Greedy Algorithm for Multiway Matching with Bounded Regret

Varun Gupta

Booth School of Business, University of Chicago, Chicago, IL 60637, varun.gupta@chicagobooth.edu

In this paper we prove the efficacy of a simple greedy algorithm for a finite horizon online resource allocation/matching problem, when the corresponding static planning linear program (SPP) exhibits a non-degeneracy condition called the *general position gap* (GPG). The key intuition that we formalize is that the solution of the reward maximizing SPP is the same as a feasibility Linear Program restricted to the optimal basic activities, and under GPG this solution can be tracked with bounded regret by a greedy algorithm, i.e., without the commonly used technique of periodically resolving the SPP.

The goal of the decision maker is to combine resources (from a finite set of resource types) into configurations (from a finite set of feasible configurations) where each configuration is specified by the number of resources consumed of each type and a reward. The resources are further subdivided into three types – *offline* (whose quantity is known and available at time 0), *online-queueable* (which arrive online and can be stored in a buffer), and *online-nonqueueable* (which arrive online and must be matched on arrival or lost). Under GPG we prove that, *(i)* our greedy algorithm gets bounded any-time regret of $\mathcal{O}(1/\epsilon_0)$ for matching reward (ϵ_0 is a measure of the GPG) when no configuration contains both an online-queueable and an online-nonqueueable resource, and *(ii)* $\mathcal{O}(\log t)$ expected any-time regret otherwise (we also prove a matching lower bound). By considering the three types of resources, our matching framework encompasses several well-studied problems such as dynamic multi-sided matching, network revenue management, online stochastic packing, and multiclass queueing systems.

Key words: Dynamic matching, Sum-of-Squares, General Position Gap, Lyapunov analysis, amortized analysis

1. Introduction

Problem Setup (informal): Consider a decision maker who seeks to maximize the total reward by combining resources of different types which arrive online into feasible matching configurations over a horizon of T steps. The set of resource types $\mathcal{I} = \{1, \ldots, n\}$ is finite, as is the set of feasible matching configurations $\mathcal{M} = \{1, \ldots, d\}$. The resource consumption matrix $\mathbf{M} \in \mathbb{N}^{n \times d}$ is used to denote the number of units of each resource participating in each match: M_{im} is the number of units of resource type i in configuration m. A match of type m generates a reward r_m. We assume that for each $i \in \mathcal{I}$, the set $\mathcal{M}$ consists of a special configuration consisting exclusively of one unit of resource i with reward 0 – which can be used to model discarding of resources. The resource types are partitioned as $\mathcal{I} = \mathcal{I}^{\text{off}} \cup \mathcal{I}^{\text{on-q}} \cup \mathcal{I}^{\text{on-nq}}$. The set $\mathcal{I}^{\text{off}}$ denotes the set of resources available offline to the decision maker; the number of available units of type $i \in \mathcal{I}^{\text{off}}$ are $\lambda_i T$. The set $\mathcal{I}^{\text{on}} = \mathcal{I}^{\text{on-q}} \cup \mathcal{I}^{\text{on-nq}}$ denotes the set of resources which arrive online *i.i.d.* from a known distribution; let λ_i denote the

probability that arrival is of type $i \in \mathcal{I}^{\mathrm{on}}$ at any time step (our results will hold a bit more generally, we defer the formal model and results until Section 4). Resources in $\mathcal{I}^{\mathrm{on\text{-}q}}$ can be queued on arrival and used to complete a matching configuration at a later time period. Resources in $\mathcal{I}^{\mathrm{on\text{-}nq}}$ can not be queued, and must be matched on arrival or are lost forever (note that a matching configuration can have at most one item from $\mathcal{I}^{\mathrm{on\text{-}nq}}$).[1]

The general framework introduced above with the three types of resources captures many online resource allocation problems studied in the literature:

1. Dynamic Multi-sided Matching (Kerimov et al. 2021a,b, Nazari and Stolyar 2019): This model is motivated by kidney matching applications (where resources correspond to donor-recipient pairs and matches correspond to directed cycles where each arc of the cycle corresponds to two pairs where the donor of the tail is compatible with the recipient of the head), ride-pooling application (where the resources correspond to riders and drivers), and assemble-to-order systems (where resources correspond to orders for assembled products as well as components used in assembly, and matches correspond to an order together with the component parts required to assemble it in some feasible manner). In dynamic multi-sided matching all the resources types are in $\mathcal{I}^{\mathrm{on\text{-}q}}$, and the goal is to maximize the matching reward while keeping the queues small in expectation. It is not necessary to match all the arriving resources. Our strongest results are for this case – we give the first parameter-free algorithm with bounded expected regret for general dynamic multi-sided matching for all times $t \in \{1, \ldots, T\}$.

2. Network Revenue Management/Online Knapsack (Jasin and Kumar 2012, Bumpensanti and Wang 2020, Vera et al. 2021): A typical example of this application is revenue management in airlines where the offline resources are plane seats on each of the legs operated by the airline. Passengers arrive online with an origin and a destination, and the decision maker offers a route between the origin and destination using legs with available capacity. To map to our setting, we can model the passengers with each origin-destination pair as an online-nonqueueable resource type and the plane seats as offline resources. Feasible configurations consist of one passenger, and a route (set of legs) matching the origin and destination of the passenger.

3. Multiclass-multiserver queueing systems with match-dependent rewards (Miller 1969, Ding et al. 2018): Here the system consists of a pool of m heterogeneous servers, and n customer types. Customers of each type arrive according to a Poisson process. The service time distribution at each server is Exponential with server-dependent mean, however, the utility of a customer-server match depends on the pair. The decision maker makes two sets of decisions: first, which customers to let into the system, and second how to match the customers and servers. To map to our setting, the customers represent online-queueable resources, and the

servers (or service opportunities) represent online-nonqueueable resources. Feasible configurations are all customer-server pairs, and are associated with the corresponding matching reward values.

4. Online Stochastic bin packing (Csirik et al. 2006, Gupta and Radovanović 2020, Banerjee and Freund 2020): Here the decision maker has access to an infinite collection of bins of integer size B. Items with scalar (integer) size arrive online and must be packed irrevocably into a feasible bin on arrival. The matching configurations correspond to multisets of resources with total size at most B, and are downward closed (any strict subset of a matching configuration is also a feasible matching configuration). We can view the items as online-queueable resources, however there are a few differences with the framework described above *(i)* this is a cost minimization setting with a cost of 1 per bin, *(ii)* we must pack every arrival in some bin, and *(iii)* the items must be packed on arrival. The first can be addressed by associating a reward of -1 with each configuration. The second can be incorporated by requiring that each item must be packed in some configuration (our proposed algorithm already enforces this by associating rejected items with singleton configurations which in this case can be assigned a reward of -2, and easily extends to setting where rejections are forbidden as long as downward closed condition for feasible configurations is satisfied). The third requirement is also met by our algorithm because it in fact commits items to configurations at the time of arrival.

Static Planning Problem and General Position Gap: The following Linear Program which finds the reward maximizing matching rates for the deterministic approximation of the online problem is called the Static Planning Linear Program (SPP). In the SPP, we relax the online aspect of the problem by assuming that all the resources can be indefinitely queued and matched at any time. The decision variable x_m corresponds to the fractional number of matches of configuration m per time step.

$$\max_{\{x_m\}_{m\in\mathcal{M}}} : \sum_{m\in\mathcal{M}} r_m \cdot x_m \qquad\qquad (\mathsf{SPP}(\boldsymbol{\lambda}))$$

subject to

$$\forall i \in \mathcal{I} : \sum_{m\in\mathcal{M}} M_{im} \cdot x_m = \lambda_i,$$

$$x_m \geq 0.$$

We denote the optimal value of $\mathsf{SPP}(\boldsymbol{\lambda})$ by $\mathsf{OPT}(\boldsymbol{\lambda})$. Recall that the set $\mathcal{M}$ includes configurations used to model discarding of resources, which is why the resource availability constraints are modeled as equality and not $\leq$. For an optimal basic feasible solution, we will call configurations m with $x_m > 0$ as *basic activities*. We say that $\boldsymbol{\lambda}$ satisfies the *general position gap* (GPG) condition if

there exists some $\epsilon > 0$, such that for an optimal basic feasible solution $\mathbf{x}^*$ to $\mathsf{SPP}(\boldsymbol{\lambda})$, for all $\hat{\boldsymbol{\lambda}}$ satisfying $|\hat{\boldsymbol{\lambda}} - \boldsymbol{\lambda}|_1 \leq \epsilon$ there also exists an optimal basic feasible solution to $\mathsf{SPP}(\hat{\boldsymbol{\lambda}})$ with the same basic activities (non-zero components) as $\mathbf{x}^*$. We present a more detailed discussion in Section 4. Algorithmically and analytically, GPG allows proving a drift bound on the potential function of interest giving bounded regret.

Sum-of-Squares algorithm: Our greedy algorithm is inspired by the Sum-of-Squares algorithm of Csirik et al. (2006) for the Online Stochastic bin packing problem we described previously. The algorithm of Csirik et al. (2006) places items greedily so as to minimize the Sum-of-Squares potential function $\Phi_{\mathsf{SS}} = \sum_{h=1}^{B-1} Q(h)^2$, where $Q(h)$ denotes the number of bins with h units of volume occupied (called level h bins). The authors prove that their algorithm gets a regret of $\mathcal{O}(\sqrt{T})$ for a special class of distributions called *Perfectly Packable* distributions, while being agnostic to the distribution. However, their algorithm gets $\mathcal{O}(\log T)$ regret compared to the hindsight optimal if the item size distribution is what is called a *Bounded Waste* distribution. Furthermore, the regret improves to $\mathcal{O}(1)$ if the algorithm knows the support of the item size distribution, and is forbidden from creating any *dead-end* level bins – these are bins into which no set of items in the support of the distribution can be packed to create a level B bin. Our starting point was the observation that this result can be generalized much more. Let $\mathcal{H} \subseteq \{1, \ldots, B-1\}$ denote the bin levels which are part of the optimal fractional packing (that is, in the optimal fractional packing, there are non-zero bins for each level $h \in \mathcal{H}$) – we will call these *basic levels* in analogy with basic activities. The sum-of-squares algorithm gets $\mathcal{O}(\log T)$ regret as long as the Sum-of-Squares potential function is modified to $\Phi_{\mathsf{SS}} = \sum_{h \notin \mathcal{H}} Q(h)^2$, and the distribution of item sizes satisfies *general position gap* (GPG) (bounded-waste distributions satisfy GPG, but also many distributions which are not Perfectly Packable satisfy GPG). Furthermore, the regret drops to $\mathcal{O}(1)$ if the algorithm is forbidden from creating bins of levels which are dead-ends for all $h \in \mathcal{H}$. At a high level, the intuition is that we can think of the optimal bin packing as the solution to a network flow feasibility problem where the source node is the empty configuration, the sink nodes correpond to $\mathcal{H}$, and every other node is an internal node where we must have flow balance (which means in the optimal solution there are no bins for levels corresponding to the internal nodes). Minimizing the potential function $\sum_{h \notin \mathcal{H}} Q(h)^2$ ensures that the expected flow-imbalance at internal nodes is bounded in expectation under GPG. The algorithm we propose is an adaptation of this observation to the multiway matching setting, where we view each match as a bin configuration. Our algorithm commits items on arrival to one of the basic activities of $\mathsf{SPP}(\boldsymbol{\lambda})$ irrevocably and greedily using a carefully chosen potential function. In addition, we treat offline resources like online resources by injecting them at a regular schedule for bookkeeping purposes; for creating matches we allow

using offline resources which have not "arrived" yet. A side effect is that the arrival distribution for our algorithm is non-stationary. We therefore prove our regret results under a more general non-stationary variant of the online matching problem combining Lyapunov drift analysis with techniques from amortized analysis of online algorithms.

Prior Work: We briefly mention the strength and weaknesses of our results in relation to the existing literature, but do not make an attempt to provide an exhaustive review of the results (for which we refer the reader to the recent papers (Vera et al. 2021) and (Kerimov et al. 2021a)). The main takeaway message of our paper is that the knowledge of basic feasible activities of the optimal offline solution is immensely powerful, and can be used to create quite simple greedy algorithms where the precise values of arrival rates $\boldsymbol{\lambda}$ and rewards $\mathbf{r}$ are immaterial and small fluctuations and non-stationarity around $\boldsymbol{\lambda}$ impact the regret gracefully. While there have been some papers which implicitly exploit this intuition, we have not seen it formalized, and not in the generality considered in the present paper. We have already mentioned that Csirik et al. (2006) prove bounded regret of Sum-of-Squares algorithm for online stochastic bin packing for a special case where effectively they are using a greedy algorithm restricted to optimal basic activities, but have not observed the general scheme to the best of our knowledge. Recently, Kerimov et al. (2021b) propose a 'greedy' algorithm with bounded regret for *bipartite* version of multiway matching under GPG. The notion of 'greedy' employed in Kerimov et al. (2021b) is that if at the time of the arrival the item can be matched to another queued item using an optimal basic feasible configuration, then this match is performed. Our results show the right generalization of 'greedy' beyond bipartite matching – the items should be committed to a matching configuration on arrival even if a match can not be completed at that time.

While our results are the state-of-the-art for multiway matching under GPG, they are not state-of-the-art for Network Revenue Management and Online Stochastic Bin Packing where there are several results which get bounded regret for these problems *without GPG*, for example, (Arlotto and Gurvich 2019, Bumpensanti and Wang 2020, Vera et al. 2021, Banerjee and Freund 2020). These results are based on frequently resolving the SPP, which we prove to be unnecessary under GPG. However we would like to point out a few differences in the flavor of results. *(i)* The benchmark against which these papers get $\mathcal{O}(1)$ regret is the expected hindsight optimal of the sample path (i.e., $\mathbf{E}[\mathsf{OPT}]$), whereas the regret guarantee in this paper also holds against the fluid optimal $\mathsf{OPT}(\boldsymbol{\lambda}) = \mathsf{OPT}(\mathbf{E})$. When GPG condition does not hold, in general $\mathsf{OPT}(\boldsymbol{\lambda}) - \mathbf{E}[\mathsf{OPT}] = \Omega(\sqrt{T})$. Therefore, the fluid benchmark is too strong and even hindsight optimal can not get $\mathcal{O}(1)$ regret against it. When GPG does hold, the expected hindsight optimal and fluid optimal are $\mathcal{O}(1)$ apart. *(ii)* All these results only get bounded regret at the terminal time T. In fact as Banerjee and Freund

(2020) show, knowledge of the horizon T is critical for the $\mathcal{O}(1)$ regret guarantee to hold. In contrast, for stochastic bin packing, we prove bounded anytime regret without the knowledge of T under GPG. If GPG holds, we believe the algorithms of (Banerjee and Freund 2020, Bumpensanti and Wang 2020) still would not get $\mathcal{O}(1)$ anytime regret since they only resolve the SPP sporadically, whereas (Vera et al. 2021) would get $\mathcal{O}(1)$ regret since it resolves a planning problem at every time step.

The General Position Gap assumption we make is equivalent to a dual non-degeneracy condition which requires that the solution to a partial dual of ($\mathsf{SPP}(\boldsymbol{\lambda})$) be unique. This dual non-degeneracy appears as a condition for small regret in many online resource allocation problems (e.g., (Huang and Neely 2009, Jasin and Kumar 2012, Chen et al. 2021)). For example, Huang and Neely (2009) study an online bandwidth allocation problem, where the evolution of the queues corresponds to a stochastic ascent algorithm for the dual function. Here dual non-degeneracy leads to the dual function being locally polyhedral around its unique optimal, which in turn gives concentration of the duals. The analysis in the current paper uses a tailor made Lyapunov function, and not the dual function. However, the spirit of the drift arguments for the Lyapunov function and the dual function are similar.

The knowledge of the basic feasible activities has been exploited in control of multiclass-multiserver queueing systems with match-dependent service rates (Stolyar and Yudovina 2013). However, in their work the emphasis is on minimizing delay under a many-servers heavy-traffic regime (there are no match-dependent rewards). The authors devise a static priority policy based on the tree induced by the basic feasible activities which is oblivious to the arrival rates (λ_i in our setup for customers) as well as service rates (the matrix $\mathbf{M}$ in our setup). Kerimov et al. (2021a) also propose a priority policy for the case of bipartite matching based on the basic feasible activities: whether priority policies beyond the bipartite case are possible is an open research question.

Summary of Contributions:
- We introduce a model of multiway matching that unifies the existing work on dynamic multi-sided matching, network revenue management, and queueing network control, among others, via the introduction of three types of resources – offline, online-queueable, and online-nonqueueable.
- We introduce a new greedy algorithm for the multiway matching problem, and formalize the intuition behind the optimality of greedy algorithms which only restrict their actions to the optimal basic feasible activities of the static planning problem. Our greedy algorithm has $\mathcal{O}(1/\epsilon_0)$ any-time regret when no configuration has both an online-queueable and online-nonqueueable resources (e.g., in dynamic multisided matching, network revenue management) and tight $\mathcal{O}(\log t)$ regret otherwise (e.g., queueing network control).

- We present a new regret analysis which combines Lyapunov function with amortized analysis to handle non-stationary arrival sequences, and should be of independent interest.

Notation: We use the notation $[i] = \{1, \ldots, i\}$. The notation $[s, t]$ is used for the discrete interval $\{s, s+1, \ldots, t\}$ as well as for the continuous interval $[s, t]$, where the use should be clear from the context. We use bold letters to denote vectors and matrices. For an $n \times d$ matrix $\mathbf{M}$, we use $\mathbf{M}_j$ to denote its jth column, and $\mathbf{M}_{\mathcal{J}}$ for $\mathcal{J} \subseteq [d]$ to denote the submatrix of $\mathbf{M}$ obtained by selecting the columns in $\mathcal{J}$. Similarly, for a vector $\mathbf{x}$, $\mathbf{x}_{\mathcal{J}}$ denotes the vector obtained by selecting the components of $\mathbf{x}$ corresponding to the subset $\mathcal{J}$. $\mathbf{e}_i$ denotes a column vector with ith entry 1 and the rest 0. For quantities which are functions of time and for stochastic processes, we use superscript t as the time index. For any process o, we use $\bar{o}^{[t_1, t_2]} := \frac{1}{t_2 - t_1 + 1} \sum_{t \in [t_1, t_2]} o^t$. The expression $(x)^+ = \max\{x, 0\}$ and $(x)^- = \max\{-x, 0\}$. For two probability distributions $\mathbf{p}, \mathbf{q}$, we denote the total variation distance by $||\mathbf{p} - \mathbf{q}||_{\mathsf{TV}} = \frac{1}{2}|\mathbf{p} - \mathbf{q}|_1$, and the total variation ball around $\mathbf{p}$ of radius δ is defined as $\mathcal{B}_{\epsilon, \mathsf{TV}}(\mathbf{p}) = \{\mathbf{q} \mid ||\mathbf{p} - \mathbf{q}||_{\mathsf{TV}} \le \epsilon\}$. For an event $\mathcal{E}$, $\mathbb{I}\{\mathcal{E}\}$ denotes the indicator random variable. For two vectors $\mathbf{x}, \mathbf{y}$, we use $\mathbf{x} \le \mathbf{y}$ to mean $\mathbf{x}$ is component-wise no larger than $\mathbf{y}$.

Outline: We begin in Section 2 by covering the somewhat simpler setting of multiway matching with only online-queueable resources and *i.i.d.* arrivals to convey the main ideas with minimum notational burden. The analysis in this setting largely uses tools already developed in (Csirik et al. 2006), but the algorithm is different than (Csirik et al. 2006). In the rest of the paper we focus on general multiway matching instances with all three types of resources (offline, online-queueable, online-nonqueueable), and non-stationary arrivals. To convert a problem with both offline and online resources into one with only online resources, our greedy algorithm is wrapped within a procedure that creates an online arrival sequence for the offline resources. In Section 3 we describe this procedure. In Section 4 we describe the general online matching model, state the General Position Gap assumptions needed for our results, and state our main results (Theorem 2 and Theorem 3). In Section 5 we present our Sum-of-Squares algorithm, and perform regret analysis in Section 6. Section 7 ends with some simulation experiments.

2. Prelude - Multiway Matching with Online-Queueable Resources and i.i.d. Arrivals

To convey the main ideas that go into the algorithm and regret analysis, in this section we consider the case of multiway matching with only online-queueable resources ($\mathcal{I} = \mathcal{I}^{\mathsf{on\text{-}q}} = [n]$) over a horizon of T time steps, with the type of item arriving at time $t \in [T]$ sampled *i.i.d.* from a known distribution $\boldsymbol{\lambda}$. Starting in Section 3, we study general online matching instances with non-stationary arrivals, and with online-nonqueueable and offline resources, which require modifying the algorithm and slightly advanced technical analysis.

We assume the set of feasible matches is denoted by the matching matrix $\mathbf{M} \in \{0,1\}^{n \times d}$. We will denote the reward of configuration m by r_m, the set of items participating in configuration m by $\mathcal{I}(m)$, and $\ell_m = |\mathcal{I}(m)|$. Let $\mathcal{M}_+$ be an optimal basis for the static planning problem ($\mathsf{SPP}(\boldsymbol{\lambda})$), and let $\mathbf{x}^*$ be the optimal solution corresponding to $\mathcal{M}_+$ ($x_m^* = 0$ for $m \notin \mathcal{M}_+$). The general position gap assumption amounts to saying that for all $\hat{\boldsymbol{\lambda}}$ in a total variational ball of radius ϵ_0 around $\boldsymbol{\lambda}$ (i.e., $\hat{\boldsymbol{\lambda}} \in \mathcal{B}_{\epsilon_0, \mathrm{TV}}(\boldsymbol{\lambda})$) there is an optimal solution to $\mathsf{SPP}(\hat{\boldsymbol{\lambda}})$ with basis $\mathcal{M}_+$.

Algorithm: As we mentioned in the introduction, the crux of our algorithm is to commit an item to a configuration in $\mathcal{M}_+$ at the time of its arrival to the system, where the identity of this configuration is decided based on greedily minimizing a Sum-of-Squares potential function. Our algorithm maintains a queue for each configuration $m \in \mathcal{M}_+$, and each resource $i \in \mathcal{I}(m)$, where this queue denotes the set of items of resource i that have been committed to configuration m but not converted into a match yet. Let $Q_{i,m}^t$ denote the queue length for the pair (i,m) just before the arrival at time $t+1$. Let $\mathbf{Q}_m^t = (Q_{i,m}^t)_{i \in \mathcal{I}}$ denote the state of the queues for configuration m, and $\mathbf{Q}^t = (\mathbf{Q}_m^t)_{m \in \mathcal{M}_+}$ denote the state of all the queues. If an item of type i arrives at time t and is committed to configuration m, then

$$\mathbf{Q}_m^t = \begin{cases} \mathbf{Q}_m^{t-1} + \mathbf{e}_i - \mathbf{M}_m & \text{if } \mathbf{Q}_m^{t-1} + \mathbf{e}_i \geq \mathbf{M}_m, \\ \mathbf{Q}_m^{t-1} + \mathbf{e}_i & \text{otherwise.} \end{cases}$$

That is, if there are sufficient number of resources in the queues $\mathbf{Q}_m^{t-1}$ along with the new arrival to complete a match then all the queues decrease accordingly. Otherwise the new arrival is added to the queue Q_{im}.

