Approximating Two-Stage Stochastic Supplier Problems

Brian Brubach
Wellesley College, MA, USA

Nathaniel Grammel
University of Maryland at College Park, MD, USA

David G. Harris
University of Maryland at College Park, MD, USA

Aravind Srinivasan
University of Maryland at College Park, MD, USA

Leonidas Tsepenekas
University of Maryland at College Park, MD, USA

Anil Vullikanti
University of Virginia, Charlottesville, VA, USA

Abstract

The main focus of this paper is radius-based (supplier) clustering in the two-stage stochastic setting with recourse, where the inherent stochasticity of the model comes in the form of a budget constraint. We also explore a number of variants where additional constraints are imposed on the first-stage decisions, specifically matroid and multi-knapsack constraints.

Our eventual goal is to provide results for supplier problems in the most general distributional setting, where there is only black-box access to the underlying distribution. To that end, we follow a two-step approach. First, we develop algorithms for a restricted version of each problem, in which all possible scenarios are explicitly provided; second, we employ a novel *scenario-discarding* variant of the standard *Sample Average Approximation (SAA)* method, in which we crucially exploit properties of the restricted-case algorithms. We finally note that the scenario-discarding modification to the SAA method is necessary in order to optimize over the radius.

2012 ACM Subject Classification Theory of computation → Design and analysis of algorithms

Keywords and phrases Approximation Algorithms, Stochastic Optimization, Two-Stage Recourse Model, Clustering Problems, Knapsack Supplier

Digital Object Identifier 10.4230/LIPIcs.APPROX/RANDOM.2021.23

Category APPROX

Related Version *Full Version*: https://arxiv.org/abs/2008.03325

Funding *Brian Brubach*: Brian Brubach was supported in part by NSF awards CCF-1422569 and CCF-1749864, and by research awards from Adobe.
Nathaniel Grammel: Nathaniel Grammel was supported in part by NSF awards CCF-1749864 and CCF-1918749, and by research awards from Amazon and Google.
Aravind Srinivasan: Aravind Srinivasan was supported in part by NSF awards CCF-1422569, CCF-1749864 and CCF-1918749, and by research awards from Adobe, Amazon, and Google.
Leonidas Tsepenekas: Leonidas Tsepenekas was supported in part by NSF awards CCF-1749864 and CCF-1918749, and by research awards from Amazon and Google.

Acknowledgements The authors want to sincerely thank Chaitanya Swamy as well as referees of earlier versions of the paper, for their precious feedback and helpful suggestions.

Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2021).
Editors: Mary Wootters and Laura Sanità; Article No. 23; pp. 23:1–23:22
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

Stochastic optimization, first introduced in the work of Beale [2] and Dantzig [5], provides a way for modeling uncertainty in the realization of the input data. In this paper, we give approximation algorithms for a family of problems in stochastic optimization, and more precisely in the 2-*stage recourse model* [22]. Our formal problem definitions follow.

We are given a set of clients $\mathcal{C}$ and a set of facilities $\mathcal{F}$, in a metric space characterized by a distance function d. We let $n = |\mathcal{C}|$ and $m = |\mathcal{F}|$. Our paradigm unfolds in two stages. In the first, each $i \in \mathcal{F}$ has a cost c^I_i, but at that time we do not know which clients from $\mathcal{C}$ will need service, and we only have a description of the distribution $\mathcal{D}$ that governs the arrivals of clients later on. In the second stage, a *scenario* $A \subseteq \mathcal{C}$ is realized with probability p_A according to $\mathcal{D}$, and now each $i \in \mathcal{F}$ has a cost c^A_i. The clients of the realized scenario are precisely those that will require service from the facilities of $\mathcal{F}$. Using only the description of the distribution $\mathcal{D}$, we can proactively open a set of facilities F_I in *stage-I*. Subsequently, when a scenario A arrives in *stage-II*, we can augment the already constructed solution by opening some additional facilities F_A.

Throughout the paper, the objective function we minimize is the *maximum covering distance or radius*. Let $d(j, S) = \min_{i \in S} d(i, j)$ for any $j \in \mathcal{C}$ and for any $S \subseteq \mathcal{F}$. We then ask for F_I and F_A, such that $d(j, F_I \cup F_A) \leq R$ for every A that materializes and all $j \in A$, **for the minimum R possible**. Furthermore, the expected opening cost of the returned solution is required to be at most some given budget B, i.e., $\sum_{i \in F_I} c^I_i + \mathbb{E}_{A \sim \mathcal{D}}\Big[\sum_{i \in F_A} c^A_i\Big] \leq B$. We call this problem *Two-Stage Stochastic Supplier* or **2S-Sup** for short.

Finally, we assume that for every $j \in \mathcal{C}$ we have $\Pr_{A \sim \mathcal{D}}[j \in A] > 0$; note that if this is not the case, then the presence of j in the input is completely redundant.

Additional Stage-I Constraints. Beyond the basic version of the problem, we also consider variants where there are additional hard constraints on the set of chosen stage-I facilities.

In *Two-Stage Stochastic Matroid Supplier* or **2S-MatSup** for short, the input also includes a matroid $\mathcal{M} = (\mathcal{F}, \mathcal{I})$, where $\mathcal{I} \subseteq 2^{\mathcal{F}}$ is the family of independent sets of $\mathcal{M}$. In this case, we additionally require $F_I \in \mathcal{I}$.

In *Two-Stage Stochastic Multi-knapsack Supplier* or **2S-MuSup** for short, L additional knapsack constraints are imposed on F_I. Specifically, we are given budgets $W_\ell \geq 0$ and weights $f^\ell_i \geq 0$ for every $i \in \mathcal{F}$ and every integer $\ell \in [L]$, such that the stage-I facilities should satisfy $\sum_{i \in F_I} f^\ell_i \leq W_\ell$ for every $\ell \in [L]$. We also call a **2S-MuSup** instance *discrete*, if all weights f^ℓ_i are integers, and for such an instance we further define a parameter $\Lambda = \prod_{\ell=1}^{L} W_\ell$.

Modeling the Stage-I Distributional Knowledge. To complete the description of a two-stage problem, one needs to define how knowledge of the distribution $\mathcal{D}$ is represented in stage-I.

The most general representation is the *black-box* model [20, 8, 17, 14, 19], where we only have access to an oracle that can sample scenarios A according to $\mathcal{D}$. In this model, every time a scenario A is revealed, either through the oracle or through an actual data realization, we also learn the facility-cost vector c^A associated with it. We also consider the more restricted *polynomial-scenarios* model [18, 11, 16, 7], where all scenarios A, together with their occurrence probabilities p_A and their corresponding facility-cost vectors c^A, are explicitly provided.

We use the suffixes **BB** and **Poly** to distinguish these settings. For example, **2S-Sup-BB** is the previously defined **2S-Sup** in the black-box model.

In both distributional settings, our algorithms must have runtime polynomial in n, m. For the polynomial-scenarios case, the runtime should also be polynomial in the number of explicitly provided scenarios.

1.1 Motivation

To our knowledge, we are the first to consider this type of radius minimization problems in the two-stage stochastic paradigm. Regarding clustering problems in this regime, most prior work has focused on *Facility Location* [18, 20, 16, 17, 9, 14, 19]. On similar lines, [1] studies a stochastic k-center variant, where points arrive independently, *but each point only needs to get covered with some given probability.* Moreover, **2S-Sup** is the natural two-stage counterpart of the well-known **Knapsack-Supplier** problem [10]. **Knapsack-Supplier** has a 3-approximation, which is also the best ratio possible unless P=NP [10].

To see a practical application for our problems, consider healthcare resource allocation, when trying to mitigate a disease outbreak through the preventive placement of testing sites. Suppose that $\mathcal{F}$ corresponds to potential locations that can host a testing center (e.g., hospitals, private clinics, university labs), $\mathcal{C}$ to populations that can be affected by a possible disease outbreak, and each scenario $A \in \mathcal{D}$ to which populations suffer the outbreak. Since immediate testing is of utmost importance, a central decision maker may prepare testing sites, such that under every scenario, each infected population has the closest possible access to a testing center. Assembling these sites in advance, i.e., in stage-I, has multiple benefits; for example, the necessary equipment and materials might be much cheaper and easier to obtain before the onset of the disease. Furthermore, the choice to minimize the maximum covering distance, as opposed to the opening cost, would reflect a policy valuing societal welfare more than economic performance.

In addition, there may be further constraints on F_I, irrespective of the stage-II decisions, which cannot be directly reduced to the budget B. For instance, we might have a constraint on the total number of personnel we want to occupy prior to the outbreak of the disease, assuming that facility i requires f_i people to keep it operational during the waiting period. To our knowledge, this is the first time additional stage-I constraints are studied in the two-stage stochastic regime.

1.2 Our Generalization Scheme and Comparison with Previous Results

Our ultimate goal is to devise algorithms for the black-box setting. As is usual in two-stage stochastic problems, we do this in three steps. First, we develop algorithms for the less complicated polynomial-scenarios model. Second, we sample a small number of scenarios from the black-box oracle and use our polynomial-scenarios algorithms to (approximately) solve the problems on them. Finally, we extrapolate this solution to the original black-box problem. This overall methodology is called *Sample Average Approximation (SAA).*

Unfortunately, standard SAA approaches [21, 4] cannot be directly applied in radius minimization problems. On a high level, the obstacle here is that we need to compute the true cost of the approximate solution, something that is impossible using already existing results. Because this is a delicate technical issue, we refer the reader to Appendix A for an in-depth discussion.

Our Sampling Framework. Since the optimal black-box radius R^* is always the distance between a client and a facility, there are at most nm different options for it. Thus, we consider each separately, and assume for now that we work with a specific guess R. Given this, we sample some N scenarios from the oracle, and let $Q = \{S_1, S_2, \dots, S_N\}$ be that sampled set. We then run our polynomial-scenarios η-approximation algorithms on Q, which are guaranteed to provide solutions that cover each client within distance ηR. Crucially, we show that if $R \geq R^*$ and N is chosen appropriately, these solutions have cost at most $(1+\epsilon)B$ on Q, for any $\epsilon > 0$. Hence, in the end we keep the minimum guess for R whose cost over the samples is at most $(1+\epsilon)B$.

For this minimum guess R (which obviously satisfies $R \leq R^*$), the polynomial-scenarios algorithm returned a stage-I set F_I, and a stage-II set F_{S_v} for each $S_v \in Q$. **Our polynomial-scenarios algorithms are also designed to satisfy two additional key properties.** First, given F_I and any $A \notin Q$, there is an *efficient* process to *extend* the algorithm's output to a stage-II solution F_A with $d(j, F_I \cup F_A) \leq \eta R$ for all $j \in A$. Second, irrespective of Q, the set $\mathcal{S}$ of possible black-box solutions the extension process might produce, has only exponential size as a function of n and m (by default, it could have size $2^{m|\mathcal{D}|}$, and note that $\mathcal{D}$ may be exponentially large or even uncountably infinite). **We call algorithms satisfying these properties *efficiently generalizable*.**

After using the extension process to construct a solution for every A that materializes, there is a final *scenario-discarding* step to our framework. Specifically, for some given $\alpha \in (0,1)$, we first determine a threshold value T corresponding to the $\lceil \alpha|Q| \rceil^{\text{th}}$ costliest scenario of Q. Then, if for an arriving A the computed set F_A has stage-II cost more than T, we perform no stage-II openings by setting $F_A = \emptyset$ (i.e., we "give up" on A). This step coupled with the bounds on $|\mathcal{S}|$ ensure that the overall opening cost of our solution is at most $(1+\epsilon)B$. At this point, note that discarding implies that there may exist scenarios A with $d(j, F_I \cup F_A) > \eta R$ for some $j \in A$. However, we show such scenarios occur with probability at most α, and the latter can be made inverse polynomially small.

