Size-Aware Dispatching to Fluid Queues

Runhan Xie
University of California, Berkeley
runhan_xie@berkeley.edu

Esa Hyytiä
University of Iceland
esa@hi.is

Rhonda Righter
University of California, Berkeley
rrighter@berkeley.edu

ABSTRACT

We develop a fluid-flow model for routing problems, where fluid consists of different size particles and the task is to route the incoming fluid to n parallel servers using the size information in order to minimize the mean latency.

The problem corresponds to the dispatching problem of (discrete) jobs arriving according to a stochastic process. In the fluid model the problem reduces to finding an optimal path to empty the system in n-dimensional space. We use the calculus of variation to characterize the structure of optimal policies. Numerical examples shed further light on the fluid routing problem and the optimal control of large distributed service systems.

1. INTRODUCTION

The problem of routing jobs to parallel FCFS (first-come first-served) servers based on job sizes and server workloads in order to minimize, e.g., mean latency is important for many applications [1, 2, 3]. Though SRPT scheduling for each server is optimal in this context, FCFS is the dominant scheduling policy in practice; see, e.g., [4], [5].

Even under Markov assumptions with homogeneous FCFS servers, there is no simple characterization of the optimal dispatching policy when both the size of an arriving job and the workloads at all servers are observed [6]. We therefore consider stable dispatching systems at the fluid limit, so instead of individual jobs arriving according to a point process in time, we model the incoming jobs as a continuum, while keeping the notion of job sizes as a control parameter of the fluid flow process.

The value function, or total latency, of the fluid system approximates the value function of the underlying stochastic system, but is more computationally tractable for determining the optimal policy and evaluating heuristics.

The main contributions of this paper are as follows:

1. We formulate the fluid control problem that captures the dynamics of the original dispatching system when the number of jobs in the system is large. To our knowledge, we are the first to study size-aware flow control for fluid models of queueing systems.

2. We show that shorter jobs should be routed to shorter queues, which reduces the problem to an optimal path problem, and that the optimal path is invariant under appropriate scaling, which reduces the number of dimensions required to determine the path. We also observe that unbalancing the workloads is beneficial, especially under heavy load.

3. For some paths, and for the optimal path in heavy traffic, the fluid value function matches the value function for the original stochastic model.

4. We show, for two servers, that latency decreases as job sizes become more variable in convex ordering.

5. We develop new heuristics for the original model based on the optimal fluid policy when workloads are large, but are adjusted to avoid idling for small workloads.

The fluid control problem reduces to the problem of determining an optimal path to empty the system. We give integral expressions for costs of arbitrary paths. Fixed control actions correspond to straight line segments, for which closed-form expressions are available.

2. RELATED WORK

The dispatching problem has been extensively studied in the queueing theory literature, and many policies have been proposed and shown to be optimal under different assumptions on the available information (server states and job sizes). If the system is homogeneous and state-aware only, then Join-the-Shortest-Queue (JSQ) [7, 8] or Least-Work-Left (LWL) [9, 10, 11, 12] have been shown to be optimal. If the system is size-aware only, then Size-Interval-Task-Assignment (SITA) [13] is shown to be optimal [14]. With the routing history as the only state information, Round-Robin (RR) is optimal, and combining RR and SITA can out-perform RR or SITA alone [15, 16, 17].

The problem is much harder when the dispatching policy is both size- *and* state-aware. Sequential dispatching heuristics, that use both state and job-size information, and that route short jobs to short queues, were introduced in [18], and a particular one, called Dice, was shown numerically to have excellent performance. Another dispatching heuristic, CARD, was proposed in [19] and proven to be delay optimal in heavy traffic, and was shown to perform well in simulations. Our work provides theoretical support for heuristics like CARD and Dice that unbalance the loads and send shorter jobs to shorter queues.

Fluid models have been applied to study staffing problems e.g. [20, 21, 22], scheduling problems e.g. [21, 23, 24, 25], and dispatching problems e.g. [26, 27, 28, 29, 30]. These

models and their controls are often state aware, i.e., they use remaining fluid in the system for decision making, but are not job-size aware. We take a first step in studying size-aware fluid dispatching problems.

3. FLUID DISPATCHING MODEL

Each of n FCFS servers has service rate $1/n$, and work comes in as a fluid flow at rate $\lambda < 1$ comprising different size particles with known density $f(x)$, CDF $F(x)$, and CCDF $\bar{F}(x) = 1 - F(x)$, and with mean $\mathrm{E}[X] = 1$, so that $\rho = \lambda \mathrm{E}[X] = \lambda < 1$. The incoming flow is to be split among servers as a function of job sizes (or size of particles).

