A queueing-theoretic model for resource allocation in one-dimensional distributed analytics network

Nitish K. Panigrahy[†], Prithwish Basu[¶], Don Towsley[†], Ananthram Swami[‡],
Kevin S. Chan[‡] and Kin K. Leung[§]
[†] University of Massachusetts Amherst, MA, USA. Email: {nitish, towsley}@cs.umass.edu
[¶]Raytheon BBN Technologies, Cambridge, MA 02138, USA. Email:prithwish.basu@raytheon.com
[‡]Army Research Laboratory, Adelphi, MD 20783, USA. Email:{ananthram.swami, kevin.s.chan}.civ@mail.mil
[§]Imperial College London, London SW72AZ, UK. Email: kin.leung@imperial.ac.uk

***Abstract*—We consider assignment policies that allocate resources to requesting users, where everyone is located on a one-dimensional line $\mathcal{L} = [0, \infty)$. First, we consider a unidirectional problem where a resource can only be allocated to a user located to its left as exemplified here. Imagine a one-way city street with traffic flowing from right to left; a ride-sharing company has distributed its vehicles along the street, which are ready to pick up users waiting at various locations. Users equipped with smartphone ride-hailing apps can register their requests on a ride allocation system. The latter then attempts to service each user by assigning a vehicle with spare capacity located to the user's right such that the average "pick up" distance is minimized. We propose the Move to Right (*MTR*) policy which assigns the nearest available resource located to the right, and contrast it with the Unidirectional Gale-Shapley (*UGS*) Matching policy known in the literature. While both these policies are optimal, we show that they are equivalent with respect to the expected distance traveled by a request (*request distance*), although *MTR* tends to be fairer, i.e., has low variance[1]. Moreover, we show that when the locations of users and resources are modeled by statistical point processes, the spatial system under unidirectional policies can be mapped to a temporal queuing system, thus allowing the application of a plethora of queuing theory results that yield closed form expressions. We also consider the bidirectional problem where there are no directional restrictions on resource allocation, e.g., assigning distributed analytics tasks to servers in vehicular sensor networks, and give an optimal policy that has lower expected time complexity than known algorithms in literature in resource rich regimes. Finally, numerical evaluation of performance of unidirectional and bidirectional allocation schemes yield design guidelines beneficial for resource placement.**

This research was sponsored by the U.S. Army Research Laboratory and the U.K. Ministry of Defence under Agreement Number W911NF-16-3-0001. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Army Research Laboratory, the U.S. Government, the U.K. Ministry of Defence or the U.K. Government. The U.S. and U.K. Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon. This document does not contain technology or technical data controlled under either the U.S. International Traffic in Arms Regulations or the U.S. Export Administration Regulations.

[1]It is well known in queueing theory that among all service disciplines the variance of the waiting time is minimized under FCFS policy [10]. Later in Section IV we will show that MTR and UGS maps to a temporal FCFS and LCFS-PR queue respectively.

I. Introduction

The past few years have witnessed significant growth in the use of distributed network analytics involving agile code, data and computational resources. In many such networked systems, for example, Internet of Things (IoT) [5], a large number of computational and storage resources are widely distributed in the physical world. These resources are accessed by various end users/applications that are also distributed over the physical space. Assigning users or applications to resources efficiently is key to the sustained high-performance operation of the system.

While in some applications the "service" occurs over a communication network, in others it occurs over a physical space. Examples of the former type of service include accessing storage resources over a wireless network to store files and requesting computational resources to run image processing tasks; whereas an example of the latter type of service is the arrival of ride-sharing vehicles to the user's location over a road transportation network.

Not surprisingly, the relative spatial distribution of resources and users in the network plays an important role in determining the overall performance of the service. A key measure of performance is the *average request distance*, that is the average distance between a user and its allocated resource/service (where the distance is measured on the network). This directly translates to the user's latency of accessing the service, which is arguably among the most important criteria in distributed service applications. An important practical constraint in such networks is finite service capacity. For example, in network analytics applications, a networked storage device can only support a finite number of concurrent users; similarly, a computational resource can only support a finite number of concurrent processing tasks. Likewise, in physical service applications like ride-sharing, a vehicle can pick up a finite number of passengers.

Therefore, a primary problem in such distributed service networks is to efficiently assign each user to a suitable resource such that the average request distance is minimized and no resource serves more users than its capacity. If the entire system is being managed by a single administrative entity

such as a ride sharing service, or a datacenter network where analytics tasks are being assigned to available CPUs, there are economic benefits in minimizing the average request distance across all (user, resource) pairs, which is tantamount to minimizing the average delay in the system.

The general version of this capacitated assignment problem can be solved by modeling it as a minimum cost flow problem on graphs [3] and running the network simplex algorithm [13]. However, if the network has a low-dimensional structure and some assumptions about the spatial distributions of users and resources hold, more efficient methods can be developed.

In this paper, we consider two one-dimensional network scenarios that motivate the study of this special case of the user-to-resource assignment problem.

The first scenario is of ride-hailing on a one-way street which allows vehicle traffic to run from right to left. If the vehicles of a ride-sharing company are distributed along the street at a certain time instant, and users equipped with smartphone ride-hailing apps request service, the system attempts to assign vehicles with spare capacity located towards the right of the users so as to minimize average "pick up" distance. Abadi et al. [1] introduced this problem and gave an optimal policy known as Unidirectional Gale-Shapley [2] matching (*UGS*), in which all users transmit rays of light toward their right in parallel and are matched with the vehicles that first receive the respective rays.

In this paper, we propose another optimal policy "Move to Right" policy (or *MTR*) which has the same "expected distance traveled by a request" (*request distance*) as UGS but has a lower variance. *MTR* sequentially allocates users to the geographically nearest available vehicle located to his/her right. If the locations of requesters and vehicles are modeled by independent Poisson processes, average request distance can be characterized in closed form by considering inter-user and inter-server distances as parameters of an M/M/1 temporal queue, or of a *bulk service* M/M/1 queue depending on server capacity, where capacity denotes the maximum number of users that can be handled by a server. We equate request distance in the spatial system to the expected *sojourn time* for the corresponding queuing model[3]. This natural mapping allows us to use well-known results from queueing theory to characterize request distances in closed form for a number of interesting situations beyond M/M/1 queues.

