPT_2. In the proof of Lemma 5, we show that if we use the optimal first stage decision D_1^* of the original problem, then we can match $\hat{S}_j$ and $\hat{S}_{\sigma(j)}^{Unmatch}$ simultaneously to $D \setminus D_1^*$ with a cost less than $3OPT_2$.
- **Maintenance.** Assume the claim is true for all values less than $i \leq \log_2 p - 1$. We show it is true for $i+1$. Since the claim is true for iteration i, we know that at the start of iteration $i+1$, for $j \in \{1, \dots, \frac{p}{2^i}\}$, $\hat{S}_j$ can be matched to $D \setminus D_1^*$ with a cost less than $3^i \cdot OPT_2$. We can therefore consider a new TSRMB problem with $\frac{p}{2^i}$ scenarios, where using D_1^* as a first stage decision ensures a second stage optimal value less than $\widehat{OPT}_2 = 3^i \cdot OPT_2$. By the proof of Lemma 5, and by using D_1^* as a first stage decision in this problem, we ensure that for $j \in \{1, \dots, \frac{p}{2^{i+1}}\}$, $\hat{S}_j$ and $\hat{S}_{\sigma(j)}^{Unmatch}$ can be simultaneously matched to $D \setminus D_1^*$ with a cost less than $3\widehat{OPT}_2 = 3^{i+1} \cdot OPT_2$. ◀

Our claim implies that in the last iteration $i = \log_2 p$:

- R_1 can be matched to D_1^* with a first stage cost of OPT_1.
- $\hat{S}_1$ can be matched to $D \setminus D_1^*$ with a second stage cost less than $3^{\log_2 p} \cdot OPT_2$.

Computing the single scenario solution for $\hat{S}_1$ will therefore yield a first stage decision D_1 that gives a total cost less than $OPT_1 + 3^{\log_2 p} \cdot OPT_2$ when the second stage is evaluated on the scenario $\hat{S}_1$. We now bound the cost of D_1 on the original scenarios $\{S_1, \dots,\ S_p\}$. Consider a scenario $S \in \{S_1, \dots,\ S_p\}$. The riders in $S \cap \hat{S}_1$ can be matched to some drivers in $D \setminus D_1$ with a cost less than $OPT_1 + 3^{\log_2 p} \cdot OPT_2$. As for other riders of $S \setminus \hat{S}_1$, they are not part of $\hat{S}_1$ because they have been matched and deleted at some iteration $i < \log_2 p$. Consider riders r in $S \setminus \hat{S}_1$ that were matched and deleted from a representative scenario at some iteration, then by statement (iii) in our claim, each r can be connected to a different rider in $\hat{S}_1 \setminus (\hat{S}_1 \cap S)$ within a path of length at most

$$\sum_{t=1}^{\log_2 p} 2 \cdot 3^{t-1} \cdot OPT_2 = (3^{\log_2 p} - 1) \cdot OPT_2.$$

We know that R_1 and $\hat{S}_1$ can be matched respectively to D_1 and $D \setminus D_1$ with a total cost less than $OPT_1 + 3^{\log_2 p} \cdot OPT_2$. Therefore, we can match R_1 and S respectively to D_1 and $D \setminus D_1$ with a total cost less than

$$OPT_1 + 3^{\log_2 p} \cdot OPT_2 + (3^{\log_2 p} - 1) \cdot OPT_2 = O(3^{\log_2 p}) \cdot OPT \simeq O(p^{1.59}) \cdot OPT.$$

Therefore, the worst-case total cost of the solution returned by Algorithm 3 is $O(p^{1.59}) \cdot OPT$. ◀

4 Implicit Scenarios

Consider an implicit model of scenarios $\mathcal{S} = \{S \subset R_2 \text{ s.t. } |S| \leq k\}$. While this model is widely used, it poses a challenge because the number of scenarios can be exponential. Therefore, even computing the worst-case second stage cost, for a given first stage solution, might not be possible in polynomial time and we can no longer assume that we can guess OPT_2. Note that the worst-case scenarios have size exactly k. Our analysis for this model depends on the balance between supply (drivers) and demand (riders). We define the surplus ℓ as the excess in the number of available drivers for matching first-stage riders and a second-stage scenario: $\ell = |D| - |R_1| - k$. As a warm-up, we study the case of no surplus ($\ell = 0$). Then, we address the more general case with a small surplus of drivers.

4.1 Warm-up: no surplus

When the number of drivers equals the number of first stage riders plus the size of scenarios (i.e., $\ell = 0$), we show a 3-approximation by simply solving a single scenario TSRMB with any of the scenarios. In fact, since $\ell = 0$, all scenarios are matched to the same set of drivers in the optimal solution. Hence, between any two scenarios, there exists a matching where all edge weights are less than $2OPT_2$. So by solving TSRMB with only one of these scenarios, we can recover a solution and bound the cost of the other scenarios within $OPT_1 + 3OPT_2$ using the triangle inequality. The algorithm and proof are presented below.

Algorithm 4 Implicit scenarios with no surplus.

Input: First stage riders R_1, second stage riders R_2, size k and drivers D.
Output: First stage decision D_1.
1: $S_1 :=$ a second stage scenario of size k.
2: $D_1 :=$ TSRMB-1-Scenario(R_1, S_1, D).
3: **return** D_1.

