A Competitive Algorithm for Random-Order Stochastic Virtual Circuit Routing

Thắng Nguyễn Kim
IBISC, Univ Evry, University Paris Saclay, Evry, France
kimthang.nguyen@univ-evry.fr

Abstract

We consider the virtual circuit routing problem in the stochastic model with uniformly random arrival requests. In the problem, a graph is given and requests arrive in a uniform random order. Each request is specified by its connectivity demand and the load of a request on an edge is a random variable with known distribution. The objective is to satisfy the connectivity request demands while maintaining the expected congestion (the maximum edge load) of the underlying network as small as possible.

Despite a large literature on congestion minimization in the deterministic model, not much is known in the stochastic model even in the offline setting. In this paper, we present an $O(\log n/\log\log n)$-competitive algorithm when optimal routing is sufficiently congested. This ratio matches to the lower bound $\Omega(\log n/\log\log n)$ (assuming some reasonable complexity assumption) in the offline setting. Additionally, we show that, restricting on the offline setting with deterministic loads, our algorithm yields the tight approximation ratio of $\Theta(\log n/\log\log n)$. The algorithm is essentially greedy (without solving LP/rounding) and the simplicity makes it practically appealing.

2012 ACM Subject Classification Theory of computation → Approximation algorithms analysis

Keywords and phrases Approximation Algorithms, Congestion Minimization

Digital Object Identifier 10.4230/LIPIcs.ISAAC.2019.39

Funding Research supported by the ANR project OATA n° ANR-15-CE40-0015-01.

1 Introduction

Congestion minimization is a fundamental problem for network operations/communication. In the former, there are connectivity requests and serving requests induces loads on network links. The load vector of each request is *deterministically* given. The objective is to satisfy the connectivity demands while maintaining the congestion of the underlying network as small as possible. The problem has been widely studied and several algorithms with performance guarantee have been designed.

In real-world scenarios, given the presence of uncertainty, request loads are rarely deterministic but vary as random variables. Uncertainty may come from different sources due to unexpected events, noise, etc. The uncertainty in the loads represents the main difficulty in designing performant algorithms in such scenarios. In this paper, we take one step closer to real-world situations by considering the congestion minimization in the stochastic model.

Stochastic Virtual Circuit Routing Problem (SVCR). Given a directed graph $G(V,E)$ where $|V| = n$, $|E| = m$ and a set of k requests. A request i (for $1 \leq i \leq k$) is specified by a origin/destination pair (o_i, d_i) and a random variable $X_{i,e}$ whose distribution is known that represents the load of request i on an edge e. Assume that $X_{i,e}$'s are bounded and without loss of generality, $X_{i,e}$'s take values in $[0,1]$. For each request i, one needs to choose a routing path connecting o_i to d_i. The *expected congestion* of a routing (connecting all requests' pairs) is $\mathbb{E}\big[\max_e \sum_{i\in T_e} X_{i,e}\big]$ where T_e is the set of requests whose routing path passes through e. The objective is to minimize the expected congestion.

30th International Symposium on Algorithms and Computation (ISAAC 2019).
Editors: Pinyan Lu and Guochuan Zhang; Article No. 39; pp. 39:1–39:12
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

In this paper, we consider the SVCR problem in the random-order setting. In the latter, requests are released over time in an *uniformly random* order and at the arrival of a request, one needs to make an irrevocable decision to satisfy the request. The random-order setting is similar to the online one; however, in the former the adversary can choose request parameters but has no influence on the request arrival order (which is uniformly random).

The congestion objective belongs to the class of ℓ_p-norm functions on load vectors. Specifically, the former corresponds to the ℓ_∞-norm and it is well-known that the ℓ_∞-norm of a m-vector can be approximated up to a constant factor by the ℓ_p-norm where $p = \log m$. In the SVCR problem, we also consider ℓ_p-norm objective functions on load vectors. Note that when we mention the SVCR problem without stating explicitly the objective, it means that the congestion objective is considered.

Stochastic algorithmic problems are common in real-world situations and have been extensively studied in different domains, including approximation algorithms. There are two classes of algorithms for stochastic problems: *non-adaptive* and *adaptive*. In the former, the decisions have been made up-front and then the realization of the randomness will be revealed. In the latter, the randomness is revealed instantaneously after each decision (so an algorithm can adapt its strategy due to the outcome of random variables observed so far). In virtual circuit routing, non-adaptive solutions are preferable and more suitable than the adaptive ones since the former is usually simpler and easier to implement. In this paper, we are interested in designing non-adaptive solutions for the SVCR problem.

The virtual circuit routing problem has been well understood in the deterministic model. Specifically, in offline setting Raghavan and Thompson [23] gave an $O(\log n/\log\log n)$-approximation algorithm and in online setting Aspnes et al. [3] provided an $O(\log n)$-competitive algorithm. The bounds are optimal up to a constant factor. However, not much in term of approximation is known in the stochastic model. A closely related problem to SVCR, the stochastic load balancing problem, has been studied in the offline setting. In the problem, given a set of jobs and machines, one needs to assign jobs to machines such that the (expected) maximum load of the assignment is minimized. Kleinberg et al. [17] first considered this problem and gave a constant approximation for identical machines, i.e., for each job j, the random loads of a job on all machines are identical. Goel and Indyk [11] provided better approximations when the job loads follow some specific distributions, for example Poisson distributions, Exponential distributions. Very recently, Gupta et al. [12] gave a constant approximation for unrelated machines. They also considered the objective of minimizing the ℓ_p-norm of machine loads and showed an $O(p/\log p)$-approximation algorithm. Their technique is based on a linear program which guarantees a strong lower bound for the stochastic load balancing problem. In their paper, Gupta et al. [12] raised an open question of designing algorithms for the SVCR problem. The main difficulty, which resists to current approaches, is to deal with the correlation of edges loads where different paths may share common edges.

1.1 Our Contribution and Approach

We give a competitive algorithm for the SVCR problem in the random-order setting. Specifically, our algorithm is $O(\log n/\log\log n)$-competitive if the congestion of the optimal solution is at least 1, i.e., informally, optimal routing is sufficiently congested. Note that even in the offline setting with deterministic loads, the problem is known to be hard to approximate within factor $\Omega(\log n/\log\log n)$ unless all problems in NP have randomized algorithms with running time $n^{\text{poly}\log n}$ [2, 8]. The result shows that in terms of approximation, one can guarantee the quality of the algorithmic solutions for the virtual circuit routing problem even with uncertainty in the request loads. Moreover, our algorithm is essentially greedy which makes it practically appealing and is easy to implement.