The second scenario involves distributed analytics in a convoy of vehicles with optional sensors and/or computing resource traveling on a road. The convoy is spatially distributed over a one-dimensional space and the communication between vehicles therein involves non-interfering single hop wireless transmission, also known as the *direct transmission model* [4]. Vehicles have heterogeneous capabilities – those with sensing/imaging resources may or may not have sufficient computational resources to do the processing, whereas others with rich computational resources allow computational tasks (e.g., deep neural network based image classification) to be assigned to them. In this case, since no directionality restrictions are imposed on the allocation algorithms, computing the optimal assignment is not as simple as in *MTR*.

We explore the special structure of the one-dimensional topology to develop an optimal algorithm that assigns requester nodes R to resource nodes S such that the total cost of assignment is minimized. Although this has been recently solved for $|R| = |S|$ [6], the $|R| < |S|$ case is still open. Our algorithm solves this case with time complexity $O(|R|^3)$. If the locations of R and S are distributed according to independent Poisson processes, we show by simulation that the sizes of groups (i.e., contiguously allocated (requester, resource) pairs) that we encounter are small and do not depend on $|R|$. Hence we conjecture without proof that the average running time of our optimal algorithm is much lower than the worst case of $O(|R|^3)$ in such realistic circumstances and is likely to be $O(|R|)$. Note that other assignment algorithms in literature such as the Hungarian primal-dual algorithm and Agarwal's variant [2] assume $|R| = |S|$ for general and Euclidean distance measures and hence applying those for the $|R| < |S|$ case would involve adding $|S| - |R|$ dummy user nodes far away. Thus in realistic resource rich environments, our optimal algorithm can be more efficient than these approaches.

Our contributions are summarized below:

1) Simple directional allocation policies such as: *MTR* and *UGS* for unidirectional assignment, and their queueing theoretic analysis yielding closed form expressions for request distance.
   - If the inter-requester and inter-resource distances are exponentially distributed, we model unidirectional policies using a temporal M/M/1 queue for basic assignment and using a bulk service M/M/1 queue for assignment under resource capacity constraints.
   - If the inter-requester distances are exponentially distributed but the inter-resource distances are generally distributed, we model the situation using an M/G/1 queue with the first customer having exceptional service time and derive expressions for exceptional service time under various inter-server distance distributions.
   - Analysis of request distance in closed form under general distance distributions when the inter-requester and inter-resource distributions have similar mean values.
2) An optimal algorithm for bidirectional assignment with time complexity $O(|R|^3)$, which can be better than known algorithms if $|R| << |S|$ or if the input distance distributions are Poisson.
3) A thorough numerical simulation study of five assignment schemes: UGS , MTR, nearest neighbor, stable assignment due to Gale and Shapley, and optimal assignment.

The paper is organized as follows. The next section discusses about the related literature. Section III contains some

[2]We rename queue matching defined in [1] as Unidirectional Gale-Shapley Matching to avoid overloading the term queue.

[3]Note that sojourn time is the sum of waiting and service times in a queue.

technical preliminaries. We show the equivalence of queue matching and MTR policy w.r.t expected request distance in Section IV, and present results associated with the case when servers are poisson distributed in Section V. We develop formulations for a general inter-server distance distribution in Section VI. The optimal allocation strategy with request distance as metric is presented in Section VII. We compare the performance of various local allocation strategies in Section VIII. We conclude the paper in Section IX.

II. Related Work

Poisson Matching: Holroyd et al. [9] first studied translation-invariant matchings between two d-dimensional Poisson processes with equal densities. Their primary focus was obtaining upper and lower bounds on matching distance for stable matchings. Abadi et al. [1] introduced UGS and derived bounds on the expected matching distance for stable matchings between two one-dimensional Poisson processes with different densities. In this paper, we propose another unidirectional allocation policy: "Move To Right" policy (*MTR*) and provide explicit expressions for the expected matching distance for both MTR and UGS when one set of points is distributed according to a renewal process.

Euclidean Bipartite Matching: The optimal requester-server assignment problem can be modeled as a minimum-weight matching on a weighted bipartite graph where weights on edges are given by the Euclidean distance between the corresponding vertices [12]. Well-known polynomial time solutions exist for this problem, such as the modified Hungarian algorithm proposed by Agarwal et al. [2] with a running time of $O(|R|^{2+\epsilon})$, where $|R|$ is the total number of requesters. In case of an equal number of requesters and servers, the optimal requester-server assignment on a real line is known [6]. In this paper, we consider the case when there are fewer requesters than servers.

III. Technical Preliminaries

Consider a set of requesters R requesting tasks with equal computation requirements and a set of computation servers S that can execute these requests. Assume that each server $j \in S$ has capacity $C_j \in \mathbb{Z}^+$ corresponding to maximum number of tasks that can be processed. Suppose requesters and servers are located in space $\hat{\mathcal{L}}$. Formally, let $r : R \to \hat{\mathcal{L}}$ and $s : S \to \hat{\mathcal{L}}$ be the location functions for requesters and servers, respectively, such that a distance measure $d_{\hat{\mathcal{L}}}(r, s)$ is well defined for all pairs $(r, s) \in R \times S$. In this paper we examine the scenario where $\hat{\mathcal{L}} = \mathbb{R}^+ = \mathcal{L}$ (say), i.e., the positive real line. We also assume that all servers have equal capacities i.e. $C_j = c \; \forall j \in S$.

A. Requester and server spatial distributions

Let $\{r_i, i \geq 1\}$ represent successive requester locations and $\{s_j, j \geq 1\}$ be the successive server locations. Let $X_j = s_j - s_{j-1}$ denote the inter-server distances and $Y_i = r_i - r_{i-1}$ be the inter-requester distances. We consider $\{X_j\}_{j\geq 1}$ to be a renewal process with cumulative distribution function (c.d.f.) satisfying

$$G_b(x) = \mathbb{P}(X_j \leq x). \tag{1}$$

We also assume Y_i is exponentially distributed with density λ i.e.

$$\mathbb{P}(Y_i \leq x) = 1 - e^{-\lambda x}. \tag{2}$$

We denote α_b and σ_b^2 to be the mean and variance associated with the inter-server distance distribution $G_b(x)$.

In our paper, we consider various inter-server distance distributions, including exponential, deterministic, uniform, hyperexponential and Weibull.

B. Allocation Policies

In this paper, one of our goal is to analyze the performance of various request allocation policies taking expected request distance as a performance metric. We define various allocation policies as follows.

Unidirectional Gale-Shapley (*UGS*): In UGS, each requester simultaneously emits ray to their right. Once the ray hits an unallocated server s, the emitter will be allocated to s.