▶ **Lemma 7.** *Algorithm 4 yields a solution with total cost less than $OPT_1 + 3OPT_2$ for TSRMB with implicit scenarios and no surplus.*

Proof of Lemma 7. Let OPT_1 and OPT_2 be the first and second stage cost of the optimal solution. Let $f(D_1)$ be the total cost of the solution returned by the algorithm. We claim that $f(D_1) \leq OPT_1 + 3OPT_2$. It is clear that $cost_1(D_1, R_1) + cost_2(D \setminus D_1, S_1) \leq OPT_1 + OPT_2$. Let $S \in \mathcal{S}$ be another scenario. Because $|D| = |R_1| + k$, the optimal solution uses exactly the same k drivers to match all the second stage scenarios. This implies that we can use the triangular inequality to find a matching between S and S_1 of bottleneck cost less than $2OPT_2$. Hence for any scenario S,

$$\begin{aligned} cost_1(D_1, R_1) + cost_2(D \setminus D_1, S) &\leq cost_1(D_1, R_1) + cost_2(D \setminus D_1, S_1) + 2OPT_2 \\ &\leq OPT_1 + 3OPT_2. \end{aligned}$$

◀

If the surplus is strictly greater than 0, the above procedure can have an approximation ratio of $\Omega(m)$. Consider the example in Figure 3, with $k = 1$ and two second stage riders. The single scenario solution for S_1 uses the optimal second stage driver of S_2. Hence, if S_2 is realized, the cost of matching S_2 to the closest available driver is $\Omega(m)$. Similarly, the single scenario problem for S_2 yields a $\Omega(m)$ cost for S_1.

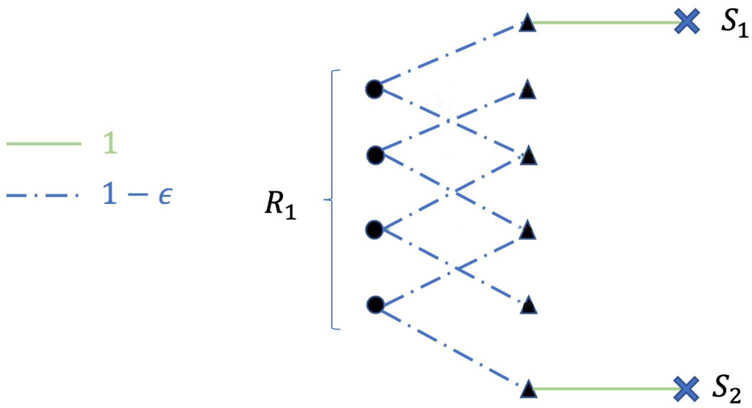

Figure 3 First stage riders are depicted as black dots and drivers as black triangles. The two second stage riders are depicted as blue crosses. Second stage optimum are depicted as solid green edges. $\mathcal{S} = \{S_1, S_2\}$, $k = 1$ and $\ell = 1$.

4.2 Small surplus

The TSRMB problem becomes challenging even with a unit surplus. Motivated by this, we focus on the case of a small surplus ℓ. In particular, we assume that $\ell < k$, i.e., the excess in the total available drivers is smaller than the size of any scenario. We present a constant approximation algorithm in this regime for the implicit model of uncertainty where the size of scenarios is relatively small with respect to the size of the universe ($k = O(\sqrt{n})$). This technical assumption is needed for our analysis but it is not too restrictive and still captures the regime where the number of scenarios can be exponential. Our algorithm attempts to cluster the second stage riders in different groups (a *ball* and a set of *outliers*) in order to reduce the number of possible worst-case configurations. We then solve a sequence of instances with representative riders from each group. In what follows, we present our construction for these groups of riders.

Our construction. First, we show that many riders are contained in a ball with radius $3OPT_2$. The center of this ball, δ, can be found by selecting the driver with the least maximum distance to its closest k second-stage riders, i.e.,

$$\delta = \arg\min_{\delta' \in D} \max_{r \in R_k(\delta')} d(\delta', r), \tag{3}$$

where $R_k(\delta')$ is the set of the k closest second stage riders to δ'. Formally, we have the following lemma. We present the proof in Appendix B.

▶ **Lemma 8.** *Suppose $k \leq \sqrt{\frac{n}{2}}$ and $\ell < k$ and let δ be the driver given by (3). Then, the ball $\mathcal{B}$ centered at δ with radius $3OPT_2$ contains at least $n - \ell$ second stage riders. Moreover, the distance between any of these riders and any rider in $R_k(\delta)$ is less than $4OPT_2$.*

Now, we focus on the rest of second stage riders. We say that a rider $r \in R_2$ is an *outlier* if $d(\delta, r) > 3OPT_2$. Denote $\{o_1, o_2, \dots, o_\ell\}$ the farthest ℓ riders from δ with $d(\delta, o_1) \geq d(\delta, o_2) \geq \dots \geq d(\delta, o_\ell)$. By Lemma 8, the $n - \ell$ riders in $\mathcal{B}$ are not outliers and the only potential outliers can be in $\{o_1, o_2, \dots, o_\ell\}$. Let j^* be the threshold such that $o_1, o_2, \dots, o_{j^*}$ are outliers and $o_{j^*+1}, \dots, o_\ell$ are not, with the convention that $j^* = 0$ if there is no outlier. There are $\ell + 1$ possible values for j^*. We call each of these possibilities a *configuration*. For $j = 0, \dots, \ell$, let C_j be the configuration corresponding to threshold candidate j. C_0 is the configuration where there is no outlier and C_{j^*} is the correct configuration.

Algorithm 5 Implicit scenarios with small surplus and $k \leq \sqrt{\frac{n}{2}}$.

Input: First stage riders R_1, second stage riders R_2, size k and drivers D.
Output: First stage decision D_1.

1: Set $\delta :=$ driver given by (3).
2: Set $S_1 :=$ the closest k second stage riders to δ.
3: Set $S_2 := \{o_1, \dots, o_\ell\}$ the farthest ℓ second stage riders from δ (o_1 being the farthest).
4: **for** $j = 0, \dots, \ell$ **do**
5: $\quad D_1(j) :=$ TSRMB-1-Scenario$(R_1, S_1 \cup \{o_1 \dots o_j\}, D)$.
6: **end for**
7: **return** $D_1 = \underset{D_1(j):\ j \in \{0,\dots,\ell\}}{\arg\min}\ cost_1\big(D_1(j), R_1\big) + \max_{S \in \{S_1, S_2\}} cost_2\big(D \setminus D_1(j), S\big)$.