For a configuration $m \in \mathcal{M}_+$, pick any permutation σ_m of $\mathcal{I}(m)$. Then the cyclic Sum-of-Squares potential function for configuration m as a function of the state $\mathbf{Q}_m$ is defined as:

$$\Phi_m(\mathbf{Q}_m) = \left(Q_{\sigma_m(1),m} - Q_{\sigma_m(\ell_m),m}\right)^2 + \sum_{k=2}^{\ell_m} \left(Q_{\sigma_m(k),m} - Q_{\sigma_m(k-1),m}\right)^2, \tag{1}$$

and the overall Sum-of-Square potential is

$$\Phi(\mathbf{Q}) = \sum_{m \in \mathcal{M}_+} \Phi_m(\mathbf{Q}_m). \tag{2}$$

Our greedy Sum-of-Squares algorithm commits the arrival at time t to a configuration so as to minimize $\Phi(\mathbf{Q}^t)$. Intuitively, for each m, one of the Q_{im}^t must always be 0 otherwise a match can be completed. By trying to minimize the sum of squared differences, the Sum-of-Squares potential function tries to keep all Q_{im}^t small.

Analysis: Let $\mathbf{A}^t = (A_i^t)_{i \in \mathcal{I}}$ denote the number of arrivals of each resource type until and including time t. Let $\mathbf{X}^t = (X_m^t)_{m \in \mathcal{M}_+}$ denote the number of completed matches of configurations in $\mathcal{M}_+$ until and including time t. The next lemma bounds the expected regret of any algorithm which commits items to configurations in $\mathcal{M}_+$ on arrival in terms of the queue lengths $\mathbf{Q}^t$. This lemma does not require general position gap assumption, however subsequent analysis then shows that $\mathbf{E}[Q_{im}^t]$ is bounded under the greedy Sum-of-Squares based algorithm if the general position gap assumption holds.

LEMMA 1. *For multiway matching with online-queueable resources and i.i.d. arrivals, the expected regret at time t for any algorithms that commits items to configurations in $\mathcal{M}_+$ on arrival is bounded as follows:*

$$\mathbf{E}\left[\sum_{m \in \mathcal{M}_+} r_m \cdot X_m^t\right] \geq \mathbf{E}[OPT(\mathbf{A}^t)] - \mathbf{r}_{\mathcal{M}_+} \mathbf{M}_{\mathcal{M}_+}^{-1} \mathbf{E}\left[\sum_{m \in \mathcal{M}_+} \mathbf{Q}_m^t\right].$$

Proof: Since $\mathcal{M}_+$ is a basis, $\mathbf{M}_{\mathcal{M}_+}$ is full rank. And since we assume that the algorithm only commits items to configurations in $\mathcal{M}_+$, $\mathbf{X}^t$ is given by:

$$\mathbf{X}_{\mathcal{M}_+}^t = \mathbf{M}_{\mathcal{M}_+}^{-1}\left(\mathbf{A}^t - \sum_{m \in \mathcal{M}_+} \mathbf{Q}_m^t\right).$$

Further, $\mathbf{E}[\mathbf{A}^t] = \boldsymbol{\lambda} \cdot t$ implies $\mathsf{OPT}(\mathbf{E}[\mathbf{A}^t]) = \mathbf{r}_{\mathcal{M}_+}^\top \mathbf{M}_{\mathcal{M}_+}^{-1} \mathbf{E}[\mathbf{A}^t]$. Therefore,

$$\mathbf{E}\left[\sum_{m \in \mathcal{M}_+} r_m \cdot X_m^t\right] = \mathbf{E}\left[\mathbf{r}_{\mathcal{M}_+}^\top \mathbf{M}_{\mathcal{M}_+}^{-1}\left(\mathbf{A}^t - \sum_{m \in \mathcal{M}_+} \mathbf{Q}_m^t\right)\right]$$

$$= \mathbf{r}_{\mathcal{M}_+}^\top \mathbf{M}_{\mathcal{M}_+}^{-1} \mathbf{E}[\mathbf{A}^t] - \mathbf{r}_{\mathcal{M}_+}^\top \mathbf{M}_{\mathcal{M}_+}^{-1} \mathbf{E}\left[\sum_{m \in \mathcal{M}_+} \mathbf{Q}_m^t\right]$$

$$= \mathsf{OPT}(\mathbf{E}[\mathbf{A}^t]) - \mathbf{r}_{\mathcal{M}_+}^\top \mathbf{M}_{\mathcal{M}_+}^{-1} \mathbf{E}\left[\sum_{m \in \mathcal{M}_+} \mathbf{Q}_m^t\right]$$

$$\geq \mathbf{E}[\mathsf{OPT}(\mathbf{A}^t)] - \mathbf{r}_{\mathcal{M}_+}^\top \mathbf{M}_{\mathcal{M}_+}^{-1} \mathbf{E}\left[\sum_{m \in \mathcal{M}_+} \mathbf{Q}_m^t\right].$$

The last step follows since $\mathsf{OPT}(\cdot)$ is concave and piece-wise linear. This fact is apparent by looking at the dual of $(\mathsf{SPP}(\boldsymbol{\lambda}))$ which is given in $(\mathsf{DSPP}(\boldsymbol{\lambda}))$. As we point out later in the proof of Proposition 1, $\mathbf{r}_{\mathcal{M}_+}^\top \mathbf{M}_{\mathcal{M}_+}^{-1}$ is the optimal solution to $(\mathsf{DSPP}(\boldsymbol{\lambda}))$, and can be thought of as the value of resources. Therefore, the second term in the right hand side of the inequality in Lemma 1 can be thought of as the value of the unmatched resources in the queues. ∎

The next lemma states that if $\Phi(\mathbf{Q}^t)$ is large and the General Position Gap condition holds, then in expectation this function has a large negative drift at time $t+1$. The proof of this lemma is almost identical to Theorem 3.4 from (Csirik et al. 2006) where authors prove a similar *Expected Decrease Condition* for stochastic bin packing with bounded waste distributions. The expected decrease condition is also central for our analysis.

LEMMA 2. *The Lyapunov function $\Psi^t := \sqrt{\Phi(\mathbf{Q}^t)}$ satisfies the following conditions:*

(a) *Bounded variation: $|\Psi^{t+1} - \Psi^t| \leq \sqrt{2}$ with probability 1*

(b) *Expected Decrease: Under the General Position Gap condition with parameter ϵ_0*

$$\mathbf{E}\left[\Phi(\mathbf{Q}^{t+1}) - \Phi(\mathbf{Q}^t) \,\middle|\, \Phi(\mathbf{Q}^t) \geq \frac{16n^4}{\epsilon_0^2}\right] \leq -\frac{\epsilon_0\sqrt{\Phi(\mathbf{Q}^t)}}{2n^2},$$

which further gives,

$$\mathbf{E}\left[\Psi^{t+1} - \Psi^t | \Psi^t \geq \frac{4n^2}{\epsilon_0}\right] \leq -\frac{\epsilon_0}{4n^2}.$$

Proof: (a) Let $m \in \mathcal{M}_+$ be the configuration to which a resource is committed at time t. Without loss of generality assume $\mathcal{I}(m) = \{1, \ldots, \ell\}$, let i be the resource added, and let the permutation σ_m used to define $\Phi_m(\mathbf{Q}_m)$ be the identity permutation, so that

$$\Phi_m(\mathbf{Q}_m) = (Q_{1,m} - Q_{\ell,m})^2 + \sum_{k=2}^{\ell} (Q_{k,m} - Q_{k-1,m})^2.$$

Assume $\Phi_m(\mathbf{Q}_m^{t+1}) \geq \Phi_m(\mathbf{Q}_m^t)$; the other case is handled in a similar manner. In this case examining the queue length differences involved in Φ_m,

$$Q_{i,m}^{t+1} - Q_{i-1,m}^{t+1} = Q_{i,m}^t - Q_{i-1,m}^t + 1, \qquad Q_{i+1,m}^{t+1} - Q_{i,m}^{t+1} = Q_{i+1,m}^t - Q_{i,m}^t - 1,$$

and all the other differences are unchanged. Therefore, for some $x, y \geq 0,$,

$$\Phi_m(\mathbf{Q}_m^{t+1}) \leq \Phi_m(\mathbf{Q}_m^t) + (x+1)^2 + (y+1)^2 - x^2 - y^2.$$

Now, following the proof of Lemma 3.5 of Csirik et al. (2006):

$$\sqrt{\Phi(\mathbf{Q}^{t+1})} - \sqrt{\Phi(\mathbf{Q}^t)} \leq \sqrt{\Phi(\mathbf{Q}^t) + (x+1)^2 + (y+1)^2 - x^2 - y^2} - \sqrt{\Phi(\mathbf{Q}^t)}$$

$$= \frac{2x + 2y + 2}{\sqrt{\Phi(\mathbf{Q}^t) + (x+1)^2 + (y+1)^2 - x^2 - y^2} + \sqrt{\Phi(\mathbf{Q}^t)}}$$

$$\leq \frac{2x + 2y + 2}{\sqrt{(x+1)^2 + (y+1)^2} + \sqrt{x^2 + y^2}}$$

which is maximized at $x = y = (x+y)/2$ for a fixed value of $x + y = z$, giving the upper bound,

$$\leq \frac{2z + 2}{\sqrt{\frac{z^2}{2} + 2z + 2} + \sqrt{\frac{z^2}{2}}}$$

$$= \sqrt{2}.$$

(b) The proof involves two steps. The first step involves showing that if $\Phi(\mathbf{Q}^t) \geq 4n^4$, then there exists some resource type $i^* \in \mathcal{I}$ such that if an item of type i^* is committed by our greedy algorithm, then the resulting state satisfies

$$\Phi(\mathbf{Q}^{t+1}) - \Phi(\mathbf{Q}^t) \leq -\frac{\sqrt{\Phi(\mathbf{Q}^t)}}{n^2}.$$

This statement is proved as Lemma EC.4 in the Appendix, and its proof follows similar to Lemma 3.7 of (Csirik et al. 2006).

The second step involves showing that for all distributions $\hat{\boldsymbol{\lambda}} \in \mathcal{B}_{\epsilon_0, \text{TV}}(\boldsymbol{\lambda})$, there exists a randomized commitment policy $\mathcal{R}(\hat{\boldsymbol{\lambda}})$ (formally defined later in Definition EC.1), such that if an item sampled from $\hat{\boldsymbol{\lambda}}$ is committed to one of the configurations in $\mathcal{M}_+$ by $\mathcal{R}(\hat{\boldsymbol{\lambda}})$ at time t then the expected change in the Sum-of-Squares potential is:

$$\mathbf{E}\big[\Phi(\mathbf{Q}^{t+1})\big] - \Phi(\mathbf{Q}^t) = 2.$$

This statement is proved as Lemma EC.3 in the Appendix, and its proof follows similar to Lemma 2.2 of (Csirik et al. 2006). Since our algorithm minimizes $\Phi(\mathbf{Q}^{t+1})$ given $\Phi(\mathbf{Q}^t)$, this gives an upper bound on $\mathbf{E}[\Phi(\mathbf{Q}^{t+1})] - \Phi(\mathbf{Q}^t)$ for our Sum-of-Squares based greedy algorithm.

Now, suppose the state at time t satisfies $\Phi(\mathbf{Q}^t) \geq \frac{16n^4}{\epsilon_0^2} \geq 4n^4$, and let i^* be the item from the first step above which leads to a large negative drift. We can decompose $\boldsymbol{\lambda}$ as:

$$\boldsymbol{\lambda} = \epsilon_0 \cdot \mathbf{e}_{i^*} + (1 - \epsilon_0) \cdot \hat{\boldsymbol{\lambda}}$$

where $\hat{\boldsymbol{\lambda}} = \frac{\boldsymbol{\lambda} - \epsilon_0 \cdot \mathbf{e}_i}{1 - \epsilon_0} \in \mathcal{B}_{\epsilon_0, \text{TV}}(\boldsymbol{\lambda})$. Combining the above two steps, the expected drift in Φ for the greedy algorithm is upper bounded by:

$$\begin{aligned}
\mathbf{E}\big[\Phi(\mathbf{Q}^{t+1})\big] - \Phi(\mathbf{Q}^t) &\leq -\epsilon_0 \cdot \frac{\sqrt{\Phi(\mathbf{Q}^t)}}{n^2} + (1 - \epsilon_0) \cdot 2 \\
&\leq -\epsilon_0 \cdot \frac{\sqrt{\Phi(\mathbf{Q}^t)}}{n^2} + \epsilon_0 \cdot \frac{\sqrt{\Phi(\mathbf{Q}^t)}}{2n^2}
\end{aligned}$$

(since $\Phi(\mathbf{Q}^t) \geq \frac{16n^4}{\epsilon_0^2}$ implies $2 \leq \epsilon_0 \cdot \frac{\sqrt{\Phi(\mathbf{Q}^t)}}{2n^2}$)

$$= -\epsilon_0 \cdot \frac{\sqrt{\Phi(\mathbf{Q}^t)}}{2n^2}.$$

Using the inequality (Lemma 3.6 of (Csirik et al. 2006)): for $x, y \geq 0$,

$$x - y \leq \frac{x^2 - y^2}{2y},$$

we get,

$$\mathbf{E}\left[\Psi^{t+1} - \Psi^t \,\bigg|\, \mathbf{Q}^t; \Psi^t \geq \frac{4n^2}{\epsilon_0}\right] \leq \frac{\mathbf{E}\left[\Phi(\mathbf{Q}^{t+1}) - \Phi(\mathbf{Q}^t) \,|\, \mathbf{Q}^t; \Psi^t \geq \frac{4n^2}{\epsilon_0}\right]}{2\sqrt{\Phi(\mathbf{Q}^t)}} \leq -\epsilon_0 \cdot \frac{\sqrt{\Phi(\mathbf{Q}^t)}}{4n^2\sqrt{\Phi(\mathbf{Q}^t)}} = -\frac{\epsilon_0}{4n^2}.$$

These final calculations are the essence of Lemma EC.5 which requires a little bit of notational overhead. We reproduced them here to keep this section self-contained to the extent possible. ∎

In (Csirik et al. 2006), after proving the expected decrease condition, the authors apply a result of Hajek (1982) to conclude that $\mathbf{E}[\Psi^t]$ and hence regret is $\mathcal{O}(1)$. However applying the result of Hajek (1982) gives a regret bound that depends on the General Position Gap parameter as $\mathcal{O}(1/\epsilon_0^2)$ and not $\mathcal{O}(1/\epsilon_0)$. To get a better regret we "upper bound" the process Ψ^t by a tractable reflected random walk with *i.i.d.* increments. We begin with the definition of multivariate increasing convex order from (Müller and Stoyan 2002, Definition 3.4.1, Theorem 3.4.2), which is the sense in which we will upper bound the process $\{\Psi^t\}_{t\in[T]}$.

DEFINITION 1 (INCREASING CONVEX ORDER). *Let* $\mathbf{X}$ *and* $\mathbf{Y}$ *be* n-*dimensional random vectors with finite expectations. Then* $\mathbf{X}$ *is less than* $\mathbf{Y}$ *in increasing convex order* $(\mathbf{X} \leq_{icx} \mathbf{Y})$ *if* $\mathbf{E}[f(\mathbf{X})] \leq \mathbf{E}[f(\mathbf{Y})]$ *for all increasing convex functions* $f : \mathbb{R}^n \to \mathbb{R}$ *such that the expectations exist. Also,* $\mathbf{X} \leq_{icx} \mathbf{Y}$ *if and only if there are random vectors* $\hat{\mathbf{X}} =_{st} \mathbf{X}$ *and* $\hat{\mathbf{Y}} =_{st} \mathbf{Y}$ *such that* $\mathbf{E}\left[\hat{\mathbf{Y}}|\hat{\mathbf{X}}\right] \geq \hat{\mathbf{X}}$ *almost surely.*

LEMMA 3. *Let* $\{\Upsilon^t\}$ *be an* $\{\mathcal{F}^t\}$-*adapted stochastic process satisfying*

(a) Bounded variation: $|\Upsilon^{t+1} - \Upsilon^t| \leq K_1$ *with probability 1,*

(b) Expected Decrease: $\mathbf{E}[\Upsilon^{t+1} - \Upsilon^t | \mathcal{F}_t; \Upsilon^t \geq K_2] \leq -2\delta' K_1.$

Let $\{\Gamma^t\}$ *be a reflected random walk with step size* K_1, *reflection boundary* $\Gamma_{\min} := K_1 (1 + \lceil K_2/K_1 \rceil)$, *initial condition* $\Gamma^\tau = \max\{\Gamma_{\min}, K_1 \lceil \Upsilon^\tau / K_1 \rceil\}$, *and*

$$\Gamma^{t+1} = \max\{\Gamma^t + \xi^{t+1}K_1, \Gamma_{\min}\}, \ t = \tau, \dots, T-1,$$

where $\xi^t \in \{-1, +1\}$ *are i.i.d. random variables taking the value* $+1$ *with probability* $\frac{1}{2} - \delta'$ *and* -1 *otherwise. Then,*

$$\left\{\Upsilon^t\right\}_{t\in[\tau,T]} \leq_{icx} \left\{\Gamma^t\right\}_{t\in[\tau,T]}.$$

We now combine Lemma 2 and Lemma 3 to prove the main regret result for this Section.

THEOREM 1. *For all* $t \geq 1$, *the Lyapunov function* $\Psi^t = \sqrt{\Phi(\mathbf{Q}^t)}$ *is upper bounded as*

$$\mathbf{E}[\Psi^t] \leq \frac{8n^2}{\epsilon_0} + \sqrt{2}.$$

Consequently, the regret for the multiway matching problem at time t is bounded as:

$$\mathbf{E}\left[\sum_{m\in\mathcal{M}_+} r_m \cdot X_m^t\right] \geq \mathbf{E}[OPT(\mathbf{A}^t)] - \mathcal{O}\left(\frac{n^3}{\epsilon_0}\right).$$

Proof: By Lemma 2 and Lemma 3, we can upper bound the stochastic process Ψ^t in increasing convex order by a random walk with *i.i.d.* increments in $\pm\sqrt{2}$, expected increment $-\frac{\epsilon_0}{4n^2}$, and

reflection boundary $\frac{4n^2}{\epsilon_0} + \sqrt{2}$. This gives the upper bound on $\mathbf{E}[\Psi^t]$ mentioned in the theorem statement since expectation is an increasing convex function. Next, we have the bound,

$$\sum_{m \in \mathcal{M}_+} \max_{i \in \mathcal{I}(m)} Q_{im}^t \leq n\Psi^t,$$

and from Lemma 1,

$$
\begin{aligned}
\mathbf{E}\left[\sum_{m \in \mathcal{M}_+} r_m \cdot X_m^t\right] &\geq \mathbf{E}[\mathsf{OPT}(\mathbf{A}^t)] - \mathbf{r}_{\mathcal{M}_+} \mathbf{M}_{\mathcal{M}_+}^{-1} \mathbf{E}\left[\sum_{m \in \mathcal{M}_+} \mathbf{Q}_m^t\right] \\
&\geq \mathbf{E}[\mathsf{OPT}(\mathbf{A}^t)] - \sum_{m \in \mathcal{M}_+} r_m \max_{i \in \mathcal{I}(m)} Q_{im}^t \\
&\geq \mathbf{E}[\mathsf{OPT}(\mathbf{A}^t)] - \max_{m \in \mathcal{M}_+} r_m \cdot \sum_{m \in \mathcal{M}_+} \max_{i \in \mathcal{I}(m)} Q_{im}^t \\
&\geq \mathbf{E}[\mathsf{OPT}(\mathbf{A}^t)] - \mathcal{O}\left(\frac{n^3}{\epsilon_0}\right).
\end{aligned}
$$

∎

Peeking ahead: When we consider multiway matching with offline and online-nonqueueable resources in the sequel, one main difference that occurs is that we need to introduce a notion of *virtual queues* in addition to the true queues Q_{im}^t we used in this section. It is the virtual queues that are used by the Sum-of-Squares based greedy algorithm to make its decisions. A second difference is that to handle offline resources in a similar manner to online resources, we inject them at a constant rate into the system. This leads to a non-stationary arrival process, which requires studying a different Lyapunov function than Ψ^t, and blending drift analysis with elements of amortized analysis since the expected decrease condition does not hold for Ψ^t anymore.

3. Intermezzo - Emulating Online Arrivals for Offline Resources

To treat offline resources in a similar manner to online resources in our greedy algorithm, we will emulate an online arrival process for them by injecting resources at a "constant rate" while at the same time ensuring that the total number of offline resources injected of each type equals their initial inventory. In this section we describe this process, which creates an instance with a non-stationary arrival process. As a result, the rest of this paper will study online matching under a somewhat general non-stationary arrival process.

Let $\mathcal{I}^{\mathsf{off}} = [n^{\mathsf{off}}]$, and $\mathcal{I}^{\mathsf{on}} = [n^{\mathsf{off}} + 1, n]$. Let T' denote the horizon for the original matching problem, and let $\lambda_i T' \in \mathbb{N}$ for $i \in \mathcal{I}^{\mathsf{off}}$ denote the number of units of offline resource i available initially. Let $\{\lambda_i\}_{i \in \mathcal{I}^{\mathsf{on}}}$ denote the probability distribution for the online resources. That is, for each $t \in [T']$, the type of the arriving item is sampled *i.i.d.* from $\{\lambda_i\}_{i \in \mathcal{I}^{\mathsf{on}}}$. Let $\boldsymbol{\lambda}^{\mathsf{on}} \in \mathbb{R}^n$ be such that $\lambda_i^{\mathsf{on}} = 0$ for $i \in \mathcal{I}^{\mathsf{off}}$ and $\lambda_i^{\mathsf{on}} = \lambda_i$ otherwise.