1.3 Outline and Contributions

In Section 2, we present our generalization scheme. We summarize it as follows:

▶ **Theorem 1.** *Suppose we have an efficiently generalizable, η-approximation algorithm for the polynomial-scenarios variant of any of the problems we study. Let $\mathcal{S}$ be the set of all potential black-box solutions its extension process may produce. Then, for any $\gamma, \epsilon, \alpha \in (0,1)$ and with $\mathcal{O}\big(\frac{1}{\epsilon\alpha} \log\big(\frac{nm|\mathcal{S}|}{\gamma}\big) \log\big(\frac{nm}{\gamma}\big)\big)$ samples, we compute a radius R and a black-box solution F_I, F_A for all $A \in \mathcal{D}$:*

1. *F_I satisfies the stage-I specific constraints of the problem (matroid or multiknapsack).*
2. *With probability at least $1-\gamma$, we have $R \leq R^*$ and $\sum_{i \in F_I} c^I_i + \mathbb{E}_{A \sim \mathcal{D}}[\sum_{i \in F_A} c^A_i] \leq (1+\epsilon)B$, where R^* the optimal radius of the black-box variant.*
3. *With probability at least $1-\gamma$, there holds $\Pr_{A \sim \mathcal{D}}[d(j, F_I \cup F_A) \leq \eta R, \ \forall j \in A] \geq 1-\alpha$.*

▶ **Theorem 2.** *We provide the following efficiently generalizable algorithms:*

- *A 3-approximation for **2S-Sup-Poly** with $|\mathcal{S}| \leq (n+1)!$.*
  For the black-box case, the sample complexity of Theorem 1 is $\tilde{O}(\frac{n}{\epsilon\alpha})$.
- *A 5-approximation for **2S-MatSup-Poly** with $|\mathcal{S}| \leq 2^m n!$.*
  For the black-box case, the sample complexity of Theorem 1 is $\tilde{O}(\frac{m+n}{\epsilon\alpha})$.
- *A 5-approximation for* discrete *instances of **2S-MuSup-Poly**, with $|\mathcal{S}| \leq 2^m$ and runtime* poly(n, m, Λ). *In the black-box case, the sample complexity of Theorem 1 is $\tilde{O}(\frac{m}{\epsilon\alpha})$.*

Here, $\tilde{O}()$ hides polylog$(n, m, 1/\gamma)$ terms. The 3-approximation for **2S-Sup-Poly** is presented in Section 3. It relies on a novel LP rounding technique, not used in clustering problems before. Notably, its approximation ratio matches the lower bound of the non-stochastic counterpart [10] (**Knapsack Supplier**), something very rare in the two-stage paradigm. The 5-approximation for **2S-MatSup-Poly** is presented in Section 4. It relies on solving an auxiliary LP, whose optimal solution is guaranteed to be integral. The 5-approximation for **2S-MuSup-Poly** is presented in Appendix C, and is based on a reduction to a deterministic supplier problem with outliers. Specifically, if we view stage-I as consisting of a deterministic robust problem, stage-II is interpreted as trying to cover all outliers left over by stage-I.

The main advantages of our generalization scheme are.

1. Unlike standard SAA approaches [4, 21], it can handle problems based on the maximum-radius objective function.
2. The approximation ratio η is preserved with high probability during the generalization. By contrast, in typical two-stage problems, the approximation ratio usually gets inflated when generalizing the polynomial-scenarios setting to the black-box one.
3. The adaptive selection of T yields crisp sample bounds in terms of α and ϵ. By contrast, simpler non-adaptive approaches (e.g., $T = \frac{B}{\alpha}$) would still give the same guarantees, but the dependence of the sample bounds on α, ϵ would be worse ($\frac{1}{\epsilon^2\alpha^2}$ compared to $\frac{1}{\epsilon\alpha}$ as we achieve). This adaptive thresholding may also be of independent interest; for instance, we conjecture that it might be able to improve the sample complexity in the SAA analysis of [4].

Remark 1. There is an important connection between the design of our generalization scheme and the design of our polynomial-scenarios approximation algorithms.

In any SAA approach, the sample complexity necessarily depends on the set of possible actions over which the generalization is performed. In Theorem 1, the sample bounds are given in terms of the *cardinality* of $\mathcal{S}$. Following the lines of [21], it may be possible to replace this dependence with a notion of dimension of the underlying convex program. However, such general bounds would lead to *significantly* larger complexities, consisting of very high order polynomials of n, m.

On the other hand, all of our polynomial-scenarios algorithms are carefully designed, so that the *cardinality* of $\mathcal{S}$ itself is small. *Indeed, one of the major contributions of this work is to show that this property can still be satisfied for sophisticated approximation algorithms using complex LP rounding.* Consequently, we can use simple generalization bounds. Besides being clear and intuitive, these lead to a much lower dependence on n, m for the sample complexity (see Theorem 2). To our knowledge, these are the first examples of non-trivial approximation algorithms for two-stage stochastic problems via directly bounding the size of the solution set $\mathcal{S}$.

Remark 2. If we assume that the maximum stage-II cost of any facility is bounded by some polynomial value Δ, then we could use standard SAA results directly for our problems. Alternatively, we can use a variant of our generalization scheme (without scenario-discarding) getting refined sample bounds. A simple modification of our Section 2 analysis yields Theorem 3. However, this additional assumption on the cost function is much stronger than what is typically used in the two-stage stochastic literature, and so our scheme aims at tackling the most general case.

▶ **Theorem 3.** *Suppose we have an efficiently generalizable, η-approximation algorithm for the polynomial-scenarios variant of any of the problems we study. Let $\mathcal{S}$ be the set of all possible black-box solutions its extension process can produce. Then, for any $\gamma, \epsilon \in (0, 1)$ and*

Algorithm 1 GreedyCluster($\mathcal{Q}, R, g$).

```
$H \leftarrow \emptyset$;
for each $j \in \mathcal{Q}$ in non-increasing order of $g(j)$ do
    $H \leftarrow H \cup \{j\}$;
    for each $j' \in \mathcal{Q}$ with $G_{j,R} \cap G_{j',R} \neq \emptyset$ do
        $\pi(j') \leftarrow j, \mathcal{Q} \leftarrow \mathcal{Q} \setminus \{j'\}$;
    end
end
Return $(H, \pi)$ ;
```

with $\mathcal{O}\big(\frac{m\Delta}{\epsilon}\log\big(\frac{nm|\mathcal{S}|}{\gamma}\big)\log\big(\frac{nm}{\gamma}\big)\big)$ samples, we get a radius R and a black-box solution F_I, F_A for all $A \in \mathcal{D}$:

1. *F_I satisfies the stage-I specific constraints of the problem (matroid or multiknapsack).*
2. *With probability at least $1-\gamma$, we have $R \leq R^*$ and $\sum_{i \in F_I} c_i^I + \mathbb{E}_{A \sim \mathcal{D}}[\sum_{i \in F_A} c_i^A] \leq (1+\epsilon)B$, where R^* the optimal radius of the black-box variant.*
3. *With probability one, we have $d(j, F_I \cup F_A) \leq \eta R$ for all $j \in A \in \mathcal{D}$.*

*In particular, with our polynomial-scenarios approximation algorithms, the sample bounds of **2S-Sup**, **2S-MatSup** and **2S-MuSup** are $\tilde{O}(\frac{nm\Delta}{\epsilon}), \tilde{O}(\frac{(n+m)m\Delta}{\epsilon})$ and $\tilde{O}(\frac{m^2\Delta}{\epsilon})$ respectively.*

1.4 Notation and Important Subroutines

For $k \in \mathbb{N}$, we use $[k]$ to denote $\{1, 2, \dots, k\}$. Also, for a vector $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_k)$ and a subset $X \subseteq [k]$, we use $\alpha(X)$ to denote $\sum_{i \in X} \alpha_i$. For a client j and $R \geq 0$, we define $G_{j,R} = \{i \in \mathcal{F} : d(i,j) \leq R\}$, $i^I_{j,R} = \arg\min_{i \in G_{j,R}} c_i^I$ and $i^A_{j,R} = \arg\min_{i \in G_{j,R}} c_i^A$ for any A.

We repeatedly use a key subroutine named GreedyCluster(), shown in Algorithm 1. Its input is a set of clients $\mathcal{Q}$, a target radius R, and an ordering function $g : \mathcal{Q} \mapsto \mathbb{R}$. Its output is a set $H \subseteq \mathcal{Q}$ along with a mapping $\pi : \mathcal{Q} \mapsto H$. The goal of this subroutine is to sparsify the given input $\mathcal{Q}$, by greedily choosing a set of representative clients H.

▶ **Observation 4.** *For $(H, \pi) = GreedyCluster(\mathcal{Q}, R, g)$, the following two properties hold: (i) for all $j, j' \in H$ with $j \neq j'$, we have $G_{j,R} \cap G_{j',R} = \emptyset$; and (ii) for all $j \in \mathcal{Q}$ with $j' = \pi(j)$, we have $G_{j,R} \cap G_{j',R} \neq \emptyset$, $d(j,j') \leq 2R$, and $g(j') \geq g(j)$.*

2 Generalizing to the Black-Box Setting

Let $\mathcal{P}$ be any of the two-stage problems we consider, with polynomial-scenarios variant $\mathcal{P}$-**Poly** and black-box variant $\mathcal{P}$-**BB**. Moreover, suppose that we have an η-approximation algorithm Alg$\mathcal{P}$ for $\mathcal{P}$-**Poly**, which we intend to use to solve $\mathcal{P}$-**BB**. Before we proceed to our generalization scheme, we present some important definitions and assumptions.

As a starting point, assume we are given a radius demand R; we later discuss how to optimize over this. Hence, we denote a $\mathcal{P}$-**BB** problem instance by the tuple $\mathfrak{I} = (\mathcal{C}, \mathcal{F}, \mathcal{M}_I, c^I, B, R)$, where $\mathcal{C}$ is the set of clients, $\mathcal{F}$ the set of facilities i, each with stage-I cost c^I_i, $\mathcal{M}_I \subseteq 2^{\mathcal{F}}$ the set of legal stage-I openings (representing the stage-I specific constraints of $\mathcal{P}$), B the budget, and R the given covering demand. In addition, there is an underlying distribution $\mathcal{D}$, where each scenario $A \in \mathcal{D}$ appears with some unknown probability p_A. Our only means of access to $\mathcal{D}$ is via a sampling oracle. Finally, when a scenario $A \in \mathcal{D}$ is revealed, we also learn the corresponding facility costs c^A_i.

▶ **Definition 5.** *We define a* strategy *s to be a tuple $(F_I^s, F_A^s \mid A \in \mathcal{D})$ of facility sets, where A ranges over $\mathcal{D}$. The set F_I^s represents the facilities s opens in stage-I, and F_A^s denotes the facilities s opens in stage-II, when the arriving scenario is A. In other words, a strategy is a just potential solution to $\mathcal{P}$-**BB**.*

▶ **Assumption 6.** *For any strategy s and $A \in \mathcal{D}$, the value $c^A(F_A^s)$ has a continuous CDF. We can assume this w.l.o.g.; we simply add a dummy facility i_d in the input, and for all s and $A \in \mathcal{D}$, we include i_d in the original F_A^s. Then, $c_{i_d}^A$ is set to be some* infinitesimal smooth noise. *Also, B and $\mathcal{M}_I$ can trivially be extended to account for i_d.* Finally, the assumption implies that for a finite set of scenarios Q, the values $c^A(F_A^s)$ for all $A \in Q$ are distinct with probability 1.

We say that a given instance $\mathfrak{I}$ is *feasible* for $\mathcal{P}$-**BB**, if there exists a strategy s^* satisfying:

$$F_I^{s^*} \in \mathcal{M}_I, \quad c^I(F_I^{s^*}) + \sum_{A \in \mathcal{D}} p_A c^A(F_A^{s^*}) \leq B, \quad \forall j \in A \in \mathcal{D} \quad d(j, F_I^{s^*} \cup F_A^{s^*}) \leq R$$

For $\mathcal{P}$-**Poly**, consider an instance $\mathfrak{J} = (\mathcal{C}, \mathcal{F}, \mathcal{M}_I, Q, \vec{q}, \vec{c}, B, R)$, where $\mathcal{C}, \mathcal{F}, \mathcal{M}_I, B, R$ are as in the $\mathcal{P}$-**BB** setting, Q is the set of provided scenarios, $\vec{c}$ the vector of stage-I and stage-II explicitly given costs, and $\vec{q}$ the vector of occurrence probabilities q_A of each $A \in Q$. We say that the instance $\mathfrak{J}$ is *feasible* for $\mathcal{P}$-**Poly**, if there exist sets $F_I \subseteq \mathcal{F}$ and $F_A \subseteq \mathcal{F}$ for every $A \in Q$, such that:

$$F_I \in \mathcal{M}_I, \quad c^I(F_I) + \sum_{A \in Q} q_A c^A(F_A) \leq B, \quad \forall j \in A \in Q \quad d(j, F_I \cup F_A) \leq R$$

We also write F for the overall collection of sets F_I and $F_A : A \in Q$.