Recall that $R_k(\delta)$ are the closest k second-stage riders to δ. For the sake of simplicity, we denote $S_1 = R_k(\delta)$ and $S_2 = \{o_1 \dots o_\ell\}$. S_2 is a feasible scenario since $\ell < k$. For every configuration C_j, we form a representative scenario using S_1 and $\{o_1 \dots o_j\}$. We solve TSRMB with this single representative scenario $S_1 \cup \{o_1 \dots o_j\}$ and denote $D_1(j)$ the corresponding optimal solution, i.e.,

$$D_1(j) = \text{TSRMB-1-Scenario}(R_1, S_1 \cup \{o_1 \dots o_j\}, D).$$

Since we can not evaluate the cost of $D_1(j)$ on all scenarios, we use the two proxy scenarios S_1 and S_2. We show that the candidate $D_1(j)$ with minimum cost over S_1 and S_2 gives a constant approximation to our original problem. The details are presented in Algorithm 5. We state the result in the next theorem.

▶ **Theorem 9.** *Algorithm 5 yields a solution with total cost less than $3OPT_1 + 17OPT_2$ for TSRMB with implicit scenarios when $k \leq \sqrt{\frac{n}{2}}$ and $\ell < k$.*

Before proving the theorem, we first introduce some notation. For all $j \in \{0, \dots, \ell\}$, denote

$$\Omega_j = cost_1\big(D_1(j), R_1\big)$$
$$\Delta_j = cost_2\big(D \setminus D_1(j), S_1 \cup \{o_1, \dots, o_j\}\big)$$
$$\beta_j = cost_1\big(D_1(j), R_1\big) + \max_{S \in \{S_1, S_2\}} cost_2\big(D \setminus D_1(j), S\big)$$

Recall that f the objective function of TSRMB. In particular,

$$f\big(D_1(j)\big) = cost_1\big(D_1(j), R_1\big) + \max_{S \in \mathcal{S}} cost_2\big(D \setminus D_1(j), S\big)$$

Our proof is based on the following two claims. Claim 10 establishes a bound on the cost of $D_1(j^*)$ when evaluated on the proxy scenarios S_1 and S_2 and on all the scenarios in $\mathcal{S}$.

Recall that j^* is the threshold index for the outliers as defined earlier in our construction. Claim 11 bounds the cost of $f(D_1(j))$ for any j.

▷ Claim 10. $\Omega_{j^*} + \Delta_{j^*} \leq OPT_1 + OPT_2$. and $f(D_1(j^*)) \leq OPT_1 + 5OPT_2$.

Proof of Claim 10.

1. In the optimal solution of the original problem, R_1 is matched to a subset D_1^* of drivers. The scenario S_1 is matched to a set of drivers D_{S_1} where $D_1^* \cap D_{S_1} = \emptyset$. Let D_o be the set of drivers that are matched to $o_1, \ldots, o_j^*$ in a scenario that contains $o_1, \ldots, o_j^*$. It is clear that $D_1^* \cap D_o = \emptyset$. We claim that $D_o \cap D_{S_1} = \emptyset$. In fact, suppose there is a driver $\rho \in D_o \cap D_{S_1}$. This implies the existence of some o_j with $j \leq j^*$ and some rider $r \in S_1$ such that $d(\rho, o_j) \leq OPT_2$ and $d(\rho, r) \leq OPT_2$. But then $d(\delta, o_j) \leq d(\delta, r) + d(\rho, r) + d(\rho, o_j) \leq 3OPT_2$ which contradicts the fact the o_j is an outlier. Therefore $D_o \cap D_{S_1} = \emptyset$. We show that D_1^* is a feasible first stage solution to the single scenario problem of $S_1 \cup \{o_1, \ldots o_j^*\}$ with a cost less than $OPT_1 + OPT_2$. In fact, D_1^* can be matched to R_1 with a cost less than OPT_1, D_{S_1} to S_1 and D_o to $\{o_1, \ldots, o_j^*\}$ with a cost less than OPT_2. Therefore $\Omega_{j^*} + \Delta_{j^*} \leq OPT_1 + OPT_2$.
2. Recall that $cost_1\big(D_1(j^*), R_1\big) = \Omega_{j^*}$. Consider a scenario S and a rider $r \in S$. Let $\mathcal{B}'$ be the set of the $n - \ell$ closest second stage riders to δ. Let $D_{S_1}(j^*)$ be set of second stage drivers matched to S_1 in the single scenario problem for scenario $S_1 \cup \{o_1, \ldots, o_{j^*}\}$. Let $D_o(j^*)$ be the set of second stage drivers matched to $\{o_1, \ldots, o_{j^*}\}$ in the single scenario problem for scenario $S_1 \cup \{o_1, \ldots, o_{j^*}\}$. Recall that the second stage cost for this single scenario problem is Δ_{j^*}. We distinguish three cases:
   a. If $r \in \mathcal{B}'$, then by Lemma 8, r is connected to every driver in $D_{S_1}(j^*)$ within a distance less than $\Delta_{j^*} + 4OPT_2$.
   b. If $r \in \{o_{j^*+1}, \ldots, o_\ell\}$, then r is connected to every driver in $D_{S_1}(j^*)$ within a distance less than $3OPT_2 + OPT_2 + \Delta_j^*$.
   c. If $r \in \{o_1, \ldots, o_{j^*}\}$ (i.e., r an outlier), then r can be matched to a different driver in $D_o(j^*)$ within a distance less than OPT_2.

   This means that in every case, we can match r to a driver in $D \setminus D_1(j^*)$ with a cost less than $4OPT_2 + \Delta_{j^*}$. This implies that

$$\max_{S \in \mathcal{S}} cost_2\big(D \setminus D_1(j^*), S\big) \leq 4OPT_2 + \Delta_{j^*}$$

   and therefore

$$\Omega_{j^*} + \max_{S \in \mathcal{S}} cost_2\big(D \setminus D_1(j^*), S\big) \leq \Omega_{j^*} + \Delta_{j^*} + 4OPT_2 \leq OPT_1 + 5OPT_2.$$

◀

▷ Claim 11. For all $j \in \{0, \ldots, l\}$ we have, $\beta_j \leq f(D_1(j)) \leq \max\{\beta_j + 4OPT_2,\ 3\beta_j + 2OPT_2\}$.