From the above primitives, we create another online matching instance of *effective horizon* $T = T'\left(1 + \sum_{i \in \mathcal{I}^{\text{off}}} \lambda_i\right)$ where at any time the arrival is either deterministic and of a type in $\mathcal{I}^{\text{off}}$, or sampled according to $\boldsymbol{\lambda}^{\text{on}}$. Define $\bar{\lambda}_i = \lambda_i \cdot T'/T$ for all $i \in \mathcal{I}$ as the mean rate of resource of type i, and $\bar{\lambda}_0 = \sum_{i \in \mathcal{I}^{\text{on}}} \bar{\lambda}_i$ as the aggregate mean arrival rate of online resources. The procedure is described in Algorithm 1.

Algorithm 1: Virtual online arrival process for offline resources

---

Initialize: $A_i^0 = 0$ for each $i \in \mathcal{I}^{\text{off}} \cup \{0\}$

for $t = 1, \ldots, T$ **do**

    Let $i = \arg\min \left\{ i \in \mathcal{I}^{\text{off}} \cup \{0\} \ : \ A_i^{t-1} - \bar{\lambda}_i \cdot (t-1) \right\}$ `// Break ties arbitrarily`

    if $i = 0$ **then**

    | $\boldsymbol{\lambda}^t = \boldsymbol{\lambda}^{\text{on}}$

    else

    | $\boldsymbol{\lambda}^t = \mathbf{e}_i$

    end

    $A_i^t = A_i^{t-1} + 1$

end

In words, the emulation algorithm looks at the deficit of the total number of resources of type i injected by time $t-1$, A_i^{t-1} ($i = 0$ for online, otherwise there is a separate type for each resource in $\mathcal{I}^{\text{off}}$), compared to the expectation, $\bar{\lambda}_i \cdot (t-1)$, and injects the type that is most in deficit. The following lemma states that the discrepancy between the non-stationary instance $\{\boldsymbol{\lambda}^t\}_{t \in [T]}$ created by Algorithm 1 and the stationary instance $\bar{\boldsymbol{\lambda}}$ is bounded at all times.

LEMMA 4. *For the emulation in Algorithm 1, for each $i \in \mathcal{I}^{\text{off}} \cup \{0\}$; $-n^{\text{off}} \leq A_i^t - \bar{\lambda}_i \cdot t \leq 1$ for each $t \in [T]$, and $A_i^T = \bar{\lambda}_i \cdot T$. As a consequence, letting $\tau = (n^{\text{off}} + 1)/\lambda_{\min}$, for all $t \in [\tau+1, T]$,*

$$\bar{\boldsymbol{\lambda}} \cdot (t - 2\tau) \leq \sum_{s=\tau+1}^{t} \boldsymbol{\lambda}^s \leq \bar{\boldsymbol{\lambda}} \cdot t$$

where $\bar{\lambda}_{\min} := \min_{i \in \mathcal{I}} \bar{\lambda}_i$; and $\sum_{s=1}^{T} \boldsymbol{\lambda}^s = \bar{\boldsymbol{\lambda}} \cdot T$.

In the next section we will define a multiway matching model with a somewhat general non-stationary arrival process satisfying a smoothed version of the general position gap assumption (Assumption 2) which we call Smoothed $\mathsf{GPG}_{\delta,\tau}$, and prove regret result for such non-stationary instances in Theorem 2. Lemma 4 goes together with Assumption 2 by showing that if the arrival process of online items is stationary and satisfies GPG with parameter ϵ_0, then the non-stationary instance created by Algorithm 1 satisfies $\mathsf{GPG}_{\delta,\tau}$ with $\tau = (n^{\text{off}} + 1)/\lambda_{\min}$ and $\delta = \epsilon_0$.

4. Main Act - Model and Results

Resource types and matching configurations: We consider an online matching instance with a horizon of T time steps. The resource types are denoted by $\mathcal{I} = [n]$ which are further partitioned into offline resources: $\mathcal{I}^{\text{off}} = [n^{\text{off}}]$, online-queueable: $\mathcal{I}^{\text{on-q}} = [n^{\text{off}} + 1, n^{\text{off}} + n^{\text{on-q}}]$, and online-nonqueueable: $\mathcal{I}^{\text{on-nq}} = [n^{\text{off}} + n^{\text{on-q}} + 1, n]$. The set of feasible matches is denoted by the matching matrix $\mathbf{M} \in \mathbb{N}^{n \times d}$ with $M_{im} = 1$ if resource i participates in m. Our algorithm and analysis extends to general $M_{im} \in \mathbb{N}$ or $M_{im} \in \mathbb{R}_+$ to capture different amounts of resource i needed by appropriately modifying the potential function, but to keep exposition clear in this paper we focus on $M_{im} \in \{0, 1\}$. We will use the notation $\mathcal{I}(m)$ to denote the set of resources that participate in configuration m, and $i \in m$ as shorthand for $i \in \mathcal{I}(m)$. Let $\ell_m = |\mathcal{I}(m)| = \sum_i M_{im}$ be the number of distinct resource types participating in m. A match of type m generates a reward r_m. To model the possibility of discarding resources, we assume that for each $i \in \mathcal{I}$, there is a configuration with resource consumption vector $\mathbf{e}_i$ and reward 0.[2]

   Arrival process: We will denote the type of the arriving resource at time t by $I^t \in \mathcal{I}$, and the distribution from which the type I^t is sampled as $\boldsymbol{\lambda}^t$. We will use $\{\mathcal{F}^t\}_{t \in [T]}$ to denote the filtration generated by the stochastic process $\{I^t\}_{t \in [T]}$, and assume that $\{\boldsymbol{\lambda}^t\}_{t \in [T]}$ is an $\{\mathcal{F}^t\}_{t \in [T]}$-predictable process. That is, the distribution $\boldsymbol{\lambda}^t$ from which the type I^t is sampled is itself a random quantity, which is $\mathcal{F}^{t-1}$-measurable. Conditioned on the sequence $\{\boldsymbol{\lambda}^t\}_{t \in [T]}$, $\{I^t\}_{t \in [T]}$ are assumed to be mutually independent. Define $Z_i^t = \mathbb{I}\{I^t = i\}$, and $A_i^t = \sum_{s=1}^{t} Z_i^s$ as the cumulative number of arrivals of resource type i up to and inclusive of time t.

   We will use $\bar{\boldsymbol{\lambda}}$ to denote a *representative* distribution for the sequence $\{\boldsymbol{\lambda}^t\}_{t \in [T]}$. The relationship between $\bar{\boldsymbol{\lambda}}$ and $\{\boldsymbol{\lambda}^t\}_{t \in [T]}$ will be clear soon when we mention the General Position Gap condition we require on the sequence $\{\boldsymbol{\lambda}^t\}_{t \in [T]}$. For now, the reader may think of $\bar{\boldsymbol{\lambda}}$ as the mean of $\{\boldsymbol{\lambda}^t\}_{t \in [T]}$. The model of the arrival process given so far is fairly general, and we need to add some constraints on the process so that it is compatible with the notion of offline resources (whose total inventory is known at time $t = 0$). Towards this end, we will require that for $i \in \mathcal{I}^{\text{off}}$, $\bar{\lambda}_i$ is deterministic, and $A_i^T = \bar{\lambda}_i \cdot T$ with probability 1.

   (Smoothed) General Position Gap assumption: We begin by formally defining the General Position Gap condition. Recall the static planning linear program ($\mathsf{SPP}(\boldsymbol{\lambda})$).

DEFINITION 2 (GENERAL POSITION GAP). *Let* $\mathbf{x}^*$ *be an optimal basic feasible solution to* $\mathsf{SPP}(\boldsymbol{\lambda})$, *and let* $\mathcal{M}_+ = \{m \in \mathcal{M} | x_m^* > 0\}$ *denote the basic activities under this optimal solution. We say that* $\boldsymbol{\lambda}$ *satisfies the* GPG_ϵ *condition if the following is true.*

   1. $\mathbf{x}^*$ *is the unique optimal solution with basis* $\mathcal{M}_+$, *and,*

2. for any $\hat{\boldsymbol{\lambda}} \in \mathcal{B}_{\epsilon, \mathsf{TV}}(\boldsymbol{\lambda})$, there exists a unique optimal solution to $\mathsf{SPP}(\hat{\boldsymbol{\lambda}})$ with $\mathcal{M}_+$ as the basic activities.

The GPG condition stated in Kerimov et al. (2021a) requires that $\mathbf{x}^*$ be unique and $x_m^* \geq \epsilon$ for all $m \in \mathcal{M}_+$. Our condition is weaker in the sense that we allow non-unique optimal basis. If the optimal solution is unique (equivalently, there is a unique optimal basis) then the two definitions are equivalent (with a different choice of ϵ). In the rest of the paper, we fix an optimal basic feasible solution $\mathbf{x}^*$ and its basis $\mathcal{M}_+$.[3]

The next assumption says that the $\mathsf{GPG}_{\epsilon_0}$ condition holds for the *representative* distribution $\bar{\boldsymbol{\lambda}}$ with probability 1.

ASSUMPTION 1. *There exists an $\epsilon_0 > 0$, such that almost surely $\bar{\boldsymbol{\lambda}}$ satisfies $\mathsf{GPG}_{\epsilon_0}$.*

While Definition 2 is more transparent for the analysis in the current work, an equivalent, and perhaps easier to check definition can be arrived at via the Dual of ($\mathsf{SPP}(\boldsymbol{\lambda})$).

$$\min_{\{\alpha_i\}_{i \in \mathcal{I}}} : \sum_{i \in \mathcal{I}} \lambda_i \cdot \alpha_i \qquad (\mathsf{DSPP}(\boldsymbol{\lambda}))$$

subject to

$$\forall m \in \mathcal{M} : \sum_{i \in \mathcal{I}} M_{im} \cdot \alpha_i \geq r_m.$$

The next proposition says that the General Position Gap condition is equivalent to the uniqueness of the optimal solution to ($\mathsf{DSPP}(\boldsymbol{\lambda})$).

PROPOSITION 1. *The distribution $\boldsymbol{\lambda}$ satisfies GPG_{ϵ} for some $\epsilon > 0$ if and only if for all $\hat{\boldsymbol{\lambda}} \in \mathcal{B}_{\epsilon, \mathsf{TV}}(\boldsymbol{\lambda})$, $\boldsymbol{\alpha}^*$ is the optimal solution to $\mathsf{DSPP}(\hat{\boldsymbol{\lambda}})$. That is, the optimal solution $\boldsymbol{\alpha}^*$ of ($\mathsf{DSPP}(\boldsymbol{\lambda})$) is unique.*

The ($\mathsf{DSPP}(\boldsymbol{\lambda})$) problem can equivalently be written as minimizing $\boldsymbol{\alpha}^\top \boldsymbol{\lambda}$ subject to $\boldsymbol{\alpha} \in \mathcal{A}$ where $\mathcal{A}$ is the polyhedron given by the inequalities in ($\mathsf{DSPP}(\boldsymbol{\lambda})$) and which is independent of $\boldsymbol{\lambda}$. For almost all $\boldsymbol{\lambda}$, except for a measure 0 set, this minima is attained uniquely at a corner of $\mathcal{A}$.

Finally, we state the assumption we need on the stochastic process $\{\boldsymbol{\lambda}^t\}_{t \in [T]}$.

ASSUMPTION 2 (**Smoothed $\mathsf{GPG}_{\delta, \tau}$**). *There exist a constant $\delta < \epsilon_0$, a smoothing window size $\tau \in \mathbb{N}$, and an $\mathcal{F}^t$-predictable stochastic process $\{\hat{\boldsymbol{\lambda}}^t\}_{t \in [T]}$, which we call smoothed distributions, such that*

1. $\hat{\boldsymbol{\lambda}}^t \in \mathcal{B}_{\epsilon_0 - \delta, \mathsf{TV}}(\bar{\boldsymbol{\lambda}})$ *for each $t \in [T]$, and*
2. $\sum_{s=1}^{t-2\tau} \hat{\boldsymbol{\lambda}}^s \leq \sum_{s=\tau+1}^{t} \boldsymbol{\lambda}^s \leq \sum_{s=1}^{t} \hat{\boldsymbol{\lambda}}^s$, *for all $t \in [\tau + 1, T]$.*

The smoothed $\mathsf{GPG}_{\delta, \tau}$ condition relaxes the requirement that the item distribution be *i.i.d.* as well as that the distribution at each time step be close enough to $\bar{\boldsymbol{\lambda}}$. We need this flexibility

to incorporate offline resources, since our emulation Algorithm 1 injects offline resources on a deterministic schedule. Instead, we require that the *smoothed* distributions $\hat{\boldsymbol{\lambda}}^t$ be close enough to $\bar{\boldsymbol{\lambda}}$, and that their cumulative sum tracks the cumulative sum of $\boldsymbol{\lambda}^t$ with bounded error. Since the smoothed $\mathsf{GPG}_{\delta,\tau}$ assumption and the arrival process model are stated quite abstractly we give a few examples where these hold:

1. Let the sequence be $\{\boldsymbol{\lambda}^t\}$ is deterministic, and each $\boldsymbol{\lambda}^t \in \mathcal{B}_{\epsilon_0-\delta,\mathrm{TV}}(\bar{\boldsymbol{\lambda}})$ for some $\bar{\boldsymbol{\lambda}}$ satisfying Assumption 1. In this case $\mathsf{GPG}_{\delta,\tau}$ condition holds with $\tau = 0$.

2. Let the online resources arrive *i.i.d.*, and the offline items 'arrive' via the construction in Section 3. Then either $\boldsymbol{\lambda}^t = \boldsymbol{\lambda}^{\mathrm{on}}$, or $\boldsymbol{\lambda}^t = \mathbf{e}_i$ for some $i \in \mathcal{I}^{\mathrm{off}}$. Lemma 4 shows that the smoothed $\mathsf{GPG}_{\delta,\tau}$ condition is true with $\delta = \epsilon_0$ (i.e., $\hat{\boldsymbol{\lambda}}^t = \bar{\boldsymbol{\lambda}}$) and $\tau = (n^{\mathrm{off}} + 1)/\bar{\lambda}_{\min}$.

3. Let the sequence of online resources be sampled *i.i.d.* from $\bar{\boldsymbol{\lambda}}$, and then the sample path be "locally permuted" by an adversary before the resources arrive at the matching system. By locally permute, we mean that there exists an integer γ so that no item is moved more than γ positions from its initial position. Such a scenario can happen for example if the items pass through a processing stage that is modeled as a queueing system before arriving at the matching system, or with batch arrivals. While the process $\boldsymbol{\lambda}^t$ might be fairly complex to model explicitly, the $\mathsf{GPG}_{\delta,\tau}$ condition still holds for this case with $\delta = \epsilon_0$ (i.e., $\hat{\boldsymbol{\lambda}}^t = \bar{\boldsymbol{\lambda}}$) and $\tau = \gamma$.

Matching process: Let X_m^t denote the number of matches of configuration m completed up to and inclusive of time t. Define:

$$N_i^t := A_i^t - \sum_{m \in \mathcal{M}} X_m^t \cdot M_{im}$$

as the units of resource i which have arrived but are unmatched up to time t.

For a configuration m which involves only offline and online-queueable resource types, a match can be completed at any time there are enough resources available. That is, if $N_i^{t-1} + Z_i^t \geq M_{im}$ for all $i \in \mathcal{I}^{\mathrm{on\text{-}q}}$, and $\bar{\lambda}_i \cdot T - \sum_{m'} X_{m'}^{t-1} \cdot M_{im'} \geq M_{im}$ for all $i \in \mathcal{I}^{\mathrm{off}}$. Note that even though we model the offline resources as arriving online, this is only for bookkeeping for our Sum-of-Squares algorithm; the feasibility of a match depends on the total availability. That is, it is possible that $N_i^t < 0$ for $i \in \mathcal{I}^{\mathrm{off}}$.

For a configuration m which involves an (exactly one) online-nonqueueable resource unit, a match can be completed only at the time of arrival of this resource as long as there are enough resources available. This criterion is the same as above: $N_i^{t-1} + Z_i^t \geq M_{im}$ for all $i \in \mathcal{I}^{\mathrm{on\text{-}q}}$, and $\bar{\lambda}_i \cdot T - \sum_{m'} X_{m'}^{t-1} \cdot M_{im'} \geq M_{im}$ for all $i \in \mathcal{I}^{\mathrm{off}}$.

Main Result: Our first main result is Theorem 2 which proves that the sum-of-squares algorithm described in Section 5 gets bounded anytime regret for online multiway matching if no

configurations contain both online-queueable and an online-nonqueueable resource, and $\mathcal{O}(\log t)$ anytime regret otherwise. The second result, Theorem 3, complements the above by proving that in the latter case $\Omega(\log t)$ anytime regret is unavoidable.

THEOREM 2. *Under the non-stationary arrival process given by the sequence $\boldsymbol{\lambda}^t$ satisfying $\mathsf{GPG}_{\epsilon_0}$ and the smoothed $\mathsf{GPG}_{\delta,\tau}$ conditions for constants $0 < \delta < \epsilon_0$, and some smoothing window bound τ, the Sum-of-Squares algorithm (Algorithm 2) satisfies the following regret guarantees:*

- *If there are no configurations in $\mathcal{M}_+$ which contain resources from both $\mathcal{I}^{on\text{-}q}$ and $\mathcal{I}^{on\text{-}nq}$, then for any time $t \in [T]$*

$$\mathbf{E}\left[\sum_m r_m \cdot X_m^t\right] \geq \mathbf{E}[OPT(\mathbf{A}^t)] - \frac{\kappa_1(n,\tau)}{\delta},$$

where $\kappa_1(n,\tau)$ is a constant that is polynomial in n,τ, independent of T.

- *If there are feasible configurations in $\mathcal{M}_+$ which contain resources from both $\mathcal{I}^{on\text{-}q}$ and $\mathcal{I}^{on\text{-}nq}$, then for any time $t \in [T]$*

$$\mathbf{E}\left[\sum_m r_m \cdot X_m^t\right] \geq \mathbf{E}[OPT(\mathbf{A}^t)] - \frac{\kappa_2(n,\tau)}{\delta}\log t,$$

where $\kappa_2(n,\tau)$ is a constant that is polynomial in n,τ, independent of T.

As mentioned earlier, in the case where online resources arrive *i.i.d.*, the smoothed $\mathsf{GPG}_{\delta,\tau}$ condition is true with $\delta = \epsilon_0$ and $\tau = (n^{off} + 1)/\lambda_{\min}$, and therefore the regret for the first case in Theorem 2 as a function of the GPG parameter ϵ_0 grows as $1/\epsilon_0$. This growth rate was shown to be necessary when all resources are only of online-queueable type in Kerimov et al. (2021a).

The next theorem shows that a regret of $\mathcal{O}(\log t)$ is unavoidable in general when configurations involve both online-queueable and online-nonqueueable resources.

THEOREM 3. *There is a multiway matching instance with online-queueable and online-nonqueueable resources where no online algorithm can guarantee for all times $t \in [T]$, and for a small enough constant $c > 0$:*

$$\mathbf{E}\left[\sum_m r_m \cdot X_m^t\right] \geq \mathbf{E}[OPT(\mathbf{A}^t)] - c\log t.$$

5. Sum-of-Squares Algorithm

Our proposed algorithm exploits the knowledge of the optimal basic activities for $\mathsf{SPP}(\bar{\boldsymbol{\lambda}})$ by only ever creating matching configurations from the set $\mathcal{M}_+$. When a resource of type i arrives at time t, it is irrevocably committed to a basic configuration $m \in \mathcal{M}_+$ based on a Sum-of-Squares potential associated with each configuration that we define shortly. For each configuration m and resource $i \in \mathcal{I}(m)$ we define two queue lengths – virtual queue lengths, which are for book-keeping

and used to define the sum-of-squares potential for configuration m; and true queue lengths which correspond to the resources that have been committed to m thus far but not departed the system due to a match. We also define two kinds of matches – virtual matches, which are for book-keeping and necessitated by the presence of online-nonqueueable resources; and true matches which are the physically realized matches.

We now describe the Sum-of-Squares potential function and the algorithm more formally.

Virtual/true matches and queue lengths: We denote by Y_{im}^t the number of resource units of type i committed by our algorithm to matching configuration $m \in \mathcal{M}_+$ up to and including time t. Recall that X_m^t denotes the number of matches completed up to time t for configuration m. In addition to X_m^t, which we will call *true* matches, we define an analogous quantity $\mathscr{X}_m^t$ for the number of *virtual matches* completed up to time t for configuration m. The notion of virtual match is primarily used for configurations with items in $\mathcal{I}^{\text{on-nq}}$, where we execute a virtual match for the purposes of bookkeeping every time the item from $\mathcal{I}^{\text{on-nq}}$ is committed to the configuration, even if enough resources are not available to complete a match. Only a subset of virtual matches are true matches – when enough resources are available. Based on the above definitions, we define the *true queue length* of type i resources for configuration m by:

$$Q_{im}^t = Y_{im}^t - X_m^t \cdot M_{im}, \tag{3}$$

and *virtual queue length* of type i resources for configuration m by:

$$\mathscr{Q}_{im}^t = Y_{im}^t - \mathscr{X}_m^t \cdot M_{im}. \tag{4}$$

($\mathscr{Q}$ should be read as script-Q; similarly $\mathscr{X}$ should be read as script-X. The scripted symbols are used for virtual versions of the true quantities). We always have $X_m^t \le \mathscr{X}_m^t$ and therefore $Q_{im}^t \ge \mathscr{Q}_{im}^t$. The virtual queue lengths for items in $\mathcal{I}^{\text{off}} \cup \mathcal{I}^{\text{on-q}}$ and configurations with a resource in $\mathcal{I}^{\text{on-nq}}$ can become negative – this is a feature since negative virtual queues incentivize our algorithm to commit more resources to such configurations.

Sum-of-Squares potential function: For a configuration m, with virtual queue length vector $\boldsymbol{\mathscr{Q}}_m = (\mathscr{Q}_{im})_{i \in \mathcal{I}}$, we define the Sum-of-Squares potential as follows:

$$\Phi_m(\boldsymbol{\mathscr{Q}}_m) = \left(\mathscr{Q}_{\sigma_m(1)m} - \mathscr{Q}_{\sigma_m(\ell_m)m} \right)^2 + \sum_{k=2}^{\ell_m} \left(\mathscr{Q}_{\sigma_m(k)m} - \mathscr{Q}_{\sigma_m(k-1)m} \right)^2, \tag{5}$$

where σ_m denotes an arbitrary permutation of $\mathcal{I}(m)$ that is fixed throughout the execution of the algorithm. Intuitively, minimizing the Sum-of-Squares potential equalizes the virtual queue lengths involved in configuration m. For every configuration one of the virtual queue lengths is always 0.