▶ **Definition 7.** *An algorithm Alg$\mathcal{P}$ is a* valid η-approximation algorithm *for $\mathcal{P}$-**Poly**, if given any problem instance $\mathfrak{J} = (\mathcal{C}, \mathcal{F}, \mathcal{M}_I, Q, \vec{q}, \vec{c}, B, R)$, one of the following two cases holds:*

(A) *If $\mathfrak{J}$ is feasible for $\mathcal{P}$-**Poly**, then Alg$\mathcal{P}$ returns a collection of sets F with $F_I \in \mathcal{M}_I$, $c^I(F_I) + \sum_{A \in Q} q_A c^A(F_A) \leq B$ and $\forall j \in A \in Q \quad d(j, F_I \cup F_A) \leq \eta R$.*

(B) *If $\mathfrak{J}$ is not feasible for $\mathcal{P}$-**Poly**, then the algorithm either returns "INFEASIBLE", or returns a collection of sets F satisfying the properties presented in A.*

▶ **Definition 8.** *A valid η-approximation algorithm Alg$\mathcal{P}$ for $\mathcal{P}$-**Poly** is* efficiently generalizable, *if for every instance $\mathfrak{J} = (\mathcal{C}, \mathcal{F}, \mathcal{M}_I, Q, \vec{q}, \vec{c}, B, R)$ for which it returns a solution F, there is an efficient procedure that implicitly extends this to a strategy $\bar{s}$, and satisfies:*

(I) *Given any $A \in \mathcal{D}$, it returns a set $F_A^{\bar{s}} \subseteq \mathcal{F}$, with $d(j, F_I^{\bar{s}} \cup F_A^{\bar{s}}) \leq \eta R$ for all $j \in A$.*

(II) *$F_I^{\bar{s}} = F_I$ and $F_A^{\bar{s}} = F_A$ for every $A \in Q$.*

(III) *Given $\mathfrak{J}$, let $\mathcal{S}$ be the set of all possible strategies that are potentially achievable using the extension procedure for any set Q. Then $|\mathcal{S}| \leq t_{\mathcal{P}}(n,m)$ for some function $t_{\mathcal{P}}(n,m)$, with $\log(t_{\mathcal{P}}(n,m)) = \text{poly}(n,m)$.*

Note that property III is not trivial, since by default $|\mathcal{S}| \leq 2^{m|\mathcal{D}|}$, and $|\mathcal{D}|$ can be exponentially large or even uncountably infinite.

The first step of our generalization is based on sampling a set Q of scenarios from $\mathcal{D}$, and then applying the efficiently-generalizable Alg$\mathcal{P}$ on Q. When running the latter, we also increase the available budget to $(1+\epsilon)B$, for some $\epsilon > 0$. The purpose of this step is to verify whether or not the given instance of $\mathcal{P}$-**BB** is feasible, and to achieve this we may have to repeat it a polynomial number of times. See Algorithm 2 for the full details.

Algorithm 2 Determining Feasibility for $\mathcal{P}$-**BB**.

```
Input : Parameters $\epsilon, \gamma, \alpha \in (0,1)$, $N \geq 1$ and a $\mathcal{P}$-BB instance
        $\mathfrak{I} = (\mathcal{C}, \mathcal{F}, \mathcal{M}_I, c^I, B, R)$.
If $\exists j \in \mathcal{C} : d(j, \mathcal{F}) > R$ then return "INFEASIBLE" ; // For points not sampled
for $h = 1, \dots, \lceil \log_{\frac{13}{12}}(1/\gamma) \rceil$ do
    Draw $N$ independent samples from the oracle, obtaining set $Q = \{S_1, \dots, S_N\}$;
    Let $\vec{c}$ the vector containing $c^I$ and the stage-II facility-cost vectors of all $S_v \in Q$;
    For every $S_v \in Q$ set $q_{S_v} \leftarrow 1/N$;
    if $Alg\mathcal{P}(\mathcal{C}, \mathcal{F}, \mathcal{M}_I, Q, \vec{q}, \vec{c}, (1+\epsilon)B, R)$ returns $F$ then
        Let $T$ be the $\lceil \alpha N \rceil^{\text{th}}$ largest value of $c^{S_v}(F_{S_v})$ among all scenarios in $Q$;
        Return $(F, T)$;
    end
end
Return "INFEASIBLE";
```

If Algorithm 2 returns "INFEASIBLE", then our approach would deem that $\mathfrak{I}$ is not feasible for $\mathcal{P}$-**BB**. Otherwise, let F be the solution returned by Alg$\mathcal{P}$ at the last "successfull" iteration of the while loop. Because Alg$\mathcal{P}$ is efficiently-generalizable, we can apply its extension procedure to any arriving scenario, and therefore implicitly construct a strategy $\bar{s}$. By the properties of Alg$\mathcal{P}$ and II, I, we have $F^{\bar{s}}_I \in \mathcal{M}_I$ and $d(j, F^{\bar{s}}_I \cup F^{\bar{s}}_A) \leq \eta R$ for every $A \in \mathcal{D}$ and $j \in A$.

However, we are not yet done. The second step of our generalization framework consists of slightly modifying the strategy $\bar{s}$. For that reason, we use the value T returned by Algorithm 2, which corresponds to the $\lceil \alpha N \rceil^{\text{th}}$ largest value $c^{S_v}(F^{\bar{s}}_{S_v})$ among all $S_v \in Q$, with Q the sampled set in the last iteration of the while loop ($F^{\bar{s}}_{S_v} = F_{S_v}$ by II). Note here that Assumption 6 ensures that the choice of T is well-defined.

If now an arriving scenario A has $c^A(F^{\bar{s}}_A) > T$, we will perform no stage-II opening. This modification eventually constructs a new strategy $\hat{s}$, with $F^{\hat{s}}_I = F^{\bar{s}}_I$, $F^{\hat{s}}_A = \emptyset$ when $c^A(F^{\bar{s}}_A) > T$, and $F^{\hat{s}}_A = F^{\bar{s}}_A$ if $c^A(F^{\bar{s}}_A) \leq T$. The latter strategy will determine our final opening actions, and hence we need to analyze its *opening cost* $C(\hat{s})$ *over* $\mathcal{D}$, and the probability with which it does not return an η-approximate solution. Regarding the latter, note that when $F^{\hat{s}}_A \neq F^{\bar{s}}_A$, we can no longer guarantee an approximation ratio of η as implied by property I for $\bar{s}$.

▶ **Lemma 9.** *If instance $\mathfrak{I}$ is feasible for $\mathcal{P}$-**BB** and $N \geq 1/\epsilon$, then with probability at least $1 - \gamma$ Algorithm 2 does not terminate with "INFEASIBLE".*

Proof. By rescaling, we assume w.l.o.g. that $B = 1$. Also, the cost of any strategy s over $\mathcal{D}$ is given by $C(s) = c^I(F^s_I) + \sum_{A \in \mathcal{D}} p_A c^A(F^s_A)$. For any specific execution of the while loop in Algorithm 2, let Y^s_v be the second-stage cost of s on sample S_v. Finally, for a fixed s the random variables Y^s_v are independent, and the empirical cost of s on Q is $\hat{C}(s) = c^I(F^s_I) + \frac{1}{N}\sum_{v=1}^{N} Y^s_v$.

If $\mathfrak{I}$ is feasible, then there exists some strategy $s^\star$ satisfying $F^{s^\star}_I \in \mathcal{M}_I$ and $d(j, F^{s^\star}_I \cup F^{s^\star}_A) \leq R$ for every $A \in Q$ and $j \in A$. We will also show that $\hat{C}(s^\star) \leq (1+\epsilon)B$ with probability at least $1/13$. In this case, the restriction of $s^\star$ to Q verifies that $(\mathcal{C}, \mathcal{F}, \mathcal{M}_I, Q, \vec{q}, \vec{c}, (1+\epsilon)B, R)$ is feasible for $\mathcal{P}$-**Poly**. Thus, since Alg$\mathcal{P}$ is a valid η-approximation for $\mathcal{P}$-**Poly**, it will not return "INFEASIBLE".

As s^* is feasible for $\mathfrak{I}$ we have $C(s^\star) \leq B$, implying $\mathbb{E}[Y_v^{s^\star}] = \sum_{A \in \mathcal{D}} p_A \cdot c^A(F_A^{s^\star}) \leq B = 1$ for all samples v. By Lemma 20 with $\delta = \epsilon BN$, this yields

$$\Pr\Big[\sum_{v=1}^N Y_v^{s^\star} < \mathbb{E}[\sum_{v=1}^N Y_v^{s^\star}] + \epsilon BN\Big] \geq \min\Big\{\frac{\epsilon BN}{1+\epsilon BN}, \frac{1}{13}\Big\}$$

When $N \geq \frac{B}{\epsilon} = \frac{1}{\epsilon}$, we see that $\epsilon BN/(1+\epsilon BN) \geq 1/13$. Hence, with probability at least $1/13$ we have $\sum_{v=1}^N Y_v^{s^\star} < \mathbb{E}[\sum_{v=1}^N Y_v^{s^\star}] + \epsilon BN$, in which case we get $\hat{C}(s^*) \leq (1+\epsilon)B$ as shown below:

$$\begin{aligned}\hat{C}(s^\star) &= c^I(F_I^{s^\star}) + \frac{1}{N}\sum_{v=1}^N Y_v^{s^\star} \leq c^I(F_I^{s^\star}) + \frac{1}{N}\sum_{v=1}^N \mathbb{E}[Y_v^{s^\star}] + \epsilon B \\ &\leq c^I(F_I^{s^\star}) + \sum_{A \in \mathcal{D}} p_A \cdot c^A(F_A^{s^\star}) + \epsilon B \leq (1+\epsilon)B\end{aligned}$$

So each iteration terminates successfully with probability at least $1/13$. To bring the error probability down to at most γ, we repeat the process for $\left\lceil \log_{\frac{13}{12}}(1/\gamma)\right\rceil$ iterations. ◀

Let $\mathcal{T}$ be the event that Algorithm 2 terminates **without returning** "INFEASIBLE", and $\mathcal{T}_h$ the event that Alg$\mathcal{P}$ found a solution F at the h^{th} iteration of the while loop. We denote by *Invalid* the event that Algorithm 2 returns an invalid output; specifically, if $\mathcal{T}$ occurs, *Invalid* is the event of having $C(\hat{s}) > (1+2\epsilon)B$, otherwise it is the event of mistakenly deciding that $\mathfrak{I}$ is not feasible. Let now Q_h be the set of scenarios sampled at the h^{th} iteration of Algorithm 2, and for any strategy s *let* T_s^h *be the* $\lceil \alpha N \rceil^{th}$ *largest value* $c^{S_v}(F_{S_v}^s)$ *among all* $S_v \in Q_h$. We then denote by $\mathcal{E}_h$ the event that for all $s \in \mathcal{S}$, we have $\Pr_{A \sim \mathcal{D}}[c^A(F_A^s) > T_s^h] \geq \frac{\alpha}{4}$. Finally, note that due to III the set $\mathcal{S}$ is deterministically given in the event $\mathcal{E}_h$.

▶ **Lemma 10.** *For any* $\gamma, \alpha \in (0,1)$ *and* $N = \mathcal{O}\Big(\frac{1}{\alpha}\log(\frac{t_{\mathcal{P}}(n,m)}{\gamma})\Big)$*, we have* $\Pr[\bar{\mathcal{E}_h}] \leq \gamma/(\log_{\frac{13}{12}}(\frac{1}{\gamma})+1)$.