Proof of Claim 11. Let α_j be the second stage cost of $D_1(j)$ on the TSRBM instance with scenarios S_1 and S_2. Formally, $\alpha_j = \max_{S \in \{S_1, S_2\}} cost_2\big(D \setminus D_1(j), S\big)$. Therefore $\beta_j = \Omega_j + \alpha_j$. Let's consider the two sets

$$O_1 = \{r \in \{o_1, \ldots, o_\ell\} \mid d(r, \delta) > 2\alpha_j + OPT_2\}.$$
$$O_2 = \{o_1, \ldots, o_\ell\} \setminus O_1.$$

Consider $D_1(j)$ as a first stage decision to TSRMB with scenarios S_1 and S_2. Let $\tilde{D}_1 \subset D \setminus D_1(j)$ be the set of drivers that are matched to O_1 when the scenario $S_2 = \{o_1, \ldots, o_\ell\}$

is realized. Similarly, let $\tilde{D}_2 \subset D \setminus D_1(j)$ be the drivers matched to scenario S_1. We claim that $\tilde{D}_1 \cap \tilde{D}_2 = \emptyset$. Suppose that there exists some driver $\rho \in \tilde{D}_1 \cap \tilde{D}_2$, this implies the existence of some $o \in O_1$ and $r \in S_1$ such that $d(\rho, o) \leq \alpha_j$ and $d(\rho, r) \leq \alpha_j$. And since $d(r, \delta) \leq OPT_2$ by definition of δ we would have

$$d(o, \delta) \leq d(\rho, o) + d(\rho, r) + d(r, \delta) \leq 2\alpha_j + OPT_2,$$

which contradicts the definition of O_1. Therefore $\tilde{D}_1 \cap \tilde{D}_2 = \emptyset$.

Now consider a scenario $S \in \mathcal{S}$. The riders of $S \cap O_1$ can be matched to $\tilde{D}_1$ with a bottleneck cost less than α_j. Recall that by Lemma 8, any rider in $R_2 \setminus \{o_1, \ldots, o_\ell\}$ is within a distance less than $4OPT_2$ from any rider in S_1. The riders $r \in S \setminus \{o_1, \ldots, o_\ell\}$ can therefore be matched to any driver $\rho \in \tilde{D}_2$ within a distance less than

$$d(r, \rho) \leq d(r, S_1) + d(S_1, \rho) \leq 4OPT_2 + \alpha_j.$$

As for riders $r \in S \cap O_2$, they can also be matched to any driver ρ of $\tilde{D}_2$ within a distance less than

$$d(r, \rho) \leq d(r, \delta) + d(\delta, S_1) + d(S_1, \rho) \leq 2\alpha_j + OPT_2 + OPT_2 + \alpha_j = 3\alpha_j + 2OPT_2.$$

Therefore we can bound the second stage cost

$$\max_{S \in \mathcal{S}} cost_2\big(D \setminus D_1(j), S\big) \leq \max\{\alpha_j + 4OPT_2, 3\alpha_j + 2OPT_2\}$$

and we get that

$$cost_1\big(D_1(j), R_1\big) + \max_{S \in \mathcal{S}} cost_2\big(D \setminus D_1(j), S\big) \leq \max\{\beta_j + 4OPT_2,\ 3\beta_j + 2OPT_2\}$$

The other inequality $\beta_j \leq cost_1\big(D_1(j), R_1\big) + \max_{S \in \mathcal{S}} cost_2\big(D \setminus D_1(j)\big)$ is trivial. ◀

We are now ready to prove the theorem.

Proof of Theorem 9. Suppose Algorithm 5 returns $D_1(\tilde{j})$ for some $\tilde{j}$. From Claim 11 and the minimality of $\beta_{\tilde{j}}$:

$$f\big(D_1(\tilde{j})\big) \leq \max\{\beta_{\tilde{j}} + 4OPT_2, 3\beta_{\tilde{j}} + 2OPT_2\} \leq \max\{\beta_{j^*} + 4OPT_2, 3\beta_{j^*} + 2OPT_2\}.$$

From Claim 10 and Claim 11, we have $\beta_{j^*} \leq f\big(D_1(j^*)\big) \leq OPT_1 + 5OPT_2$. We conclude that,

$$f\big(D_1(\tilde{j})\big) \leq \max\big\{OPT_1 + 9OPT_2, 3OPT_1 + 17OPT_2\big\} = 3OPT_1 + 17OPT_2.$$

◀

5 Conclusion

In this paper, we present a new two-stage robust optimization framework for matching problems under both explicit and implicit models of uncertainty. Our problem is motivated by real-life applications in the ride-hailing industry. We study the Two-Stage Robust Matching Bottleneck problem, prove its hardness, and design approximation algorithms under different settings. Our algorithms give a constant approximation if the number of scenarios is fixed, but require additional assumptions when there are polynomially or exponentially many scenarios. It is an interesting question if there exists a constant approximation in the general case that does not depend on the number of scenarios.