In order to design algorithms for the SVCR problem, we study the more general objective of minimizing the ℓ_p-norm of edge loads. We consider the primal-dual technique with configuration LPs [25]. This approach provides a clean way to deal with non-linear objective functions and intuitive constructions of dual variables. Our algorithm is a generalized version of Greedy Restart algorithms introduced by Molinaro [22] in the context of machine load balancing (which can be seen as a special case of the SVCR problem where the network consists of two nodes and parallel edges connecting these two nodes). Informally, for every request the algorithm selects a routing path greedily with respect to some function $\psi_{\kappa,p}$ (defined later) which depends on the current load vector. However, when half of the requests have been considered, the algorithm *restarts* the procedure: it still chooses a routing path greedily with respect to the function $\psi_{\kappa,p}$ but now the function $\psi_{\kappa,p}$ depends on the load vector induced *only* by the second half of the requests. Building on the primal-dual technique with configuration LPs [25] and useful probability inequalities together with insightful observations by Molinaro [22], we prove the competitiveness of the algorithm in the online random-order setting.

Besides, we revisit the classic virtual circuit routing problem in offline setting with deterministic loads (where $X_{i,e}$'s are deterministic values w_i for every e). We show that our algorithm achieves the tight approximation ratio of $\Theta(\log n/ \log\log n)$. Remark that our greedy algorithm is simpler than the algorithms by Raghavan and Thompson [23], Srinivasan [24] which are based on LP-rounding techniques or the recent algorithm by Chekuri and Idleman [6] which relies on the notion of multiroute flows [16].

1.2 Further Related Works

In the offline setting, the virtual circuit routing problem is also known under the name of the congestion minimization problem. The latter is a relaxation of the classsic edge-disjoint paths problem: given a graph and a collection of source-sink pairs, can the pairs be connected via edge-disjoint paths. For the variant of the congestion minimization problem where $d_i = 1$ and $w_i \equiv 1$ for every $1 \leq i \leq k$, Raghavan and Thompson gave an $O(\log n/ \log\log n)$-approximation algorithm via their influential randomized rounding technique [23]. This ratio is subsequently proved by Chuzhoy et al. [8] to be tight assuming some complexity hypothesis. Srinivasan [24] considered the multipath congestion minimization problem corresponding to the setting where $d_i \geq 1$ and $w_i \equiv 1$ for every $1 \leq i \leq k$. Srinivasan presented an $O(\log n/ \log\log n)$-approximation algorithm by developing a dependent rounding technique for cardinality constraints [24] . The technique is extended in subsequent works for handling more general constraints [10, 9, 7]. Recently, Chekuri and Idleman [6] gave a simple algorithm for the multipath congestion minimization problem. They showed the $O(\log n/ \log\log n)$ approximation ratio via the notion of multiroute flows which were originally introduced by Kishimoto and Takeuchi [16]. That enables a simple solution without using dependent rounding and also allows them to improve the results in some particular cases.

The congestion minimization problem has been also studied in online setting where requests arrive online. Aspnes et al. [3] gave an $O(\log n)$-competitive algorithm and proved that this bound is optimal up to a constant factor. For the more general objective of ℓ_p-norm, Awerbuch et al. [4] considered the load balancing problem and proved that greedy algorithm achieved the bound of $O(p)$, also optimal up to a constant factor. Caragiannis [5] strengthened and significantly simplified the analysis of the greedy algorithm and showed the optimal bound of $\frac{1}{2^{1/p}-1}$.

Stochastic combinatorial optimization problems such as shortest paths, minimum spanning trees, knapsack, bin-packing etc have been considered by Li and Deshpande [18] and Li and Yuan [19] and Kleinberg et al. [17]. In these problems, parameters (length, weights, etc) are given as random variables with known distributions and the objective is to optimize the expected value of some cost/utility functions. In this paper, we are interested in the class of non-adaptive algorithms. Several works [21, 20, 15, 14] have considered adaptive algorithms where the decisions of algorithms depend on the current state of the solutions.

2 Preliminaries

In this section, we give some definitions and technical lemmas which are useful in our analysis. This part is drawn significantly from Molinaro [22]. Recall that in the random-order model, the cost of a routing is the expected ℓ_p-norm of the load vector where the expectation is taken over the random order and the random vectors $X_{i,e}$'s.

Given $p > 1$, its Hölder conjugate q is the number that satisfies $\frac{1}{p} + \frac{1}{q} = 1$. The dual of the ℓ_p-norm is the ℓ_q-norm. Let ℓ_q^+ be the set of non-negative vectors in $\mathbb{R}^m$ with ℓ_q-norm at most 1. Given a constant κ and p, define function $\psi_{\kappa,p} : \mathbb{R}^m \to \mathbb{R}$ as $\psi_{\kappa,p}(\boldsymbol{u}) = \frac{p}{\kappa}\left(\left\|\boldsymbol{1} + \frac{\kappa}{p}\boldsymbol{u}\right\|_p - 1\right)$. The function $\psi_{\kappa,p}$ can be equivalently written as

$$\psi_{\kappa,p}(\boldsymbol{u}) = f_{\kappa,p}^{-1}\left(\sum_{h=1}^{m} f_{\kappa,p}(u_h)\right) \qquad \text{where } f_{\kappa,p}(u_h) = \left(1 + \frac{\kappa}{p}u_h\right)^p$$

Recall that $\|\boldsymbol{u}\|_p = g^{-1}\left(\sum_{h=1}^m g(u_h)\right)$ where $g(u_h) = (u_h)^p$. Informally, $\psi_{\kappa,p}(\cdot)$ is a smooth approximation of $\|\cdot\|_p$ as shown later in Lemma 1. In the paper, we are interested in the congestion which is the ℓ_∞-norm of the load vectors. It is well-known that the ℓ_∞-norm of any vector can be approximated by ℓ_p-norm of that vector where m is the number of coordinates and $p = \log m$. Molinaro [22] introduced the function $\psi_{\kappa,p}$ as a smoother version of ℓ_p-norm and showed that using function $\psi_{\kappa,p}$, one can obtain tighter bound then using directly the ℓ_p-norm function for the scheduling problem of minimizing the ℓ_p-norm of the load vectors in the random-order model.