Proof. Focus on a specific iteration h. Consider a strategy $s \in \mathcal{S}$, and for each $S_v \in Q_h$ let X_v be an indicator random variable that is 1 iff $c^{S_v}(F_{S_v}^s) > T_s^h$. Also let $X = \sum_{v=1}^N X_v$, and note that by Assumption 6 we have $X = \lceil \alpha N \rceil - 1$. This implies that the empirical probability of scenarios with stage-II cost more than T_s^h is $q_s^h = (\lceil \alpha N \rceil - 1)/N$. Finally, let $p_s^h = \Pr_{A \sim \mathcal{D}}[c^A(F_A^s) > T_s^h]$.

If $p_s^h \geq \alpha$ then we immediately get $\Pr[p_s^h < \alpha/4] = 0$. Therefore, assume that $p_s^h < \alpha$. If $N \geq 4/\alpha$ then we have:

$$q_s^h - \frac{\alpha}{2} = \frac{\lceil \alpha N \rceil - 1}{N} - \frac{\alpha}{2} \geq \frac{\alpha N - 1}{N} - \frac{\alpha}{2} = \frac{\alpha}{2} - \frac{1}{N} \geq \frac{\alpha}{2} - \frac{\alpha}{4} = \frac{\alpha}{4}$$

Hence, if $p_s^h \geq q_s^h - \frac{\alpha}{2}$ and $N \geq 4/\alpha$, we get $p_s^h \geq \frac{\alpha}{4}$. Using Lemma 22 with $p_s^h < \alpha$, $\delta = \alpha/2$ and $N = \frac{8}{\alpha}\log\Big(\frac{t_{\mathcal{P}}(n,m)}{\gamma}(\log_{\frac{13}{12}}(\frac{1}{\gamma})+1)\Big) \geq 4/\alpha$ yields the following:

$$\Pr[p_s^h < \frac{\alpha}{4}] \leq \Pr[p_s^h < q_s^h - \alpha/2] \leq e^{-\Big(\frac{t_{\mathcal{P}}(n,m)}{\gamma}(\log_{\frac{13}{12}}(\frac{1}{\gamma})+1)\Big)} = \frac{\gamma}{t_{\mathcal{P}}(n,m)(\log_{\frac{13}{12}}(1/\gamma)+1)}$$

A union bound over all $s \in \mathcal{S}$ and property III will finally give $\Pr[\bar{\mathcal{E}_h}] \leq \gamma/(\log_{\frac{13}{12}}(\frac{1}{\gamma})+1)$. ◀

▶ **Theorem 11.** *For any* $\epsilon, \gamma, \alpha \in (0,1)$ *and* $N = \mathcal{O}\Big(\frac{1}{\epsilon\alpha}\log(\frac{t_{\mathcal{P}}(n,m)}{\gamma})\Big)$, $\Pr[\mathit{Invalid}] \leq 3\gamma$.

Proof. Using the definition of the Invalid event and Lemmas 9, 10 we get the following.

$$\begin{aligned}
\Pr[\text{Invalid}] &= \Pr[\text{Invalid} \mid \bar{\mathcal{T}}]\Pr[\bar{\mathcal{T}}] + \Pr[\text{Invalid} \mid \mathcal{T}]\Pr[\mathcal{T}] \leq \gamma + \sum_h \Pr[\text{Invalid} \wedge \mathcal{T}_h] \\
&= \gamma + \sum_h \Big(\Pr[\text{Invalid} \wedge \mathcal{T}_h \mid \mathcal{E}_h]\Pr[\mathcal{E}_h] + \Pr[\text{Invalid} \wedge \mathcal{T}_h \mid \bar{\mathcal{E}_h}]\Pr[\bar{\mathcal{E}_h}] \Big) \\
&\leq 2\gamma + \sum_h \Pr[\text{Invalid} \wedge \mathcal{T}_h \wedge \mathcal{E}_h] \qquad (1)
\end{aligned}$$

For each $s \in \mathcal{S}$, let t_s be value such that $\Pr_{A \sim \mathcal{D}}[c^A(F^s_A) > t_s] = \frac{\alpha}{4}$. Note that the existence of t_s is guaranteed by Assumption 6. Further, for each $s \in \mathcal{S}$, $A \in \mathcal{D}$, define $\tilde{c}^A(F^s_A)$ to be $c^A(F^s_A)$ if $c^A(F^s_A) \leq t_s$, and 0 otherwise. In addition, for an iteration h let $Y^s_{v,h}$ be a random variable denoting the second-stage $\tilde{c}$ cost of s for the v-th sample of h, and $Z^s_{v,h}$ be an indicator random variable that is 1 iff the original second-stage cost of s on the v-th sample of h is greater than t_s. We use the following cost functions:

$$\hat{C}_h(s) = c^I(F^s_I) + \frac{1}{N}\sum_{v=1}^{N} Y^s_{v,h} + \frac{t_s}{N}\sum_{v=1}^{N} Z^s_{v,h} \text{ and } \tilde{C}(s) = c^I(F^s_I) + \sum_{A \in \mathcal{D}} p_A \cdot \tilde{c}^A(F^s_A)$$

Also, if $p_s = \Pr_{A \sim \mathcal{D}}[c^A(F^s_A) > t_s]$, then $\mathbb{E}[\hat{C}_h(s)] = \tilde{C}(s) + p_s t_s$. Finally, let $\hat{C}^{II}_h(s) = \hat{C}_h(s) - c^I(F^s_I)$ and $\tilde{C}^{II}(s) = \tilde{C}(s) - c^I(F^s_I)$.

Now observe that if $\text{Invalid} \wedge \mathcal{T}_h \wedge \mathcal{E}_h$ occurs, then there must exist some $s \in \mathcal{S}$ with $\hat{C}_h(s) \leq (1+\epsilon)B$ and $\tilde{C}(s) > (1+2\epsilon)B$. Specifically we have $\hat{C}_h(\bar{s}) \leq (1+\epsilon)B$ and $\tilde{C}(\bar{s}) > (1+2\epsilon)B$. To see why $\hat{C}_h(\bar{s}) \leq (1+\epsilon)B$ is true, note than under this event Alg$\mathcal{P}$ finds a solution in iteration h. The empirical cost of this solution (which corresponds to a restriction of $\bar{s}$) is at most $(1+\epsilon)B$, and the pruning based on the value t_s can only decrease this cost. Regarding $\tilde{C}(\bar{s}) > (1+2\epsilon)B$, under $\text{Invalid} \wedge \mathcal{T}_h \wedge \mathcal{E}_h$ we at first have $C(\hat{s}) > (1+2\epsilon)B$. In addition, $\tilde{C}(\bar{s}) \leq C(\hat{s})$, because by the definitions of t_s and $\mathcal{E}_h$ we have $t_s \geq T^h_s$. Hence, we upper bound the probability of $\text{Invalid} \wedge \mathcal{T}_h \wedge \mathcal{E}_h$ as follows:

$$\begin{aligned}
\Pr[\text{Invalid} \wedge \mathcal{T}_h \wedge \mathcal{E}_h] &\leq \Pr[\exists s \in \mathcal{S} : \ \hat{C}_h(s) \leq (1+\epsilon)B \wedge \tilde{C}(s) > (1+2\epsilon)B] \\
&\leq \Pr[\exists s \in \mathcal{S} : \ \hat{C}_h(s) \leq (1+\epsilon)B \wedge \tilde{C}(s) + p_s t_s > (1+2\epsilon)B + p_s t_s] \\
&\leq \Pr[\exists s \in \mathcal{S} : \ \hat{C}_h(s) \leq (1+\epsilon)B \wedge \mathbb{E}[\hat{C}_h(s)] > (1+2\epsilon)B + p_s t_s] \\
&\leq \Pr[\exists s \in \mathcal{S} : \ \hat{C}^{II}_h(s) \leq (1-\delta_s)\mathbb{E}[\hat{C}^{II}_h(s)]] \\
&\leq \sum_{s \in \mathcal{S}} \Pr\Big[\hat{C}^{II}_h(s) \leq (1-\delta_s)\mathbb{E}[\hat{C}^{II}_h(s)]\Big] \\
&= \sum_{s \in \mathcal{S}} \Pr\Big[N \cdot \hat{C}^{II}_h(s)/t_s \leq (1-\delta_s)N \cdot \mathbb{E}[\hat{C}^{II}_h(s)]/t_s\Big] \qquad (2)
\end{aligned}$$

In the above we defined δ_s such that $\delta_s \geq \frac{\epsilon + p_s t_s}{1+2\epsilon+p_s t_s}$, and also we made use of $B = 1$ and $\mathbb{E}[\hat{C}_h(s)] = \tilde{C}(s) + p_s t_s > 1 + 2\epsilon + p_s t_s$. Applying Lemma 21 gives

$$\Pr[N \cdot \hat{C}^{II}_h(s)/t_s \leq (1-\delta_s)N \cdot \mathbb{E}[\hat{C}^{II}_h(s)]/t_s] \leq e^{\frac{-N(\epsilon+p_s t_s)^2}{2t_s(1+2\epsilon+p_s t_s)}} \qquad (3)$$

We now focus on the quantity $\frac{(\epsilon+p_s t_s)^2}{2t_s(1+2\epsilon+p_s t_s)}$, and consider two distinct cases for $p_s t_s$.

- Suppose $p_s t_s \geq \epsilon$. Then $\frac{(\epsilon+p_s t_s)^2}{2t_s(1+2\epsilon+p_s t_s)} \geq \frac{p_s^2 t_s^2}{2t_s(1+3p_s t_s)} \geq \frac{p_s}{2}\frac{p_s t_s}{1+3p_s t_s} \geq \frac{\epsilon \cdot p_s}{2(1+3\epsilon)}$,where the last inequality follows because $x/(1+3x)$ is increasing and in our case $x \geq \epsilon$.

- Suppose $p_s t_s < \epsilon$. Then $\frac{(\epsilon+p_s t_s)^2}{2t_s(1+2\epsilon+p_s t_s)} \geq \frac{\epsilon^2}{2t_s(1+3\epsilon)} \geq \frac{\epsilon \cdot p_s}{2(1+3\epsilon)}$,where in the last inequality we used the fact that in this case $t_s < \epsilon/p_s$.

Therefore, by definition of p_s, we have $\frac{(\epsilon+p_s t_s)^2}{2t_s(1+2\epsilon+p_s t_s)} \geq \frac{\epsilon \cdot \alpha}{8(1+3\epsilon)}$ in every case. Plugging that in (3), (2), and setting $N = \frac{8(1+\epsilon)}{\epsilon\alpha} \log\left(\frac{t_{\mathcal{P}}(n,m)}{\gamma}(\log_{\frac{13}{12}}(\frac{1}{\gamma})+1)\right)$ gives $\Pr[\text{Invalid} \wedge \mathcal{T}_h \wedge \mathcal{E}_h] \leq \gamma/(\log_{\frac{13}{12}}(\frac{1}{\gamma})+1)$. Finally, using this in (1) gives the desired error probability of at most 3γ. ◀

▶ **Theorem 12.** *For any $\gamma, \alpha \in (0,1)$ and $N = \mathcal{O}\left(\frac{1}{\alpha}\log(\frac{t_{\mathcal{P}}(n,m)}{\gamma})\right)$, the solution strategy $\hat{s}$ satisfies $\Pr_{A\sim\mathcal{D}}[d(j, F_I^{\hat{s}} \cup F_A^{\hat{s}}) \leq \eta R,\ \forall j \in A] \geq 1-2\alpha$ with probability at least $1-\gamma$.*