In the case of a configuration with a resource in $\mathcal{I}^{\text{on-nq}}$ the virtual queue length of this resource is 0. For the other configurations, some virtual queue length must be 0 otherwise a virtual match would be completed until this was the case. This fact, combined with the spirit of trying to equalize queue lengths through the potential function (6), achieves the goal of keeping all queues small. This is also precisely the intuition behind the back-pressure algorithm of Tassiulas and Ephremides (1990) for routing packets/flows in networks. However, unlike network flow, in our case there is no natural ordering of virtual queues, and therefore we use the cyclic Sum-of-Squares potential function.

As a quick remark, in the case where M_{im} take general values (other than 0 or 1), the Sum-of-Squares potential for configuration would be defined as:

$$\Phi_m(\mathcal{Q}_m) = \left(\frac{\mathcal{Q}_{\sigma_m(1)m}}{M_{\sigma_m(1)m}} - \frac{\mathcal{Q}_{\sigma_m(\ell_m)m}}{M_{\sigma_m(\ell_m)m}} \right)^2 + \sum_{k=2}^{\ell_m} \left(\frac{\mathcal{Q}_{\sigma_m(k)m}}{M_{\sigma_m(k)m}} - \frac{\mathcal{Q}_{\sigma_m(k-1)m}}{M_{\sigma_m(k-1)m}} \right)^2. \tag{6}$$

We denote the overall Sum-of-Squares potential as:

$$\Phi(\mathcal{Q}) = \sum_{m \in \mathcal{M}_+} \Phi_m(\mathcal{Q}_m). \tag{7}$$

The change in potential function when an item of type i is potentially committed to m is computed as follows:

$$\Delta \Phi_m(\mathcal{Q}_m, i) = \Phi_m(\mathcal{Q}'_m) - \Phi_m(\mathcal{Q}_m) \tag{8}$$

where $\mathcal{Q}'_m$ is defined as

$$\mathcal{Q}'_m = \begin{cases} \mathcal{Q}_m + \mathbf{e}_i - \mathbf{M}_m & \text{if virtual\_feasible}(m, \mathcal{Q}_m, i), \\ \mathcal{Q}_m + \mathbf{e}_i & \text{otherwise.} \end{cases}$$

The boolean function virtual_feasible$(m, \mathcal{Q}_m, i)$ is given by:

$$\text{virtual\_feasible}(m, \mathcal{Q}_m, i) = \begin{cases} \text{True} & \text{if } i \in \mathcal{I}^{\text{on-nq}}, \\ \text{True} & \text{if } \mathcal{I}^{\text{on-nq}} \cap \mathcal{I}(m) = \emptyset \text{ and } \mathcal{Q}_m + \mathbf{e}_i \geq \mathbf{M}_m, \\ \text{False} & \text{otherwise.} \end{cases} \tag{9}$$

In words, we always execute a virtual match whenever an online-nonqueueable resource is committed to the configuration. For configurations without any online-nonqueueable resource, we only execute a virtual match when there are sufficient number of true resources available in the queues. Note that even in this case, a virtual match does not always imply a true match since the resources in $\mathcal{I}^{\text{off}}$ could have been exhausted.

Analogous to virtual_feasible, we define the boolean function true_feasible:

$$\text{true\_feasible}(m, \mathbf{Q}_m, \mathbf{X}, i) = \begin{cases} \text{False} & \text{if } \exists i' \in \mathcal{I}^{\text{off}} \cap \mathcal{I}(m) : \bar{\lambda}_{i'} T < M_{i'm} + \sum_{m'} X_{m'} M_{i'm'}, \\ \text{False} & \text{if } \exists i' \in \mathcal{I}^{\text{on}} \cap \mathcal{I}(m) : Q_{i'm} + \mathbb{I}\{i'=i\} < M_{im}, \\ \text{True} & \text{otherwise.} \end{cases} \tag{10}$$

In words, a true match is *infeasible*, if either the total number of offline resources is insufficient (hence the dependence on **X**, the number of true matches executed so far), or the true queue length for any online resource in $\mathcal{I}(m)$ is insufficient; otherwise the true match is feasible. In our algorithm, we check the feasibility of a true match only if a virtual match is executed.

Algorithm 2 is our proposed greedy Sum-of-Squares (SS) algorithm.

Algorithm 2: Sum-of-Squares algorithm (SS) for Multiway matching

Initialize: $X_m^0 = \mathscr{X}_m^0 = 0$ for each $m \in \mathcal{M}_+$

   $Y_{im}^0 = 0$ for all $m \in \mathcal{M}_+, i \in \mathcal{I}(m)$

Definition: $Q_{im}^t, \mathscr{Q}_{im}^t$ in (3)-(4); $\Delta\Phi_m$ in (8); virtual_feasible, true_feasible in (9)-(10)

for $t = 1, \ldots, T$ **do**

    Observe arrival type I^t

    Let $m^t = \arg\min_m \{\Delta\Phi_m(\mathscr{Q}_m^{t-1}, I^t) \mid m \in \mathcal{M}_+, I^t \in \mathcal{I}(m)\}$ // Find configuration to commit

    $Y_{I^t m^t}^t = Y_{I^t m^t}^{t-1} + 1$ // Increment resource-configuration counter

    if virtual_feasible$(m^t, \mathscr{Q}_{m^t}^{t-1}, I^t)$ **then**

        $\mathscr{X}_{m^t}^t = \mathscr{X}_{m^t}^{t-1} + 1$ // Increment virtual match counter

        if true_feasible$(m^t, \mathbf{Q}_{m^t}^{t-1}, \mathbf{X}^{t-1}, I^t)$ **then**

            $X_{m^t}^t = X_{m^t}^{t-1} + 1$ // Increment true match counter

        end

    end

    for $m \neq m^t$ **do**

        $X_m^t = X_m^{t-1}; \; Y_m^t = Y_m^{t-1}$

    end

    for $(i,m) \neq (I^t, m^t)$ **do**

        $X_{im}^t = X_{im}^{t-1}$

    end

end

REMARK 1. If we had used the true queue length $Q_{i'm}$ as the criterion for offline resources in true_feasible, instead of the global availability, then our algorithm would get a regret of $\mathcal{O}(\log t)$ for configurations with online-nonqueueable and offline resources. In particular, we would not get $\mathcal{O}(1)$ regret for Network Revenue Management application.

REMARK 2. Contrast with use of virtual queues in Nazari and Stolyar (2019): In (Nazari and Stolyar 2019), the authors look at multiway matching with only online-queueable items, and also use a notion of virtual queues. However, their use differs from the use of virtual queues in this paper. In the algorithm of Nazari and Stolyar (2019), there is one true queue and one virtual queue per item type. Both increase on item arrivals. However, virtual queues decrease whenever virtual matches are executed, and the identity of the virtual match configuration is decided based

on optimizing a combination of the revenue r_m and the current virtual queue lengths. A virtual match is added to a list of incomplete matches which are converted into true matches when there are enough resources available in the true queues. There are two differences in our paper: (i) there is a virtual queue for each item-configuration pair, and (ii) it is the assignment of items to match configurations that happens based on virtual queues; virtual/true matches execute whenever there are enough virtual/true resources committed to a match configuration. The use of virtual queues in (Nazari and Stolyar 2019) allows them to adapt to unknown arrival rates λ without explicitly learning them, as the virtual queue lengths converge to the dual variables of the SPP. However, their algorithm requires tuning a parameter β, roughly corresponding to the inverse of the virtual queue lengths, to be "as large as possible, but not larger."

6. Regret Analysis

In Section 6.1 we prove Theorem 2 which upper bounds the any-time regret for Algorithm 2. In Section 6.2 we prove Theorem 3 which lower bounds the anytime regret for any online algorithm in the case of configurations with both $\mathcal{I}^{\text{on-q}}$ and $\mathcal{I}^{\text{on-nq}}$ resources by $\Omega(\log t)$.

6.1. Regret Upper Bound for Sum-of-Squares Algorithm

We prove the regret bound in three steps.

Step 1 (Bounding the gap between true reward and virtual reward): Note that for any configuration m and time t, the virtual reward $r_m \mathscr{X}_m^t$ is an upper bound on the true reward $r_m X_m^t$. Our first lemma bounds this gap.

LEMMA 5. *For a configuration m and time t, the gap between online and offline matches is bounded as follows:*

Case 1: m with both online-queueable and online-nonqueueable resources $(\mathcal{I}(m) \cap \mathcal{I}^{\text{on-q}} \neq \emptyset$, and $\mathcal{I}(m) \cap \mathcal{I}^{\text{on-nq}} \neq \emptyset)$:

$$\mathscr{X}_m^t - X_m^t \leq \max_{i \in \mathcal{I}(m) \cap \mathcal{I}^{\text{on-q}}} \max_{s \in [t]} (\mathscr{Q}_{im}^s)^- + \sum_{i \in \mathcal{I}(m) \cap \mathcal{I}^{\text{off}}} \sum_{m' : i \in \mathcal{I}(m')} (\mathscr{Q}_{im'}^t)^- + |\mathcal{I}(m)|.$$

Case 2: Otherwise; m without online-nonqueueable resource, or without online-queueable resource:

$$\mathscr{X}_m^t - X_m^t \leq \sum_{i \in \mathcal{I}(m) \cap \mathcal{I}^{\text{off}}} \sum_{m' : i \in \mathcal{I}(m')} (\mathscr{Q}_{im'}^t)^- + |\mathcal{I}(m)|.$$

Case 2 implies that for configurations with only $\mathcal{I}^{\text{on-q}}$ resources, the number of virtual and true matches are the same.

Lemma 5 was not required for the case studied in Section 2 since we did not need virtual queues in that setting.

Step 2 (Bounding the gap between virtual reward and offline optimal reward): The next sample path lemma states that if $\mathbf{A}^t/t$ is close to $\bar{\boldsymbol{\lambda}}$, then the size of the virtual queues give a bound on the gap between the offline optimal reward and the virtual reward.

LEMMA 6. *If $\mathbf{A}^t/t \in \mathcal{B}_{\epsilon_0, TV}(\bar{\boldsymbol{\lambda}})$, then,*

$$OPT(\mathbf{A}^t) - \sum_m r_m \cdot \mathscr{X}_m^t = \mathbf{r}_{\mathcal{M}_+}^\top \mathbf{M}_{\mathcal{M}_+}^{-1} \sum_{m \in \mathcal{M}_+} \boldsymbol{\mathscr{Q}}_m^t.$$

In the above, $\mathbf{r}_{\mathcal{M}_+}$ denotes the vector $(r_m)_{m \in \mathcal{M}_+}$, and $\mathbf{M}_{\mathcal{M}_+}$ denotes the matrix $(\mathbf{M}_m)_{m \in \mathcal{M}_+}$.

As a corollary of this sample path lemma, we get the following bound on the gap between the expected offline optimal reward and the expected virtual reward, which generalizes Lemma 1 to non-stationary arrival instances.

LEMMA 7. *If $\{\boldsymbol{\lambda}^t\}$ satisfies the smoothed $\mathsf{GPG}_{\delta, \tau}$ condition for $\delta > 0$, then,*

$$\mathbf{E}[OPT(\mathbf{A}^t)] - \mathbf{E}\left[\sum_m r_m \cdot \mathscr{X}_m^t\right] \le \mathbf{r}_{\mathcal{M}_+}^\top \mathbf{M}_{\mathcal{M}_+}^{-1} \sum_{m \in \mathcal{M}_+} \mathbf{E}[\boldsymbol{\mathscr{Q}}_m^t] + \mathcal{O}(\tau).$$

Step 3 (Bounding $\mathscr{Q}_{im}^t$ and $\min_{s \in [t]} \mathscr{Q}_{im}^s$): To apply Lemma 5 we need to show that the virtual queues remain small (either in expectation for any fixed time t, or the minimum over the interval $[t]$). As a first step towards this, in the next lemma we show that a carefully constructed Lyapunov function that upper bounds the sum-of-squares potential $\Phi(\boldsymbol{\mathscr{Q}}^t)$ exhibits a useful negative drift condition. Proving this Lemma is the main technical novelty of the paper (compared to the drift analysis in Section 2 which followed in straightforward manner from (Csirik et al. 2006)).

LEMMA 8. *There exists a non-negative stochastic process Ξ^t, and a potential function $\Psi^t = \sqrt{\Phi(\boldsymbol{\mathscr{Q}}^t) + \Xi^t}$ for Algorithm 2, satisfying, for all $t \in [T]$,*

$$0 \le \Xi^t \le 2\tau\left(\sqrt{8\Phi(\boldsymbol{\mathscr{Q}}^t)} + 8\tau + 2\ln(1 + 4\tau)\right),$$

and such that Ψ^t obeys the following Lyapunov-drift conditions for $t \ge \tau + 1$ under the smoothed $\mathsf{GPG}_{\delta, \tau}$ assumption:

(a) Bounded variation: $|\Psi^{t+1} - \Psi^t| \le K_1(\tau)$

(b) Expected Decrease:

$$\mathbf{E}\left[\Psi^{t+1} - \Psi^t | \Psi^t \ge K_2(\tau, \delta))\right] \le -\frac{\delta}{8\sqrt{2}n^2}.$$

In the above,

$$K_1(\tau) := 6\tau^2 + 9\tau + 2\tau \log(3 + 4\tau) + 8,$$

$$K_2(\tau, \delta) := \sqrt{2 \max\{c_1(\tau, \delta), c_2(\tau)\}} = \mathcal{O}\left(n^2 \tau^2 / \delta\right).$$

where $c_1(\tau, \delta), c_2(\tau)$ are constants defined in (EC.17) *and* (EC.18), *respectively.*

A result of Hajek (1982) shows that under the conditions of Lemma 8, for some constant $\eta > 0$, $\mathbf{E}\left[e^{\eta \Psi^t}\right]$ is bounded for all t, as well as that this process is a super-martingale in the sense $\mathbf{E}\left[e^{\eta \Psi^{t+1}} | \Psi^t \geq K_2(\tau, \delta)\right] \leq \rho e^{\eta \Psi^t}$ for some $0 \leq \rho < 1$. However a direct application of the bounds from Hajek (1982) give us an expected regret of $\mathcal{O}(1/\delta^2)$, not $\mathcal{O}(1/\delta)$. Instead we will use the bound we proved in Lemma 3 to upper bound the process Ψ^t in increasing convex order by a simple reflected random walk to finish the proof.

Proof of Theorem 2: We apply Lemma 3 by choosing $\Upsilon^t := \Psi^t, K_1 := K_1(\tau), K_2 := K_2(\tau, \delta), \delta' := \frac{\delta}{16\sqrt{2}n^2 K_1(\tau)}$ to conclude that $\{\Psi^t\}_{t \in [\tau, T]} \leq_{icx} \{\Gamma^t\}_{t \in [\tau, T]}$. Choosing $f(\mathbf{X}) = X^t$ in Definition 1, and using Lemma 3, we get $\mathbf{E}[\Psi^t] \leq \mathbf{E}[\Gamma^t]$ for all $t \geq \tau$. Defining $\rho = \frac{1 - 2\delta'}{1 + 2\delta'}$, a straightforward calculation shows that the expectation of the steady state distribution for the random walk $\{\Gamma^t\}$ is $\mathbf{E}[\Gamma^\infty] = \Gamma_{\min} + K_1 \frac{\rho}{1-\rho}$. Since the random walk $\{\Gamma^t\}$ starts at Γ^τ, and not in steady state, a simple coupling argument shows that we can bound

$$\mathbf{E}[\Gamma^t] \leq \max\{\Gamma^\tau, \Gamma_{\min}\} + K_1 \frac{\rho}{1 - \rho} = \mathcal{O}(n^2 \tau^4 / \delta).$$

Since for each $m \in \mathcal{M}_+$, $i \in \mathcal{I}(m)$, $\mathbf{E}[\mathcal{Q}_{im}^t] \leq \mathbf{E}[\Phi^t] \leq \mathbf{E}[\Psi^t]$, the above implies that the expected length of virtual queues is bounded by a constant depending on (n, τ, δ), and combined with Lemma 5 (Case 2) and Lemma 6 gives bounded anytime regret for multiway matching when no configuration has both online-queueable and online-nonqueueable resources.

We next proceed to prove logarithmic regret for the case when some configurations have both online-queueable and online-nonqueueable resources, in which case according to Case 1 of Lemma 5, we need to bound $\mathbf{E}\left[\max_{s \in [t]} (\mathcal{Q}_{im}^s)^-\right]$, which is in turn bounded by $\mathbf{E}\left[\max_{s \in [t]} \Psi^s\right]$. Since max is an increasing convex function, we further have the bound, $\mathbf{E}\left[\max_{s \in [t]} \Psi^s\right] \leq \mathbf{E}\left[\max_{s \in [t]} \Gamma^s\right]$.

To bound $\mathbf{E}\left[\max_{s \in [t]} \Gamma^s\right]$, consider a random walk on $\mathbb{N}$ with initial state 1, reflection boundary 0, and probabilities of step size of $+1, -1$ being $\frac{1}{2} - \delta'$ and $\frac{1}{2} + \delta'$, respectively. Let S denote the random variable for the supremum of this random walk until the process first hits 0. We can write the following recurrence inequality for the distribution of S:

$$\mathbf{Pr}[S \geq 1] = 1,$$

$$\mathbf{Pr}[S \geq 2] = \left(\frac{1}{2} - \delta'\right),$$

$$\text{for } x \geq 3, \ \mathbf{Pr}[S \geq x] \leq \left(\frac{1}{2} + \delta'\right) \cdot 0 + \left(\frac{1}{2} - \delta'\right)(\mathbf{Pr}[S \geq x-1] + \mathbf{Pr}[S \geq x])$$

$$\leq \rho\mathbf{Pr}[S \geq x-1] \leq \left(\frac{1}{2} - \delta'\right)\rho^{x-2}.$$

In words, the supremum of Γ^t in one excursion above $\Gamma_{\min}$ until it hits this level again is stochastically bounded from above by $3K_1$ plus a geometrically distributed random variable with mean $\frac{K_1}{1-\rho} \leq \frac{K_1}{\delta'}$. Since there can be at most t such excursions in the interval $\{1, \ldots, t\}$, and the expectation of the maximum of t i.i.d. geometrically distributed random variables, each with mean K_1/δ', is given by $H_t \frac{K_1}{\delta'}$ ($H_t \sim \log t$ is the tth harmonic number), we get,

$$\mathbf{E}\left[\max_{s \in [t]} \Psi^s\right] \leq \mathbf{E}\left[\max_{s \in [t]} \Gamma^s\right] \leq \Gamma_{\min} + 3K_1 + \frac{K_1}{\delta'}H_t = K_2(\tau, \delta) + \mathcal{O}\left(\frac{n^2\tau^4 \log t}{\delta}\right).$$

Combined with Lemma 6 (Case 1) and Lemma 5, the above proves the second part of Theorem 2. ∎

6.2. Regret Lower Bound

Our goal is to show that, in general, for matching instances where configurations involve both online-queueable and online-nonqueueable resources, there must be some time t at which an expected regret of $\Omega(\log t)$ compared to the offline optimal for the instance seen until t is unavoidable. To prove this, we consider one specific instance, illustrated in Figure 1. There are two matching configurations, $\mathcal{M} = [2]$ and three resource types with $\mathcal{I}^{\text{on-q}} = \{1\}$ and $\mathcal{I}^{\text{on-nq}} = \{2, 3\}$. The matching matrix is

$$\mathbf{M} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

The solution of $\mathsf{SPP}(\boldsymbol{\lambda})$ is $x_1 = 0.3, x_2 = 0.05$.

In words, only resource 1 can be queued and both resource type 2 and 3 are only available for matching on arrival. Since matches with resource type 2 generate higher reward, and resource type 2 are scarcer than resource type 1, any time a resource of type 2 arrives and there are no resources of type 1 in the queue, we incur additional regret. The key intuition behind the regret lower bound of $\Omega(\log t)$ is that if at some time t, the expected number of units of resource 1 in the queue is $\Omega(\log t)$, then we have already incurred a regret of $\Omega(\log t)$ *at this time*, since the offline optimal restricted to arrivals in $[t]$ would have matched all resources of type 1. Instead, if we have an expected queue length of $o(\log t)$ at all times, then with high probability, there can be a sequence of $\Omega(\log T)$ consecutive arrivals of type 2, most of which are lost due to insufficient type 1 resources. This causes us to incur $\Omega(\log T)$ regret at the terminal time T.
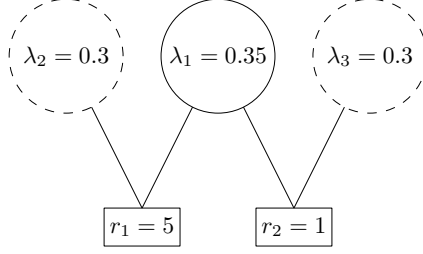
Figure 1   Instance used for proving Theorem 3. Online-queueable resources correspond to solid circles, and online-nonqueueable to dashed circles. The matching configurations are represented by rectangles.
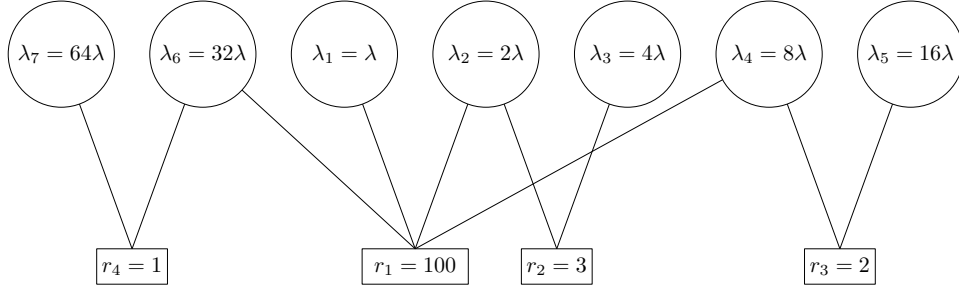
Figure 2   Multiway matching instance with only online-queueable resources (Figure 9 from Kerimov et al. (2021a)). λ is the normalization constant.