References

1. Gagan Aggarwal, Gagan Goel, Chinmay Karande, and Aranyak Mehta. Online vertex-weighted bipartite matching and single-bid budgeted allocations. In *Proceedings of the twenty-second annual ACM-SIAM symposium on Discrete Algorithms*, pages 1253–1264. SIAM, 2011.
2. Nikhil Bansal, Niv Buchbinder, Anupam Gupta, and Joseph Seffi Naor. An $o(\log k^2)$-competitive algorithm for metric bipartite matching. In *European Symposium on Algorithms*, pages 522–533. Springer, 2007.
3. Piotr Berman, Bhaskar DasGupta, and Eduardo Sontag. Randomized approximation algorithms for set multicover problems with applications to reverse engineering of protein and gene networks. *Discrete Applied Mathematics*, 155(6-7):733–749, 2007.
4. Niv Buchbinder, Kamal Jain, and Joseph Seffi Naor. Online primal-dual algorithms for maximizing ad-auctions revenue. In *European Symposium on Algorithms*, pages 253–264. Springer, 2007.
5. Nikhil R Devanur and Thomas P Hayes. The adwords problem: online keyword matching with budgeted bidders under random permutations. In *Proceedings of the 10th ACM conference on Electronic commerce*, pages 71–78, 2009.
6. Nikhil R Devanur, Kamal Jain, and Robert D Kleinberg. Randomized primal-dual analysis of ranking for online bipartite matching. In *Proceedings of the twenty-fourth annual ACM-SIAM symposium on Discrete algorithms*, pages 101–107. SIAM, 2013.
7. Kedar Dhamdhere, Vineet Goyal, R Ravi, and Mohit Singh. How to pay, come what may: Approximation algorithms for demand-robust covering problems. In *46th Annual IEEE Symposium on Foundations of Computer Science (FOCS'05)*, pages 367–376. IEEE, 2005.
8. Omar El Housni and Vineet Goyal. Beyond worst-case: A probabilistic analysis of affine policies in dynamic optimization. In *Advances in neural information processing systems*, pages 4756–4764, 2017.
9. Bruno Escoffier, Laurent Gourvès, Jérôme Monnot, and Olivier Spanjaard. Two-stage stochastic matching and spanning tree problems: Polynomial instances and approximation. *European Journal of Operational Research*, 205(1):19–30, 2010.
10. Uriel Feige. A threshold of ln n for approximating set cover. *Journal of the ACM (JACM)*, 45(4):634–652, 1998.
11. Uriel Feige, Kamal Jain, Mohammad Mahdian, and Vahab Mirrokni. Robust combinatorial optimization with exponential scenarios. In *International Conference on Integer Programming and Combinatorial Optimization*, pages 439–453. Springer, 2007.
12. Jon Feldman, Aranyak Mehta, Vahab Mirrokni, and Shan Muthukrishnan. Online stochastic matching: Beating 1-1/e. In *2009 50th Annual IEEE Symposium on Foundations of Computer Science*, pages 117–126. IEEE, 2009.
13. Moran Feldman, Ola Svensson, and Rico Zenklusen. Online contention resolution schemes. In *Proceedings of the twenty-seventh annual ACM-SIAM symposium on Discrete algorithms*, pages 1014–1033. SIAM, 2016.
14. Yiding Feng and Rad Niazadeh. Batching and optimal multi-stage bipartite allocations. In *12th Innovations in Theoretical Computer Science Conference (ITCS 2021)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2021.
15. Yiding Feng, Rad Niazadeh, and Amin Saberi. Two-stage stochastic matching with application to ride hailing. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2862–2877. SIAM, 2021.
16. Harold N Gabow and Robert E Tarjan. Algorithms for two bottleneck optimization problems. *Journal of Algorithms*, 9(3):411–417, 1988.
17. Robert S Garfinkel and KC Gilbert. The bottleneck traveling salesman problem: Algorithms and probabilistic analysis. *Journal of the ACM (JACM)*, 25(3):435–448, 1978.
18. Gagan Goel and Aranyak Mehta. Online budgeted matching in random input models with applications to adwords. In *SODA*, volume 8, pages 982–991, 2008.

19 Anupam Gupta, Viswanath Nagarajan, and Ramamoorthi Ravi. Thresholded covering algorithms for robust and max-min optimization. In *International Colloquium on Automata, Languages, and Programming*, pages 262–274. Springer, 2010.

20 Bernhard Haeupler, Vahab S Mirrokni, and Morteza Zadimoghaddam. Online stochastic weighted matching: Improved approximation algorithms. In *International workshop on internet and network economics*, pages 170–181. Springer, 2011.

21 Dorit S Hochbaum and David B Shmoys. A unified approach to approximation algorithms for bottleneck problems. *Journal of the ACM (JACM)*, 33(3):533–550, 1986.

22 Omar El Housni, Vineet Goyal, and David Shmoys. On the power of static assignment policies for robust facility location problems. *arXiv preprint arXiv:2011.04925*, 2020.

23 Patrick Jaillet and Xin Lu. Online stochastic matching: New algorithms with better bounds. *Mathematics of Operations Research*, 39(3):624–646, 2014.

24 Bala Kalyanasundaram and Kirk Pruhs. Online weighted matching. *Journal of Algorithms*, 14(3):478–488, 1993.

25 Viggo Kann. Maximum bounded 3-dimensional matching is max snp-complete. *Information Processing Letters*, 37(1):27–35, 1991.

26 Chinmay Karande, Aranyak Mehta, and Pushkar Tripathi. Online bipartite matching with unknown distributions. In *Proceedings of the forty-third annual ACM symposium on Theory of computing*, pages 587–596, 2011.

27 Richard M Karp, Umesh V Vazirani, and Vijay V Vazirani. An optimal algorithm for on-line bipartite matching. In *Proceedings of the twenty-second annual ACM symposium on Theory of computing*, pages 352–358, 1990.

28 Irit Katriel, Claire Kenyon-Mathieu, and Eli Upfal. Commitment under uncertainty: Two-stage stochastic matching problems. *Theoretical Computer Science*, 408(2-3):213–223, 2008.

29 Samir Khuller, Stephen G Mitchell, and Vijay V Vazirani. On-line algorithms for weighted bipartite matching and stable marriages. *Theoretical Computer Science*, 127(2):255–267, 1994.

30 Nan Kong and Andrew J Schaefer. A factor 12 approximation algorithm for two-stage stochastic matching problems. *European Journal of Operational Research*, 172(3):740–746, 2006.

31 Nitish Korula and Martin Pál. Algorithms for secretary problems on graphs and hypergraphs. In *International Colloquium on Automata, Languages, and Programming*, pages 508–520. Springer, 2009.

32 Euiwoong Lee and Sahil Singla. Maximum matching in the online batch-arrival model. In *International Conference on Integer Programming and Combinatorial Optimization*, pages 355–367. Springer, 2017.