First, observe that

$$\nabla\psi_{\kappa,p}(\boldsymbol{u}) = \frac{p}{\kappa} \cdot \nabla\left\|\boldsymbol{1} + \frac{\kappa}{p}\boldsymbol{u}\right\|_p \in \ell_q^+ \tag{1}$$

where $q = p/(p-1)$ since

$$\begin{aligned} \frac{p}{\kappa} \cdot \frac{\partial}{\partial u_h}\left\|\boldsymbol{1} + \frac{\kappa}{p}\boldsymbol{u}\right\|_p &= \frac{\left(1 + \frac{\kappa}{p}u_h\right)^{p-1}}{\left(\sum_{h=1}^m \left(1 + \frac{\kappa}{p}u_h\right)^p\right)^{1-1/p}} \quad \forall 1 \leq h \leq m \\ \Rightarrow \quad \left\|\nabla\psi_{\kappa,p}(\boldsymbol{u})\right\|_q &= 1. \end{aligned}$$

The following lemma shows useful properties of functions $\psi_{\kappa,p}$'s and relates them to the ℓ_p-norm function.

▶ **Lemma 1** ([22]). *For arbitrary $\kappa > 0$, it holds that*

- *For all $\boldsymbol{u} \in \mathbb{R}_+^m$,*

$$\|\boldsymbol{u}\|_p \leq \psi_{\kappa,p}(\boldsymbol{u}) \leq \|\boldsymbol{u}\|_p + \frac{p(m^{1/p} - 1)}{\kappa} \tag{2}$$

- *For all $\boldsymbol{u} \in \mathbb{R}_+^m$ and $\boldsymbol{v} \in [0,1]^m$, for every coordinate $1 \leq h \leq m$,*

$$e^{-\kappa}\left(\nabla\psi_{\kappa,p}(\boldsymbol{u})\right)_h \leq \left(\nabla\psi_{\kappa,p}(\boldsymbol{u}+\boldsymbol{v})\right)_h \leq e^{\kappa}\left(\nabla\psi_{\kappa,p}(\boldsymbol{u})\right)_h \tag{3}$$

The following key inequality is proved in [22, Lemma 3.1].

▶ **Lemma 2** ([22]). *Consider a set of vector $\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_k\} \in [0,1]^m$ and let $V_1, \ldots, V_t$ be sample without replacement from this set for $1 \leq t \leq k$. Let U be a random vector in ℓ_q^+ that depends only on $V_1, \ldots, V_{t-1}$. Then, for all $\kappa > 0$,*

$$\mathbb{E}\left[\left\langle V^t, U\right\rangle\right] \leq e^{\kappa} \left\|\mathbb{E}V_t\right\|_p + \frac{1}{k-(t-1)} \cdot \frac{p(m^{1/p}-1)}{\kappa}$$

The following corollary is a direct consequence by replacing κ by $\kappa \cdot \frac{1}{4} \log\log n$.

▶ **Corollary 3.** *Consider a set of vector $\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_k\} \in [0,1]^m$ and let $V_1, \ldots, V_t$ be sample without replacement from this set for $1 \leq t \leq k$. Let U be a random vector in ℓ_q^+ that depends only on $V_1, \ldots, V_{t-1}$. Then, for all $\kappa > 0$,*

$$\mathbb{E}\left[\left\langle V^t, U\right\rangle\right] \leq e^{\kappa}(\log^{1/4} n) \left\|\mathbb{E}V_t\right\|_p + \frac{1}{k-(t-1)} \cdot \frac{p(m^{1/p}-1)}{\kappa} \frac{1}{\frac{1}{4}\log\log n}$$

▶ Remark. We emphasize that Lemma 2 and Corollary 3 hold with arbitrary $\kappa > 0$ (not necessarily $0 < \kappa < 1$). Molinaro [22] proved Lemma 2 using the regret-minimization technique from online learning. It has been observed that there is an interesting connection between regret minimization and the random-order model: regret minimization techniques can be used to prove probability inequalities. This direction has been recently explored in [1, 13, 22]. In particular, employing Lemma 2 and other powerful inequalities, Molinaro [22] proved competitive algorithms for the load balancing problem in the random-order model.

3 An $O(\log n/\log\log n)$-Competitive Algorithm in Random-Order Setting

We consider the SVCR problem in the random-order setting with the objective of minimizing the ℓ_p-norm of edge loads. The algorithm for the congestion objective will be deduced by choosing appropriate parameters.

Formulation. We say that C is a *configuration* if C is a partial feasible solution of the problem. In other words, a configuration C is a set $\{(i, P_{ij}) : 1 \leq i \leq k, P_{ij} \in \mathcal{P}_i\}$ where the couple (i, P_{ij}) represents request i and the selected $o_i - d_i$ path P_{ij} in configuration C to satisfy request i. Given an arrival order (a permutation) π, denote $\pi(t)$ the request which is released at step t in the order π. For any permutation π, let $x^{\pi}_{\pi(t),j}$ be a variable indicating whether the selected path for request $\pi(t)$ is $P_{\pi(t),j}$. For a configuration C and a permutation π, let z^{π}_{C} be a variable such that $z^{\pi}_{C} = 1$ if and only if for every $(\pi(t), P_{\pi(t),j}) \in C$, $x^{\pi}_{\pi(t),j} = 1$. In other words, $z^{\pi}_{C} = 1$ iff the selected solution is C when the request arrival order is π.

Let $\boldsymbol{\ell}(i, P_{ij}) \in \mathbb{R}^m$ be the load random vector of path P_{ij}, i.e., $\boldsymbol{\ell}(i, P_{ij})_e = X_{i,e}$ for every $e \in P_{ij}$ and equals 0 otherwise ($e \notin P_{ij}$). Moreover, let $\boldsymbol{\ell}(C)$ be the load random vector of configuration C, i.e., $\boldsymbol{\ell}(C) = \sum_{(i,P_{ij}) \in C} \boldsymbol{\ell}(i, P_{ij})$. The expected cost ($\ell_p$-norm objective) of configuration C is $\mathbb{E}_X\left[\left\|\boldsymbol{\ell}(C)\right\|_p\right]$ where the expectation is taken over the random vectors $X_{i,e}$'s. We consider the following formulation (left-hand side) and the dual of its relaxation.