Proof. Consider some iteration h and strategy $s \in \mathcal{S}$. Let $p_s^h = \Pr_{A\sim\mathcal{D}}[c^A(F_A^s) > T_s^h]$, and $\mathcal{B}_s^h$ the event of having $p_{T_s^h} > 2\alpha$. Suppose that $p_s^h > \alpha$, otherwise $\mathcal{B}_s^h$ cannot occur. Let X_v an indicator random variable that is 1 iff s has stage-II cost larger than T_s^h in the v-th sample. Also, let $X = \sum_{v=1}^{N} X_v$, and recall that $X = \lceil \alpha N \rceil - 1 \leq \alpha N$. Moreover, we have $\mathbb{E}[X] = p_s^h N$ and notice that $2X > \mathbb{E}[X]$ implies $p_s^h < 2\alpha$. Using Lemma 21 with $\delta = 1/2$ we get $\Pr[X \leq \mathbb{E}[X]/2] \leq e^{-p_s^h N/8}$. Because $p_s^h > \alpha$, setting $N = \frac{8}{\alpha}\log(\frac{t_{\mathcal{P}}(n,m)}{\gamma}(\log_{\frac{13}{12}}(\frac{1}{\gamma})+1))$ gives $\sum_h \sum_{s\in\mathcal{S}} \Pr[\mathcal{B}_s^h] \leq \gamma$. ◀

Finally, by optimizing over the radius, we get our main generalization result:

▶ **Theorem 13.** *Assume we have an efficiently generalizable η-approximation for $\mathcal{P}$-**Poly**. Then, using $\mathcal{O}\left(\frac{1}{\epsilon\alpha}\log(\frac{nm\cdot t_{\mathcal{P}}(n,m)}{\gamma})\log\frac{nm}{\gamma}\right)$ samples, we obtain a strategy $\hat{s}$ and a radius R, such that with probability at least $1-\mathcal{O}(\gamma)$ the following hold: (i) $C(\hat{s}) \leq (1+2\epsilon)B$, (ii) $F_I^{\hat{s}} \in \mathcal{M}_I$; (iii) $R \leq R^*$, where R^* is the optimal radius for $\mathcal{P}$-**BB**; (iv) $\Pr_{A\sim\mathcal{D}}[d(j, F_I^{\hat{s}} \cup F_A^{\hat{s}}) \leq \eta R,\ \forall j \in A] \geq 1-2\alpha$.*

Proof. Because R^* is the distance between some facility and some client, there are at most nm alternatives for it. Thus, we can run Algorithm 2 for all possible nm target radius values, using error parameter $\gamma' = \frac{\gamma}{nm}$. We then return the smallest radius that did not yield "INFEASIBLE". By a union bound over all radius choices, the probability of the *Invalid* event in any of them is at most 3γ. Thus, with probability at least $1-3\gamma$, the chosen radius R satisfies $R \leq R^*$, and the opening cost of the corresponding strategy is at most $(1+2\epsilon)B$. Finally, for the returned strategy Theorem 12 holds as well, and the sample bound accounts for all iteration of Algorithm 2.

Additionally, note that we do not need fresh samples for each radius guess R; we can draw an appropriate number of samples N upfront, and test all guesses in "parallel" with the same data. ◀

In light of Theorem 13 and the generic search step for the radius R, we assume for all our $\mathcal{P}$-poly problems that a target radius R is given explicitly.

We conclude with some final remarks. At first, III guarantees $N = \text{poly}(n, m, \frac{1}{\epsilon}, \frac{1}{\alpha}, \log\frac{1}{\gamma})$. Also, the probability 2α of not returning an η-approximate solution can be made inverse polynomially small, without affecting the polynomial nature of the sample complexity.

3 Approximation Algorithm for 2S-Sup-BB

In this section we tackle **2S-Sup-BB**, by first designing a 3-approximation algorithm for **2S-Sup-Poly**, and then proving that the latter is efficiently generalizable.

Algorithm 3 Correlated LP-Rounding Algorithm for **2S-Sup-Poly**.

```
Solve LP (4)-(6) to get a feasible solution $y^I, y^A : A \in Q$;
if no feasible LP solution exists then
    Return "INFEASIBLE";
end
$(H_I, \pi^I) \leftarrow \text{GreedyCluster}(\mathcal{C}, R, g^I)$, where $g^I(j) = y^I(G_j)$ ;
for each scenario $A \in Q$ do
    $(H_A, \pi^A) \leftarrow \text{GreedyCluster}(A, R, g^A)$, where $g^A(j) = -y^I(G_{\pi^I(j)})$ ;
end
Order the clients of $H_I$ as $j_1, j_2, \dots, j_h$ such that $y^I(G_{j_1}) \leq y^I(G_{j_2}) \leq \dots \leq y^I(G_{j_h})$;
Consider an additional "dummy" client $j_{h+1}$ with $y^I(G_{j_{h+1}}) > y^I(G_{j_\ell})$ for all $\ell \in [h]$;
for all integers $\ell = 1, 2, \dots, h+1$ do
    $F^\ell_I \leftarrow \{i^I_{j_k} \mid j_k \in H_I \text{ and } y^I(G_{j_k}) \geq y^I(G_{j_\ell})\}$;
    for each $A \in Q$ do
        $F^\ell_A \leftarrow \{i^A_j \mid j \in H_A \text{ and } F^\ell_I \cap G_{\pi^I(j)} = \emptyset\}$;
    end
    $S_\ell \leftarrow c^I(F^\ell_I) + \sum_{A \in Q} p_A \cdot c^A(F^\ell_A)$;
end
Return $F^{\ell^*}_I, F^{\ell^*}_A : A \in Q$ such that $\ell^* = \arg\min_\ell S_\ell$;
```

3.1 A 3-Approximation Algorithm for 2S-Sup-Poly

We are given a list of scenarios Q together with their probabilities p_A and cost vectors c^A, a target radius R, and let $G_j = G_{j,R}$, $i^I_j = i^I_{j,R}$, $i^A_j = i^A_{j,R}$ for every $j \in \mathcal{C}$ and $A \in Q$. Consider LP (4)-(6).

$$\sum_{i \in \mathcal{F}} y^I_i \cdot c^I_i + \sum_{A \in Q} p_A \sum_{i \in \mathcal{F}} y^A_i \cdot c^A_i \leq B \tag{4}$$

$$\sum_{i \in G_j} (y^I_i + y^A_i) \geq 1, \quad \forall j \in A \in Q \tag{5}$$

$$0 \leq y^I_i, y^A_i \leq 1 \tag{6}$$

Constraint (4) captures the total expected cost, and constraint (5) the fact that for all $A \in Q$, every $j \in A$ must have an open facility within distance R from it. In addition, note that if the LP is infeasible, then there cannot be a solution of radius at most R for the given **2S-Sup-Poly** instance. The rounding algorithm appears in Algorithm 3.

▶ **Theorem 14.** *For any scenario $A \in Q$ and every $j \in A$, we have $d(j, F^{\ell^*}_I \cup F^{\ell^*}_A) \leq 3R$.*

Proof. Focus on some $A \in Q$. Recall that $d(j, \pi^I(j)) \leq 2R$ and $d(j, \pi^A(j)) \leq 2R$ for any $j \in A$. For $j \in H_A$ the statement is clearly true, because either $G_{\pi^I(j)} \cap F^{\ell^*}_I \neq \emptyset$ or $G_j \cap F^{\ell^*}_A \neq \emptyset$. So consider some $j \in A \setminus H_A$. If $G_{\pi^A(j)} \cap F^{\ell^*}_A \neq \emptyset$, then any facility $i \in G_{\pi^A(j)} \cap F^{\ell^*}_A$ will be within distance $3R$ from j. If on the other hand $G_{\pi^A(j)} \cap F^{\ell^*}_A = \emptyset$, then our algorithm guarantees $G_{\pi^I(\pi^A(j))} \cap F^{\ell^*}_I \neq \emptyset$. Further, the stage-II greedy clustering yields $g^A(\pi_A(j)) \geq g^A(j) \implies y^I(G_{\pi^I(j)}) \geq y^I(G_{\pi^I(\pi^A(j))})$. Therefore, from the way we formed $F^{\ell^*}_I$ and the fact that $G_{\pi^I(\pi^A(j))} \cap F^{\ell^*}_I \neq \emptyset$, we infer that $G_{\pi^I(j)} \cap F^{\ell^*}_I \neq \emptyset$. The latter ensures that $d(j, G_{\pi^I(j)} \cap F^{\ell^*}_I) \leq 3R$. ◀

▶ **Theorem 15.** *The opening cost S_{ℓ^*} of Algorithm 3 is at most B.*

Algorithm 4 Generalization Procedure for **2S-Sup-Poly**.

```
Input : Returned sets $F_I$, $F_A : A \in Q$ and inner execution details of Algorithm 3
Let $\bar{s}$ the strategy we will define, and for the stage-I actions set $F^{\bar{s}}_I \leftarrow F_I$;
Suppose scenario $A \in \mathcal{D}$ arrived in the second stage;
For every $j \in A$ set $g(j) \leftarrow -y^I(G_{\pi^I(j)})$, where $y^I, \pi^I$ are the LP solution vector and
  stage-I mapping computed in Algorithm 3;
$(H_A, \pi^A) \leftarrow$ GreedyCluster$(A, R, g)$;
$F^{\bar{s}}_A \leftarrow \{i^A_j \mid j \in H_A$ and $F_I \cap G_{\pi^I(j)} = \emptyset\}$;
```

Proof. Consider the following process to generate a random solution: we draw a random variable β uniformly from $[0,1]$, and then set $F^\beta_I = \{i^I_j \mid j \in H_I \text{ and } y^I(G_j) \geq \beta\}$, $F^\beta_A = \{i^A_j \mid j \in H_A \text{ and } F_I \cap G_{\pi^I(j)} = \emptyset\}$ for all $A \in Q$. For each possible draw for β, the resulting sets F^β_I, F^β_A correspond to sets F^ℓ_I, F^ℓ_A for some integer $\ell \in [h+1]$. Hence, in order to show the existence of an ℓ with $S_\ell \leq B$, it suffices to show $\mathbb{E}_{\beta\sim[0,1]}[c^I(F^\beta_I) + \sum_{A\in Q} p_A \cdot c^A(F^\beta_A)] \leq B$.

We start by calculating the probability of opening a given facility i^I_j with $j \in H_I$ in stage-I. This will occur only if $\beta \leq y^I(G_j)$, and so $\Pr[i^I_j \text{ is opened at stage-I}] \leq \min(y^I(G_j), 1)$. Therefore, due to $G_j \cap G_{j'} = \emptyset$ for all distinct $j, j' \in H_I$, we get:

$$\mathbb{E}_{\beta\sim[0,1]}[c^I(F^\beta_I)] \leq \sum_{j\in H_I} c^I_{i^I_j} \cdot y^I(G_j) \leq \sum_{i\in\mathcal{F}} y^I_i \cdot c^I_i \tag{7}$$

Moreover, for any $j \in H_A$ and any $A \in Q$ we have $\Pr[i^A_j \text{ is opened at stage-II} \mid A] = 1 - \min(y^I(G_{\pi^I(j)}), 1) \leq 1 - \min(y^I(G_j), 1) \leq y^A(G_j)$. The first inequality results from the greedy clustering of stage-I that gives $y^I(G_{\pi^I(j)}) \geq y^I(G_j)$, and the second follows from (5). Thus, due to $G_j \cap G_{j'} = \emptyset$ for all distinct $j, j' \in H_A$, we get:

$$\mathbb{E}_{\beta\sim[0,1]}[c^A(F^\beta_A)] \leq \sum_{j\in H_A} c^A_{i^A_j} \cdot y^A(G_j) \leq \sum_{i\in\mathcal{F}} y^A_i \cdot c^A_i \tag{8}$$

Combining (7), (8) and (4) gives $\mathbb{E}_{\beta\sim[0,1]}[c^I(F^\beta_I)] + \sum_{A\in Q} p_A \cdot \mathbb{E}_{\beta\sim[0,1]}[c^A(F^\beta_A)] \leq B$. ◀

3.2 Generalizing to the Black-Box Setting

To show that Algorithm 3 fits the framework of Section 2, we must show that it is efficiently generalizable as in Definition 8. For one thing, it is obvious that Algorithm 3 satisfies the properties of Definition 7, and therefore is a valid 3-approximation. Hence, we only need a process to efficiently extend its output to any arriving scenario $A \in \mathcal{D}$, where $\mathcal{D}$ the black-box distribution. This is demonstrated in Algorithm 4, which mimics the stage-II actions of Algorithm 3. Here we crucially exploit the fact that the stage-II decisions of Algorithm 3 only depend on information from the LP about stage-I variables.