LEMMA 9. *For the instance shown in Figure 1, there exists a constant $c > 0$, and an integer T_0, such that for all $T \geq T_0$, for any online algorithm, either*

1. *there are exists some $t \in [T/2, T]$ such that $\mathbf{E}[N_1^t] \geq 2c \log T$, and therefore,*

$$\mathbf{E}[r_1 X_1^t + r_2 X_2^t] \leq \mathbf{E}[OPT(\mathbf{A}^t)] - c \log T,$$

2. *or $\mathbf{E}[X_1^T] \leq A_2^T - c \log T$, and therefore,*

$$\mathbf{E}\left[r_1 X_1^T + r_2 X_2^T\right] \leq \mathbf{E}\left[OPT(\mathbf{A}^T)\right] - 2c \log T.$$

Proof of Theorem 3:   The proof immediately follows from Lemma 9. If the first case of the lemma applies, then the regret at such a time is $\Omega(\log T)$, otherwise the second case applies and the regret at time T is $\Omega(\log T)$. ∎

7. Experiments

Instance 1: The first instance in our experiments is an instance with online-queueable resources which we borrow from Kerimov et al. (2021a) and reproduce in Figure 2. The instance was mentioned in that work as one where a natural greedy policy does not give bounded regret. The intuition behind the observation is that the match of type 1 generates the highest rewards, but involves resources which are part of other matching configurations. A greedy algorithm may not

leave enough resources for type 1 matches. This is where the idea that the right view of designing a greedy algorithm is that it should be greedy in committing resources to configurations, rather than only at the time of matching is critical. A second notable feature of the experiments in Kerimov et al. (2021a) is that depending on the choice of the tuning parameter which controls the frequency of the resolving algorithm in Kerimov et al. (2021a), the optimality gap could be potentially quite large. Figure 3 shows one simulation run of the Sum-of-Squares algorithm for this instance from Kerimov et al. (2021a) which highlights the bounded regret of the proposed algorithm.
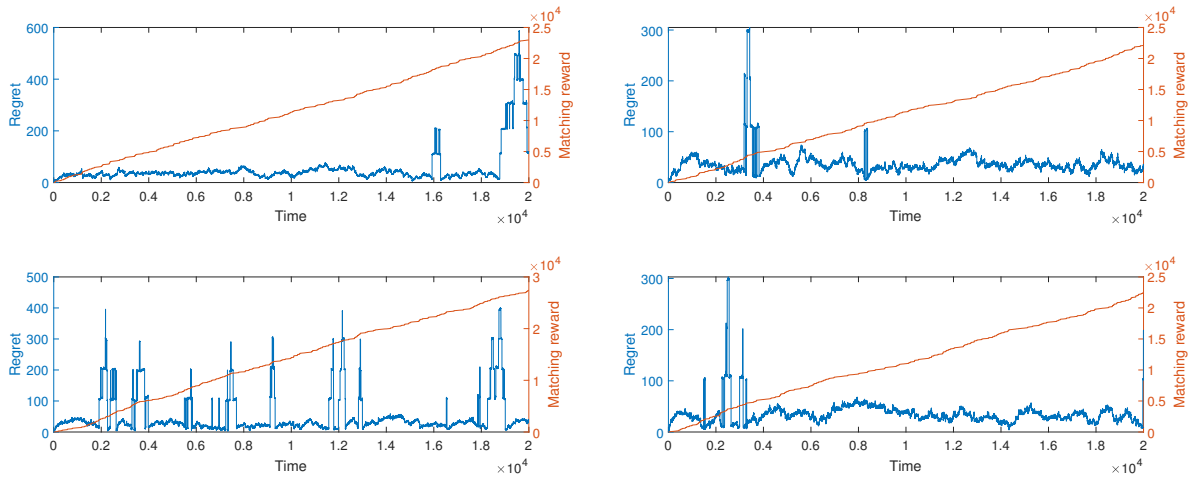
Figure 3   Four simulation paths for the Sum-of-Squares algorithm applied to the instance in Figure 9 of (Kerimov et al. 2021a) (reproduced in Figure 2). The red curves show the cumulative matching reward for the offline optimal policy. The blue curves show the regret of the proposed Sum-of-Squares algorithm.

Instance 2: The second instance we simulate is the lower bound instance from Figure 1 used to prove Theorem 3 for the case where configurations have both online-queueable and online-nonqueueable resources. In Figure 4 we compare our algorithm (blue curve) against the offline optimal (which is allowed to rematch all arrivals, even if the matches may not be realizable online), and the performance of a Dynamic Programming optimal policy (black curve) which aims to maximize the terminal matching reward. While the terminal regret of the DP optimal policy is superior to the terminal regret of the Sum-of-Squares algorithm, Sum-of-Squares performs comparably to the DP optimal policy for the entire interval $[T]$ as Theorems 2 and 3 prove.

Instance 3: The third instance we simulate is for the stochastic bin packing problem, where our goal is to compare the performance of our SS algorithm compared to the Sum-of-Squares algorithm of Csirik et al. (2006), and the modification where dead-end levels are forbidden. To be able to use the algorithm of Csirik et al. (2006) as is, we will focus on a packing instance corresponding to Bounded Waste distribution (which are a subclass of distributions exhibiting GPG). In the instance
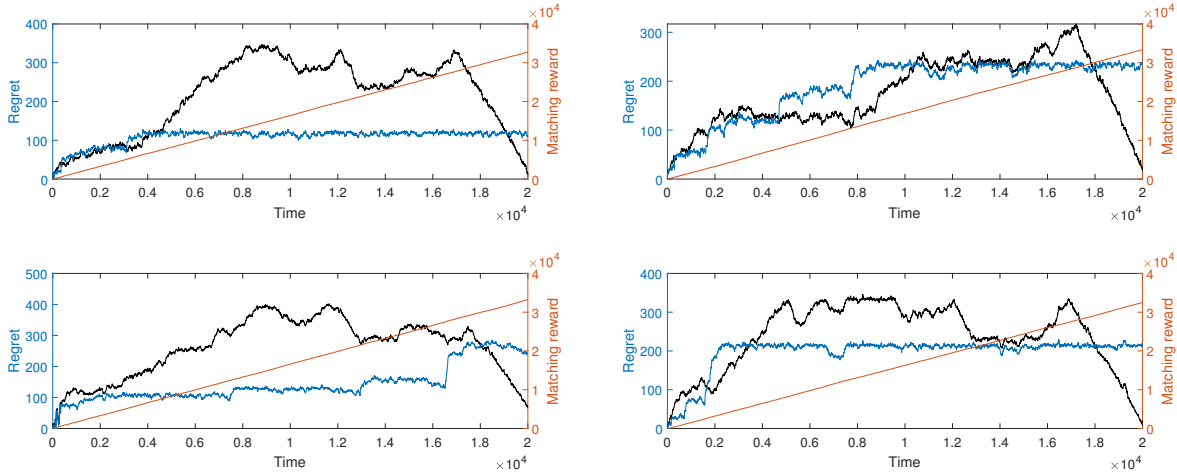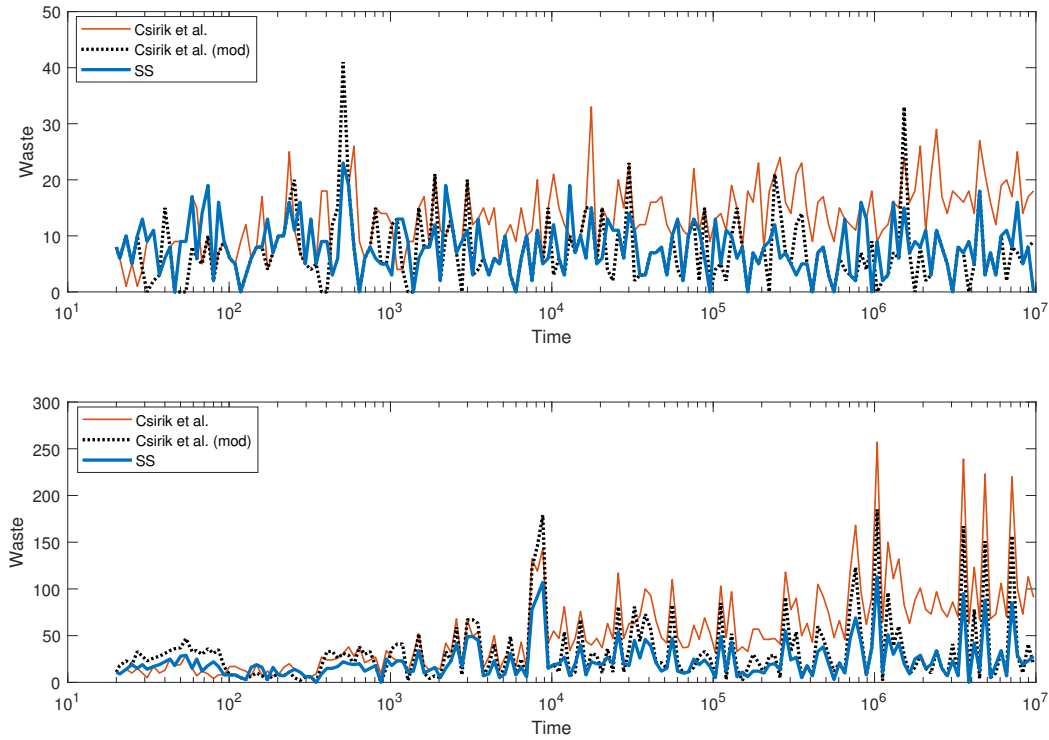
Figure 4   Four simulation paths for the Sum-of-Squares algorithm applied to the instance in Figure 1. The offline optimal computed here is an upper bound on the matching reward under the assumption that all arrivals can be queued and rematched. The black curve shows the regret of the Dynamic Programming based policy for maximizing expected reward at the terminal time. The blue curve shows the regret of the Sum-of-Squares algorithm.

we simulate (adapted from (Csirik et al. 2006)), items of size 2 or 3 arrive over the packing horizon, and must be packed on arrival into bins of size $B = 9$. The size of the arriving item is sampled i.i.d., with λ_2 denoting the probability that the item is of size 2, and $\lambda_3 = 1 - \lambda_2$ that it is of size 3. Any distribution with $\lambda_3 \geq 1/4$ is Perfectly Packable, and is a Bounded Waste distribution if the inequality is strict.

Figure 5 shows the simulation results, where we plot the sample path of the waste incurred by the algorithm of Csirik et al. (2006) and our proposed SS algorithm. Waste stands for the unused space in the open bins used by the online packing algorithms. The solution to the SPP would correspond to a waste of 0, so our plots are an upper bound on the regret. We compare our algorithm with the basic Sum-of-Squares algorithm of Csirik et al. (2006) (labeled *Csirik et al.* in the plots), and the modification where dead-end levels are never created (labeled *Csirik et al. (mod)*). For the instance we simulate, level 8 is the only dead-end level. We simulate two distributions: with $\lambda_3 = 1/4 + 1/8$ (top) and $\lambda_3 = 1/4 + 1/64$. As we mentioned earlier, we get a Bounded Waste distribution when $\lambda_3 > 1/4$, and the distance to $1/4$ is a measure of the general position gap.

The first observation we can make is that the basic algorithm of Csirik et al. (2006) incurs a waste that grows as $\Theta(\log t)$, as proved by the authors. This is more apparent in the bottom plot, and occurs due to accumulation of level 8 bins. The second observation is that both, our SS algorithm, as well as the modified Sum-of-Squares algorithm of Csirik et al. (2006), have waste which does not seem to grow with t, and have comparable performance.

Figure 5   Simulation paths for the Sum-of-Squares algorithm applied to a stochastic bin packing instance with Bounded Waste item size distribution. The bin size is $B = 9$, and the item size distribution is $1 - \lambda_2 = \lambda_3 = 1/4 + \epsilon$, with $\epsilon = 1/8$ for the top plot and $\epsilon = 1/64$ for the bottom plot. The curves show the wasted space in opened bins used by three algorithms: *SS* corresponds to the algorithm proposed in the current paper, *Csirik et al.* corresponds to the basic Sum-of-Squares algorithm of Csirik et al. (2006), and *Csirik et al. (mod)* corresponds to the variation where bins of level 8 (a dead-end level) are never created.

8. Concluding Remarks

The main purpose of this research note is to highlight the power of knowing partial information, i.e., the basic feasible activities, about the optimal offline solution. While some of the ideas in this paper have perhaps been hidden in some of the existing literature, we believe they have not been expressed explicitly. A second contribution we make is to present a slightly more unified view of online multiway matching by considering three types of resources. Finally, the potential function analysis we conduct for non-stationary distributions by borrowing some tricks from amortized analysis is likely to be of broader interest. We would like to reiterate that for some of the problems which fall within our unified framework, stronger results are known (bounded regret without GPG), and whether these results can be obtained via pure greedy algorithms would be an interesting direction to explore. Further, whether knowledge of basic feasible activities alone can be used to create priority policies which are oblivious to the entries of matrix $\mathbf{M}$ (other than using whether

they are zero or non-zero, as is done by Stolyar and Yudovina (2013) for queueing systems) is interesting to explore.

Endnotes

1. Our results extend to settings where configurations require multiple units of resources from $\mathcal{I}^{\text{on-nq}}$, but these resources must be committed on arrival to partially completed configurations which have their demand from $\mathcal{I}^{\text{off}} \cup \mathcal{I}^{\text{on-q}}$ met.

2. Our results extend to the case where discarding is prohibited as long as the set of configurations is downward closed, that is, strict subsets of feasible configurations are also feasible.

3. An example where the solution of the primal (SPP($\boldsymbol{\lambda}$)) is non-unique is the following. Consider stochastic bin packing instance with bins of size $B = 10$, and items of sizes 2,6 and 7. In particular $\lambda_6 = \lambda_7 = 0.4$ and $\lambda_2 = 0.2$ denote the probabilities with which an arrival is of size 6,7 and 2, respectively. In this case there are multiple optimal basis, for example $\{(2,2,6),(6),(7)\}$, $\{(2,6),(6),(7)\}$, $\{(6),(2,7),(7)\}$, where, e.g., $(2,2,6)$ denotes the configuration with two size 2 items and one size 6 item. However, the optimal solution to (DSPP($\boldsymbol{\lambda}$)) is unique: $\alpha_2^* = 0, \alpha_6^* = -1, \alpha_7^* = -1$.

References

Arlotto, Alessandro, Itai Gurvich. 2019. Uniformly bounded regret in the multisecretary problem. *Stochastic Systems* **9**(3) 231–260.

Banerjee, Siddhartha, Daniel Freund. 2020. Uniform loss algorithms for online stochastic decision-making with applications to bin packing. *Abstracts of the 2020 SIGMETRICS/Performance Joint International Conference on Measurement and Modeling of Computer Systems*. 1–2.

Bumpensanti, Pornpawee, He Wang. 2020. A re-solving heuristic with uniformly bounded loss for network revenue management. *Management Science* **66**(7) 2993–3009.

Chen, Guanting, Xiaocheng Li, Yinyu Ye. 2021. Fairer LP-based Online Allocation via Analytic Center. *arXiv preprint arXiv:2110.14621* .

Csirik, Janos, David S Johnson, Claire Kenyon, James B Orlin, Peter W Shor, Richard R Weber. 2006. On the Sum-of-Squares algorithm for bin packing. *Journal of the ACM (JACM)* **53**(1) 1–65.

Ding, Yichuan, S McCormick, Mahesh Nagarajan. 2018. A fluid model for an overloaded bipartite queueing system with heterogeneous matching utility. *Available at SSRN 2854492* .

Gupta, Varun, Ana Radovanović. 2020. Interior-point-based online stochastic bin packing. *Operations Research* **68**(5) 1474–1492.

Hajek, Bruce. 1982. Hitting-time and occupation-time bounds implied by drift analysis with applications. *Advances in Applied probability* **14**(3) 502–525.

Huang, Longbo, Michael J Neely. 2009. Delay reduction via Lagrange multipliers in stochastic network optimization. *2009 7th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks*. IEEE, 1–10.

Jasin, Stefanus, Sunil Kumar. 2012. A re-solving heuristic with bounded revenue loss for network revenue management with customer choice. *Mathematics of Operations Research* **37**(2) 313–345.

Kerimov, Süleyman, Itai Ashlagi, Itai Gurvich. 2021a. Dynamic Matching: Characterizing and Achieving Constant Regret. *Available at SSRN 3824407* .

Kerimov, Süleyman, Itai Ashlagi, Itai Gurvich. 2021b. On the Optimality of Greedy Policies in Dynamic Matching. *Available at SSRN 3918497* .

Lavrov, Mikhail. 2020. Lecture notes in Linear Programming - Duality and the Simplex Tableau. `https://faculty.math.illinois.edu/~mlavrov/docs/482-spring-2020/lecture14.pdf`. Accessed: July 16, 2022.

Miller, Bruce L. 1969. A queueing reward system with several customer classes. *Management Science* **16**(3) 234–245.

Müller, Alfred, Dietrich Stoyan. 2002. *Comparison Methods for Stochastic Models and Risks*, vol. 389. Wiley.

Nazari, Mohammadreza, Alexander L Stolyar. 2019. Reward maximization in general dynamic matching systems. *Queueing Systems* **91**(1) 143–170.

Stolyar, Alexander L, Elena Yudovina. 2013. Tightness of invariant distributions of a large-scale flexible service system under a priority discipline. *Stochastic Systems* **2**(2) 381–408.

Tassiulas, Leandros, Anthony Ephremides. 1990. Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks. *29th IEEE Conference on Decision and Control*. IEEE, 2130–2132.

Vera, Alberto, Siddhartha Banerjee, Itai Gurvich. 2021. Online allocation and pricing: Constant regret via Bellman inequalities. *Operations Research* .

Online Appendix

Appendix A: Proofs

A.1. Proof of Lemma 3

First observe that for any random variable Δ with support on $[-K_1, +K_1]$ and mean $\mathbf{E}[\Delta]$, the random variable Δ' with support on $\{-K_1, +K_1\}$ and mean $\mathbf{E}[\Delta'] \geq \mathbf{E}[\Delta]$ satisfies $\Delta \leq_{icx} \Delta'$.

To prove the lemma, we give a construction that satisfies the second part of Definition 1. That is, we construct $\{\hat{\Upsilon}^t\}$ and $\{\hat{\Gamma}^t\}$ on a common probability space so that $\mathbf{E}\left[\hat{\Gamma}|\hat{\Upsilon}\right] \geq \hat{\Upsilon}$ almost surely. In our construction $\hat{\Upsilon}^t = \Upsilon^t$ for all $t \in [T]$, and the only interesting part is the construction of $\{\hat{\Gamma}^t\}$ on the probability space of $\{\Upsilon^t\}$.

Base case: $\hat{\Gamma}^\tau = \Gamma^\tau$. By the choice of Γ^τ, we have $\hat{\Gamma}^\tau \geq \hat{\Upsilon}^\tau$ almost surely.

Inductive Step: Assume that we have constructed on a joint probability space $\{\hat{\Upsilon}^s\}_{s \in [\tau, t]}$ and $\{\hat{\Gamma}^s\}_{s \in [\tau, t]}$ satisfying $\mathbf{E}\left[\hat{\Gamma}|\hat{\Upsilon}\right] \geq \hat{\Upsilon}$, obeying the marginals. Denote

$$\hat{\Delta}_\Upsilon^{t+1} = \hat{\Upsilon}^{t+1} - \hat{\Upsilon}^t.$$

The random variable $\hat{\Gamma}^{t+1}$ is given by:

$$\hat{\Gamma}^{t+1} = \begin{cases} \max\left\{\Gamma_{min}, \hat{\Gamma}^t + \hat{\Delta}_\Gamma^{t+1}\right\} & \text{if } \hat{\Upsilon}^t \geq K_2, \\ \max\left\{\Gamma_{min}, \hat{\Gamma}^t + \Delta_\Gamma^{t+1}\right\} & \text{if } \hat{\Upsilon}^t < K_2. \end{cases}$$

In the above $\hat{\Delta}_\Gamma^{t+1}$ is a random variable with support $\{-K_1, +K_1\}$, mean $\mathbf{E}\left[\hat{\Delta}_\Gamma^{t+1}\right] = -2\delta' K_1$, that is coupled with $\hat{\Delta}_\Upsilon^{t+1}$ so that *(i)* conditioned on $\hat{\Delta}_\Upsilon^{t+1}$, $\hat{\Delta}_\Gamma^{t+1}$ is independent of $\{\hat{\Gamma}^s\}_{s \in [t]}$ and $\{\hat{\Upsilon}^s\}_{s \in [t]}$, and *(ii)* and

$$\mathbf{E}\left[\hat{\Delta}_\Gamma^{t+1}|\hat{\Delta}_\Upsilon^{t+1}\right] \geq \hat{\Delta}_\Upsilon^{t+1}.$$

The random variable $\Delta_\Gamma^{t+1} = \xi^{t+1}\gamma$ with $\xi^{t+1} \in \{-1, +1\}$ denoting a random variable with distribution given in the Lemma, and independent of $\hat{\Upsilon}$ and $\{\hat{\Gamma}^s\}_{s \in [t]}$.

Note that the marginal distributions of Δ_Γ^{t+1} and $\hat{\Delta}_\Gamma^{t+1}$ are identical, and both random variables are independent of $\{\hat{\Gamma}_s\}_{s \in [t]}$, thus proving that the marginal distribution of $\hat{\Gamma}$ agrees with Γ.

It remains to show that we have

$$\mathbf{E}\left[\hat{\Gamma}^{t+1}|\hat{\Upsilon}\right] = \mathbf{E}\left[\hat{\Gamma}^{t+1}|\{\hat{\Upsilon}^s\}_{s \in [t+1]}\right] \geq \hat{\Upsilon}^{t+1}.$$

On the event $\{\hat{\Upsilon}^t < K_2\}$ we have $\hat{\Upsilon}^{t+1} \leq \hat{\Upsilon}^t + K_1 \leq K_2 + K_1 \leq \Gamma_{\min} \leq \hat{\Gamma}^{t+1}$.

On the event $\{\hat{\Upsilon}^t \geq K_2\}$:

$$\mathbf{E}\left[\hat{\Gamma}^{t+1}|\{\hat{\Upsilon}^s\}_{s\in[t+1]}, \hat{\Upsilon}^t \geq K_2\right] \geq \mathbf{E}\left[\hat{\Gamma}^t|\{\hat{\Upsilon}^s\}_{s\in[t+1]}, \hat{\Upsilon}^t \geq K_2\right] + \mathbf{E}\left[\hat{\Delta}_\Gamma^{t+1}|\{\hat{\Upsilon}^s\}_{s\in[t+1]}, \hat{\Upsilon}^t \geq K_2\right]$$
$$\geq \hat{\Upsilon}^t + \mathbf{E}\left[\hat{\Delta}_\Gamma^{t+1}|\hat{\Delta}_\Upsilon^{t+1}, \hat{\Upsilon}^t \geq K_2\right]$$
$$\geq \hat{\Upsilon}^t + \hat{\Delta}_\Upsilon^{t+1}$$
$$= \hat{\Upsilon}^{t+1},$$

as required. The first inequality follows since $\hat{\Gamma}^{t+1} = \max\{\Gamma_{\min}, \hat{\Gamma}^t + \hat{\Delta}_\Gamma^{t+1}\}$, the second follows by the induction hypothesis $\mathbf{E}\left[\hat{\Gamma}^t|\{\hat{\Upsilon}^s\}_{s\in[t+1]}\right] \geq \hat{\Upsilon}^t$, and the third inequality follows by the coupling between $\hat{\Delta}_\Upsilon^{t+1}$ and $\hat{\Delta}_\Gamma^{t+1}$.