$$
\begin{aligned}
&\min \mathbb{E}_{\pi}\Big[\sum_{C} \mathbb{E}_X\big[\, \|\boldsymbol{\ell}(C)\|_p\big] z_C^{\pi}\Big] \\
&\sum_{j: P_{\pi(t),j} \in \mathcal{P}_{\pi(t)}} x^{\pi}_{\pi(t),j} = 1 \quad \forall \pi, t \\
&\sum_{C:(\pi(t),P_{\pi(t),j}) \in C} z_C^{\pi} = x^{\pi}_{\pi(t),j} \quad \forall \pi, t, j \\
&\sum_{C} z_C^{\pi} = 1 \quad \forall \pi \\
&x^{\pi}_{\pi(t),j}, z_C^{\pi} \in \{0,1\} \quad \forall \pi, t, j, C
\end{aligned}
$$

$$
\begin{aligned}
&\max \sum_{\pi} \Big(\sum_{t} \alpha_t^{\pi} + \gamma^{\pi} \Big) \\
&\alpha_t^{\pi} \leq \beta_{t,j}^{\pi} \quad \forall \pi, t, j && (4) \\
&\gamma^{\pi} + \sum_{(\pi(t),P_{\pi(t),j}) \in C} \beta_{t,j}^{\pi} \leq \\
&\qquad \leq \mathbb{P}[\pi] \cdot \mathbb{E}_X\big[\|\boldsymbol{\ell}(C)\|_p\big] \quad \forall \pi, C && (5)
\end{aligned}
$$

In the primal, the first constraint guarantees that for any arrival order π, request $\pi(t)$ has to be satisfied by some path $P_{\pi(t),j} \in \mathcal{P}_{\pi(t)}$. The second constraint ensures that if request $\pi(t)$ selects path $P_{\pi(t),j}$ then the couple $(\pi(t), P_{\pi(t),j})$ must be in the solution. The third constraint says that one always has to output a solution for the problem.

Algorithm. The algorithm is primarily a form of Greedy Restart introduced by Molinaro [22] in the context of machine load balancing. We consider a generalized version for the SVCR problem in the angle of a primal-dual method with configuration LPs. Informally, for every request the algorithm selects a routing path *greedily* with respect to the function $\psi_{\kappa,p}$ which depends on the current load vector. However, when half of the requests have been considered, the algorithm *restarts* the procedure: it still chooses a routing path greedily with respect to a function $\psi_{\kappa,p}$ but now the function $\psi_{\kappa,p}$ depends on the load vector induced *only* by the second half of the requests. The intuition is the following. In the worst-case lower bound construction [3, 4, 5], at every time given the current routing the adversary traps every algorithm to accumulate the loads on links which become congested later. The restart step in the algorithm avoids accumulating the loads on potentially-congested links. The formal description of the algorithm is the following.

Let $\kappa > 0$ be a fixed parameter to be determined later. Let A_t be the configuration (partial solution) of the algorithm before the arrival of the t^{th} request. Initially, $A_0 = B_0 = \emptyset$. At the arrival of the t^{th} request, denoted as i, select a path P_{i,j^*} that is an optimal solution of

$$
\min_{P_{ij} \in \mathcal{P}_i} \Big\{ \psi_{\kappa',p}\big(\boldsymbol{\ell}(B_t) + \boldsymbol{\ell}(i, P_{ij})\big) - \psi_{\kappa',p}\big(\boldsymbol{\ell}(B_t)\big) \Big\}
$$

where $\boldsymbol{\ell}$ is the load function (defined in the formulation) and $\kappa' = \kappa \cdot \frac{1}{4} \log\log n$. Update $A_{t+1} = A_t \cup (i, P_{i,j^*})$ and $B_{t+1} = B_t \cup (i, P_{i,j^*})$. If $t = k/2 + 1$, reset $B_t = \emptyset$.

In the above description of the algorithm, we need the knowledge of k – the number of requests – in order to reset B_t at $t = k/2 + 1$. In fact, one can implement the algorithm without the knowledge of k as the following. Initially, $A_0 = B_{\text{odd}} = B_{\text{even}} = \emptyset$. At the arrival of the t^{th} request, denoted as i, select a path P_{i,j^*} that is an optimal solution of

$$
\begin{cases}
\min_{P_{ij} \in \mathcal{P}_i} \big\{ \psi_{\kappa',p}\big(\boldsymbol{\ell}(B_{\text{odd}}) + \boldsymbol{\ell}(i, P_{ij})\big) - \psi_{\kappa',p}\big(\boldsymbol{\ell}(B_{\text{odd}})\big) \big\} & \text{if } t \text{ is odd} \\
\min_{P_{ij} \in \mathcal{P}_i} \big\{ \psi_{\kappa',p}\big(\boldsymbol{\ell}(B_{\text{even}}) + \boldsymbol{\ell}(i, P_{ij})\big) - \psi_{\kappa',p}\big(\boldsymbol{\ell}(B_{\text{even}})\big) \big\} & \text{if } t \text{ is even}
\end{cases}
$$

where $\boldsymbol{\ell}$ is the load function (defined in the formulation) and $\kappa' = \kappa \cdot \frac{1}{4} \log\log n$. Update $A_{t+1} = A_t \cup (i, P_{i,j^*})$ and update B_{odd} or B_{even} depending on whether t is odd or even.

Analysis

For the sake of simplicity, we will analyze the algorithm using its first description. In the sequel, we will define the dual variables, prove the feasibility and show the competitive ratio. As κ (so κ') and p are fixed, for simplicity, we drop the indices κ' and p in $\psi_{\kappa',p}$.

Dual variables. For any permutation σ, denote A_t^σ and B_t^σ as the configurations A_t and B_t (respectively) in the execution of algorithm (before the arrival of the t^{th}) request assuming that the request arrival order is σ. Define the dual variables as follows.

$$\begin{aligned}
\beta_{t,j}^\pi &:= \frac{\mathbb{P}[\pi]}{e^{2\kappa}(\log^{1/4} n)}\mathbb{E}_X\mathbb{E}_\sigma\left[\psi\big(\boldsymbol{\ell}(B_t^\sigma) + \boldsymbol{\ell}(\sigma(t), P_{\sigma(t),j})\big) - \psi\big(\boldsymbol{\ell}(B_t^\sigma)\big)\right],\\
\alpha_t^\pi &:= \frac{\mathbb{P}[\pi]}{e^{2\kappa}(\log^{1/4} n)}\mathbb{E}_X\mathbb{E}_\sigma\left[\min_j\left\{\psi\big(\boldsymbol{\ell}(B_t^\sigma) + \boldsymbol{\ell}(\sigma(t), P_{\sigma(t),j})\big) - \psi\big(\boldsymbol{\ell}(B_t^\sigma)\big)\right\}\right]\\
&= \frac{\mathbb{P}[\pi]}{e^{2\kappa}(\log^{1/4} n)}\mathbb{E}_X\mathbb{E}_\sigma\left[\psi\big(\boldsymbol{\ell}(B_t^\sigma) + \boldsymbol{\ell}(\sigma(t), P_{\sigma(t),j^*})\big) - \psi\big(\boldsymbol{\ell}(B_t^\sigma)\big)\right],\\
\gamma^\pi &:= -\frac{\mathbb{P}[\pi]}{2e^{2\kappa}(\log^{1/4} n)}\mathbb{E}_X\mathbb{E}_\sigma\left[\|\boldsymbol{\ell}(A^\sigma)\|_p\right].
\end{aligned}$$

Informally, $\beta_{t,j}^\pi$ is proportional (up to a factor $\mathbb{P}[\pi] = 1/n!$) to the expected marginal increase (over random order σ) of the objective at the arrival of request $\sigma(t)$ assuming that the selected strategy to serve $\sigma(t)$ is $P_{\sigma(t),j}$. Variable α_t^π is also proportional (up to a factor $\mathbb{P}[\pi] = 1/n!$) to the expected marginal increase of the objective at the arrival of request $\sigma(t)$ due to the algorithm.