Since Algorithm 4 exactly imitates the stage-II actions of Algorithm 3, it is easy to see that property II is satisfied. Further, the arguments in Theorem 14 would still apply, and eventually guarantee $d(j, F^{\bar{s}}_I \cup F^{\bar{s}}_A) \leq 3R$ for all $j \in A$ and any $A \in \mathcal{D}$, thus verifying property I. To conclude, we only need to prove III. Let $\mathcal{S}_K$ the set of strategies achievable via Algorithm 4.

▶ **Lemma 16.** *Algorithm 3 satisfies property III with $|\mathcal{S}_K| \leq (n+1)!$.*

Algorithm 5 Rounding Algorithm for **2S-MatSup-Poly**.

```
Solve LP (9)-(12) to get a feasible solution $y^I, y^A$ for all $A \in Q$;
if no feasible LP solution exists then
    Return “INFEASIBLE”;
end
$(H_I, \pi^I) \leftarrow \text{GreedyCluster}(\mathcal{C}, R, g^I)$ where $g^I(j) = y^I(G_j)$ ;
Let $g^{II} : \mathcal{C} \mapsto [n]$ be some fixed and given bijective mapping;
for each scenario $A \in Q$ do
    $(H_A, \pi^A) \leftarrow \text{GreedyCluster}(A, R, g^{II})$ ;
end
Solve LP (13)-(16) and get an optimal integral solution $z^*$, such that
 $z^*_i \in \{0,1\}\ \forall i \in \mathcal{F}$;
$F_I \leftarrow \{i \in \mathcal{F} \mid z^*_i = 1\}$;
$F_A \leftarrow \{i^A_j \in \mathcal{F} \mid j \in H_A$ and $G_{\pi^I(j)} \cap F_I = \emptyset\}$ for every $A \in Q$.
```

Proof. The constructed final strategy is determined by 1) the sorted order of $y^I(G_j)$ for all $j \in \mathcal{C}$, and 2) a minimum threshold ℓ' such that $G_{j_{\ell'}} \cap F_I \neq \emptyset$ with $j_{\ell'} \in H_I$. Given those, we know exactly what H_I and H_A for every $A \in \mathcal{D}$ will be, as well as F_I and F_A for every $A \in \mathcal{D}$. The set of all possible such options is also independent Q. Since there are $n!$ total possible orderings for the $y^I(G_j)$ values, and the threshold parameter ℓ' can take at most $n+1$ values, we get $|\mathcal{S}_K| \leq (n+1)!$. ◀

4 Approximation Algorithm for 2S-MatSup-BB

The outline of this section is similar to that of Section 3. We begin with a 5-approximation algorithm for **2S-MatSup-Poly**, and then show that it is also efficiently generalizable.

4.1 A 5-Approximation Algorithm for 2S-MatSup-Poly

We are given a radius R, and a list of scenarios Q together with their probabilities p_A and cost vectors c^A. Moreover, assume that $r_{\mathcal{M}}$ is the rank function of the input matroid $\mathcal{M} = (\mathcal{F}, \mathcal{I})$. We also use the notation $G_j = G_{j,R}$, and $i^A_j = i^A_{j,R}$ for every $j \in \mathcal{C}$ and $A \in Q$. Consider LP (9)-(12).

$$\sum_{i \in \mathcal{F}} y^I_i \cdot c^I_i + \sum_{A \in Q} p_A \sum_{i \in \mathcal{F}} y^A_i \cdot c^A_i \leq B \tag{9}$$

$$\sum_{i \in G_j} (y^I_i + y^A_i) \geq 1, \quad \forall j \in A \in Q \tag{10}$$

$$\sum_{i \in U} y^I_i \leq r_{\mathcal{M}}(U), \quad \forall U \subseteq \mathcal{F} \tag{11}$$

$$0 \leq y^I_i, y^A_i \leq 1 \tag{12}$$

Compared to LP (4)-(6), the only difference lies in constraint (11), which exactly represents the stage-I matroid requirement. Hence, it is a valid relaxation for the problem. Although the LP has an exponential number of constraints, it can be solved in polynomial time via the Ellipsoid algorithm, with a separation oracle based on minimizing a submodular function [13].

Assuming LP feasibility, our algorithm (presented in full detail in Algorithm 5), begins with two greedy clustering steps, one for each stage, that produce sets H_I, $H_A : A \in Q$ with corresponding mappings π^I and π^A. We then set up and solve the auxiliary LP shown in (13)-(16), and use this solution to determine sets F_I and F_A.

$$\text{minimize} \sum_{i \in \mathcal{F}} z_i \cdot c_i^I + \sum_{A \in Q} p_A \sum_{j \in H_A} c^A_{i^A_j}(1 - z(G_{\pi^I(j)})) \tag{13}$$

$$\text{subject to } z(G_j) \leq 1, \ \forall j \in H_I \tag{14}$$

$$z(U) \leq r_{\mathcal{M}}(U), \ \forall U \subseteq \mathcal{F} \tag{15}$$

$$0 \leq z_i \leq 1 \tag{16}$$

▶ **Lemma 17.** *If LP (9)-(12) is feasible, then the optimal solution z^* of the auxiliary LP (13)-(16) has objective function value at most B, and is integral (i.e. for all $i \in \mathcal{F}$ we have $z^*_i \in \{0,1\}$).*

Proof. Solution z^* is integral since the LP (13)-(16) is the intersection of two matroid polytopes, namely, the polytope correspondind to $\mathcal{M}$, and a partition matroid polytope over all G_j with $j \in H_I$. (Recall that sets G_j for $j \in H_I$ are pairwise disjoint.)

Now let y^I, y^A be a feasible solution of (9)-(12). For all $j \in H_I$ with $y^I(G_j) \leq 1$, set $z_i = y^I_i$ for all $i \in G_j$. For all $j \in H_I$ with $y^I(G_j) > 1$, set $z_i = y^I_i / y^I(G_j)$ for all $i \in G_j$. For the rest of the facilities set $z_i = 0$. This solution obviously satisfies (14). Also, because y^I satisfies (11) and $z_i \leq y^I_i$ for all i, we know that z satisfies (15) too. Finally, regarding the objective function:

$$\sum_{i \in \mathcal{F}} z_i \cdot c_i^I \leq \sum_{i \in \mathcal{F}} y_i \cdot c_i^I \tag{17}$$

For the second-stage cost we then get:

$$\begin{aligned}\sum_{A \in Q} p_A \sum_{j \in H_A} c^A_{i^A_j}(1 - z(G_{\pi^I(j)})) &\leq \sum_{A \in Q} p_A \sum_{\substack{j \in H_A: \\ y^I(G_{\pi^I(j)}) \leq 1}} c^A_{i^A_j}(1 - y^I(G_{\pi^I(j)})) \\ &\leq \sum_{A \in Q} p_A \sum_{\substack{j \in H_A: \\ y^I(G_{\pi^I(j)}) \leq 1}} c^A_{i^A_j}(1 - y^I(G_j)) \\ &\leq \sum_{A \in Q} p_A \sum_{\substack{j \in H_A: \\ y^I(G_{\pi^I(j)}) \leq 1}} c^A_{i^A_j} y^A(G_j) \leq \sum_{A \in Q} p_A \sum_{i \in \mathcal{F}} y^A_i c^A_i \end{aligned} \tag{18}$$

The second line follows from the stage-I greedy clustering, which ensures $y^I(G_{\pi^I(j)}) \geq y^I(G_j)$ for all $j \in \mathcal{C}$. The last line is due to (10), and the fact that for all $A \in Q$ and all distinct $j, j' \in H_A$ we have $G_j \cap G_{j'} = \emptyset$. Finally, combining (9), (17) and (18) we get the desired bound on the cost. ◀

▶ **Theorem 18.** *For the sets F_I, $F_A : A \in Q$ returned by Algorithm 5 the following three properties hold: (i) $F_I \in \mathcal{I}$, (ii) $c^I(F_I) + \sum_{A \in Q} p_A c^A(F_A) \leq B$, and (iii) $d(j, F_I \cup F_A) \leq 5R$ for all $j \in A \in Q$.*

Proof. (i) is obvious, since z^* satisfies constraint (15). For (ii), the opening cost of the solution coincides with the value of the objective (13) for z^*, and hence by Lemma 17 it is at most B.

For (iii), consider $A \in Q$, and recall that $d(j, \pi^I(j)) \leq 2R$ and $d(j, \pi^A(j)) \leq 2R$ for any $j \in A$. For $j \in H_A$ the bound (iii) holds, because either $G_{\pi^I(j)} \cap F_I \neq \emptyset$ or $G_j \cap F_A \neq \emptyset$. So suppose that $j \in A \setminus H_A$. If $G_{\pi^A(j)} \cap F_A \neq \emptyset$, then any facility $i \in G_{\pi^A(j)} \cap F_A$ will be within distance $3R$ from j. If on the other hand $G_{\pi^A(j)} \cap F_A = \emptyset$, then there exists $i \in G_{\pi^I(\pi^A(j))} \cap F_I$. Therefore, $d(i,j) \leq d(i, \pi^I(\pi^A(j))) + d(\pi^I(\pi^A(j)), \pi^A(j)) + d(\pi^A(j), j) \leq 5R$. ◀

Algorithm 6 Generalization Procedure for **2S-MatSup-Poly**.

```
Input : Returned sets $F_I$, $F_A : A \in Q$ and inner execution details of Algorithm 5
Let $\bar{s}$ the strategy we will define, and for the stage-I actions set $F_I^{\bar{s}} \leftarrow F_I$;
Suppose scenario $A \in \mathcal{D}$ arrived in the second stage;
Let $\pi^I$ the stage-I mapping and $g^{II}$ the bijective function, both used in Algorithm 5;
Set $(H_A, \pi^A) \leftarrow \text{GreedyCluster}(A, R, g^{II})$;
Open the set $F_A^{\bar{s}} = \{i_j^A \mid j \in H_A \text{ and } F_I \cap G_{\pi^I(j)} = \emptyset\}$;
```

4.2 Generalizing to the Black-Box Setting

It is clear that Algorithm 5 satisfies Definition 7, and therefore is a valid 5-approximation. Consider now Algorithm 6 to efficiently extend its output to any arriving scenario $A \in \mathcal{D}$. Since Algorithm 6 exactly imitates the stage-II actions of Algorithm 5, it is easy to see that property II is satisfied. Furthermore, the arguments in Theorem 18 would still go through, and eventually guarantee $d(j, F_I^{\bar{s}} \cup F_A^{\bar{s}}) \leq 5R$ for all $j \in A$ and any $A \in \mathcal{D}$, thus verifying property I. To conclude, we only need to prove III. Let $\mathcal{S}_M$ the set of strategies achievable via Algorithm 6.

▶ **Lemma 19.** *Algorithm 5 satisfies property III with* $|\mathcal{S}_M| = 2^m \cdot n!$.

Proof. Since g^{II} can be thought of as part of the input, $\bar{s}$ depends only on 1) the set F_I returned by Algorithm 5, and 2) the sorted order of $y^I(G_j)$ for all $j \in \mathcal{C}$, which ultimately dictates the mapping π^I. Given those, we can determine the stage-II openings for every possible scenario $A \in \mathcal{D}$. These options do not depend on scenarios Q. The total number of possible outcomes for F_I is 2^m, and the total number of orderings for the clients of $\mathcal{C}$ is $n!$. Hence, $|\mathcal{S}_M| = 2^m \cdot n!$. ◀

References

1 Shipra Agrawal, Amin Saberi, and Yinyu Ye. Stochastic combinatorial optimization under probabilistic constraints, 2008. arXiv:0809.0460.

2 E. M. L. Beale. On minimizing a convex function subject to linear inequalities. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 173–184, 1955.

3 Deeparnab Chakrabarty and Maryam Negahbani. Generalized center problems with outliers. *ACM Trans. Algorithms*, 2019.

4 Moses Charikar, Chandra Chekuri, and Martin Pal. Sampling bounds for stochastic optimization. In *Approximation, Randomization and Combinatorial Optimization. Algorithms and Techniques*, pages 257–269, 2005.