Move To Right (*MTR*): In MTR, starting from left, each requester is allocated sequentially to the nearest available server to its right.

Nearest Neighbor (*NN*) [14]: In this matching, each requester greedily selects the nearest available server and we take that pair out, and continue.

Gale-Shapley (*GS*) [7]: In this matching, each requester selects the nearest server and each server selects its nearest requester. We only remove reciprocating pairs, and continue.

Optimal Matching: This matching provides the minimum average request distance among all feasible allocation policies.

IV. Unidirectional Allocation Policies

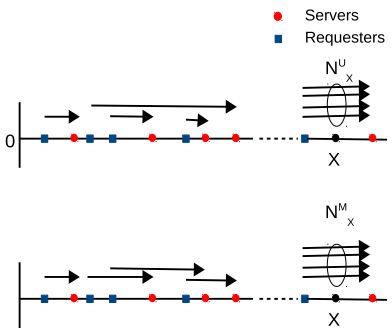

Fig. 1: Allocation of requesters to servers on the 1-D network. Top: UGS, Bottom: MTR allocation policy.

In this Section, we show the equivalence of UGS and MTR w.r.t the expected queue length distribution and the expected request distance.

Define N_x^U and N_x^M to be the number of requests that traverse point $X \in \mathcal{L}$ under UGS and in MTR respectively as shown in Figure 1. Let d_i^U and $d_i^M, i \in R$ be the corresponding distances between requester i and its allocated server under both policies. We have the following theorem.

Theorem 1. *The distributions N_x^U and N_x^M and hence the expected request distance are the same for both UGS and MTR.*

Proof. Due to the unidirectional nature of matching, both UGS and MTR have the same set of busy cycles. Busy cycles may be defined as groups of requests that are assigned to groups of servers with unallocated servers between groups. Consider any such busy cycle and let X be a point on it. Let $L_{x,R}^U$ and $L_{x,S}^U$ denote the number of requesters and number of servers to the left of point X in the busy cycle under UGS matching. Due to the unidirectional nature of matching, $N_x^U = L_{x,R}^U - L_{x,S}^U$. Similarly define $L_{x,R}^M$ and $L_{x,S}^M$ for MTR policy. Again due to unidirectional nature, $N_x^M = L_{x,R}^M - L_{x,S}^M$. As both matchings have the same set of busy cycles we have $L_{x,R}^U = L_{x,R}^M$ and $L_{x,S}^U = L_{x,S}^M$. Thus we get

$$N_x^U = N_x^M,\ X \in \mathbb{R}^+, \tag{3}$$

Thus applying Little's law we have

$$\mathbb{E}[d_i^U] = \mathbb{E}[d_i^M]. \tag{4}$$

□

Remark 1. *Note that Theorem 1 applies to any inter-server or inter-requester distance distribution. It also applies to the case where servers have capacities $c > 1$.*

V. Unidirectional Poisson Matching

In this section, we characterize request distance statistics under unidirectional policies when both requesters and servers are distributed according to two independent Poisson processes. Let λ be the requester density and μ the server density. We first analyze MTR as follows.

A. MTR

We first consider the case when each server has unit capacity i.e. $c = 1$.

1) Server capacity: $c = 1$: When server capacity is one, the service network can be modeled as an M/M/1 queue. An M/M/1 queue consists of a single server with customer arrivals described by a Poisson process and customer service times by an exponential distribution. Thus the distance between two consecutive requesters in the service network can be thought of as inter-arrival time between customers in an M/M/1 queue. Similarly the distance between consecutive servers corresponds to a customer service time. In the service network, random variable d_i corresponds to the sojourn time of the i^{th} customer in the M/M/1 queue and N_x denotes the number of customers in the queue at time instant X. If $\lambda < \mu$, then d_i converges to a random variable $d \sim \text{Expo}(\mu - \lambda)$ and N_x converges to a random variable $N \sim \text{Geo}(1 - \lambda/\mu)$. Thus we can evaluate request distance as

$$\mathbb{E}[d] = \frac{1}{\mu - \lambda}. \tag{5}$$

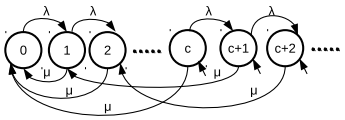

Fig. 2: State space diagram for server capacity c with $c > 1$.

2) Server capacity: $c > 1$: When server capacity is greater than one, the service network maps to a bulk service M/M/1 queue. A bulk service M/M/1 queue provides service to a group of c customers. The server serves a bulk of at most c customers whenever it becomes free. The service time for the group is exponentially distributed and customer arrivals are described by a Poisson process. Similar to the previous case, the distance between two consecutive requesters in the service network can be thought of as inter-arrival time between customers in the bulk service M/M/1 queue. However, the distance between two consecutive servers should be mapped to a bulk service time.

Having established an analogy between the service network and the bulk service M/M/1 queue, we now define the state space for the service network. Consider the definition of N_x as the number of requests that traverse point $X \in \mathcal{L}$ under MTR. In steady state, N_x converges to a random variable N provided $\lambda < c\mu$. Let π_k denote $\Pr[N = k]$ where $k = 0, 1, \ldots$. The state space diagram for such a system is shown in Figure 2. Thus we have the following balance equations similar to that of a bulk service M/M/1 queue [11]

$$\begin{aligned}(\lambda + \mu)\pi_k &= \mu\pi_{k+c} + \lambda\pi_{k-1},\ k \geq 1,\\ \lambda\pi_0 &= \mu(\pi_1 + \pi_2 + \ldots + \pi_c).\end{aligned} \tag{6}$$

By taking the z-transform and following the procedure in [11], we obtain the steady state probability vector $\pi = [\pi_0, \pi_1, \ldots]$. By applying Little's formula, we obtain the following expression for the request distance

$$\mathbb{E}[d] = \frac{r_0}{\lambda(1 - r_0)}, \tag{7}$$

where r_0 is the only root in the interval $(0, 1)$ of the following characteristic equation (with r as the variable)

$$\mu r^{c+1} - (\lambda + \mu)r + \lambda = 0. \tag{8}$$

B. UGS

When both requesters and servers are Poisson distributed and servers have unit capacity then the request distance in UGS has the same distribution as the busy cycle in the corresponding Last-Come-First-Served Preemptive-Resume (*LCFS-PR*) queue having the density function [1]

$$f_{\lambda,\mu}(x) = \frac{1}{x\sqrt{\rho}} e^{(\lambda+\mu)x} I_1(2x\sqrt{\lambda\mu}),\ x > 0, \tag{9}$$