▶ **Lemma 4.** *For any permutation σ, denote A^σ as the final configuration of the algorithm in case that the request arrival order is σ. Suppose that the cost of the algorithm $\mathbb{E}_X\mathbb{E}_\sigma\left[\|\boldsymbol{\ell}(A^\sigma)\|_p\right] \geq \frac{4e^\kappa p(m^{1/p}-1)}{\kappa\cdot\frac{1}{4}\log\log n}$. Then the variables defined above constitute a dual feasible solution.*

Proof. The first dual constraint (4) follows immediately the definitions of α_t^π and $\beta_{t,j}^\pi$. In the remaining of the proof, we prove the second dual constraint (5). Fix a configuration C and a permutation π. Let $P_{i,c(i)}$ be the path of request i in configuration C. In other words, configuration C consists of couples $(i, P_{i,c(i)})$ for all requests i.

By the definition of dual variables, the second constraint reads: for any given permutation π and any given configuration C,

$$-\frac{1}{2}\mathbb{P}[\pi]\cdot\mathbb{E}_X\mathbb{E}_\sigma\left[\|\boldsymbol{\ell}(A^\sigma)\|_p\right] + \sum_{t=1}^k \mathbb{P}[\pi]\cdot\mathbb{E}_X\mathbb{E}_\sigma\left[\psi\big(\boldsymbol{\ell}(B_t^\sigma) + \boldsymbol{\ell}(\sigma(t), P_{\sigma(t),j})\big) - \psi\big(\boldsymbol{\ell}(B_t^\sigma)\big)\right] \\ \leq e^{2\kappa}(\log^{1/4} n)\cdot\mathbb{P}[\pi]\cdot\mathbb{E}_X\left[\|\boldsymbol{\ell}(C)\|_p\right]$$

where for any permutation σ, the path $P_{\sigma(t),c(\sigma(t))}$ of request $\sigma(t)$ is completely determined in configuration C, i.e., $(\sigma(t), P_{\sigma(t),c(\sigma(t))}) \in C$. This is equivalent to

$$\sum_{t=1}^k \mathbb{E}_X\mathbb{E}_\sigma\left[\psi\big(\boldsymbol{\ell}(B_t^\sigma) + \boldsymbol{\ell}(\sigma(t), P_{\sigma(t),j})\big) - \psi\big(\boldsymbol{\ell}(B_t^\sigma)\big)\right] \\ \leq e^{2\kappa}(\log^{1/4} n)\cdot\mathbb{E}_X\left[\|\boldsymbol{\ell}(C)\|_p\right] + \frac{1}{2}\cdot\mathbb{E}_X\mathbb{E}_\sigma\left[\|\boldsymbol{\ell}(A^\sigma)\|_p\right]. \tag{6}$$

We prove Inequality (6). First we bound the sum in the left-hand side for all $1 \leq t \leq k/2$.

$$\begin{aligned}
&\mathbb{E}_X \sum_{t=1}^{k/2} \mathbb{E}_\sigma \Big[\psi\left(\boldsymbol{\ell}(B_t^\sigma) + \boldsymbol{\ell}\left(\sigma(t), P_{\sigma(t),c(\sigma(t))}\right)\right) - \psi\left(\boldsymbol{\ell}(B_t^\sigma)\right)\Big] \\
&\leq \mathbb{E}_X \sum_{t=1}^{k/2} \mathbb{E}_\sigma \Big[\Big\langle \nabla\psi\left(\boldsymbol{\ell}(B_t^\sigma) + \boldsymbol{\ell}\left(\sigma(t), P_{\sigma(t),c(\sigma(t))}\right)\right), \boldsymbol{\ell}\left(\sigma(t), P_{\sigma(t),c(\sigma(t))}\right)\Big\rangle\Big] \\
&\leq e^\kappa \sum_{t=1}^{k/2} \mathbb{E}_X \mathbb{E}_\sigma \Big[\Big\langle \nabla\psi\left(\boldsymbol{\ell}(B_t^\sigma)\right), \boldsymbol{\ell}\left(\sigma(t), P_{\sigma(t),c(\sigma(t))}\right)\Big\rangle\Big] \\
&\leq e^\kappa \cdot \sum_{t=1}^{k/2} \Bigg(e^\kappa(\log^{1/4} n) \cdot \mathbb{E}_X \left\| \mathbb{E}_\sigma \Big[\boldsymbol{\ell}\Big(\sigma(t), P_{\sigma(t),c(\sigma(t))}\Big)\Big] \right\|_p + \frac{1}{k-t+1} \cdot \frac{p(m^{1/p}-1)}{\kappa \cdot \frac{1}{4}\log\log n}\Bigg) \\
&= e^{2\kappa}(\log^{1/4} n) \cdot \frac{k}{2} \cdot \mathbb{E}_X \left\| \frac{\boldsymbol{\ell}(C)}{k} \right\|_p + e^\kappa \sum_{t=1}^{k/2} \frac{1}{k-t+1} \cdot \frac{p(m^{1/p}-1)}{\kappa \cdot \frac{1}{4}\log\log n} \\
&\leq \frac{e^{2\kappa}(\log^{1/4} n)}{2} \mathbb{E}_X\left[\|\boldsymbol{\ell}(C)\|_p\right] + e^\kappa \cdot \frac{p(m^{1/p}-1)}{\kappa \cdot \frac{1}{4}\log\log n} \\
&< \frac{e^{2\kappa}(\log^{1/4} n)}{2} \mathbb{E}_X\left[\|\boldsymbol{\ell}(C)\|_p\right] + \frac{1}{4} \cdot \mathbb{E}_X \mathbb{E}_\sigma\left[\|\boldsymbol{\ell}(A^\sigma)\|_p\right].
\end{aligned} \tag{7}$$

Recall that $\boldsymbol{\ell}\left(\sigma(t), P_{\sigma(t),c(\sigma(t))}\right) \in [0,1]^m$. The first and second inequalities follow the convexity of ψ and Lemma 1 (Inequality (3)), respectively. The third inequality holds by Corollary 3 and note that $\nabla\psi\left(\boldsymbol{\ell}(B_t^\sigma)\right) \in \ell_q^+$ by observation (1). The next equality is due to the fact that σ is an uniform random order. The last inequality follows the assumption of the algorithm cost.