5 George B. Dantzig. Linear programming under uncertainty. *Management Science*, pages 197–206, 1955.

6 Uriel Feige. On sums of independent random variables with unbounded variance and estimating the average degree in a graph. *SIAM Journal on Computing*, 35(4):964–984, 2006.

7 A. Gupta, R. Ravi, and A. Sinha. An edge in time saves nine: Lp rounding approximation algorithms for stochastic network design. In *45th Annual IEEE Symposium on Foundations of Computer Science*, pages 218–227, 2004.

8 Anupam Gupta, Martin Pál, R. Ravi, and Amitabh Sinha. Boosted sampling: Approximation algorithms for stochastic optimization. In *Proceedings of the Thirty-Sixth Annual ACM Symposium on Theory of Computing*, STOC '04, page 417–426, 2004.

9 Anupam Gupta, Martin Pal, R. Ravi, and Amitabh Sinha. Sampling and cost-sharing: Approximation algorithms for stochastic optimization problems. *SIAM J. Comput.*, pages 1361–1401, 2011.

10 Dorit S. Hochbaum and David B. Shmoys. A unified approach to approximation algorithms for bottleneck problems. *J. ACM*, 1986.

11 Nicole Immorlica, David Karger, Maria Minkoff, and Vahab S. Mirrokni. On the costs and benefits of procrastination: Approximation algorithms for stochastic combinatorial optimization problems. In *Proceedings of the Fifteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 691–700, 2004.

12 Anton J. Kleywegt, Alexander Shapiro, and Tito Homem-de Mello. The sample average approximation method for stochastic discrete optimization. *SIAM Journal on Optimization*, 12(2):479–502, 2002.

13 Lap-Chi Lau, R. Ravi, and Mohit Singh. *Iterative Methods in Combinatorial Optimization*. Cambridge University Press, USA, 1st edition, 2011.

14 Andre Linhares and Chaitanya Swamy. Approximation algorithms for distributionally-robust stochastic optimization with black-box distributions. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 768–779, 2019.

15 Andrea Pietracaprina, Geppino Pucci, and Federico Solda. Coreset-based strategies for robust center-type problems, 2020. `arXiv:2002.07463`.

16 R. Ravi and Amitabh Sinha. Hedging uncertainty: Approximation algorithms for stochastic optimization problems. In *Integer Programming and Combinatorial Optimization*, 2004.

17 David Shmoys and Chaitanya Swamy. An approximation scheme for stochastic linear programming and its application to stochastic integer programs. *J. ACM*, 53:978–1012, November 2006.

18 Aravind Srinivasan. Approximation algorithms for stochastic and risk-averse optimization. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1305–1313, 2007.

19 C. Swamy and D. B. Shmoys. Sampling-based approximation algorithms for multi-stage stochastic optimization. In *46th Annual IEEE Symposium on Foundations of Computer Science (FOCS'05)*, pages 357–366, 2005.

20 Chaitanya Swamy. Risk-averse stochastic optimization: Probabilistically-constrained models and algorithms for black-box distributions: (extended abstract). *Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1627–1646, 2011.

21 Chaitanya Swamy and David Shmoys. The sample average approximation method for 2-stage stochastic optimization. *Survey Paper*, April 2008.

22 Chaitanya Swamy and David B. Shmoys. Approximation algorithms for 2-stage stochastic optimization problems. *SIGACT News*, pages 33–46, 2006.

A Applying the Standard SAA Method in Supplier Problems

Consider the standard two-stage stochastic setting. In the first stage, we are allowed to take some proactive actions and commit to an anticipatory part of the solution x, which will incur some cost $c(x)$. In the second stage, a scenario A is sampled from the distribution $\mathcal{D}$, and we can take some *stage-II* recourse actions y_A with cost $f_A(x, y_A)$. If X is the set of stage-I actions and Y the set of recourse actions, the goal is to find a solution $x^\star \in X$ to minimize $f(x) = c(x) + \mathbb{E}_{A\sim\mathcal{D}}[q_A(x)]$, where $q_A(x) = \min_{y\in Y}\{f_A(x,y) \mid (x,y) \text{ is a valid solution for } A\}$.

The Standard SAA Method. Consider minimizing $f(x)$ in the black-box model. If S is a set of scenarios sampled from the black-box oracle, let $\hat{f}(x) = c(x) + \big(\sum_{A\in S} q_A(x)\big)/|S|$ be the empirical estimate of $f(x)$. Also, let x^* and $\bar{x}$ be the minimizers of $f(x)$ and $\hat{f}(x)$ respectively.

The work [21] shows that if $f(x)$ is modeled as a convex program, then for any $\epsilon, \gamma \in (0,1)$ and with $|S| = \text{poly}(n, m, \lambda, \epsilon, 1/\gamma)$, we have $f(\bar{x}) \leq (1+\epsilon)f(x^*)$ with probability at least $1-\gamma$ (λ is the maximum multiplicative factor by which an element's cost is increased in stage-II). An alternate proof of this appeared in [4], which also covered the case of $f(x)$ being an integer program. Moreover, [4] proves that if $\bar{x}$ is an α-approximate minimizer of $\hat{f}(x)$, then a slight modification to the sampling still gives $f(\bar{x}) \leq (\alpha + \epsilon)f(x^*)$ with probability at least $1 - \gamma$.

The result of [4] further implies that the black-box model can be effectively reduced to the polynomial-scenarios one, via the following process. Assuming that $f(x)$ corresponds to the integer program modeling our problem, first find an α-approximate minimizer $\bar{x}$ of $\hat{f}(x)$, and treat $\bar{x}$ as the stage-I actions. Then, given any arriving A, re-solve the problem using any known ρ-approximation algorithm for the non-stochastic counterpart, with $\bar{x}$ as a fixed part of the solution. This process eventually leads to an overall approximation ratio of $\alpha\rho + \epsilon$.

Roadblocks for the Standard SAA Analysis in Supplier Problems. A natural way to fit our models within the existing framework, is to first assume knowledge of the optimal radius R^* and then use the opening cost as the objective function $f_{R^*}(x)$, by turning the radius requirement into a simple covering constraint. In other words, we set $f_{R^*}(x) = c^I(x) + \mathbb{E}_{A \sim \mathcal{D}}[q_{A,R^*}(x)]$ with $q_{A,R^*}(x) = \min_y\{c^A(y) \mid (x, y) \text{ covers all } j \in A \text{ within distance } R^*\}$. Note that $f_{R^*}(x)$ may represent both the convex and the integer program corresponding to the underlying problem.

To avoid any overhead in the approximation ratio (from re-solving the problem in stage-II), one should apply SAA to the function $f_{R^*}(x)$ corresponding to the convex program describing the problem (the roadblock described here trivially extends to the case of $f_{R^*}(x)$ being an integer function as well). If there exists a rounding that turns the empirical minimizer $\bar{x}_{R^*}$ into a solution that covers each client within distance αR^*, while also having an opening cost of at most $f_{R^*}(\bar{x}_{R^*})$, we get the desired result because $f_{R^*}(\bar{x}_{R^*}) \leq (1+\epsilon)f_{R^*}(x^*_{R^*})$ and $f_{R^*}(x^*_{R^*}) \leq B$. With slight modifications, all our polynomial-scenarios algorithms can be interpreted as such rounding procedures.

Nonetheless, we still have to identify a good guess for R^*, **and this constitutes an unavoidable roadblock in applying standard SAA** in supplier problems. Since R^* is one of nm alternative options, one can test each of those individually. Hence, assume we work with some guess R, and define the corresponding cost functions $f_R, \hat{f}_R$ with minimizers $x^*_R, \bar{x}_R$ respectively. Observe that R is a good guess iff $f_R(x^*_R) \leq (1 + \mathcal{O}(\epsilon))B$, since in this way vanilla SAA combined with our rounding procedures yields an opening cost of $f_R(\bar{x}_R) \leq (1+\epsilon)f_R(x^*_R)$, and minimizing over the radius is just a matter of finding the minimum good guess. However, because $f_R(x)$ is not efficiently computable, the only way to test if R is a good guess, is through $\hat{f}_R(x)$. Unfortunately, empirically estimating $f_R(x)$ within an $(1+\epsilon)$ factor may require a super-polynomial number of samples [12]. The reason for this is the existence of scenarios with high stage-II cost appearing with small probability, which drastically increase the variance of $\hat{f}_R(x)$. **On a high level, the obstacle in supplier problems stems from the need to not only find a minimizer $\boldsymbol{\bar{x}_R}$, but also compute its corresponding value $\boldsymbol{f_R(\bar{x}_R)}$**. This makes it impossible to know which guesses R are good, and consequently there is no way to optimize over the radius.

Finally, note that if the stage-II cost of every scenario is polynomially bounded, the variance of $\hat{f}_R(x)$ is also polynomial, and standard SAA arguments go through without difficulties. However, this assumption is much stronger than is typically used for the two-stage stochastic model.

B Auxiliary Lemmas

▶ **Lemma 20** ([6]). *Let $X_1, \dots, X_K$ be non-negative independent random variables, with expectations $\mu_1, \dots, \mu_K$ respectively, where $\mu_k \leq 1$ for every k. Let $X = \sum_{k=1}^{K} X_i$, and let $\mu = \sum_{k=1}^{K} \mu_i = \mathbb{E}[X]$. Then for every $\delta > 0$ we have $\Pr[X < \mu + \delta] \geq \min\{\frac{\delta}{1+\delta}, \frac{1}{13}\}$.*

The two following lemmas are standard Chernoff bounds.

▶ **Lemma 21.** *Let $X_1, X_2, \dots, X_K$ be independent random variables with $X_k \in [0,1]$ for every k. For $X = \sum_{k=1}^{K} X_k$ with $\mu = \mathbb{E}[X]$ and any $\delta > 0$, we have $\Pr[X \leq (1-\delta)\mu] \leq e^{\frac{-\mu\delta^2}{2}}$.*

▶ **Lemma 22.** *Let $X_1, X_2, \dots, X_K$ be independent Bernoulli random variables with parameter p. Let $X = \sum_{k=1}^{K} X_k$ the corresponding binomial random variable. If for the realization of X we have $X = qK$, then for any $\delta > 0$ we have $\Pr[p < q - \delta] \leq e^{-K\delta^2/2p}$*

C Approximation Algorithm for 2S-MuSup-BB

To tackle this, we construct an efficiently generalizable algorithm for **2S-MuSup-Poly**, via an intriguing reduction to a non-stochastic clustering problem with outliers. Specifically, if we view stage-I as consisting of a deterministic robust problem, stage-II is interpreted as covering all outliers left over by stage-I. Formally, we use the following robust problem:

Robust Weighted Multi-Knapsack-Supplier. We are given a set of clients $\mathcal{C}$ and a set of facilities $\mathcal{F}$, in a metric space with distance function d. The input also includes parameters $V, R \in \mathbb{R}_{\geq 0}$, and for every client $j \in \mathcal{C}$ an associated weight $v_j \in \mathbb{R}_{\geq 0}$. In addition, we have the same types of multi-knapsack constraints as in **2S-MuSup**: there are L in total budgets W_ℓ, and every facility $i \in \mathcal{F}$ has costs f^ℓ_i for $\ell \in [L]$. The goal is to choose a set of facilities $S \subseteq \mathcal{F}$, such that $\sum_{j \in \mathcal{C}: d(j,S) > R} v_j \leq V$ and $f^\ell(S) \leq W_\ell$ for every $\ell \in [L]$. Clients j with $d(j,S) > R$ are called outliers. Finally, an instance of this problem is called *discrete*, if the values f^ℓ_i are all integers.

We first show that any ρ-approximation for **Robust Weighted Multi-Knapsack-Supplier** can be used in order to get an efficiently generalizable $(\rho+2)$-approximation algorithm for **2S-MuSup-Poly**. In addition, we argue that already existing work [3, 15] gives a 3-approximation for discrete instances of **Robust Weighted Multi-Knapsack-Supplier**, thus leading to an efficiently generalizable 5-approximation for discrete instances of **2S-MuSup-Poly**.