Distribution	Parameters	$G_b(x)$	$\mathcal{B}(x)$
Exponential	μ: rate	$1-e^{-\mu x}$	$\frac{1}{\lambda}\left[1-e^{-\lambda x}\right]-\frac{1}{\lambda+\mu}\left[1-e^{-(\lambda+\mu)x}\right]$
Uniform	b : maximum value	$x/b, \quad 0 \le x \le b$	$\frac{1}{\lambda^2 b}\left[1-e^{-\lambda b}\right]-\frac{e^{-\lambda x}}{\lambda}$
Deterministic	d_0 : constant	$1, \quad x \ge d_0$	$\frac{e^{-\lambda d_0}-e^{-\lambda x}}{\lambda}$
Hyper -exponential	l: order p_j : phase probability μ_j : phase rate	$1-\sum_{j=1}^{l} p_j e^{-\mu_j x}$	$\frac{1}{\lambda}\left[1-e^{-\lambda x}\right]-\sum_{j=1}^{l}\frac{p_j}{\lambda+\mu_j}\left[1-e^{-(\lambda+\mu_j)x}\right]$
Weibull	k : shape, θ: scale	$1-e^{-(x/\theta)^k}$	$\frac{1}{\lambda}\left[1-e^{-\lambda x}\right]-\int_0^x e^{-(x/\theta)^k}e^{-\lambda x}dx$

TABLE I: Properties of specific inter-server distance distributions.

where $\rho=\lambda/\mu$ and I_1 is the modified Bessel function of the first kind. Thus the expected request distance is equivalent to the average busy cycle duration in LCFS-PR queue given by $1/(\mu-\lambda)$.

When servers have capacities $c>1$ then it is difficult to characterize the expected request distance explicitly with an LCFS preemptive approach. However, by Theorem 1, the expected request distance under UGS is same as that of MTR given by Equation (7).

VI. Unidirectional General Matching

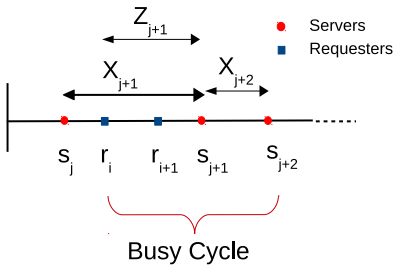

Fig. 3: Allocation of requesters to servers for renewal inter-server distance distribution.

In this Section, we derive expressions for the expected request distance when servers are distributed according to a renewal process. Assume each server has capacity $c=1$. Above spatial model can be mapped to a temporal M/G/1 queue where the first customer in a busy period receives different service from the others, [15]. To illustrate this consider a busy cycle: $[r_i, s_{j+2}]$ as shown in Figure 3. r_i can be thought of as the first customer in the busy period while r_{i+1} sees the system busy when arrives. Let s_j be the server placed on the immediate left of requester r_i. Clearly the service distance (spatial analogy of temporal service time) for requester r_i is described by the random variable Z_{j+1} while for second requester r_{i+1} by random variable X_{j+2}. When $\{X_j\}_{j\ge1}$ are distributed exponentially, both Z_{j+1} and X_{j+1} are exponentially distributed with the same density due to the memoryless property of the exponential distribution. However, this is not true when X_j is described by a renewal process. Denote $G_e(x)$ and $g_e(x)$ as the distribution and density functions for the random variable Z_{j+1}. Then the expected queue length and the expected sojourn time of a M/G/1 queue where the first customer in a busy period receives different service from the others is given by [15]

$$\overline{Q}=\frac{\lambda\alpha_e}{[1-\lambda(\alpha_b-\alpha_e)]}+\frac{\lambda^2(\sigma_e^2+\alpha_e^2-\sigma_b^2-\alpha_b^2)}{2[1-\lambda(\alpha_b-\alpha_e)]}+\frac{\lambda^2(\sigma_b^2+\alpha_b^2)}{2[1-\lambda\alpha_b]}, \tag{10}$$

$$\overline{d}=\frac{\overline{Q}}{\lambda}. \tag{11}$$

where α_b and α_e are the means associated with the service time distribution functions $G_b(x)$ and $G_e(x)$ respectively. Similarly σ_b^2 and σ_e^2 are the variances associated with $G_b(x)$ and $G_e(x)$ respectively. We assume $\lambda\alpha_b<1$ in the spatial setting.

A. Evaluation of the distribution function: $G_e(x)$

In this Section, we compute explicit expressions for the distribution function $G_e(x)$.

In the spatial version, when a busy period starts, requests other than the first request gets a service distance $X\sim G_b(x)$. Let $Y\sim$ Expo(λ) be the inter-request distance distribution. Using memoryless property of Y, the distribution function for the distance the first request in a busy period travels can be computed as

$$\begin{aligned}G_e(x)&=\Pr(X-Y<x|Y<X)\\&=\Pr(X-Y<x|X-Y>0)\\&=\frac{\Pr(X-Y<x)-\Pr(X-Y<0)}{1-\Pr(X-Y<0)}\\&=\frac{D_{XY}(x)-D_{XY}(0)}{1-D_{XY}(0)},\ x\ge0,\end{aligned} \tag{12}$$

where $D_{XY}(x)$ is the distribution of the random variable $X-Y$ (also known as difference distribution). $D_{XY}(x)$ can be expressed as

$$\begin{aligned}D_{XY}(x)&=\Pr(X-Y\le x)=\int_0^\infty \Pr(X-y\le x)\Pr(Y=y)dy\\&=\int_0^\infty G_b(x+y)\lambda e^{-\lambda y}dy=\int_x^\infty G_b(z)\lambda e^{-\lambda(z-x)}dz\\&=\lambda e^{\lambda x}\left[\int_0^\infty G_b(z)e^{-\lambda z}dz-\int_0^x G_b(z)e^{-\lambda z}dz\right]\\&=\lambda e^{\lambda x}\left[\mathcal{A}_{G_b}(\lambda)-\mathcal{B}(x)\right],\end{aligned} \tag{13}$$

where $\mathcal{A}$ is the Laplace Transform operator on the function G_b and $\mathcal{B}(x) = \int_0^x G_b(z)e^{-\lambda z}dz$. Clearly $\mathcal{B}(0) = 0$. Thus combining (12) and (13) yields

$$G_e(x) = \frac{\lambda e^{\lambda x}\left[\mathcal{A}_{G_b}(\lambda) - \mathcal{B}(x)\right] - \lambda \mathcal{A}_{G_b}(\lambda)}{1 - \lambda \mathcal{A}_{G_b}(\lambda)}, \tag{14}$$

$$g_e(x) = \frac{\lambda^2 e^{\lambda x}\left[\mathcal{A}_{G_b}(\lambda) - \mathcal{B}(x)\right] - \lambda G_b(x)}{1 - \lambda \mathcal{A}_{G_b}(\lambda)}, \tag{15}$$

$$\alpha_e = \int_0^\infty x g_e(x)dx,\ \sigma_e^2 = \left[\int_0^\infty x^2 g_e(x)dx\right] - \alpha_e^2. \tag{16}$$