Now we bound the sum of the left-hand side of Inequality (6) for $k/2 < t \leq k$. That can be done similarly with a subtle observation. For completeness, we show all steps.

$$\begin{aligned}
&\mathbb{E}_X \sum_{t=k/2+1}^{k} \mathbb{E}_\sigma \Big[\psi\left(\boldsymbol{\ell}(B_t^\sigma) + \boldsymbol{\ell}\left(\sigma(t), P_{\sigma(t),c(\sigma(t))}\right)\right) - \psi\left(\boldsymbol{\ell}(B_t^\sigma)\right)\Big] \\
&\leq \mathbb{E}_X \sum_{t=k/2+1}^{k} \mathbb{E}_\sigma \Big[\Big\langle \nabla\psi\left(\boldsymbol{\ell}(B_t^\sigma) + \boldsymbol{\ell}\left(\sigma(t), P_{\sigma(t),c(\sigma(t))}\right)\right), \boldsymbol{\ell}\left(\sigma(t), P_{\sigma(t),c(\sigma(t))}\right)\Big\rangle\Big] \\
&\leq e^\kappa \sum_{t=k/2+1}^{k} \mathbb{E}_X \mathbb{E}_\sigma \Big[\Big\langle \nabla\psi\left(\boldsymbol{\ell}(B_t^\sigma)\right), \boldsymbol{\ell}\left(\sigma(t), P_{\sigma(t),c(\sigma(t))}\right)\Big\rangle\Big] \\
&\leq e^\kappa \cdot \sum_{t=k/2+1}^{k} \Bigg(e^\kappa(\log^{1/4} n) \cdot \mathbb{E}_X \left\| \mathbb{E}_\sigma \Big[\boldsymbol{\ell}\left(\sigma(t), P_{\sigma(t),c(\sigma(t))}\right)\Big] \right\|_p \\
&\qquad\qquad + \frac{1}{k-(t-k/2-1)} \cdot \frac{p(m^{1/p}-1)}{\kappa \cdot \frac{1}{4}\log\log n}\Bigg) \\
&= e^{2\kappa}(\log^{1/4} n) \cdot \frac{k}{2} \cdot \mathbb{E}_X \left\| \frac{\boldsymbol{\ell}(C)}{k} \right\|_p + e^\kappa \sum_{t=k/2+1}^{k} \frac{1}{k-(t-k/2-1)} \cdot \frac{p(m^{1/p}-1)}{\kappa \cdot \frac{1}{4}\log\log n} \\
&\leq \frac{e^{2\kappa}(\log^{1/4} n)}{2} \mathbb{E}_X\left[\|\boldsymbol{\ell}(C)\|_p\right] + e^\kappa \cdot \frac{p(m^{1/p}-1)}{\kappa \cdot \frac{1}{4}\log\log n} \\
&< \frac{e^{2\kappa}(\log^{1/4} n)}{2} \mathbb{E}_X\left[\|\boldsymbol{\ell}(C)\|_p\right] + \frac{1}{4} \cdot \mathbb{E}_X \mathbb{E}_\sigma\left[\|\boldsymbol{\ell}(A^\sigma)\|_p\right].
\end{aligned} \tag{8}$$

All the above equalities and inequalities follow by the same arguments as before except the third inequality. In the latter, we apply Corollary 3 with the observation that $\nabla\psi\left(\boldsymbol{\ell}(B_t^\sigma)\right)$ depends only on $(t - k/2 - 1)$ random load variables due to the fact that the algorithm restarts at $t = k/2$. This interesting idea has been observed by Molinaro [22]. Note that this is the only place we use the restart property of the algorithm.

Hence, summing Inequalities (7) and (8), Inequality (6) follows. ◀

▶ **Theorem 5.** *For any arbitrary $\kappa > 0$, the algorithm has expected cost at most $2e^{2\kappa}(\log^{1/4} n)$ times the optimal value plus an additive constant $\frac{4e^\kappa p(m^{1/p}-1)}{\kappa\cdot\frac{1}{4}\log\log n}$ for the SVCR problem with ℓ_p-norm objective in the random-order setting.*

Proof. Consider first the case where the (expected) cost of the algorithm $\mathbb{E}_X\mathbb{E}_\sigma\left[\|\boldsymbol{\ell}(A^\sigma)\|_p\right] \geq \frac{4e^\kappa p(m^{1/p}-1)}{\kappa\cdot\frac{1}{4}\log\log n}$. Then, by the algorithm and the definition of dual variables, the dual objective equals

$$\begin{aligned}
&\sum_\pi \Big(\sum_t \alpha_t^\pi + \gamma^\pi \Big) \\
&= \frac{\mathbb{P}[\pi]}{e^{2\kappa}(\log^{1/4} n)} \sum_{\pi,t} \mathbb{E}_X\mathbb{E}_\sigma\left[\psi\big(\boldsymbol{\ell}(B_t^\sigma) + \boldsymbol{\ell}(\sigma(t), P_{\sigma(t),j^*})\big) - \psi\big(\boldsymbol{\ell}(B_t^\sigma)\big)\right] \\
&\qquad\qquad - \frac{\mathbb{P}[\pi]}{2e^{2\kappa}(\log^{1/4} n)} \sum_\pi \mathbb{E}_X\mathbb{E}_\sigma\left[\|\boldsymbol{\ell}(A^\sigma)\|_p\right] \\
&= \frac{1}{e^{2\kappa}(\log^{1/4} n)} \mathbb{E}_X\mathbb{E}_\sigma\left[\psi\big(\boldsymbol{\ell}(B_{n/2}^\sigma)\big) + \psi\big(\boldsymbol{\ell}(B_n^\sigma)\big)\right] - \frac{1}{2e^{2\kappa}(\log^{1/4} n)} \cdot \mathbb{E}_X\mathbb{E}_\sigma\left[\|\boldsymbol{\ell}(A^\sigma)\|_p\right] \\
&\geq \frac{1}{e^{2\kappa}(\log^{1/4} n)} \mathbb{E}_X\mathbb{E}_\sigma\left[\left\|\boldsymbol{\ell}(B_{n/2}^\sigma)\right\|_p + \|\boldsymbol{\ell}(B_n^\sigma)\|_p\right] - \frac{1}{2e^{2\kappa}(\log^{1/4} n)} \cdot \mathbb{E}_X\mathbb{E}_\sigma\left[\|\boldsymbol{\ell}(A^\sigma)\|_p\right] \\
&\geq \frac{1}{e^{2\kappa}(\log^{1/4} n)} \mathbb{E}_X\mathbb{E}_\sigma\left[\left\|\boldsymbol{\ell}(B_{n/2}^\sigma) + \boldsymbol{\ell}(B_n^\sigma)\right\|_p\right] - \frac{1}{2e^{2\kappa}(\log^{1/4} n)} \cdot \mathbb{E}_X\mathbb{E}_\sigma\left[\|\boldsymbol{\ell}(A^\sigma)\|_p\right] \\
&= \frac{1}{e^{2\kappa}(\log^{1/4} n)} \mathbb{E}_X\mathbb{E}_\sigma\left[\|\boldsymbol{\ell}(A^\sigma)\|_p\right] - \frac{1}{2e^{2\kappa}(\log^{1/4} n)} \cdot \mathbb{E}_X\mathbb{E}_\sigma\left[\|\boldsymbol{\ell}(A^\sigma)\|_p\right] \\
&= \frac{1}{2e^{2\kappa}(\log^{1/4} n)} \cdot \mathbb{E}_X\mathbb{E}_\sigma\left[\|\boldsymbol{\ell}(A^\sigma)\|_p\right].
\end{aligned}$$