C.1 Reducing 2S-MuSup-Poly to Robust Weighted Multi-Knapsack-Supplier

We first suppose that the costs c^I_i are polynomially bounded integers, and claim that this restriction will be removed when we generalize to the black-box setting. Once more, let Q be a set of provided scenarios, R a target radius, and $G_j = G_{j,R}$, $i^A_j = i^A_{j,R}$ for all $j \in \mathcal{C}$ and $A \in Q$. Furthermore, suppose that we have a ρ-approximation algorithm RW for **Robust Weighted Multi-Knapsack-Supplier**. For a feasible instance $\mathfrak{I}'$ of the latter problem, RW returns a solution S satisfying all knapsack constraints and also $\sum_{j \in \mathcal{C}: d(j,S) > \rho R} v_j \leq V$. Otherwise, it either returns "INFEASIBLE", or again a solution with the previous properties.

If the provided instance $\mathfrak{I}$ of **2S-MuSup-Poly** is feasible, the first step in tackling the problem is figuring out the portion of the budget, say B_I, that is used in the first stage of a

Algorithm 7 Approximation Algorithm for **2S-MuSup-Poly**.

```
Let $g^{II} : \mathcal{C} \mapsto [n]$ be some fixed and given bijective mapping;
for each scenario $A \in Q$ do
    $(H_A, \pi^A) \leftarrow$ GreedyCluster$(A, R, g^{II})$;
end
Construct instance $\mathfrak{I}'$ of Robust Weighted Multi-Knapsack-Supplier as
  discussed;
if $RW(\mathfrak{I}')$ = "INFEASIBLE" then
    Return "INFEASIBLE";
end
$F_I \leftarrow RW(\mathfrak{I}')$;                                   // Stage-I facilities
for each scenario $A \in Q$ do
    $F_A \leftarrow \{i^A_j \mid j \in H_A$ with $d(j, F_I) > \rho R\}$;        // Stage-II facilities
end
```

feasible solution. Since the costs c^I_i are polynomially bounded integers, we can guess B_I in polynomial time through solving the problem for all different alternatives for it. So from this point on, assume w.l.o.g. that we have the correct B_I, and also let $B_{II} = B - B_I$.

Algorithm 7 shows how to use RW to approximate **2S-MuSup-Poly**. It begins with greedy clustering steps for each A, and given H_A, π^A it constructs an instance $\mathfrak{I}'$ of **Robust Weighted Multi-Knapsack-Supplier** as follows. $\mathcal{C}$, $\mathcal{F}$, d, and R are the same for both problems. For all $j \in \mathcal{C}$ we set $v_j = \sum_{A \in Q : j \in H_A} p_A \cdot c^A_{i^A_j}$ and also $V = B_{II}$. Finally, the instance $\mathfrak{I}'$ has $L' = L + 1$ knapsack constraints, where the first L are the stage-I constraints of **2S-MuSup-Poly** ($f^\ell(S) \leq W_\ell$), and the last is $c^I(S) \leq B_I$.

▶ **Lemma 23.** *If the original **2S-MuSup-Poly** instance $\mathfrak{I}$ is feasible, then the **Robust Weighted Multi-Knapsack-Supplier** instance $\mathfrak{I}'$ is also feasible.*

Proof. Consider some feasible solution $F^\star_I, F^*_A$ for **2S-MuSup-Poly**. We claim that $F^\star_I$ is a valid solution for $\mathfrak{I}'$. It clearly satisfies the L knapsack constraints of the form $f^\ell(F^\star_I) \leq W_\ell$, and if our guess B_I is the right one, it also satisfies $c^I(F^\star_I) \leq B_I$. Now, for any $A \in Q$, any client $j \in H_A$ with $d(j, F^\star_I) > R$ must be covered by some facility $x^A_j \in G_j \cap F^\star_A$. Since B_{II} is the second-stage portion of the budget used by $F^\star_I, F^*_A$, and $G_{j'} \cap G_{j''} = \emptyset$ for all distinct $j', j'' \in H_A$ we have:

$$B_{II} \geq \sum_A p_A \sum_{i \in F^\star_A} c^A_i \geq \sum_A p_A \sum_{\substack{j \in H_A: \\ d(j, F^\star_I) > R}} c^A_{x^A_j} \geq \sum_A p_A \sum_{\substack{j \in H_A: \\ d(j, F^\star_I) > R}} c^A_{i^A_j} = \sum_{\substack{j \in \mathcal{C}: \\ d(j, F^\star_I) > R}} v_j$$

This implies that $S = F^\star_I$ satisfies the constraint $\sum_{j : d(j,S) > R} v_j \leq B_{II}$ of instance $\mathfrak{I}'$. ◀

▶ **Theorem 24.** *Algorithm 7 is a valid $(\rho + 2)$-approximation for **2S-MuSup-Poly**.*

Proof. First of all, Lemma 23 guarantees that if the given instance of **2S-MuSup-Poly** is feasible, we will get a solution F_I, F_A. By specification of RW, $c^I(F_I) \leq B_I$ and $f^\ell(F_I) \leq W_\ell$ for every ℓ. The stage-II cost C_{II} of this solution is given by:

$$C_{II} = \sum_A p_A \sum_{\substack{j \in H_A: \\ d(j, F_I) > \rho R}} c^A_{i^A_j} = \sum_{\substack{j \in \mathcal{C}: \\ d(j, F_I) > \rho R}} v_j \leq B_{II},$$

Algorithm 8 Generalization Procedure for **2S-MuSup-Poly**.

```
Input : Returned sets $F_I$, $F_A : A \in Q$ and inner execution details of Algorithm 7
Let $\bar{s}$ the strategy we will define, and for the stage-I actions set $F_I^{\bar{s}} \leftarrow F_I$;
Suppose scenaio $A$ arrived in the second stage;
$(H_A, \pi^A) \leftarrow \text{GreedyCluster}(A, R, g^{II})$, where $g^{II}$ the bijective function used in
  Algorithm 7;
Open the set $F_A^{\bar{s}} \leftarrow \{i_j^A \mid j \in H_A \text{ and } d(j, F_I) > \rho R\}$;
```

where the last inequality follows because F_I is the output of $RW(\mathfrak{I}')$.

Consider now a $j \in A$ for some $A \in Q$. The distance of j to its closest facility will be at most $d(\pi^A(j), F_I \cup F_A) + d(j, \pi^A(j))$. Since $\pi^A(j) \in H_A$, there will either be a stage-I open facility within distance ρR from it, or we perform a stage-II opening in $G_{\pi(j)}$, which results in a covering distance of at most R. Also, by the greedy clustering step, we have $d(j, \pi^A(j)) \leq 2R$. So in the end we get $d(j, F_I \cup F_A) \leq (\rho + 2)R$. ◀

By combining Algorithm 7 with existing 3-approximation algorithms for **Robust Weighted Multi-Knapsack-Supplier**, we get the following result:

▶ **Theorem 25.** *There is a* 5*-approximation algorithm for discrete instances of* ***2S-MuSup-Poly****, where additionally all c_i^I are polynomially bounded integers. The runtime of it is* poly(n, m, Λ)*.*

Proof. The results of [3] give a 3-approximation for discrete instances of **Robust Weighted Multi-Knapsack-Supplier**, when $v_j = 1$ for all j. The work of [15] extends this to allow arbitrary v_j values. Note that by our assumption that the values c^I are polynomially bounded integers, the instance $\mathfrak{I}'$ is discrete, and hence the algorithm of [15] can be utilized in Algorithm 7 and give a 5-approximation for **2S-Sup-Poly**. Finally, given the results in [3, 15], the runtime of the whole process will be poly(n, m, Λ). ◀

C.2 Generalizing to the Black-Box Setting

Since the algorithm of Section C.1 is a valid $(\rho + 2)$-approximation, consider the process in Algorithm 8, which efficiently extends its output to any arriving scenario $A \in \mathcal{D}$.

Because Algorithm 8 exactly mimics the stage-II actions of Algorithm 7, it is easy to see that property II is satisfied. Further, the arguments of Theorem 24 would still ensure $d(j, F_I \cup F_A) \leq (\rho + 2)R$ for every $j \in A$ and $A \in \mathcal{D}$, thus guaranteeing property I. To conclude, we again only need to prove property III. Let $\mathcal{S}_{MK}$ the set of strategies achievable via Algorithm 8.

▶ **Lemma 26.** *Algorithm 7 satisfies property III with $|\mathcal{S}_{MK}| = 2^m$.*

Proof. The returned final strategy depends solely on the set F_I. Given that, we can exactly determine all possible stage-II openings, since every H_A for $A \in \mathcal{D}$ can be computed using the fixed function g^{II}. There are 2^m choices for F_I, and therefore $|\mathcal{S}_{MK}| = 2^m$. Finally, it is easy to see that the set $\mathcal{S}_{MK}$ is independent of Q . ◀

Our algorithm for **2S-MuSup-Poly** requires the values c_i^I to be polynomially bounded integers. As we show next, this assumption can be removed by a standard rescaling trick:

▶ **Theorem 27.** *Suppose that the c^I_i are arbitrary numbers. By appropriate cost-quantization for any $\epsilon \in (0,1)$, Algorithm 7 can be modified to give a solution $F_I, F_A : A \in Q$ for **2S-MuSup-Poly**, where $d(j, F_I \cup F_A) \leq (\rho+2)R$ for all $A \in Q$, $j \in A$, and also $c^I(F_I) + \sum_{A \in Q} p_A c^A(F_A) \leq (1+\epsilon)B$.*

Proof. For convenience, let us assume that $B = 1$, and suppose that all facilities have $c^I_i \leq B = 1$ (as otherwise they can never be opened). Given some $\epsilon > 0$, let us define $q = \epsilon/m$, and form new costs by $\tilde{c}^I_i = \lceil c^I_i/q \rceil, \tilde{c}^A_i = c^A_i/q, B' = B(1+\epsilon)/q$. The costs $\tilde{c}^I_i$ are at most $\lceil 1/q \rceil$, and hence are polynomially-bounded integers. Therefore, the reduction of Section C.1 can be applied.

Suppose now that F_I, F_A is a solution to the original instance of **2S-MuSup-Poly**, with opening cost at most B. For the modified cost of this solution we then have:

$$\tilde{c}^I(F_I) + \sum_A p_A \tilde{c}^A(F_A) \leq (c^I(F_I) + \sum_A p_A c^A(F_A))/q + \sum_{i \in \mathcal{F}} 1 \leq B/q + m \leq B'$$

Thus, F_I, F_A is also a solution to the modified instance, implying that the latter is feasible. Hence, consider any solution $\tilde{F}_I, \tilde{F}_A$ to the modified instance, that we would get after running Algorithm 7 with the new costs; its opening cost in the original instance is $c^I(\tilde{F}_I) + \sum_A p_A c^A(\tilde{F}_A) \leq q\tilde{c}^I(\tilde{F}_I) + q\sum_A p_A \tilde{c}^A(\tilde{F}_A) \leq qB' = B(1+\epsilon)$. Therefore, since $\tilde{F}_I, \tilde{F}_A$ is a $(\rho+2)$-approximate solution, we get the desired result. ◀

Note that applying our generalization framework on this solution would make the overall cost over $\mathcal{D}$ be at most $(1+\mathcal{O}(\epsilon))(1+\epsilon)B = (1+\mathcal{O}(\epsilon))B$, which implies that in the black-box setting we do not need the initial assumption for the costs c^I_i.

C.3 Connections to 2S-MatSup

Suppose we define our non-stochastic robust problem as having one knapsack and one matroid constraint, instead of L knapsack constraints. Then the reduction of Section C.1 would yield a $(\rho+2)$-approximation for **2S-MatSup-Poly** in the exact same manner, where ρ the ratio of the algorithm used to solve the corresponding deterministic outliers problem.

A result of [3, Theorem 16] gives a 3-approximation for this outliers problem, which in turn would give a 5-approximation for **2S-MatSup-Poly**. However, the algorithm obtained in this way would be randomized (its solution may not be a valid one), would only work for polynomially bounded values v_j, and would also be significantly more complex than the algorithm of Section 4.