1) ***$G_b(x) \sim$ Exponential(μ)****:* In this case, both X and Y are exponentially distributed. Thus the difference distribution is given by

$$D_{XY}(x) = 1 - \frac{\lambda}{\lambda+\mu}e^{-\mu x}, \text{when } x \geq 0 \tag{17}$$

Combining (12) and (17), we get

$$G_e(x) = \frac{1 - \frac{\lambda}{\lambda+\mu}e^{-\mu x} - 1 + \frac{\lambda}{\lambda+\mu}}{\frac{\lambda}{\lambda+\mu}} = 1 - e^{-\mu x}. \tag{18}$$

Thus we obtain $G_b(x) = G_e(x) \sim$ Exponential(μ). Putting $\alpha_b = \alpha_e = 1/\mu$, $\sigma_b^2 = \sigma_e^2 = 1/\mu^2$ and $\lambda/\mu = \rho$ in Equation (10) and (11), we get

$$\overline{Q} = \frac{\rho}{1-\rho}, \quad \overline{d} = \frac{1}{\mu - \lambda} \tag{19}$$

2) ***$G_b(x) \sim$ Uniform$(0, b)$****:* The c.d.f. for uniform distribution is

$$G_b(x) = \begin{cases} \frac{x}{b}, & 0 \leq x \leq b; \\ 1, & x > b, \end{cases} \tag{20}$$

where b is the uniform parameter. Thus we have

$$\begin{aligned} D_{XY}(x) &= \int_0^\infty G_b(x+y)\lambda e^{-\lambda y}dy \\ &= \left[\int_0^{b-x} \frac{x+y}{b}\lambda e^{-\lambda y}dy\right] + \left[\int_{b-x}^\infty 1\ \lambda e^{-\lambda y}dy\right] \\ &= \frac{\lambda x - e^{-\lambda(b-x)} + e^{-\lambda b}}{b\lambda + e^{-\lambda b} - 1} \end{aligned} \tag{21}$$

Taking $k_\lambda = 1/(b\lambda + e^{-\lambda b} - 1)$ and using Equation (12) we have

$$\begin{aligned} G_e(x) &= k_\lambda\left[\lambda x + e^{-\lambda b}(1 - e^{\lambda x})\right], \\ g_e(x) &= \lambda k_\lambda\left[1 - e^{-\lambda b}e^{\lambda x})\right]. \end{aligned} \tag{22}$$

Taking $\alpha_e = \int_0^b x g_e(x)dx$ and $\sigma_e^2 = [\int_0^b x^2 g_e(x)dx] - \alpha_e^2$ we have

$$\begin{aligned} \alpha_e &= \frac{b^2\lambda}{2}k_\lambda - \frac{1}{\lambda}, \\ \sigma_e^2 &= \frac{b^3\lambda}{3}k_\lambda - \frac{k_\lambda}{\lambda}\left[b(b\lambda - 2) + \frac{2}{\lambda}(1 - e^{-\lambda b})\right] - \alpha_e^2, \\ \alpha_b &= b/2, \quad \sigma_b^2 = b^2/12. \end{aligned} \tag{23}$$

We can substitute the values of $\alpha_e, \alpha_b, \sigma_e^2, \sigma_b^2$ in Equation (11) and obtain $\overline{d}$.

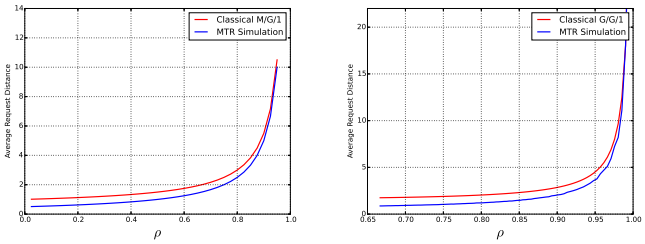

Fig. 4: Average request distance comparison for classical vs MTR through simulation for Poisson distributed *(Left)* and General distributed *(Right)* requesters.

3) ***$G_b(x) \sim$ Deterministic(d_0)****:* Another interesting scenario is when servers are equally spaced at a distance d_0 from each other i.e. when $G_b(x) \sim$ Deterministic(d_0). The c.d.f. for deterministic distribution is

$$G_b(x) = \begin{cases} 0, & 0 \leq x < d_0; \\ 1, & x \geq d_0, \end{cases} \tag{24}$$

where d_0 is the deterministic parameter. A similar analysis as that of uniform distribution yields

$$G_e(x) = c_\lambda\left[e^{-\lambda(d_0 - x)} - e^{\lambda d_0}\right];\ g_e(x) = \lambda c_\lambda\left[e^{-\lambda(d_0-x)}\right], \tag{25}$$

where $c_\lambda = 1/(1 - e^{-\lambda d_0})$. Thus we have

$$\begin{aligned} \alpha_e &= c_\lambda \frac{d_0\lambda + e^{-\lambda d_0} - 1}{\lambda}, \\ \sigma_e^2 &= \frac{c_\lambda}{\lambda}\left[d_0(d_0\lambda - 2) + \frac{2}{\lambda}(1 - e^{-\lambda d_0})\right] - \alpha_e^2, \\ \alpha_b &= d_0, \quad \sigma_b^2 = 0. \end{aligned} \tag{26}$$

We have also considered other distributions, such as Hyper-exponential and Weibull distributions. The expressions for $\mathcal{B}(x)$ are presented in Table I. We can evaluate $\mathcal{A}_{G_b}(\lambda)$ by setting $\mathcal{A}_{G_b}(\lambda) = B(\infty)$.