The first inequality follows the properties of ψ (Lemma 1, Inequality (2)). The second inequality is due to the norm inequality $\|\boldsymbol{a}\|_p + \|\boldsymbol{b}\|_p \geq \|\boldsymbol{a}+\boldsymbol{b}\|_p$. The subsequent equality holds since $B_{n/2}^\sigma \uplus B_n^\sigma = A^\sigma$ (note that $B_{n/2+1}^\sigma$ was re-initialized as an empty set).

Besides, the primal is $\mathbb{E}_X\mathbb{E}_\sigma\left[\|\boldsymbol{\ell}(A^\sigma)\|_p\right]$. Therefore, by weak duality, $\mathbb{E}_X\mathbb{E}_\sigma\left[\|\boldsymbol{\ell}(A^\sigma)\|_p\right] \leq 2e^{2\kappa}(\log^{1/4} n)OPT$ where OPT is the value of an optimal solution.

Now consider the case that the expected cost of the algorithm $\mathbb{E}_X\mathbb{E}_\sigma\left[\|\boldsymbol{\ell}(A^\sigma)\|_p\right]$ is at most $\frac{4e^\kappa p(m^{1/p}-1)}{\kappa\cdot\frac{1}{4}\log\log n}$. Obviously, $\mathbb{E}_X\mathbb{E}_\sigma\left[\|\boldsymbol{\ell}(A^\sigma)\|_p\right] < OPT + \frac{4e^\kappa p(m^{1/p}-1)}{\kappa\cdot\frac{1}{4}\log\log n}$. Therefore, combining the cases we deduce that

$$\mathbb{E}_X\mathbb{E}_\sigma\left[\|\boldsymbol{\ell}(A^\sigma)\|_p\right] \leq 2e^{2\kappa}(\log^{1/4} n)OPT + \frac{4e^\kappa p(m^{1/p}-1)}{\kappa\cdot\frac{1}{4}\log\log n}. \qquad ◀$$

▶ **Corollary 6.** *Assume that the optimum solution is at least 1 (i.e., the optimal routing is sufficiently congested). Then the algorithm with parameters $p = O(\log n)$ and $\kappa = 1$ is $O(\log n/\log\log n)$-approximation for the SVCR problem.*

Proof. Recall that the congestion (ℓ_∞-norms over edge loads) can be approximated up to a constant factor by the ℓ_p-norm function for $p = \log m = O(\log n)$. Applying Theorem 5 for $p = O(\log n)$ and $\kappa = 1$, we have the following upper-bound on the congestion of the algorithm:

$$O\big(e^{2\kappa}(\log^{1/4} n)\big)OPT + \frac{4e^{\kappa}p(m^{1/p}-1)}{\kappa \cdot \frac{1}{4}\log\log n} \leq O\left(e^{2\kappa}(\log^{1/4} n) + \frac{e^{\kappa}\log n}{\kappa \cdot \frac{1}{4}\log\log n}\right)OPT$$
$$= O\left(\log^{1/4} n + \frac{\log n}{\log\log n}\right)OPT = O\left(\frac{\log n}{\log\log n}\right)OPT \qquad (9)$$

where OPT is the value of an optimal solution. As the optimum solution is at least 1, the corollary follows. ◀

4 A Simple $\Theta(\log n/\log\log n)$-Approximation Algorithm for Virtual Circuit Routing

In this section, we revisit the classic virtual circuit routing problem and provide a simple algorithm with tight approximation guarantee (assuming some complexity hypothesis).

Virtual Circuit Routing. In the problem, there is a directed graph $G(V, E)$ where $|V| = n$ and a collection of k requests. A request i for $1 \leq i \leq k$ is specified by a origin-destination pairs $o_i, d_i \in V$, and a positive weight w_i representing the (deterministic) load of request i on an edge e if it is used by request i. The goal is to choose for each request i a routing path connecting o_i and d_i so that the *congestion* induced by the collection of all paths is minimized. The load of an edge e is equal to the total weight of requests routing through e, i.e., $\sum_i w_i$ where the sum is taken over all requests i whose some path contains e. The congestion of a collection of paths is the maximum load over all edges.

Approximation algorithm.

1. Normalize all request weights by dividing every weight by $\max_{i'} w_{i'}$. The new *normalized* weights $\widetilde{w}_i = \frac{w_i}{\max_{i'} w_{i'}}$ satisfy $\widetilde{w}_i \in [0, 1]$.
2. Define the parameters $p = O(\log n)$, $\kappa = 1$ and $\kappa' = \frac{1}{4}\log\log n$.
3. Sample an uniform random order of the requests and consider requests in this order.
4. Let A_t be the configuration (partial solution) of the algorithm before the arrival of the t^{th} request. Initially, $A_0 = B_0 = \emptyset$. At the arrival of the t^{th} request, denoted as i, select a path P_{i,j^*} that is an optimal solution of

$$\min_{P_{ij} \in \mathcal{P}_i} \psi_{\kappa',p}\big(\tilde{\boldsymbol{\ell}}(B_t) + \tilde{\boldsymbol{\ell}}(i, P_{ij})\big) - \psi_{\kappa',p}\big(\tilde{\boldsymbol{\ell}}(B_t)\big)$$

where $\tilde{\boldsymbol{\ell}}$ is the load function with respect to the normalized weights. Update $A_{t+1} = A_t \cup (i, P_{i,j^*})$ and $B_{t+1} = B_t \cup (i, P_{i,j^*})$. If $t = k/2 + 1$, reset $B_t = \emptyset$.