33 Lyft. Matchmaking in lyft line - part 1. `https://eng.lyft.com/matchmaking-in-lyft-line-9c2635fe62c4`, 2016.

34 Mohammad Mahdian and Qiqi Yan. Online bipartite matching with random arrivals: an approach based on strongly factor-revealing lps. In *Proceedings of the forty-third annual ACM symposium on Theory of computing*, pages 597–606, 2011.

35 Vahideh H Manshadi, Shayan Oveis Gharan, and Amin Saberi. Online stochastic matching: Online actions based on offline statistics. *Mathematics of Operations Research*, 37(4):559–573, 2012.

36 Jannik Matuschke, Ulrike Schmidt-Kraepelin, and José Verschae. Maintaining perfect matchings at low cost. *arXiv preprint arXiv:1811.10580*, 2018.

37 Aranyak Mehta. Online matching and ad allocation. *Theoretical Computer Science*, 8(4):265–368, 2012.

38 Aranyak Mehta, Amin Saberi, Umesh Vazirani, and Vijay Vazirani. Adwords and generalized online matching. *Journal of the ACM (JACM)*, 54(5):22–es, 2007.

39 Aranyak Mehta, Bo Waggoner, and Morteza Zadimoghaddam. Online stochastic matching with unequal probabilities. In *Proceedings of the twenty-sixth annual ACM-SIAM symposium on Discrete algorithms*, pages 1388–1404. SIAM, 2014.

40 Adam Meyerson, Akash Nanavati, and Laura Poplawski. Randomized online algorithms for minimum metric bipartite matching. In *Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm*, pages 954–959. Society for Industrial and Applied Mathematics, 2006.
41 Sharath Raghvendra. A robust and optimal online algorithm for minimum metric bipartite matching. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2016)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2016.
42 Uber. Uber marketplace and matching. `https://marketplace.uber.com/matching`, 2020.
43 Vijay V Vazirani. *Approximation algorithms*. Springer Science & Business Media, 2013.
44 Lingyu Zhang, Tao Hu, Yue Min, Guobin Wu, Junying Zhang, Pengcheng Feng, Pinghua Gong, and Jieping Ye. A taxi order dispatch model based on combinatorial optimization. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 2151–2159, 2017.

A NP-Hardness proofs for TSRMB

We start by presenting the 3-Dimensional Matching (3-DM) and Set Cover problems, that we use in our reductions to show Theorem 1. Both problems are known to be strongly NP-hard [10, 25].

3-Dimensional Matching (3-DM): Given three sets U, V, and W of equal cardinality n, and a subset T of $U \times V \times W$, is there a subset M of T with $|M| = n$ such that whenever (u, v, w) and (u', v', w') are distinct triples in M, $u \neq u'$, $v \neq v'$, and $w \neq w'$?

Set Cover Problem: Given a set of elements $\mathcal{U} = \{1, 2, ..., n\}$ (called the universe), a collection $S_1, \dots, S_m$ of m sets whose union equals the universe and an integer p.
Question: Is there a set $C \subset \{1, \dots, m\}$ such that $|C| \leq p$ and $\bigcup_{i \in C} S_i = \mathcal{U}$?

Proof of Theorem 1.
Explicit uncertainty. Consider an instance of the 3-Dimensional Matching Problem. We can use it to construct (in polynomial time) an instance of TSRMB with 2 scenarios as follows:

- Create two scenarios of size n: $S_1 = U$ and $S_2 = V$.
- Set $D = T$, every driver corresponds to a triple in T.
- For every $w \in W$, let $d_T(w)$ be the number of sets in T that contain w. We create $d_T(w) - 1$ first stage riders, that are all copies of w. The total number of first stage riders is therefore $|R_1| = |T| - n$.
- For $(w, e) \in R_1 \times D$, $d(w, e) = \begin{cases} 1 & \text{if } w \in e \\ 3 & \text{otherwise.} \end{cases}$
- For $(u, e) \in S_1 \cup S_2 \times D$, $d(u, e) = \begin{cases} 1 & \text{if } u \in e \\ 3 & \text{otherwise.} \end{cases}$
- For $u, v \in R_1 \cup S_1 \cup S_2, d(u, v) = \min_{e \in D} d(u, e) + d(v, e)$.
- For $e, f \in D, d(e, f) = \min_{u \in R_1 \cup S_1 \cup S_2} d(u, e) + d(u, f)$.

This choice of distances induces a metric graph. We claim that there exists a 3-dimensional matching if and only if there exists a solution to this TSRMB instance with total cost equal to 2. Suppose that $M = \{e_1, \dots, e_n\} \subset T$ is a 3-Dimensional matching. Let $e_1, \dots, e_n$ be the drivers that correspond to M in the TSRMB instance. We show that by using $D_1 = D \setminus \{e_1, \dots, e_n\}$ as a first stage decision, we ensure that the total cost for the TSRMB instance is equal to 2. For any rider u in scenario S_1, by definition of M, there exits a unique edge $e_i \in M$ that covers u. The corresponding driver $e_i \notin D_1$ can be matched to u with a distance equal to 1. Furthermore, e_i cannot be matched to any other rider in S_1 with a cost less than 1. Similarly, for any rider v in scenario S_2, since there exits a unique edge $e_j \in M$ that covers v, the corresponding driver can be matched to v with a cost of 1. The second stage cost is therefore equal to 1. As for the first stage cost, we know by definition of M, that every element $w \in W$ is covered exactly once. Therefore, for every $w \in W$, there exists $d_T(w) - 1$ edges that contain w in $T \setminus M$. This means that every 1st stage rider can be matched to a driver in D_1 with a cost equal to 1. Hence the total cost of this two-stage matching is equal to 2.