B. Comparision with classical M/G/1 systems

In this section, we compare the results obtained for the unidirectional general matching to that of a classical M/G/1 system. The expected sojourn time of a classical M/G/1 queue is given by [8]:

$$\mathbb{E}[d] = \alpha_b + \lambda\frac{\sigma_b^2 + \alpha_b^2}{2(1-\rho)}, \tag{27}$$

where $\rho = \lambda\alpha_b$. Thus we have the following theorem.

Theorem 2. *When $\rho \to 1$, the classical M/G/1 queue and the M/G/1 queue with exceptional service time have identical expected sojourn time given by the expression $\lambda\frac{\sigma_b^2+\alpha_b^2}{2(1-\rho)}$ under finite mean and variance assumption.*

Proof. When $\rho \to 1$, the expected sojourn time in the M/G/1 queue with exceptional service time obtained from Equation (11) simplifies to

$$\begin{aligned}\overline{d}_{spatial} &= 1 + \frac{\lambda}{2\alpha_e}(\sigma_e^2 + \alpha_e^2 - \sigma_b^2 - \alpha_b^2) + \lambda\frac{\sigma_b^2 + \alpha_b^2}{2(1-\rho)} \\ &\approx \lambda\frac{\sigma_b^2 + \alpha_b^2}{2(1-\rho)} \end{aligned} \tag{28}$$

Similarly when $\rho \to 1$, Equation (27) simplifies to $\overline{d}_{classical} \approx \lambda\frac{\sigma_b^2+\alpha_b^2}{2(1-\rho)}$. □

Thus we expect the average request distance to exhibit similar behavior.

We validate the correctness of Theorem 2 through simulation as follows. We consider a deterministic inter-server distance distribution with parameter $d_0 = 1$. We compare the request distance obtained for MTR through simulation to that of in Equation (27) for $|R| = |S| = 10^6$. The comparison is presented in Figure 4 (Left). It is evident that when $\rho \to 1$, the average request distance in both the systems are exact. However, for moderate traffic, the average request distance in the classical version serves as an upper bound to the spatial version. We get similar results when inter-server distances are distributed according to a uniform distribution.

Remark 2. *Note that if $G_e(x)$ does not have a closed form expression, for example when $G_b(x) \sim$Weibull, Equation* (27) *can be used to evaluate the expected request distance at heavy traffic.*

C. General request and server spatial distributions

Now, consider the case when the inter-requester distances have a general distribution with mean α_c and variance σ_c^2. We let $\rho = \alpha_b/\alpha_c$ with $\rho < 1$. As $\rho \to 1$, we can show that the behavior of MTR approaches that of the FCFS G/G/1 queue. It is known that the distribution of the waiting time in a G/G/1 queue will be approximately an exponentially distributed random variable and the mean sojourn time would then be given by [8]

$$\mathbb{E}[d] = \alpha_b + \frac{\sigma_b^2 + \sigma_c^2}{2\alpha_c(1-\rho)}. \tag{29}$$

We expect the average request distance to exhibit similar behavior.

We study the behavior of spatial G/G/1 system when $\rho \to 1$ as follows. We consider deterministic inter-server distances with parameter $d_0 = 1$. We compare the average request distance obtained from simulation to that of in Equation (29) for $|R| = |S| = 10^6$. We assume the inter-requester distance distribution to be uniform. The comparison is presented in Figure 4 (Right). It is evident that as $\rho \to 1$, the average request distance in both spatial system and temporal G/G/1 system are exact. Thus we have the following conjecture.

Conjecture 1. *At heavy traffic i.e. when $\rho \to 1$, the classical G/G/1 queue and the G/G/1 queue with exceptional service time have identical expected sojourn time and is given by the expression $\frac{\sigma_b^2+\sigma_c^2}{2\alpha_c(1-\rho)}$.*

VII. Bidirectional Allocation Policies

Both UGS and MTR minimize average request distance among all unidirectional policies. In this section we formulate the optimal request allocation policy for a bi-directional system with average request distance as the metric. Our objective is to find a function $\pi^* : R \to S$, such that

$$\begin{aligned} &\arg\min_{\pi} \sum_{i \in R} d_{\mathcal{L}}(r_i, s_{\pi(i)}) \\ &s.t. \quad \sum_{i \in R} \mathbb{1}_{\pi(i)=j} \leq c, \forall j \in S \end{aligned} \tag{30}$$

Algorithm 1 Optimal Request Assignment in 1-D Grid

Input: $r_1 \leq \cdots \leq r_{|R|}$; $s_1 \leq \cdots \leq s_{|S|}$
Output: The optimal assignment π
procedure OptCurr(r, s, π, i, j, k)
  if $i == j$ or $d_{\mathcal{L}}(r_i, s_j) \leq d_{\mathcal{L}}(r_i, s_k)$ **then**
    $\pi(i) = j$
    return π
  end if
  $\text{cost}_j = \text{ComputeCost}(\pi, i-1) + d_{\mathcal{L}}(r_i, s_j)$
  $\pi_{new}, s_u = \text{ShiftLeft}(\pi, i-1)$
  $\text{cost}_k = \text{ComputeCost}(\pi_{new}, i-1) + d_{\mathcal{L}}(r_i, s_k)$
  if $\text{cost}_j < \text{cost}_k$ **then**
    $\pi(i) = j$
    return π
  else
    $\pi_{new}(i) = k$
    if $s_k < r_i$ or $s_u ==$ NULL **then**
      return π_{new}
    else
      $l = \text{LastAllocated}(\pi_{new}, i-1)$
      return $\text{OptCurr}(r, s, \pi, i, k, l)$
    end if
  end if
end procedure
procedure OptimalAssignment(r, s)
  $\pi(i) = i,\ \forall i \in \{1, \cdots, |R|\}$
  if $|R| < |S|$ **then**
    for $i = 1, \cdots, |R|$ **do**
      if $i == 1$ **then**
        $\pi(i) = \text{FindNearest}(i)$
      else
        $k = \text{LastAllocated}(\pi, i-1)$
        $j = \text{NearestUnallocatedToRight}(i, k)$
        $\pi = \text{OptCurr}(r, s, \pi, i, j, k)$
      end if
    end for
  end if
  return π
end procedure

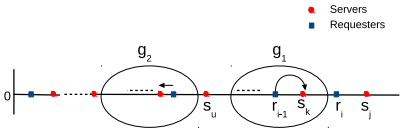

Fig. 5: An optimal assignment to request allocation problem and formation of groups.