A.2. Proof of Lemma 4

It is an easy observation that for each $t \in [T]$, $\sum_{i\in\mathcal{I}^{\text{off}}\cup\{0\}} \left(A_i^t - \bar{\lambda}_i \cdot t\right) = \sum_{i\in\mathcal{I}^{\text{off}}\cup\{0\}} A_i^t - t = 0$. Therefore, if for some i, $A_i^{t-1} - \bar{\lambda}_i \cdot (t-1) > 0$, then there exists another i' with $A_{i'}^{t-1} - \bar{\lambda}_i \cdot (t-1) < 0$, and therefore

$$A_i^t - \bar{\lambda}_i \cdot t < A_i^{t-1} - \bar{\lambda}_i \cdot (t-1).$$

Since $A_i^t - \bar{\lambda}_i \cdot t$ increases by at most $1 - \bar{\lambda}_i$ in any time step, and decreases whenever it is strictly positive, it takes a maximum value of $1 - \bar{\lambda}_i$.

To lower bound A_i^t, we note that $A_i^t = t - \sum_{i'\in\mathcal{I}^{\text{off}}\cup\{0\}\backslash\{i\}} A_{i'}^t \geq t - \sum_{i'\in\mathcal{I}^{\text{off}}\cup\{0\}\backslash\{i\}} (\bar{\lambda}_{i'}\cdot t + 1 - \bar{\lambda}_{i'}) \geq \bar{\lambda}_i \cdot t - n^{\text{off}}$.

Finally, $A_i^T = \bar{\lambda}_i \cdot T$ follows because A_i^T is at most $\bar{\lambda}_i \cdot T + (1 - \bar{\lambda}_i)$, and hence being integral, at most $\bar{\lambda}_i \cdot T$. Combining $\sum_i A_i^T = T$, $A_i^T \leq \bar{\lambda}_i T$ for all $i \in \mathcal{I}^{\text{off}} \cup \{0\}$, and $\sum_i \bar{\lambda}_i T = T$, we get $A_i^T = \bar{\lambda} \cdot T$.

A.3. Proof of Proposition 1

We first show that Definition 2 implies uniqueness of the optimal solution $\boldsymbol{\alpha}^*$ of (DSPP($\boldsymbol{\lambda}$)). Let $\mathcal{M}_+$ be an optimal basis. By the assumption in Definition 2, the optimal primal variables for $\hat{\boldsymbol{\lambda}} \in \mathcal{B}_{\epsilon,\text{TV}}(\boldsymbol{\lambda})$ with the basis $\mathcal{M}_+$ are given by:

$$\mathbf{x}_{\mathcal{M}_+}^*(\hat{\boldsymbol{\lambda}}) = \mathbf{M}_{\mathcal{M}_+}^{-1}\hat{\boldsymbol{\lambda}},$$

with optimal objective function value

$$\mathbf{r}_{\mathcal{M}_+}^\top \mathbf{M}_{\mathcal{M}_+}^{-1}\hat{\boldsymbol{\lambda}}.$$

It follows that $\boldsymbol{\alpha}^* = \left(\mathbf{r}_{\mathcal{M}_+}^\top \mathbf{M}_{\mathcal{M}_+}^{-1}\right)^\top$ is an optimal solution to the dual DSPP($\hat{\boldsymbol{\lambda}}$) (see, e.g., the lecture notes by Lavrov (2020)). Since $\boldsymbol{\alpha}^*$ is an optimal dual solution for all $\hat{\boldsymbol{\lambda}} \in \mathcal{B}_{\epsilon,\text{TV}}(\boldsymbol{\lambda})$, and observing that we wrote the dual program (DSPP($\boldsymbol{\lambda}$)) as

$$\max_{\boldsymbol{\alpha}\in\mathcal{A}} \boldsymbol{\alpha}^\top \boldsymbol{\lambda}$$

for a polyhedron $\mathcal{A}$ which depends on $\mathbf{r}$ and $\mathbf{M}$ but not on $\boldsymbol{\lambda}$, $\boldsymbol{\alpha}^*$ must be a vertex of $\mathcal{A}$, and hence the unique optimal solution to (DSPP($\boldsymbol{\lambda}$)).

Conversely, since $\boldsymbol{\alpha}^*$ is an optimal solution for all $\hat{\boldsymbol{\lambda}} \in \mathcal{B}_{\epsilon,\mathrm{TV}}(\boldsymbol{\lambda})$, pick any $|\mathcal{I}|$ linear independent constraints that bind at $\boldsymbol{\alpha}^*$, and call these the basis $\mathcal{M}_+$. It follows that $\mathbf{x}^*_{\mathcal{M}_+}(\hat{\boldsymbol{\lambda}}) = \mathbf{M}^{-1}_{\mathcal{M}_+}\hat{\boldsymbol{\lambda}}$ is an optimal solution to DSPP($\hat{\boldsymbol{\lambda}}$).

A.4. Proof of Lemma 5

Case 1: For a configuration m with online-nonqueueable resources, a virtual match happens whenever an online-nonqueueable resource is committed to the configuration. However, a true match only happens if the offline resources are available (globally), and online-queueable resources are available (locally, per the true queue length).

To account for the gap in matches due to lack of online-queueable resources, we will use the difference between the true and virtual queue lengths for such resources. More formally, let $i \in \mathcal{I}(m) \cap \mathcal{I}^{\mathrm{on\text{-}nq}}$. Then the virtual queue length $\mathcal{Q}_{im}$ decreases by 1 whenever a virtual match happens, and the true queue length Q_{im} decreases by 1 only when a true match happens. However, both $\mathcal{Q}_{im}$ and Q_{im} increase when a unit of resource i is committed to m. Therefore,

$$\mathscr{X}^t_m - X^t_m = Q^t_{im} - \mathcal{Q}^t_{im}.$$

Now consider the last time before t that a virtual match happened but a true match did not happen because of lack of an online-queueable resource. Denote this time by θ, and let us continue to denote by i the type of this online-queueable resource. Therefore we must have $Q^\theta_{im} = Q^{\theta-1}_{im} \le 0$, and $\mathcal{Q}^\theta_{im} = \mathcal{Q}^{\theta-1}_{im} - 1$. Therefore, the gap between virtual and online matches up to and inclusive of time θ is

$$\mathscr{X}^\theta_m - X^\theta_m = Q^\theta_{im} - \mathcal{Q}^\theta_{im} \le 1 - \mathcal{Q}^\theta_{im} \le 1 + \max_{s\in[t]}\left(\mathcal{Q}^s_{im}\right)^- \le 1 + \max_{i\in\mathcal{I}(m)\cap\mathcal{I}^{\mathrm{on\text{-}nq}}}\max_{s\in[t]}\left(\mathcal{Q}^s_{im}\right)^-.$$

The gap in matches due to time interval $[\theta+1, t]$ is solely due to lack of offline resources globally, which we bound next. We first observe that the total number of units of a resource $i \in \mathcal{I}^{\mathrm{off}}$ used by all configurations can be bounded by the units consumed due to virtual matches

$$\sum_{m\in\mathcal{M}_+} M_{im} X^t_m \le \sum_{m\in\mathcal{M}_+} M_{im} \mathscr{X}^t_m.$$

Next, let us view Y^t_{im} as the "quota" of resource i that is assigned to configuration m. If all configurations consume less units of resource i than their quota, then there would be no lost true matches caused by lack of resource type i, and in this case all the virtual queues $\mathcal{Q}^t_{im}$ would be non-negative. Therefore, we can upper bound the gap between virtual and true matches of configuration m due to resource i into *(i)* the excess virtual consumption of resource i by m beyond its own quota –

this overconsumption is bounded from above by $(\mathscr{Q}_{im}^t)^-$; and *(ii)* the over-consumption of resource i by other configurations, thereby eating into configuration m's quota – this over-consumption is upper bounded by

$$\sum_{m':i\in\mathcal{I}(m),m\neq m'}(\mathscr{Q}_{im'}^t)^-.$$

This in turn leads to an upper bound on the gap between true and virtual matches of configuration m due to resource $i\in\mathcal{I}^{\text{off}}$ of

$$1+\sum_{m':i\in\mathcal{I}(m)}(\mathscr{Q}_{im'}^t)^-.$$

Summing the gaps due to online-queueable and offline resources, we get the bound in the Lemma:

$$\mathscr{X}_m^t-X_m^t\leq\max_{i\in\mathcal{I}(m)\cap\mathcal{I}^{\text{on-q}}}\max_{s\in[t]}(\mathscr{Q}_{im}^s)^-+\sum_{i\in\mathcal{I}(m)\cap\mathcal{I}^{\text{off}}}\sum_{m'|i\in\mathcal{I}(m')}(\mathscr{Q}_{im'}^t)^-+|\mathcal{I}(m)|.$$

Case 2: If configuration m does not have online-nonqueueable resources, then the gap between virtual and true matches occurs only due to lack of offline-resources (since virtual match is not performed if there are not enough virtual online-queueable resources, and the true queue length is always larger than virtual queue length for these resources). The gap due to lack of offline resources can be bound as in Case 1.

If the configuration m does not have online-queueable resources, then again a virtual match is performed as soon as the online-nonqueueable resource arrives, and the gap between virtual and true matches is only due to lack of offline resources. This again can be bounded as in Case 1.

A.5. Proof of Lemma 6

The final virtual queue lengths for all the resources are given by $\sum_m\mathscr{Q}_m^t$, and therefore the quantity of resources consumed to generate the virtual matches are $\mathbf{A}^t-\sum_m\mathscr{Q}_m^t$. Since we only generate matches in the set $\mathcal{M}_+$, and by our general position gap assumption $\mathbf{M}_{\mathcal{M}_+}$ is full rank, the number of virtual matches is given by the unique solution to the following linear system:

$$\forall i\in\mathcal{I}\ :\ \sum_{m\in\mathcal{M}}M_{im}\cdot\mathscr{X}_m^t=A_i^t-\sum_m\mathscr{Q}_{im}^t,$$

or,

$$\mathscr{X}^t=(\mathbf{M}_{\mathcal{M}_+})^{-1}\left(\mathbf{A}^t-\sum_m\mathscr{Q}_m^t\right),$$

or in terms of matching reward,

$$\mathbf{r}_{\mathcal{M}_+}^\top\mathscr{X}^t=\mathbf{r}_{\mathcal{M}_+}^\top(\mathbf{M}_{\mathcal{M}_+})^{-1}\left(\mathbf{A}^t-\sum_m\mathscr{Q}_m^t\right). \tag{EC.1}$$

Assuming $\mathbf{A}^t/t\in\mathcal{B}_{\epsilon_0,\text{TV}}(\bar{\boldsymbol{\lambda}})$, $\mathsf{GPG}_{\epsilon_0}$ condition implies, $\mathbf{r}_{\mathcal{M}_+}^\top(\mathbf{M}_{\mathcal{M}_+})^{-1}\mathbf{A}^t=\mathsf{OPT}(\mathbf{A}^t)$, and therefore,

$$\sum_m r_m\cdot\mathscr{X}_m^t=\mathsf{OPT}(\mathbf{A}^t)-\mathbf{r}_{\mathcal{M}_+}^\top(\mathbf{M}_{\mathcal{M}_+})^{-1}\sum_m\mathscr{Q}_m^t.$$

A.6. Proof of Lemma 7

Starting from (EC.1),

$$\mathbf{E}\left[\sum_{m \in \mathcal{M}_+} r_m \mathscr{X}_m^t\right] = \mathbf{r}_{\mathcal{M}_+}^\top (\mathbf{M}_{\mathcal{M}_+})^{-1} \mathbf{E}[\mathbf{A}^t] - \mathbf{r}_{\mathcal{M}_+}^\top (\mathbf{M}_{\mathcal{M}_+})^{-1} \mathbf{E}\left[\sum_m \mathscr{Q}_m^t\right]. \qquad \text{(EC.2)}$$

Our goal is to relate $\mathbf{r}_{\mathcal{M}_+}^\top (\mathbf{M}_{\mathcal{M}_+})^{-1} \mathbf{E}[\mathbf{A}^t]$ and $\mathbf{E}[\mathsf{OPT}(\mathbf{A}^t)]$. In the rest of the proof we will consider $t \geq \tau + 1$, otherwise the Lemma is straightforward since all the terms involved are $\mathcal{O}(\tau)$. Using the dual ($\mathsf{DSPP}(\boldsymbol{\lambda})$) of ($\mathsf{SPP}(\boldsymbol{\lambda})$) it is easy to see that $\mathsf{OPT}(\boldsymbol{\lambda})$ is a concave, piece-wise linear, and Lipschitz function. Therefore,

$$\mathbf{E}[\mathsf{OPT}(\mathbf{A}^t)] \leq \mathsf{OPT}(\mathbf{E}[\mathbf{A}^t])$$

$$= \mathsf{OPT}\left(\sum_{s=1}^{t} \boldsymbol{\lambda}^s\right)$$

$$= \mathsf{OPT}\left(\sum_{s=1}^{t-\tau} \hat{\boldsymbol{\lambda}}^s\right) + \left[\mathsf{OPT}\left(\sum_{s=1}^{t} \boldsymbol{\lambda}^s\right) - \mathsf{OPT}\left(\sum_{s=1}^{t-\tau} \hat{\boldsymbol{\lambda}}^s\right)\right]$$

$$= \mathbf{r}_{\mathcal{M}_+}^\top (\mathbf{M}_{\mathcal{M}_+})^{-1} \sum_{s=1}^{t-\tau} \hat{\boldsymbol{\lambda}}^s + \left[\mathsf{OPT}\left(\sum_{s=1}^{t} \boldsymbol{\lambda}^s\right) - \mathsf{OPT}\left(\sum_{s=1}^{t-\tau} \hat{\boldsymbol{\lambda}}^s\right)\right]$$

$$= \mathbf{r}_{\mathcal{M}_+}^\top (\mathbf{M}_{\mathcal{M}_+})^{-1} \sum_{s=1}^{t} \boldsymbol{\lambda}^s + \mathbf{r}_{\mathcal{M}_+}^\top (\mathbf{M}_{\mathcal{M}_+})^{-1} \left[\sum_{s=1}^{t-\tau} \hat{\boldsymbol{\lambda}}^s - \sum_{s=1}^{t} \boldsymbol{\lambda}^s\right] + \left[\mathsf{OPT}\left(\sum_{s=1}^{t} \boldsymbol{\lambda}^s\right) - \mathsf{OPT}\left(\sum_{s=1}^{t-\tau} \hat{\boldsymbol{\lambda}}^s\right)\right]$$

$$= \mathbf{r}_{\mathcal{M}_+}^\top (\mathbf{M}_{\mathcal{M}_+})^{-1} \mathbf{E}[\mathbf{A}^t] + \mathcal{O}(\tau) \qquad \text{(EC.3)}$$

where in the third step we have used the smoothed GPG assumption on $\{\boldsymbol{\lambda}^t\}$, in the fourth step we have used $\frac{1}{t-\tau} \sum_{s=1}^{t-\tau} \hat{\boldsymbol{\lambda}}^s \in \mathcal{B}_{\epsilon_0, \mathsf{TV}}(\bar{\boldsymbol{\lambda}})$, and in last step we have used $\sum_{s=1}^{t-2\tau} \hat{\boldsymbol{\lambda}}^s \leq \sum_{s=\tau+1}^{t} \boldsymbol{\lambda}^s \leq \sum_{s=1}^{t} \hat{\boldsymbol{\lambda}}^s$, and that OPT is Lipschitz. Combining (EC.2) and (EC.3),

$$\mathbf{E}\left[\sum_{m \in \mathcal{M}_+} r_m \mathscr{X}_m^t\right] = \mathbf{E}[\mathsf{OPT}(\mathbf{A}^t)] - \mathbf{r}_{\mathcal{M}_+}^\top (\mathbf{M}_{\mathcal{M}_+})^{-1} \mathbf{E}\left[\sum_m \mathscr{Q}_m^t\right] - \mathcal{O}(\tau).$$

A.7. Proof of Lemma 8

Our proof combines the Lyanunov drift analysis from Section 2 and amortized analysis to handle non-stationary arrival distributions. We begin with a recap of the proof from Section 2, where this analysis fails, and how the amortized analysis idea helps.

We will use the following notation throughout: for a stochastic process f^t (think of this as a Lyapunov/potential function), let

$$\mathbf{E}_{\boldsymbol{\lambda}, \pi}\left[\Delta^t f\right] := \mathbf{E}_{I^t \sim \boldsymbol{\lambda}, \pi}\left[f^t - f^{t-1} | \mathcal{F}^{t-1}\right]$$

be a shorthand for the expected change in f from time $t-1$ to t if the type of the resource arriving at time t is sampled from distribution $\boldsymbol{\lambda}$ and "placed" according to policy π. By placing according

to π, we mean the choice of m^t in Algorithm 2 is replaced by policy π. The policy π can be randomized, as well as can depend on the entire history until time t of the arrivals and its own randomized actions. We will us SS to denote the policy in Algorithm 2.

A particular randomized policy that will be key in the analysis is the policy $\mathcal{R}(\boldsymbol{\lambda})$ that we mentioned in the proof of Lemma 2.

DEFINITION EC.1 (RANDOMIZED POLICY $\mathcal{R}(\boldsymbol{\lambda})$). Suppose the item arriving at time t is sampled from a distribution $\boldsymbol{\lambda} \in \mathcal{B}_{\epsilon_0,\text{TV}}(\bar{\boldsymbol{\lambda}})$. The randomized policy, $\mathcal{R}(\boldsymbol{\lambda})$, first solves SPP$(\boldsymbol{\lambda})$ with a given optimal basis $\mathcal{M}_+$. Let $\mathbf{x}$ denote the unique optimal solution with basis $\mathcal{M}_+$. Given the type of the arrival $I^t = i$, policy $\mathcal{R}(\boldsymbol{\lambda})$ commits the item to configuration m with probability $\frac{M_{im}x_m}{\sum_{m'} M_{im'}x_{m'}}$. That is, the resource is committed to a configuration with probability distribution consistent with the solution $\mathbf{x}$ to SPP$(\boldsymbol{\lambda})$.

A.7.1. Proof Sketch

Let us first attempt to reproduce the analysis in Section 2 for the general case (non-stationary arrivals, all three types of resources). Consider the Lyapunov function $f^t = \sqrt{\Phi(\boldsymbol{\mathcal{Q}}^t)}$. We first proved that since the queue lengths can change by ± 1 in any one time step, the change in this potential function is bounded by $\sqrt{2}$. This step still holds for the general case, since the virtual queues also change by ± 1.

Next, to prove expected decrease condition, we meed to bound $\mathbf{E}_{\boldsymbol{\lambda}^t,\text{SS}}[\Delta^t f]$ when f^{t-1} is large. This is done by constructing a probabilistic policy, π^t, which is easier to study and upper bounds the drift for policy SS since for any policy π^t, $\mathbf{E}_{\boldsymbol{\lambda}^t,\text{SS}}[\Delta^t f] \leq \mathbf{E}_{\boldsymbol{\lambda}^t,\pi^t}[\Delta^t f]$.

To construct π^t, we first need to prove that if $\Phi(\boldsymbol{\mathcal{Q}}^{t-1})$ for the current state is large, then there must exist some resource type i^*, such that if this item is packed by SS then the change in the sum-of-squares potential is proportional to $-\sqrt{\Phi(\boldsymbol{\mathcal{Q}}^{t-1})}$ (Lemma EC.4). Next, one show that for any $\mathring{\boldsymbol{\lambda}} \in \mathcal{B}_{\epsilon_0,\text{TV}}(\bar{\boldsymbol{\lambda}})$, the randomized policy $\mathcal{R}(\mathring{\boldsymbol{\lambda}})$ gives an expected change in the Sum-of-Squares potential is bounded above by a constant (Lemma EC.3). The policy π^t is then constructed by splitting

$$\boldsymbol{\lambda}^t = \delta \mathbf{e}_{i^*} + (1-\delta) \underbrace{\left(\frac{\boldsymbol{\lambda}^t - \delta \mathbf{e}_{i^*}}{1-\delta} \right)}_{=: \mathring{\boldsymbol{\lambda}}^t},$$

for some $\delta > 0$. We now bound the drift under SS by arguing as follows: with probability $(1-\delta)$ the item distribution is $\mathring{\boldsymbol{\lambda}}^t$ which can be packed using $\mathcal{R}(\mathring{\boldsymbol{\lambda}}^t)$ so that the expected change in the sum-of-squares potential is $\mathcal{O}(1)$, and with probability δ the item distribution is $\mathbf{e}_i$ which can be packed greedily with a large negative change in the potential. We call this hybrid randomized+greedy policy Hyb$(\boldsymbol{\lambda}^t, \boldsymbol{\mathcal{Q}}^{t-1})$, and abbreviate it as Hyb^t for this particular instantiation of parameters.

Since Hyb^t is greedy only for one item type, its drift is easier to study, and can be shown to be proportional to $-\sqrt{\Phi(\boldsymbol{\mathscr{Q}}^{t-1})}$ for $\Phi(\boldsymbol{\mathscr{Q}}^{t-1})$ large enough (Lemma EC.5).