Suppose now that there exists a solution to the TSRMB instance with a cost equal to 2. This means that the first and second stage costs are both equal to 1. Let $M = \{e_1, \ldots, e_n\}$ be the set of drivers used in the second stage of this solution. We show that M is a 3-dimensional matching. Let $e_i = (u, v, w)$ and $e_j = (u', v', w')$ be distinct triples in M. Since the second stage cost is equal to 1, the driver e_i (resp. e_j) must be matched to u (resp. u') in S_1. Since we have exactly n second stage drivers and n riders in S_1, this means that e_i and e_j have to be matched to different second stage riders in S_1. Therefore we get $u' \neq u$. Similarly we see that $v' \neq v$. Assume now that $w = w'$, this means that the TSRMB solution has used two drivers (triples) e_i and e_j that contain w in the second stage. It is therefore impossible to match all the $d_T(w) - 1$ copies of w in the first stage with a cost equal to 1. Therefore $w \neq w'$. The above construction can be performed in polynomial time of the 3-DM input, and therefore shows that TSRMB with two scenarios is NP-hard.

Now, to show that TSRMB is hard to approximate within a factor better than 2, we consider three scenarios. Consider an instance of 3-DM. We can use it to construct an instance of TSRMB with 3 scenarios as follows:

- Create 3 scenarios of size n: $S_1 = U$, $S_2 = V$ and $S_3 = W$.
- Set $D = T$.
- Create $|R_1| = |T| - n$ first stage riders.
- For $(w, e) \in R_1 \times D$, $d(w, e) = 1$.
- For $(u, e) \in S_1 \cup S_2 \cup S_3 \times D$, $d(u, e) = \begin{cases} 1 & \text{if } u \in e \\ 3 & \text{otherwise.} \end{cases}$
- For $u, v \in R_1 \cup S_1 \cup S_2 \cup S_3$, $d(u, v) = \min_{e \in D} d(u, e) + d(v, e)$.
- For $e, f \in D, d(e, f) = \min_{u \in R_1 \cup S_1 \cup S_2 \cup S_3} d(u, e) + d(u, f)$.

This choice of distances induces a metric graph. Similarly to the proof of 2 scenarios, we can show that there exists a 3-dimensional matching if and only if there exists a TSRMB solution with cost equal to 2. Furthermore, any solution for this TSRMB instance has either a total cost of 2 or 4 (the first stage cost is always equal to 1). We show that if a $(2 - \epsilon)$-approximation (for some $\epsilon > 0$) to the TSRMB exists then 3-Dimensional Matching is decidable. We know that this instance of TSRMB has a solution with total cost equal to 2 if and only if there is a 3-dimensional matching. Furthermore, if there is no 3-dimensional matching, the cost of the optimal solution to TSRMB must be 4. Therefore, if an algorithm guarantees a ratio of $(2 - \epsilon)$ and a 3-dimensional matching exists, the algorithm delivers a solution with total cost equal to 2. If there is no 3-dimensional matching, then the solution produced by the algorithm has a total cost of 4.

Implicit uncertainty. We prove the hardness for $k = 1$. We start from an instance of the Set Cover problem and construct an instance of the TSRMB problem. Consider an instance of the decision problem of set cover. We can use it to construct the following TSRMB instance:

- Create m drivers $D = \{1, \ldots, m\}$. For each $j \in \{1, \ldots, m\}$, driver j corresponds to set S_j.
- Create $m - p$ first stage riders, $R_1 = \{1, \ldots, m - p\}$.
- Create n second stage riders, $R_2 = \{1, \ldots, n\}$.
- Set $\mathcal{S} = \{\{1\}, \ldots, \{n\}\}$. Every scenario is of size 1.

As for the distances between riders and drivers, we define them as follows:

- For $(i, j) \in R_1 \times D$, $d(i, j) = 1$.
- For $(i, j) \in R_2 \times D$, $d(i, j) = \begin{cases} 1 & \text{if } i \in S_j \\ 3 & \text{otherwise.} \end{cases}$
- For $i, i' \in R_1 \cup R_2$, $d(i, i') = \min\limits_{j \in D} d(i, j) + d(i', j)$.
- For $j, j' \in D$, $d(j, j') = \min\limits_{i \in R_1 \cup R_2} d(i, j) + d(i, j')$.

This choice of distances induces a metric graph. Moreover, every feasible solution to this TSRMB instance has a first stage cost of exactly 1. We show that a set cover of size $\leq p$ exists if and only if there is a TSRMB solution with total cost equal to 2. Suppose without loss of generality that $S_1, \ldots, S_p$ is a set cover. Then by using the drivers $\{1, \ldots, p\}$ in the second stage, we ensure that every scenario is matched with a cost of 1. This implies the existence of a solution with total cost equal to 2. Now suppose there is a solution to the TSRMB problem with cost equal to 2. Let D_2 be the set of second stage drivers of this solution, then we have $|D_2| = p$. We claim that the sets corresponding to drivers in D_2 form a set cover. In fact, since the total cost of the TSRMB solution is equal to 2, the second stage cost is equal to 1. This means that for every scenario $i \in \{1, \ldots, n\}$, there is a driver $j \in D_2$ within a distance 1 from i. Therefore $i \in S_j$ and $\{S_j : j \in D_2\}$ is a set cover.

Next we show that if $(2 - \epsilon)$-approximation (for some $\epsilon > 0$) to the TSRMB exists then Set Cover is decidable. We know that the TSRMB problem has a solution of cost 2 if and only if there is a set cover of size less than p. Furthermore, if there is no such set cover, the cost of the optimal solution must be 4. Therefore, if the algorithm guarantees a ratio of $(2 - \epsilon)$ and there is a set cover of size less than p, the algorithm delivers a solution with a total cost of 2. If there is no set cover, then clearly the solution produced by the algorithm has a cost of 4. ◀

▶ Remark 12. For $k \geq 2$, we can use a generalization of Set Cover to show that the problem is hard for any k. We use a reduction from the Set MultiCover Problem ([3, 43]) defined below.