▶ **Theorem 7** ([23, 24, 6]). *The algorithm has approximation ratio $O(\log n/\log\log n)$.*

Proof. By Corollary 6, specifically Inequality (9), we have the bound on the congestion of the algorithm (after normalizing the weights):

$$\mathbb{E}[\widetilde{ALG}] \leq O\left(\frac{\log n}{\log\log n}\right)\widetilde{OPT}$$

where $\widetilde{ALG}$ and $\widetilde{OPT}$ are the congestions of the algorithm and the optimal solution with normalized weights, respectively. Multiplying both sides by the normalizing factor, the theorem follows. ◀

5 Conclusion

In the paper, we have provided a competitive algorithm for the SCVR problem and prove that the quality of approximation solutions to the problem can be preserved even with the presence of uncertainty. Through the paper, we also show that primal-dual approaches are robust in the stochastic model and the random-order model can be used to design/simplify randomized approximation algorithms. A direction is to design randomized algorithms for other (stochastic) problems using primal-dual techniques and random-order request sequences.

References

1 Shipra Agrawal and Nikhil R Devanur. Fast algorithms for online stochastic convex programming. In *Proc. 26th ACM-SIAM symposium on Discrete algorithms*, pages 1405–1424, 2014.

2 Matthew Andrews and Lisa Zhang. Logarithmic hardness of the directed congestion minimization problem. In *Proc. 38th Symposium on Theory of Computing*, pages 517–526, 2006.

3 James Aspnes, Yossi Azar, Amos Fiat, Serge Plotkin, and Orli Waarts. On-line routing of virtual circuits with applications to load balancing and machine scheduling. *Journal of the ACM (JACM)*, 44(3):486–504, 1997.

4 Baruch Awerbuch, Yossi Azar, Edward F Grove, Ming-Yang Kao, P Krishnan, and Jeffrey Scott Vitter. Load balancing in the ℓ_p-norm. In *Proc. 36th Foundations of Computer Science*, pages 383–391, 1995.

5 Ioannis Caragiannis. Better bounds for online load balancing on unrelated machines. In *Proc. 19th Symposium on Discrete Algorithms*, pages 972–981, 2008.

6 Chandra Chekuri and Mark Idleman. Congestion minimization for multipath routing via multiroute flows. In *Proc. 1st Symposium on Simplicity in Algorithms*, 2018.

7 Chandra Chekuri, Jan Vondrak, and Rico Zenklusen. Dependent randomized rounding via exchange properties of combinatorial structures. In *Proc. 51st Annual IEEE Symposium on Foundations of Computer Science*, pages 575–584, 2010.

8 Julia Chuzhoy, Venkatesan Guruswami, Sanjeev Khanna, and Kunal Talwar. Hardness of routing with congestion in directed graphs. In *Proc. 39th ACM Symposium on Theory of Computing*, pages 165–178, 2007.

9 Benjamin Doerr. Randomly rounding rationals with cardinality constraints and derandomizations. In *Symposium on Theoretical Aspects of Computer Science*, pages 441–452, 2007.

10 Rajiv Gandhi, Samir Khuller, Srinivasan Parthasarathy, and Aravind Srinivasan. Dependent rounding and its applications to approximation algorithms. *Journal of the ACM*, 53(3):324–360, 2006.

11 Ashish Goel and Piotr Indyk. Stochastic load balancing and related problems. In *Proc. 40th Symposium on Foundations of Computer Science*, pages 579–586, 1999.

12 Anupam Gupta, Amit Kumar, Viswanath Nagarajan, and Xiangkun Shen. Stochastic load balancing on unrelated machines. In *Proc. 29th Symposium on Discrete Algorithms*, pages 1274–1285, 2018.

13 Anupam Gupta and Marco Molinaro. How the experts algorithm can help solve LPs online. *Mathematics of Operations Research*, 41(4):1404–1431, 2016.

14 Varun Gupta, Benjamin Moseley, Marc Uetz, and Qiaomin Xie. Stochastic online scheduling on unrelated machines. In *Conference on Integer Programming and Combinatorial Optimization*, pages 228–240, 2017.

15 Sungjin Im, Benjamin Moseley, and Kirk Pruhs. Stochastic scheduling of heavy-tailed jobs. In *Proc. 32nd Symposium on Theoretical Aspects of Computer Science*, pages 474–486, 2015.

16 Wataru Kishimoto and Masashi Takeuchi. m-route flows in a network. *Electronics and Communications in Japan (Part III: Fundamental Electronic Science)*, 77(5):1–18, 1994.

17 Jon Kleinberg, Yuval Rabani, and Éva Tardos. Allocating bandwidth for bursty connections. *SIAM Journal on Computing*, 30(1):191–217, 2000.

18 Jian Li and Amol Deshpande. Maximizing expected utility for stochastic combinatorial optimization problems. *Mathematics of Operations Research*, 2018.

19 Jian Li and Wen Yuan. Stochastic combinatorial optimization via poisson approximation. In *Proc. 45th Symposium on Theory of Computing*, pages 971–980, 2013.

20 Nicole Megow, Marc Uetz, and Tjark Vredeveld. Models and algorithms for stochastic online scheduling. *Mathematics of Operations Research*, 31(3):513–525, 2006.

21 Rolf H Möhring, Andreas S Schulz, and Marc Uetz. Approximation in stochastic scheduling: the power of LP-based priority policies. *Journal of the ACM*, 46(6):924–942, 1999.

22 Marco Molinaro. Online and random-order load balancing simultaneously. In *Proc. 28th ACM-SIAM Symposium on Discrete Algorithms*, pages 1638–1650, 2017.

23 Prabhakar Raghavan and Clark D Thompson. Randomized rounding: a technique for provably good algorithms and algorithmic proofs. *Combinatorica*, 7(4):365–374, 1987.

24 Aravind Srinivasan. Distributions on level-sets with applications to approximation algorithms. In *Proc. 42nd IEEE Symposium on Foundations of Computer Science*, pages 588–597, 2001.

25 Nguyen Kim Thang. Online primal-dual algorithms with configuration linear programs. *arXiv*, 2017. arXiv:1708.04903.