However, to be able to claim that under policy $\mathcal{R}(\mathring{\boldsymbol{\lambda}}^t)$ the expected change in the Sum-of-Squares potential is $\mathcal{O}(1)$, we require the stricter condition: $\boldsymbol{\lambda}^t \in \mathcal{B}_{\epsilon_0-\delta,\mathsf{TV}}(\bar{\boldsymbol{\lambda}})$, so that

$$||\mathring{\boldsymbol{\lambda}}^t - \bar{\boldsymbol{\lambda}}||_{\mathsf{TV}} \leq ||\mathring{\boldsymbol{\lambda}}^t - \boldsymbol{\lambda}^t||_{\mathsf{TV}} + ||\boldsymbol{\lambda}^t - \bar{\boldsymbol{\lambda}}||_{\mathsf{TV}} \leq \epsilon_0.$$

The obstacle in using the above proof approach is that $\boldsymbol{\lambda}^t$ need not be in $\mathcal{B}_{\epsilon_0-\delta,\mathsf{TV}}(\bar{\boldsymbol{\lambda}})$, but the smoothed distributions $\hat{\boldsymbol{\lambda}}^t$ are. Therefore, while the following is true:

$$\mathbf{E}_{\boldsymbol{\lambda}^t,\mathsf{SS}}[\Delta^t f] \leq \mathbf{E}_{\boldsymbol{\lambda}^t,\pi^t}[\Delta^t f],$$

the right hand side of the above need not give a negative drift for a t where $\Phi(\boldsymbol{\mathscr{Q}}^{t-1})$ is large. Instead, we redefine $\mathsf{Hyb}^t := \mathsf{Hyb}(\hat{\boldsymbol{\lambda}}^t, \boldsymbol{\mathscr{Q}}^{t-1})$, and rewrite the right hand side as

$$\mathbf{E}_{\boldsymbol{\lambda}^t,\mathsf{SS}}[\Delta^t f] \leq \mathbf{E}_{\hat{\boldsymbol{\lambda}}^t,\mathsf{Hyb}^t}[\Delta^t f] + \left(\mathbf{E}_{\boldsymbol{\lambda}^t,\pi^t}[\Delta^t f] - \mathbf{E}_{\hat{\boldsymbol{\lambda}}^t,\mathsf{Hyb}^t}[\Delta^t f]\right). \qquad (\text{EC.4})$$

The first term above has the desired properties of being bounded, and being a large negative value when $\Phi(\boldsymbol{\mathscr{Q}}^{t-1})$ is large. For the second term, we will use a trick from amortized analysis of online algorithms. We add another term Ξ^t to the potential function so that $f^t = \Psi^t = \sqrt{\Phi(\boldsymbol{\mathscr{Q}}^t) + \Xi^t}$. The term Ξ^t acts like a non-negative "bank balance," so that whenever the second term of (EC.4) is positive leading to an increase in Φ^t, then Ξ^t decreases by enough (i.e., we can draw on our bank balance to pay for this extra cost) so that,

$$\Delta^t \Xi \approx \mathbf{E}_{\hat{\boldsymbol{\lambda}}^t,\mathsf{Hyb}^t}[\Delta^t \Phi] - \mathbf{E}_{\boldsymbol{\lambda}^t,\pi^t}[\Delta^t \Phi]$$

and consequently,

$$\mathbf{E}_{\boldsymbol{\lambda}^t,\mathsf{SS}}[\Delta^t \Phi + \Delta^t \Xi] \lessapprox \mathbf{E}_{\hat{\boldsymbol{\lambda}}^t,\mathsf{Hyb}^t}[\Delta^t \Phi].$$

A crucial component of this is a careful construction of the policy π^t (which is only for the purposes of analysis), where we use the fact that by the smoothed $\mathsf{GPG}_{\delta,\tau}$ assumption, $\boldsymbol{\lambda}^t$ satisfy

$$\sum_{s=1}^{t-2\tau} \hat{\boldsymbol{\lambda}}^s \leq \sum_{s=\tau+1}^{t} \boldsymbol{\lambda}^s \leq \sum_{s=1}^{t} \hat{\boldsymbol{\lambda}}^s.$$

Therefore, we can *couple* the probability mass of arrival at time t under $\boldsymbol{\lambda}^t$ to some probability mass of arrival during interval $[t-2\tau,t]$ under $\hat{\boldsymbol{\lambda}}^s$. To be slightly more precise, the randomized policy π^t follows some randomized action taken by policy Hyb^s during the interval $s \in [t-2\tau,t]$. This ensures that if $\mathbf{E}_{\boldsymbol{\lambda}^t,\pi^t}[\Delta^t \Phi]$ is large, then it was already paid for by $\mathbf{E}_{\hat{\boldsymbol{\lambda}}^s,\mathsf{Hyb}^s}[\Delta^s \Phi]$ at an earlier time, which is part of our bank balance Ξ^t. One final wrinkle is that, this randomized action is taken by Hyb^s and π^t at different times, and at different states of the virtual queues. However, since $|s-t| \leq 2\tau$, the difference in the change in Φ due to this can be bounded by 16τ (Lemma EC.6). We now proceed with the formal proof.

A.7.2. Preliminary lemmas

We begin with some preliminary lemmas which are proved at the end of this Section.

The first lemma gives the expression in the change of the sum-of-squares potential for a configuration when an item is added.

LEMMA EC.1 (**Elementary Cost Functions**). *Consider a configuration m with virtual state $\mathscr{Q}_m$. Without loss of generality, let $\mathcal{I}(m) = [\ell]$, and $\sigma_m = (1, 2, \dots, \ell)$. Define $\delta_i = \mathscr{Q}_{im} - \mathscr{Q}_{(i-1)m}$ for $i \in \{2, \dots, \ell\}$, and $\delta_1 = \mathscr{Q}_{1m} - \mathscr{Q}_{\ell m}$, with the convention $\delta_{\ell+1} := \delta_1$. Let one unit of resource type i be committed to m resulting in new virtual state $\mathscr{Q}'_m$ (potentially after a virtual match). Then,*

$$\mathsf{cost}(\mathscr{Q}, i, m) := \Phi_m(\mathscr{Q}'_m) - \Phi_m(\mathscr{Q}_m) = 2(\delta_i - \delta_{i+1}) + 2.$$

The elementary cost is bounded as $|\mathsf{cost}(\mathscr{Q}, i, m)| \le \sqrt{8\Phi(\mathscr{Q})} + 2$.

The next lemma gives an almost sure bound on the change of the potential function Ψ^t. This turns out to be a bit weak for our purposes, we prove a strengthened version during the proof of Lemma 8.

LEMMA EC.2 (**Bounded Increment Lemma**). *Let $\mathscr{Q}$ be an arbitrary state of virtual queues, into which an item is packed to yield new virtual state $\mathscr{Q}'$. Then for $\Xi, \Xi' \ge 0$,*

$$\left| \sqrt{\Phi(\mathscr{Q}') + \Xi'} - \sqrt{\Phi(\mathscr{Q}) + \Xi} \right| \le \sqrt{2} + |\Xi' - \Xi|.$$

The next three lemmas prove that the expected drift of the Sum-of-Square function Φ is negative if the potential function is large. Since these lemmas only deal with Φ, and not the combined Lyapunov function Ψ^t, the proofs of these Lemmas largely follow the tools developed by Csirik et al. (2006). It is also these three lemmas which suffice for the analysis in Section 2 for the simpler setting of *i.i.d.* arrivals and online-queueable resources.

LEMMA EC.3. *Consider a state $\mathscr{Q}$ encountered during the execution of Algorithm 2, into which an item sampled from a distribution $\boldsymbol{\lambda} \in \mathcal{B}_{\epsilon_0, TV}(\bar{\boldsymbol{\lambda}})$ is packed according to $\mathcal{R}(\boldsymbol{\lambda})$ to obtain a state $\mathscr{Q}'$. Then the expected change in Sum-of-squares potential is:*

$$\mathbf{E}_{\boldsymbol{\lambda}, \mathcal{R}(\boldsymbol{\lambda})} [\Phi(\mathscr{Q}') - \Phi(\mathscr{Q})] = 2.$$

LEMMA EC.4. *Consider a state $\mathscr{Q}$ encountered during the execution of SS such that $\Phi(\mathscr{Q}) \ge B^2$ where $B := 2n^2$. Then there exists a resource type i^*, such that if an item of type i^* is packed by SS, the resulting virtual queue state satisfies*

$$\Phi(\mathscr{Q}') - \Phi(\mathscr{Q}) \le -2\frac{\sqrt{\Phi(\mathscr{Q})}}{B}.$$

As a consequence of Lemma EC.3 and Lemma EC.4, the next lemma shows the existence of a Hybrid randomized+greedy policy with large negative drift, and which by definition also bounds the drift of SS.

LEMMA EC.5 (**Drift Lemma**). *Consider a state $\mathcal{Q}$ such that $\Phi(\mathcal{Q}) \geq \frac{4B^2}{\delta^2}$ where $B := 2n^2$, into which an item of type sampled from $\boldsymbol{\lambda}$, $\boldsymbol{\lambda} \in \mathcal{B}_{\epsilon_0 - \delta, TV}(\bar{\boldsymbol{\lambda}})$ has to be packed. Then, there exists a policy $\mathsf{Hyb}(\boldsymbol{\lambda}, \mathcal{Q})$, such that the resulting virtual queue state satisfies*

$$\mathbf{E}_{\boldsymbol{\lambda}, SS}\left[\Phi(\mathcal{Q}') - \Phi(\mathcal{Q})\right] \leq \mathbf{E}_{\boldsymbol{\lambda}, Hyb(\boldsymbol{\lambda}, \mathcal{Q})}\left[\Phi(\mathcal{Q}') - \Phi(\mathcal{Q})\right] \leq -\frac{\delta\sqrt{\Phi(\mathcal{Q})}}{B}.$$

LEMMA EC.6 (**Lipschitz Cost Lemma**). *Let $\mathcal{Q}^s$ and $\mathcal{Q}^t$ be two states of the virtual queues at times s, t respectively under some matching algorithm. Fix any configuration m, and resource type $i \in \mathcal{I}(m)$. Let $\hat{\mathcal{Q}}^s$ and $\hat{\mathcal{Q}}^t$ be the states of the virtual queues after committing i to configuration m in states $\mathcal{Q}^s$ and $\mathcal{Q}^t$, respectively. Then*

$$\left| \left(\Phi_m(\hat{\mathcal{Q}}^t_m) - \Phi_m(\mathcal{Q}^t_m)\right) - \left(\Phi_m(\hat{\mathcal{Q}}^s_m) - \Phi_m(\mathcal{Q}^s_m)\right) \right| \leq 8|s - t|.$$

A.7.3. Formal Proof

As we mentioned in the proof sketch, we bound the (amortized) drift in $\Phi + \Xi$ at time t by the drift under the hybrid policy $\mathsf{Hyb}(\hat{\boldsymbol{\lambda}}^t, \mathcal{Q}^{t-1})$, and use a cleverly constructed sequence of policies $\{\pi^t\}$ *coupled* to the sequence $\mathsf{Hyb}(\hat{\boldsymbol{\lambda}}^t, \mathcal{Q}^{t-1})$ to upper bound the drift of SS. We first explain the construction of this sequence $\{\pi^t\}$.

Construction of π^t: We begin by defining a sequence of functions for each $i \in \mathcal{I}$ and $m \in \mathcal{M}_+$ based on the actions of $\mathsf{Hyb}^t := \mathsf{Hyb}(\hat{\boldsymbol{\lambda}}^t, \mathcal{Q}^{t-1})$. Note that Hyb^t is $\mathcal{F}^{t-1}$-measurable. Let the probability that Hyb^t commits resource type i to configuration m be $\hat{p}^t_{im}$. Based on these, we define the piecewise constant functions $\hat{q}_{im} : [0, T] \to [0, 1]$ as follows:

$$\hat{q}_{im}(s) = \hat{\lambda}_i^{\lceil s \rceil} \cdot \hat{p}_{im}^{\lceil s \rceil}, \qquad \forall \, s \in [0, T]. \tag{EC.5}$$

The following helper functions will be useful for defining π^t. We use $\hat{\Lambda}_i(t) : [0, T] \to [0, T]$ to denote the cumulative (fractional) expected number of arrivals of type i by time t under the sequence of distributions $\{\hat{\boldsymbol{\lambda}}^t\}$ (here $[0, T]$ is the continuous interval). By $\Lambda_i(t) : [\tau, T] \to [0, T]$ we denote the cumulative expected number of arrivals of type i in the interval $\{\tau + 1, \ldots, t\}$ according to the sequence $\{\boldsymbol{\lambda}^t\}$:

$$\hat{\Lambda}_i(t) = \sum_{s=1}^{\lfloor t \rfloor} \hat{\lambda}_i^s + (t - \lfloor t \rfloor)\hat{\lambda}_i^{\lceil t \rceil}, \qquad \forall t \in [0, T], \tag{EC.6}$$

$$\Lambda_i(t) = \sum_{s=\tau+1}^{t} \lambda_i^s, \qquad \forall t \in \{\tau + 1, \ldots, T\}. \tag{EC.7}$$

By the smoothed $\mathsf{GPG}_{\delta,\tau}$ condition we have,

$$t - 2\tau \le \hat{\Lambda}_i^{-1}(t) \le t. \tag{EC.8}$$

The policy π^t is only defined for $t \in \{\tau + 1, \ldots, T\}$ (the potential is bounded until time τ, and our drift analysis only begins at time $\tau + 1$). Recall that the policy π^t sees a resource with type sampled from $\boldsymbol{\lambda}^t$. The probability that policy π^t commits an item of type i to configuration m is given by

$$p_{im}^t = \frac{1}{\lambda_i^t} \int_{s=\hat{\Lambda}_i^{-1}(\Lambda_i(t-1))}^{s=\hat{\Lambda}_i^{-1}(\Lambda_i(t))} \hat{q}_{im}(s) ds. \tag{EC.9}$$

Intuitively, policy π^t looks at the cumulative probability with which Hyb sends resource i to configuration m during the (real-valued) interval $\left[\hat{\Lambda}_i^{-1}(\Lambda_i(t)), \hat{\Lambda}_i^{-1}(\Lambda_i(t-1))\right]$. In this sense, the randomized action of the π^t for resource type i is coupled with the randomized action of Hyb for resource type i during this interval. The intervals for different resource types need not be (and in general will not be) the same.

Relating cost of π^t and Hyb: By definition, the expected change of the potential under policy π^t at time t is

$$\mathbf{E}_{\boldsymbol{\lambda}^t, \pi^t}[\Delta^t \Phi] = \sum_{i,m} p_{im}^t \cdot \mathsf{cost}(\boldsymbol{\mathscr{Q}}^{t-1}, i, m).$$

We define the cost incurred by Hyb^s for the actions contributing to π^t as:

$$\mathsf{cost}_{\mathsf{Hyb},t} = \sum_i \int_{s=\hat{\Lambda}_i^{-1}(\Lambda_i(t-1))}^{s=\hat{\Lambda}_i^{-1}(\Lambda_i(t))} \sum_{m \in \mathcal{M}_+} \mathsf{cost}(\boldsymbol{\mathscr{Q}}^{s-1}, i, m) \hat{q}_{im}(s) ds. \tag{EC.10}$$

In words, for every i, if the π^t policy incorporates the action taken by Hybrid at time $s \le t$, then $\mathsf{cost}_{\mathsf{Hyb},t}$ incorporates the cost paid by Hyb^s due to item i (in proportion to amount of contribution of time s to q_{im}^t). The key idea is that due to this coupling of π^t and Hyb, (EC.8) and Lemma EC.6 yield the key inequality:

$$\mathbf{E}_{\boldsymbol{\lambda}^t, \pi^t}[\Delta^t \Phi] \le \mathsf{cost}_{\mathsf{Hyb},t} + 16\tau. \tag{EC.11}$$

Amortized Lyapunov Drift analysis: We first define the compensator Ξ^t, and the Lyapunov function Ψ^t for Lemma 8:

$$\Xi^t := \sum_{s=1}^t \mathbf{E}_{\hat{\boldsymbol{\lambda}}^s, \mathsf{Hyb}^s}[\Delta^s \Phi] - \sum_{s=\tau+1}^t \mathsf{cost}_{\mathsf{Hyb},s} + \tau \max_{t-2\tau \le s \le t} \left(\sqrt{8\Phi(\boldsymbol{\mathscr{Q}}^s)} + 2 \right), \tag{EC.12}$$

$$\Psi^t = \sqrt{\Phi(\boldsymbol{\mathscr{Q}}^t) + \Xi^t}. \tag{EC.13}$$

Our goal is to show that with the above definition of Ξ^t, *(i)* Ξ^t is non-negative for all t (so that Ψ^t bounds Φ^t), *(ii)* Ξ^t is not too large (so that the bound is not loose), and *(iii)* that the bounded increment and expected decrease conditions from Lemma 8 hold for Ψ^t.

To show non-negativity, focus on the first part of Ξ^t:

$$\Xi_1^t := \sum_{s=1}^{t} \mathbf{E}_{\hat{\boldsymbol{\lambda}}^s, \mathsf{Hyb}^s}\left[\Delta^s \Phi\right] - \sum_{s=\tau+1}^{t} \mathsf{cost}_{\mathsf{Hyb}, s}.$$

Both the first and second term in the summation involve the elementary cost functions for the hybrid policy; the first sum is the cost in the interval $\{1, \ldots, t\}$, and the second term involves the coupling used for the policies $\pi^{\tau+1}, \ldots, \pi^t$, and therefore omits for each i, some time interval in $[0, t]$. Crucially, by (EC.8), the support of the missing mass is entirely on $\{t - 2\tau, \ldots, t\}$, and the total missing mass is bounded by τ. By Lemma EC.1, the absolute value of the elementary cost for any action at time s is bounded by $\sqrt{8\Phi(\boldsymbol{\mathcal{Q}}^s)} + 2$, and hence $\max_{t-2\tau \le s \le t} \sqrt{8\Phi(\boldsymbol{\mathcal{Q}}^s)} + 2$ uniformly upper bounds the norm of missing elementary cost function mass in the summation.

To show that the Ξ^t is small, the above argument also immediately implies:

$$\Xi^t \le 2\tau \max_{t-2\tau \le s \le t} \left(\sqrt{8\Phi(\boldsymbol{\mathcal{Q}}^s)} + 2\right), \tag{EC.14}$$

and applying Lemma EC.7 (proved at the end of this subsection), we can further bound the above in terms of $\Phi(\boldsymbol{\mathcal{Q}}^t)$ as

$$\Xi^t \le 2\tau \left(\sqrt{8\Phi(\boldsymbol{\mathcal{Q}}^t)} + 8\tau + 2\ln(1 + 4\tau)\right). \tag{EC.15}$$

We now prove a drift bound for $\Phi(\boldsymbol{\mathcal{Q}}^t) + \Xi^t$. Since SS minimizes the drift in Φ, the change in Φ under SS can be upper bounded by the drift under π^t:

$$\mathbf{E}_{\boldsymbol{\lambda}^t, \mathsf{SS}}\left[\Delta^t \Phi + \Delta^t \Xi\right] \le \mathbf{E}_{\boldsymbol{\lambda}^t, \pi^t}\left[\Delta^t \Phi\right] + \mathbf{E}_{\boldsymbol{\lambda}^t, \mathsf{SS}}\left[\Delta^t \Xi\right]$$
$$\overset{(\text{EC.11})}{\le} \mathsf{cost}_{\mathsf{Hyb}, t} + 16\tau + \mathbf{E}_{\boldsymbol{\lambda}^t, \mathsf{SS}}\left[\Delta^t \Xi\right],$$

adding and subtracting $\mathbf{E}_{\hat{\boldsymbol{\lambda}}^t, \mathsf{Hyb}^t}\left[\Delta^t \Phi\right]$:

$$= \mathbf{E}_{\hat{\boldsymbol{\lambda}}^t, \mathsf{Hyb}^t}\left[\Delta^t \Phi\right] + 16\tau + \mathsf{cost}_{\mathsf{Hyb}, t} - \mathbf{E}_{\hat{\boldsymbol{\lambda}}^t, \mathsf{Hyb}^t}\left[\Delta^t \Phi\right] + \mathbf{E}_{\boldsymbol{\lambda}^t, \mathsf{SS}}\left[\Delta^t \Xi\right],$$

using (EC.12) to expand $\mathbf{E}_{\boldsymbol{\lambda}^t, \mathsf{SS}}\left[\Delta^t \Xi\right]$:

$$= \mathbf{E}_{\hat{\boldsymbol{\lambda}}^t, \mathsf{Hyb}^t}\left[\Delta^t \Phi\right] + 16\tau + \mathsf{cost}_{\mathsf{Hyb}, t} - \mathbf{E}_{\hat{\boldsymbol{\lambda}}^t, \mathsf{Hyb}^t}\left[\Delta^t \Phi\right]$$
$$+ \mathbf{E}_{\hat{\boldsymbol{\lambda}}^t, \mathsf{Hyb}^t}\left[\Delta^t \Phi\right] - \mathsf{cost}_{\mathsf{Hyb}, t} + \tau \left(\max_{t-2\tau \le s \le t} \sqrt{8\Phi(\boldsymbol{\mathcal{Q}}^s)} - \max_{t-2\tau-1 \le s \le t-1} \sqrt{8\Phi(\boldsymbol{\mathcal{Q}}^s)}\right)$$
$$= \mathbf{E}_{\hat{\boldsymbol{\lambda}}^t, \mathsf{Hyb}^t}\left[\Delta^t \Phi\right] + 16\tau + \tau \left(\max_{t-2\tau \le s \le t} \sqrt{8\Phi(\boldsymbol{\mathcal{Q}}^s)} - \max_{t-2\tau-1 \le s \le t-1} \sqrt{8\Phi(\boldsymbol{\mathcal{Q}}^s)}\right),$$

applying Lemma EC.7 to bound the term in parenthesis above,

$$\leq \mathbf{E}_{\hat{\boldsymbol{\lambda}}^t,\mathsf{Hyb}^t}\left[\Delta^t\Phi\right] + 16\tau + \tau\left(8\tau + 4 + 2\log(3+4\tau)\right)$$

$$\leq \mathbf{E}_{\hat{\boldsymbol{\lambda}}^t,\mathsf{Hyb}^t}\left[\Delta^t\Phi\right] + 8\tau^2 + 2\tau\log(3+4\tau) + 20\tau. \tag{EC.16}$$

Bounded increment for Ψ^t: Now we turn to proving the bounded increment property for Ψ^t, which improves upon Lemma EC.2. Lemma EC.2 has $\Xi' - \Xi$ on the right hand side, which is in fact not bounded since it involves $\mathbf{E}_{\hat{\boldsymbol{\lambda}}^t,\mathsf{Hyb}^t}\left[\Delta^t\Phi\right] - \mathsf{cost}_{\mathsf{Hyb},t}$ which is bounded by $2\tau\sqrt{8\Phi(\boldsymbol{\mathcal{Q}})^t}$.