Set MultiCover Problem: Given a set of elements $\mathcal{U} = \{1, 2, ..., n\}$ (called the universe) and a collection $S_1, \ldots, S_m$ of m sets whose union equals the universe. A "coverage factor" (positive integer) k and an integer p. Is there a set $C \subset \{1, \ldots, m\}$ such that $|C| \leq p$ and for each element $x \in \mathcal{U}$, $|j \in C \ : \ x \in S_j| \geq k$?

We can create an instance of TSRMB from a Set MultiCover instance similarly to Set Cover with the exception that $\mathcal{S} = \{S \subset R_2 \ s.t. \ |S| = k\}$. The hardness result follows similarly.

B Implicit scenarios: small surplus

Proof of Lemma 8. Let δ be the driver given by (3). We claim that the k closest riders to δ are all within a distance less than OPT_2 from δ. Consider D_2^* to be the $k + \ell$ drivers left for the second stage in the optimal solution. Every driver in D_2^* can be matched to a set of different second stage riders over different scenarios. Let us rank the drivers in D_2^* according to how many different second stage riders they are matched to over all scenarios, in descending order. Formally, let $D_2^* = \{\delta_1, \delta_2, \dots, \delta_{k+\ell}\}$ and let $R^*(\delta_i)$ be the second stage riders that are matched to δ_i in the optimal solution in some scenario, such that

$$|R^*(\delta_1)| \geq \ldots \geq |R^*(\delta_{k+\ell})|.$$

We claim that $|R^*(\delta_1)| \geq k$. In fact, we have $\sum_{i=1}^{k+\ell} |R^*(\delta_i)| \geq n$ because every second stage rider is matched to at least one driver in some scenario. Therefore

$$|R^*(\delta_1)| \geq \frac{n}{k+\ell} \geq \frac{n}{2k} \geq k.$$

We know that all the second stage riders in $R^*(\delta_1)$ are within a distance less than OPT_2 from δ_1. Therefore $\max_{r \in R_k(\delta_1)} d(\delta_1, r) \leq OPT_2$. But we know that by definition of δ,

$$\max_{r \in R_k(\delta)} d(\delta, r) \leq \max_{r \in R_k(\delta_1)} d(\delta_1, r) \leq OPT_2$$

This proves that the k closest second stage riders to δ are within a distance less than OPT_2. Let $R(\delta)$ be the set of all second stage riders that are within a distance less than OPT_2 from δ. Recall that $R_k(\delta)$ is the set of the k closest second stage riders to δ. In the optimal solution, the scenario $R_k(\delta)$ is matched to a set of at least new $k-1$ drivers $\{\delta_{i_1}, \ldots \delta_{i_{k-1}}\} \subset D_2^* \setminus \{\delta\}$. We show a lower bound on the size of $R(\delta)$ and the number of riders matched to $\{\delta_{i_1}, \ldots \delta_{i_{k-1}}\}$ over all scenarios in the optimal solution.

▷ Claim 13. $\left|R(\delta) \bigcup_{j=1}^{k-1} R^*(\delta_{i_j})\right| \geq n - \ell$

Proof. Suppose the opposite, suppose that at least $\ell + 1$ riders from R_2 are not in the union. Let F be the set of these $\ell + 1$ riders. Since $\ell + 1 \leq k$, we can construct a scenario S that includes F. In the optimal solution, and in particular, in the second stage matching of S, at least one rider from F needs to be matched to a driver from $\{\delta, \delta_{i_1}, \ldots \delta_{i_{k-1}}\}$. Otherwise there are only ℓ second stage drivers left to match all of F. Therefore there exists $r \in F$ such that either $r \in R(\delta)$ or there exists $j \in \{1, \ldots, k-1\}$ such that $r \in R^*(\delta_{i_j})$. This shows that $r \in R(\delta) \bigcup_{j=1}^{k-1} R^*(\delta_{i_j})$, which is a contradiction. Therefore, at most ℓ second stage riders are not in the union. ◀

▷ Claim 14. For any rider $r \in R(\delta) \bigcup_{j=1}^{k-1} R^*(\delta_{i_j})$, we have $d(r, \delta) \leq 3OPT_2$.

Proof. If $r \in R(\delta)$ then by definition we have $d(r, \delta) \leq OPT_2$. Now suppose $r \in R^*(\delta_{i_j})$ for $j \in [k-1]$. Let r' be the rider from scenario $R_k(\delta)$ that was matched to δ_{i_j} in the optimal solution. Then by the triangular inequality

$$d(r, \delta) \leq d(r, \delta_{i_j}) + d(\delta_{i_j}, r') + d(r', \delta) \leq 3OPT_2.$$

◀

From Claim 14, we see that the ball centered at δ, with radius $3OPT_2$, contains at least $n - \ell$ second stage riders in $R(\delta) \bigcup_{j=1}^{k-1} R^*(\delta_{i_j})$. This proves the first part of the lemma. The second part is proved in the next claim.

▷ Claim 15. For $r_1 \in R_k(\delta)$ and $r_2 \in R(\delta) \bigcup_{j=1}^{k-1} R^*(\delta_{i_j})$, we have $d(r_1, r_2) \leq 4OPT_2$.

Proof. Let $r_1 \in R_k(\delta)$. If $r_2 \in R(\delta)$ then $d(r_1, r_2) \leq d(r_1, \delta) + d(\delta, r_2) \leq 2OPT_2$. If $r_2 \in R^*(\delta_{i_j})$ for some j, and r' is the rider from scenario $R_k(\delta)$ that was matched to δ_{i_j}

$$d(r_1, r_2) \leq d(r_1, \delta) + d(\delta, r') + d(r', \delta_{i_j}) + d(\delta_{i_j}, r_2) \leq 4OPT_2.$$

◀

Claim 13 shows that the number of riders included in $R(\delta) \bigcup_{j=1}^{k-1} R^*(\delta_{i_j})$ is at least $n - \ell$. Claim 14 shows that each one of this rider has distance less than $3OPT_2$ from δ. Finally, Claim 15 shows that the distance between any one of this riders and any rider in $R_k(\delta)$ is less than $3OPT_2$. This concludes the proof of Lemma 8. ◀