Let $|R|$ and $|S|$ denote number of requests and number of servers respectively. Let $r_1 \leq r_2 \leq \cdots \leq r_i \leq \cdots \leq r_{|R|}$ be locations of requests and $s_1 \leq s_2 \leq \cdots \leq s_i \leq \cdots \leq s_{|S|}$ be locations of servers. We first focus on the case when $c = 1$. We consider the following two scenarios.

Case 1: $|R| = |S|$
When number of requesters and servers are equal, an optimal allocation strategy is given by the following theorem [6].

Theorem 3. *When $|R| = |S|$ an optimal assignment is obtained by the policy: $\pi(i) = i,\ \forall i \in \{1, \cdots, |R|\}$ i.e. allocating the i^{th} request to the i^{th} server and the average request distance is given by*

$$\mathbb{E}[d] = \frac{1}{|R|} \sum_{i=1}^{|R|} |s(i) - r(i)|. \tag{31}$$

Case 2: $|R| < |S|$ This is the case where there are fewer requesters than servers. In this case, the optimal assignment algorithm is presented in Algorithm 1. Before proceeding to the details of Algorithm 1, consider the following definition of group and a corresponding lemma.

Definition 1. *Let π be an assignment of requesters to servers. Under policy π, a group can be defined as a collection of consecutive allocated servers and requesters separated by an unallocated server on either end.*

Lemma 1. *Let g be a group with set of requesters R_g and set of servers S_g. Let both R_g and S_g be sorted in increasing order of their positions. An optimal assignment policy for the group π_g^{opt} is obtained by allocating the i^{th} request in R_g to the i^{th} server in S_g.*

Proof. Clearly, by definition of group, $|R_g| = |S_g|$. Thus applying Theorem 3 yields an optimal policy. □

For example, consider the formation of groups g_1 and g_2 due to an optimal assignment as shown in Figure 5. The groups g_1 and g_2 are separated by an unallocated server u. Clearly, there can not be any matching edges between g_1 and g_2 as it would violate the optimality criterion. We exploit this particular structure of the optimal allocation and propose an optimal allocation policy as follows.

Let us now discuss Algorithm 1 in detail. Without loss of generality (W.l.o.g), let us denote π_{i-1}^{opt} as the optimal allocation for the set of requesters: $\{1, 2, \cdots, i-1\}$. Let g_1 denote the group containing requester $i-1$ as shown in Figure 5. Assume that Algorithm 1 has correctly produced optimal output π_{i-1}^{opt} at the $i-1^{th}$ iteration. Let k be the last allocated server in g_1 i.e. the right most server in group g_1. We now wish to find an optimal allocation policy π_i^{opt} for the set of requesters: $\{1, 2, \cdots, i\}$. Let j be the nearest unallocated server to requester i s.t. $s_k < s_j$. If $d_{\mathcal{L}}(r_i, s_j) \leq d_{\mathcal{L}}(r_i, s_k)$ then we allocate server j to requester i. This might result in creation of a new group or extension of the existing group g_1. Let u be the nearest unallocated server immediately left of group g_1. The case when $d_{\mathcal{L}}(r_i, s_j) > d_{\mathcal{L}}(r_i, s_k)$, we shift the allocation of each requester in g_1 to one server left of their current allocations while allocating the first requester in g_1 to u forming a new policy π_{new}. We then compare the costs associated $\pi_{new} \cup \{\pi_{new}(i) = k\}$ to that of $\pi_{i-1}^{opt} \cup \{\pi_{i-1}^{opt}(i) = j\}$. If the policy π_{new} has lower cost, we repeat the process of shifting left until either is more expensive to perform a shift or there are no more unallocated servers to the left of group g_1.

A. Proof of correctness

It is trivial to check that the stopping criteria in Lines 4 and 11 in Algorithm 1 ensures optimality. Hence we shift our focus to the criterion on Line 16. Consider the definitions of i, j, k, π_{i-1}^{opt} and π_{new} introduced earlier. Let R_{g_1} and S_{g_1} be the set of requesters and the set of servers for group g_1. By Lemma 1 an optimal allocation for requesters R_{g_1} and servers $S_{g_1} \setminus \{k\} \cup \{u\}$ corresponds to allocating them sequentially in increasing order of their locations. The *ShiftLeft* operation in Line 9 ensures this order. Thus the value of cost_k in Line 10 is optimal w.r.t all other policies given i is assigned to k. Now consider the stopping criterion on Line 16. Clearly if $s_k < r_i$ then $s_{k-1} < s_k < r_i$. Thus we have $d_{\mathcal{L}}(r_i, s_{k-1}) \leq d_{\mathcal{L}}(r_i, s_k)$. Hence traversing further to the left of server k will not yield a better solution.

B. Time complexity

The modules *LastAllocated, NearestUnallocatedToRight* and *FindNearest* can be executed in $O(1)$ time by simple book keeping and preprocessing such as merging the sorted arrays r and s. The comparison between costs $\text{cost}_j, j \in \{j, j-1, \cdots u+1\}$ and $\text{cost}_k, k \in \{k, k-1, \cdots, u+1, u\}$ in Line 10 can be implemented through incurring a time complexity of $O(|g_{max}|)$ where g_{max} is the group of maximum cardinality in the optimal solution. The *ShiftLeft* operation in Line 9 incurs a time complexity of $O(|g_{max}|)$. Thus the time complexity for module *OptCurr* is $O(|g_{max}|^2)$ and the total complexity of Algorithm 1 is $O(|R||g_{max}|^2)$. In the worst case, $|g_{max}| = |R|$. Thus the total worst case time complexity of Algorithm 1 is $O(|R|^3)$. However, in the next Section, we show that the average group sizes in optimal solutions are very small and the expected running time of Algorithm 1 is conjectured to be $O(|R|)$, significantly less than what is described above.

Remark 3. *Note that, Algorithm 1 can easily be modified to handle the case where $c > 1$. In this case, we place c replicas of a server j on the location s_j. When $c \ll |R|$ the worst case running time complexity of Algorithm 1 remains the same.*

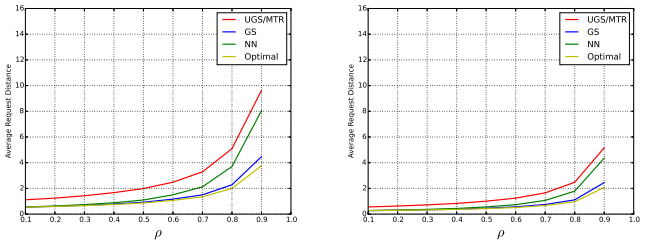

Fig. 6: Comparison of different policies under Exponential (Left) and Deterministic (Right) inter-server distance distributions.