First note that, we can upper bound

$$\Xi^t - \Xi^{t-1} = \mathbf{E}_{\hat{\boldsymbol{\lambda}}^t,\mathsf{Hyb}^t}\left[\Delta^t\Phi\right] - \mathsf{cost}_{\mathsf{Hyb},t} + \tau\left(\max_{t-2\tau\leq s\leq t}\sqrt{8\Phi(\boldsymbol{\mathcal{Q}}^s)} - \max_{t-2\tau-1\leq s\leq t-1}\sqrt{8\Phi(\boldsymbol{\mathcal{Q}}^s)}\right)$$

$$\leq 2\tau\sqrt{8\Phi(\boldsymbol{\mathcal{Q}})^t} + 8\tau^2 + 2\tau\log(3+4\tau) + 4\tau$$

$$\leq 2\tau\sqrt{8\Phi(\boldsymbol{\mathcal{Q}})^{t-1}} + 8\tau^2 + 2\tau\log(3+4\tau) + 4\tau + 8.$$

Now we proceed through the steps of proof of Lemma EC.2: First we assume $\Phi(\boldsymbol{\mathcal{Q}}^t) \geq \Phi(\boldsymbol{\mathcal{Q}}^{t-1})$. We have, for some $x, y \geq 0$,

$$\Phi(\boldsymbol{\mathcal{Q}}^t) \leq \Phi(\boldsymbol{\mathcal{Q}}^{t-1}) + (x+1)^2 + (y+1)^2 - x^2 - y^2.$$

Using this,

$$\sqrt{\Phi(\boldsymbol{\mathcal{Q}}^t) + \Xi^t} - \sqrt{\Phi(\boldsymbol{\mathcal{Q}}^{t-1}) + \Xi^{t-1}} \leq \sqrt{\Phi(\boldsymbol{\mathcal{Q}}^t) + (x+1)^2 + (y+1)^2 - x^2 - y^2 + \Xi^t} - \sqrt{\Phi(\boldsymbol{\mathcal{Q}}^{t-1}) + \Xi^{t-1}}$$

$$= \frac{2x + 2y + 2 + \Xi^t - \Xi^{t-1}}{\sqrt{\Phi(\boldsymbol{\mathcal{Q}}) + (x+1)^2 + (y+1)^2 - x^2 - y^2 + \Xi^t} + \sqrt{\Phi(\boldsymbol{\mathcal{Q}}^{t-1}) + \Xi^{t-1}}}$$

$$\leq \frac{2x + 2y + 2}{\sqrt{(x+1)^2 + (y+1)^2} + \sqrt{x^2 + y^2}} + \frac{\Xi^t - \Xi^{t-1}}{\sqrt{2} + \sqrt{\Phi(\boldsymbol{\mathcal{Q}}^{t-1})}}$$

$$\leq \sqrt{2} + \frac{2\tau\sqrt{8\Phi(\boldsymbol{\mathcal{Q}})^{t-1}} + 8\tau^2 + 2\tau\log(3+4\tau) + 4\tau + 8}{\sqrt{2} + \sqrt{\Phi(\boldsymbol{\mathcal{Q}}^{t-1})}}$$

$$\leq \sqrt{2} + 2\tau\sqrt{8} + \frac{8}{\sqrt{2}}\tau^2 + 2\tau\log(3+4\tau) + \frac{4}{\sqrt{2}}\tau + \frac{8}{\sqrt{2}}$$

$$\leq 6\tau^2 + 9\tau + 2\tau\log(3+4\tau) + 8$$

$$=: K_1(\tau).$$

Expected Decrease for Ψ^t: We use the following three facts established previously,

(i) From (EC.16),

$$\mathbf{E}_{\boldsymbol{\lambda}^t,\mathsf{SS}}\left[\Delta^t\Phi + \Delta^t\Xi\right] \leq \mathbf{E}_{\hat{\boldsymbol{\lambda}}^t,\mathsf{Hyb}^t}\left[\Delta^t\Phi\right] + 8\tau^2 + 2\tau\log(3+4\tau) + 20\tau.$$

(ii) From Lemma EC.5, if $\Phi(\mathcal{Q}^{t-1}) \geq \frac{4B^2}{\delta^2}$, where $B = 2n^2$, then,

$$\mathbf{E}_{\hat{\boldsymbol{\lambda}}^t,\mathsf{Hyb}^t}\left[\Delta^t\Phi\right] \leq -\frac{\delta\sqrt{\Phi(\mathcal{Q}^{t-1})}}{B}.$$

From (i) and (ii), if $\Phi(\mathcal{Q}^{t-1}) \geq c_1(\tau,\delta)$ where

$$c_1(\tau,\delta) := \frac{4B^2(8\tau^2 + 2\tau\log(3+4\tau)+20\tau+1)^2}{\delta^2}, \qquad (\text{EC.17})$$

then

$$\mathbf{E}_{\boldsymbol{\lambda}^t,\mathsf{SS}}\left[\Delta^t\Phi + \Delta^t\Xi\right] \leq -\frac{\delta\sqrt{\Phi(\mathcal{Q}^{t-1})}}{2B}.$$

(iii) From (EC.15),

$$\Xi^{t-1} \leq 2\tau\left(\sqrt{8\Phi(\mathcal{Q}^{t-1})}+8\tau+2\ln(1+4\tau)\right),$$

and therefore if $\Phi(\mathcal{Q}^{t-1}) \geq c_2(\tau)$ where

$$c_2(\tau) := 32\tau^2 + 4\tau(8\tau+2\ln(1+4\tau)), \qquad (\text{EC.18})$$

is large enough so that

$$c_2(\tau) \geq 2\tau\left(\sqrt{8c_2(\tau)}+8\tau+2\ln(1+4\tau)\right),$$

then $(\Psi^{t-1})^2 = \Phi(\mathcal{Q}^{t-1}) + \Xi^{t-1} \leq 2\Phi(\mathcal{Q}^{t-1})$.

Using the inequality (Lemma 3.6 of Csirik et al. (2006)), for $x,y \geq 0$,

$$x - y \leq \frac{x^2 - y^2}{2y},$$

substituting $x = \Psi^t, y = \Psi^{t-1}$, when $\Phi(\mathcal{Q}^{t-1}) \geq \max\{c_1(\tau,\delta), c_2(\tau)\}$,

$$\mathbf{E}[\Delta^t\Psi] \leq \frac{\mathbf{E}[\Delta^t\Phi + \Delta^t\Xi]}{2\Psi^{t-1}} \leq \frac{-\delta\sqrt{\Phi(\mathcal{Q}^{t-1})}/2B}{2\sqrt{2\Phi(\mathcal{Q}^{t-1})}} \leq -\frac{\delta}{4\sqrt{2}B}.$$

∎

LEMMA EC.7. *Let $a_0 \geq 0$, and consider the series $a_t = a_{t-1} + \sqrt{8a_{t-1}} + 2$. Then,*

$$\sqrt{8a_t} - \sqrt{8a_0} \leq 4t + 2\log(1+2t).$$

Proof: We can upper bound a_t by $x(t)$ which solves the following differential equation:

$$\frac{dx}{dt} = \sqrt{8x} + 2$$

with initial condition $x(0) = a_0$. The solution of this differential equation is given by

$$\sqrt{8x(t)} - \sqrt{8x(0)} = 4t + 2\log\frac{\sqrt{8x(t)}+2}{\sqrt{8x(0)}+2}.$$

To upper bound the ratio inside the log, we consider the solution of the differential equation with initial condition $x(0) = 0$, which gives an upper bound on $\sqrt{8x(t)}$ of $4t$. Therefore, we get the bound $\frac{\sqrt{8x(t)}+2}{\sqrt{8x(0)}+2} \leq \frac{4 \cdot t + 2}{4 \cdot 0 + 2} = 1 + 2t$ for all $x(0) \geq 0$. Substituting this upper bound into the solution of the differential equation

$$\sqrt{8x(t)} - \sqrt{8x(0)} \leq 4t + 2\log(1 + 2t).$$

∎

A.7.4. Proofs of supporting lemmas

Proof of Lemma EC.1: The proof is a straightforward computation from the definition (6). If an resource of type i arrives and no virtual match is complete then $\mathscr{Q}'_{im} = \mathscr{Q}_{im} + 1$, and the rest of the virtual queue lengths are unchanged. Therefore, δ_i increases by 1, and δ_{i+1} decreases by 1. Since

$$\Phi_m(\boldsymbol{\mathscr{Q}}_m) = \sum_{k=1}^{\ell} \delta_k^2,$$

the overall change in Φ_m is $2(\delta_i - \delta_{i+1}) + 2$.

Now consider the case that type i resource arrives and a virtual match happens. In this case $\mathscr{Q}'_{im} = \mathscr{Q}_{im}$, and the rest of the virtual queue lengths decrease by 1. Net effect is still that δ_i increases by 1, δ_{i+1} decreases by 1, and the rest of the δ_k are unchanged.

Since $|\delta_i|, |\delta_{i+1}| \leq \Phi(\boldsymbol{\mathscr{Q}}) \leq \sqrt{|\delta_i|^2 + \delta_{i+1}^2}$, it follows that $|\delta_i| + |\delta_{i+1}| \leq \sqrt{2\Phi(\boldsymbol{\mathscr{Q}})}$, from which the bound on $\mathsf{cost}(\boldsymbol{\mathscr{Q}}, i, m)$ follows. ∎

Proof of Lemma EC.2: Let m be the configuration into which type i resource is added. Consider the case in which $\Phi_m(\boldsymbol{\mathscr{Q}}'_m) \geq \Phi_m(\boldsymbol{\mathscr{Q}}_m)$. As described in the proof of Lemma EC.1, in this case δ_i increases and δ_{i+1} decreases. In the worst case, δ_i was positive, and δ_{i+1} was negative, so both changes lead to an increase in Φ_m. Let the original value of $\delta_i = x$ and $\delta_{i+1} = -y$ for $x, y \geq 0$. Then,

$$\Phi_m(\boldsymbol{\mathscr{Q}}'_m) \leq \Phi_m(\boldsymbol{\mathscr{Q}}_m) + (x+1)^2 + (y+1)^2 - x^2 - y^2.$$

Now, following the proof of Lemma 3.5 of Csirik et al. (2006):

$$
\begin{aligned}
\sqrt{\Phi(\boldsymbol{\mathscr{Q}}') + \Xi'} - \sqrt{\Phi(\boldsymbol{\mathscr{Q}}) + \Xi} &\leq \sqrt{\Phi(\boldsymbol{\mathscr{Q}}) + (x+1)^2 + (y+1)^2 - x^2 - y^2 + \Xi'} - \sqrt{\Phi(\boldsymbol{\mathscr{Q}}) + \Xi} \\
&= \frac{2x + 2y + 2 + \Xi' - \Xi}{\sqrt{\Phi(\boldsymbol{\mathscr{Q}}) + (x+1)^2 + (y+1)^2 - x^2 - y^2 + \Xi'} + \sqrt{\Phi(\boldsymbol{\mathscr{Q}}) + \Xi}} \\
&\leq \frac{2x + 2y + 2 + \Xi' - \Xi}{\sqrt{(x+1)^2 + (y+1)^2} + \sqrt{x^2 + y^2}} \\
&= \frac{2x + 2y + 2 + \Xi' - \Xi}{\sqrt{x^2 + y^2 + 2x + 2y + 2} + \sqrt{x^2 + y^2}}
\end{aligned}
$$

and letting $z = x + y$ and minimizing $x^2 + y^2$ to get $x = y = z/2$, we further upper bound the above by,

$$\leq \frac{2z + 2 + \Xi' - \Xi}{\sqrt{\frac{z^2}{2} + 2z + 2} + \sqrt{\frac{z^2}{2}}}$$

$$= \sqrt{2} \frac{2z + 2 + \Xi' - \Xi}{(z + 2) + z}$$

$$\leq \sqrt{2} + |\Xi' - \Xi|.$$

The case $\Phi_m(\mathscr{Q}'_m) \leq \Phi_m(\mathscr{Q}_m)$ is handled similarly. ∎

Proof of Lemma EC.3: It suffices to prove the result for the case where the configuration m to which an item has to be committed is fixed, and the probability that the type of arriving item $I = i$ is $p_i = M_{im}/\sum_{i'} M_{i'm} = 1/\ell_m$. We can then view the distribution $\boldsymbol{\lambda}$ as first sampling a configuration m according to the solution of $\mathsf{SPP}(\boldsymbol{\lambda})$, and then sampling the type of the item i as in the previous sentence. Since by Lemma EC.1, the change in Φ_m is $2(\delta_i - \delta_{i+1}) + 2$ when a resource a type i arrives, the expected change if a resource $i \in \mathcal{I}(m)$ arrives uniformly at random,

$$\mathbf{E}[\Phi_m(\mathscr{Q}'_m) - \Phi_m(\mathscr{Q}_m)] = \frac{1}{\ell_m} \sum_{i=1}^{\ell_m} (2(\delta_i - \delta_{i+1}) + 2) = 2.$$

∎

Proof of Lemma EC.4: Let $\Phi := \sum_m \Phi_m(\mathscr{Q}_m)$. Since $|\mathcal{M}_+| = n$, there must exist a configuration m such that $\Phi_m(\mathscr{Q}_m) \geq \Phi/n$. Using the notation and setup from Lemma EC.1, since $\Phi_m(\mathscr{Q}_m) = \sum_{k=1}^{\ell} \delta_k^2$, there must exist a j such that $|\delta_j| \geq \sqrt{\Phi}/n$, and let $i = \arg\max_{k \leq j-1} \delta_{k+1} - \delta_k$. Since $\sum_k \delta_k = 0$, it must be the case that $\delta_{i+1} - \delta_i \geq \frac{\sqrt{\Phi}}{n^2}$. By Lemma EC.1, if the arriving item is of type $i^* = i$, then the change in Φ_m is

$$\Phi_m(\mathscr{Q}'_m) - \Phi_m(\mathscr{Q}_m) = -2(\delta_{i+1} - \delta_i) + 2 \leq -\frac{2\sqrt{\Phi}}{n^2} + 2 \leq -4\frac{\sqrt{\Phi}}{B} + 2 \leq -2\frac{\sqrt{\Phi}}{B}$$

where the second inequality follows from $B = 2n^2$, and the next from the assumption $\Phi \geq B^2$. ∎

Proof of Lemma EC.5: By Lemma EC.4, there is a resource type i^* which if packed into $\mathscr{Q}$ according to the SS algorithm leads to a change in sum-of-squares potential of at least $-2\frac{\sqrt{\Phi}}{B}$. We decompose $\boldsymbol{\lambda}$ into

$$\boldsymbol{\lambda} = \delta \mathbf{e}_{i^*} + (1 - \delta) \underbrace{\left(\frac{\boldsymbol{\lambda} - \mathbf{e}_{i^*}}{1 - \delta} \right)}_{=: \mathring{\boldsymbol{\lambda}}}.$$

The policy $\mathsf{Hyb}(\boldsymbol{\lambda}, \mathscr{Q})$ is defined to be the mixture of $\mathcal{R}(\mathring{\boldsymbol{\lambda}})$ and SS: If the item type is $i \neq i^*$, Hyb picks the distribution to commit i to based on the randomized policy $\mathcal{R}(\mathring{\boldsymbol{\lambda}})$. If the item type is $i = i^*$, then Hyb follows SS with probability δ/λ_i, and $\mathcal{R}(\mathring{\boldsymbol{\lambda}})$ otherwise.

Then since SS greedily minimizes Φ,

$$\mathbf{E}_{\boldsymbol{\lambda},\mathsf{SS}}\left[\Phi(\mathscr{Q}') - \Phi(\mathscr{Q})\right] \leq (1-\delta)\mathbf{E}_{\mathring{\boldsymbol{\lambda}},\mathcal{R}(\mathring{\boldsymbol{\lambda}})}\left[\Phi(\mathscr{Q}') - \Phi(\mathscr{Q})\right] + \delta\mathbf{E}_{\mathbf{e}_{i^*},\mathsf{SS}}\left[\Phi(\mathscr{Q}') - \Phi(\mathscr{Q})\right].$$

We have $||\mathring{\boldsymbol{\lambda}} - \bar{\boldsymbol{\lambda}}||_{\mathsf{TV}} \leq ||\mathring{\boldsymbol{\lambda}} - \boldsymbol{\lambda}||_{\mathsf{TV}} + ||\boldsymbol{\lambda} - \bar{\boldsymbol{\lambda}}||_{\mathsf{TV}} \leq \epsilon_0$, and therefore $\mathbf{E}_{\mathring{\boldsymbol{\lambda}},\mathcal{R}(\mathring{\boldsymbol{\lambda}})}\left[\Phi(\mathscr{Q}') - \Phi(\mathscr{Q})\right] \leq 2$ by Lemma EC.3. Therefore,

$$\begin{aligned}
\mathbf{E}_{\boldsymbol{\lambda},\mathsf{SS}}\left[\Phi(\mathscr{Q}') - \Phi(\mathscr{Q})\right] &\leq (1-\delta)2 - \delta\frac{2\sqrt{\Phi(\mathscr{Q})}}{B} \\
&\leq 2 - 2\delta\frac{\sqrt{\Phi(\mathscr{Q})}}{B} \\
&\leq -\delta\frac{\sqrt{\Phi(\mathscr{Q})}}{B}
\end{aligned}$$

when $\Phi(\mathscr{Q}) \geq \frac{4B^2}{\delta^2}$. ∎

Proof of Lemma EC.6: Recall the setup and notation of Lemma EC.1. Let $s < t$, and $\tau = t - s$. Since at most τ items have been packed in configuration m during the times steps $[s+1, t]$, each $\mathscr{Q}_{km}$ could have changed by at most τ, and therefore each δ_i can change by at most 2τ, and $2(\delta_i - \delta_{i+1})$ can change by at most 8τ. ∎

A.8. Proof of Lemma 9

First consider the case where for some $t \in [T/2, T]$, $\mathbf{E}[N_1^t] \geq 2c\log T$. We can lower bound the regret as,

$$\mathsf{OPT}(\mathbf{A}^t) - (5X_1^t + X_2^t) \geq \mathbb{I}\{A_2^t + A_3^t \geq A_1^t\}N_1^t,$$

or,

$$\mathbf{E}[OPT(\mathbf{A}^t) - (5X_1^t + X_2^t)] \geq \mathbf{Pr}[A_2^t + A_3^t \geq A_1^t]\mathbf{E}[N_1^t | A_2^t + A_3^t \geq A_1^t].$$

Since $A_2^t + A_3^t - A_1^t$ can be written as the sum of t i.i.d. random variables with take the value $+1$ with probability 0.64 and -1 with probability 0.36, a simple application of Hoeffding bound gives

$$\mathbf{Pr}[A_1^t - A_2^t - A_3^t \geq 0] \leq e^{-\frac{2(0.28t)^2}{4t}} \leq e^{-T/60}.$$

Also,

$$\begin{aligned}
\mathbf{Pr}[A_2^t + A_3^t \geq A_1^t]\mathbf{E}[N_1^t | A_2^t + A_3^t \geq A_1^t] &= \mathbf{E}[N_1^t] - \mathbf{E}[N_1^t | A_2^t + A_3^t < A_1^t] \cdot \mathbf{Pr}[A_2^t + A_3^t < A_1^t] \\
&\geq 2c\log T - T \cdot \mathbf{Pr}[A_1^t - A_2^t - A_3^t \geq 0] \\
&\geq 2c\log T - Te^{-\frac{T}{60}}.
\end{aligned}$$

For $T \geq T_0$ large enough, we get $2c\log T - Te^{-\frac{T}{60}} \geq c\log T$.

Next, consider the case that for all $t \in [T/2, T]$, $\mathbf{E}[N_1^t] \leq \frac{c}{4}\log T$ (where we will choose the constant c later). We first create $K = \frac{T}{2c\log T} - 1$ disjoint sub-intervals of $[T/2 + 1, T]$ where the kth subinterval

is $[\tau_k + 1, \tau_k + c \log T]$ where $\tau_k = T/2 + (k-1) \cdot c \log T$. Let $\mathcal{E}_k$ denote the event that all the arrivals in kth sub-interval are of resource type 2, and let $\mathcal{G}_k$ denote the event $\{N_1^{\tau_k}\} \leq \frac{c}{2} \log T$. Because N_1^t is adapted to the arrivals until time t, $\mathcal{E}_k$ and $\mathcal{G}_k$ are independent for all k. Further, $\mathcal{E}_k$ are *i.i.d.*. Let $\alpha = \mathbf{Pr}[\mathcal{E}_k]$, and let $\beta_k = \mathbf{Pr}[\mathcal{G}_k]$. Since by our assumption, $\mathbf{E}[N_1^t] \leq \frac{c}{4} \log T$, we have $\mathbf{Pr}\left[N_1^t \geq \frac{c}{2} \log T\right] \leq \frac{1}{2}$, or $\mathbf{Pr}\left[N_1^t \leq \frac{c}{2} \log T\right] \geq 1/2$. In other words,

$$\beta_k \geq \frac{1}{2} \text{ for all } k.$$

Further $\mathbf{Pr}[\mathcal{E}_k] = \frac{1}{T^{-c \log 0.32}}$. Let c be such that $-c \log 0.32 = 0.5$, so that

$$\alpha = \frac{1}{\sqrt{T}}.$$

Our goal is to show that $\mathbf{Pr}[\cup_k \mathcal{E}_k \mathcal{G}_k]$ is at least a constant, since on this event $\frac{c}{2} \log T$ arrivals of type 2 resources are lost due to insufficient queue of type 1 resource, leading to a regret of $\frac{5c}{2} \log T$ at time T.

Let $B = \sum_k \mathcal{E}_k \mathcal{G}_k$ count the number of such events. We will use the second moment method to bound $\mathbf{Pr}[B \geq 1]$ which says that

$$\mathbf{Pr}[B > 0] \geq \frac{\mathbf{E}[B]^2}{\mathbf{E}[B^2]}.$$

In our case, since B is non-negative and integer valued, $\mathbf{Pr}[B > 0] = \mathbf{Pr}[B \geq 1]$.

We have

$$\mathbf{E}[B] = \mathbf{E}\left[\sum_k \mathcal{E}_k \mathcal{G}_k\right] = \sum_k \mathbf{E}[\mathcal{E}_k \mathcal{G}_k] = \sum_k \mathbf{E}[\mathcal{E}_k]\mathbf{E}[\mathcal{G}_k] = \alpha \cdot \sum_k \beta_k.$$

Next,

$$\begin{aligned}
\mathbf{E}\left[B^2\right] &= \sum_k \mathbf{E}[\mathcal{E}_k \mathcal{G}_k] + 2 \sum_{k < \ell} \mathbf{E}[\mathcal{E}_k \mathcal{E}_\ell \mathcal{G}_k \mathcal{G}_\ell] \\
&\leq \sum_k \mathbf{E}[\mathcal{E}_k] + 2 \sum_{k < \ell} \mathbf{E}[\mathcal{E}_k \mathcal{E}_\ell] \\
&= K\alpha + K(K-1)\alpha^2.
\end{aligned}$$

Finally,

$$\mathbf{Pr}[B \geq 1] \geq \frac{\mathbf{E}[B]^2}{\mathbf{E}[B^2]} \geq \frac{K^2\alpha^2/4}{K^2\alpha^2 + K\alpha} \geq 1/8,$$

where in the last step we have used $K\alpha \geq 1$ for our choice of c, and $T \geq T_0$ large enough.