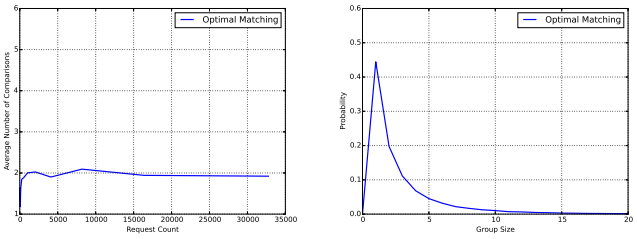

Fig. 7: Expected Complexity Analysis of Algorithm 1: Average Number of comparisons of module *OptCurr* (Left) and Probability distribution for size of a group (Right) in an optimal policy.

VIII. Performance Comparison

In this section, we compare the performance of various greedy allocation strategies along with the unidirectional policies to the optimal strategy.

A. Experimental Setup

In our experiments, we consider requesters to be poisson distributed with density $\lambda \in (0, 1)$. We consider various inter-server distance distributions with an expected value of 1. In particular, (i) for exponential distributions, the density is set to $\mu = 1$; (ii) for deterministic distributions, we assign parameter $d_0 = 1$. We consider a collection of 10^5 requesters and 10^5 servers, i.e. $|R| = |S| = 10^5$. Then, we assign requesters according to the UGS. Let $R_U \subseteq R$ be the set of requesters allocated under UGS matching. Clearly $|R_U| < |R|$. We then run optimal and other greedy policies on the set R_U and S. For each of the experiments, the expected request distance for the corresponding policy is averaged over 50 trials.

B. Comparison of different allocation policies

We first consider the case when both requesters and servers are distributed according to Poisson processes. From Figure 6 (Left), we observe that due to its directional nature UGS or MTR has a larger expected request distance as compared to other policies. At low loads i.e. when $\rho \ll 1$, the bidirectional greedy allocation policies perform similar to the optimal policy. But as $\rho \to 1$, the Nearest Neighbor policy perform worse and approach to that of UGS or MTR. The Gale-Shapley greedy allocation works reasonably well in both cases. Next we consider the case when the inter-server distance distribution is deterministic. We observe similar trends as that of the Poisson case as shown in Figure 6 (Right). However, under equal load, all the policies have smaller expected request distance as compared to their Poisson counterpart.

Remark 4. *Above discussion advocates for placing equidistant servers in a bidirectional system with Poisson distributed requesters to minimize expected request distance.*

C. Distribution of group size

In Section VII, we claimed that the expected running time of Algorithm 1 should be $O(|R|)$. We verify the claim through simulation.

M/M/1 arrivals: When servers and requesters are located according to Poisson processes with densities μ and λ with $\lambda < \mu$, there is a stationary regime where groups of requests are assigned to groups of servers with unallocated servers between groups. These would correspond to "busy periods" in temporal queuing processes. Note that the expected number of consecutively allocated servers in the unidirectional policy MTR is $1/(1-\rho)$ where $\rho = \lambda/\mu$. We conjecture this to be the case for optimal assignment as well. First we plot the average number of program variable comparisons inside module *OptCurr* in Algorithm 1 as shown in Figure 7 (left). Clearly, the average number of comparisons are small and do not depend on $|R|$. We also plot the probability distribution of group sizes for an optimal policy with requester density $\lambda = 0.5$ and server density $\mu = 1$ as shown in Figure 7 (Right). It is evident that the distribution is concentrated around 2 while the expression $1/(1-\rho) = 2$.

IX. Conclusion

In this paper, we introduced a queuing theoretic model for analyzing the behavior of unidirectional policies to allocate tasks to servers on the real line. We show the equivalence of UGS and MTR w.r.t the expected request distance and present results associated with the case when servers are Poisson distributed and with a general inter-server distance distribution. We also proposed an algorithm to obtain an optimal allocation policy in a bi-directional system and compared the performance of various greedy allocation strategies along with the unidirectional policies to that of optimal policy. Going further, we aim to extend our results in two ways. First we would like to extend the analysis for unidirectional policies to a two-dimensional geographic region. Second we would also consider analyzing the case when servers have different capacities.

References

[1] H. K. Abadi and B. Prabhakar. Stable Matchings in Metric Spaces: Modeling Real-World Preferences using Proximity. *arXiv:1710.05262*, 2017.

[2] P. Agarwal, A. Efrat, and M. Sharir. Vertical Decomposition of Shallow Levels in 3-Dimensional Arrangements and Its Applications. *SOCG*, 1995.

[3] R. Ahuja, T. Magnanti, and J. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice-Hall, Inc, 1993.

[4] A. Asadi, Q. Wang, and V. Mancuso. A Survey on Device-to-device Communication in Cellular Networks. In *IEEE Commun. Surv. Tut*, 2013.

[5] L. Atzori, A. Iera, and G. Morabito. The Internet of Things: A Survey. *Computer Networks*, 54(15):2787–2805, 2010.

[6] J. Bukac. Matching On a Line. *arXiv:1805.00214*, 2018.

[7] D. Gale and L. Shapley. College Admissions and Stability of Marriage. *Amer. Math. Monthly 69*, pages 9–15, 1962.

[8] D. Gross and C. Harris. Fundamentals of Queueing Theory. *Wiley Series in Probability and Statistics*, 1998.

[9] A. E. Holroyd, R. Pemantle, R. Peres, and O. Schramm. Poisson Matching. *Annales de l IHP Probabilites et Statistiques*, 45:266–287, 2009.

[10] F. Kingman. The Effect of Queue Discipline on Waiting Time Variance. *Math. Proc. Cambridge Phil. Soc.*, 58:163164, 1962.

[11] L. Kleinrock. *Queueing Systems*. John Wiley and Sons, 1976.

[12] M. Mezard and G. Parisi. The Euclidean Matching Problem. *J. Phys. France*, 49:2019–2025, 1988.

[13] J. Orlin. A Polynomial Time Primal Network Simplex Algorithm for Minimum Cost Flows. *Mathematical Programming*, 78:109–129, 1997.

[14] E. Stuart. Matching Methods for Causal Inference: a Review and a Look Forward. *Stat. Sci.*, 25:1–21, 2010.

[15] P. Welch. On a Generalized m/g/1 Queuing Process in Which The First Customer of Each Busy Period Receives Exceptional Service. *Operations Research*, 12:736–752, 1964.