The state information, assumed known, is denoted by $\mathbf{u} = (u_1, \ldots, u_n)$, where u_i defines the backlog, or remaining workload, in queue i. Thus, with no arrivals, queue i would empty at time $u_i n$. A control policy (job dispatcher) π splits the incoming flow of jobs λ into n sub-flows, based on job sizes and the current backlog. That is, the policy chooses arbitrary subsets of job sizes from non-negative reals, $A_i(\mathbf{u})$, and routes jobs of size $x \in A_i(\mathbf{u})$ to server i. We will show that for an optimal policy, each subset comprises a single non-overlapping interval, and moreover, shorter jobs Should be routed to servers with smaller backlog.

The instantaneous rate of jobs, $\lambda_i(\mathbf{u})$, and the work flow, $\rho_i(\mathbf{u})$, routed to server i is then

$$\lambda_i(\mathbf{u}) := \lambda \int_{A_i(\mathbf{u})} f(x)\,dx;\ \rho_i(\mathbf{u}) := \lambda \int_{A_i(\mathbf{u})} x f(x)\,dx$$

where $\sum \lambda_i(\mathbf{u}) = \lambda = \rho = \sum \rho_i(\mathbf{u})$. For ease of notation, we often omit "$(\mathbf{u})$" and write λ_i and ρ_i, but we remind the reader that these rates are constantly changing according to the control policy.

Given our objective of minimizing latency, each arriving job assigned to server i incurs cost $u_i n$, so server i incurs waiting time cost at rate $\lambda_i u_i n$, and the total cost rate is

$$c(\mathbf{u}) := n \sum_i \lambda_i u_i. \tag{1}$$

For all meaningful policies $\mathbf{u} = 0$ is an absorbing state where no further costs are incurred, but there may be others. Let n_0 denote the number of empty servers, $n_0 := ||\{i : u_i = 0\}|| = \sum_i \mathbf{1}(u_i = 0)$. Whenever $n_0 \cdot (1/n) \geq \rho$, the empty servers can process all incoming jobs without increasing their backlogs, incurring no further costs, so we call any state with $n_0 \geq n\rho$ an absorbing state. We can assume, without loss of generality, that the system stops (is absorbed) in such a state. The question is how to move from the current state $\mathbf{u}$ to an absorbing state at minimal cost. It is easy to show that before absorption, incoming fluid should be routed to servers with zero fluid at rate $\rho_i = 1/n$, and all servers should work at rate $1/n$.

When fewer than n^* servers are empty, we have

$$\dot{u}_i = \rho_i - \frac{1}{n} =: -\nu_i,$$

where ν_i is queue i's instantaneous drainage rate, with $\rho_i = 1/n$ if $u_i = 0$. The cost (value function) of policy $\pi = (A_1(\mathbf{u}), ..., A_n(\mathbf{u}))$, with $\tau_\pi(\mathbf{u})$ denoting the time to absorption, is

$$v_\pi(\mathbf{u}) := \int_0^\infty c(\mathbf{u}_\pi(t))\,dt = \int_0^{\tau_\pi(\mathbf{u})} c(\mathbf{u}_\pi(t))\,dt \forall \mathbf{u}.$$

Remark 1 (Random Split) *With load-balancing random split (RND), the total cost incurred is*

$$C_i = \int_0^{\frac{n}{1-\rho}u_i} c_i(t)\,dt = n\int_0^{\frac{n}{1-\rho}u_i} \frac{1-\rho}{n} t\,dt = \frac{\rho n u_i^2}{2(1-\rho)},$$

$$v_{RND}(\mathbf{u}) = \sum_{i=1}^n C_i = \frac{\rho n}{2(1-\rho)}\left(u_1^2 + \ldots + u_n^2\right),$$

which, for $\rho < 1$ is finite for any job size distribution, regardless of $\mathrm{E}[X^2]$. Moreover, this matches the value function of the stochastic system; for each M/G/1-FCFS queue, $v(u) = \rho u^2/(2(1-\rho))$ [31, 32].

4. ANALYSIS WITH n SERVERS

We first show that the optimal policy is a dynamic SITA-type policy, which, for a given $\mathbf{u}$, splits job sizes into n non-overlapping intervals, sending shorter jobs to servers with less work. That is, with $u_1 \geq u_2 \geq \ldots \geq u_n$, the optimal policy is determined by size thresholds $0 = h_0(\mathbf{u}) \leq \cdots \leq h_n(\mathbf{u}) = \infty$, so that $A_{n+1-i}(\mathbf{u}) = [h_{i-1}(\mathbf{u}), h_i(\mathbf{u}))$.

Proposition 1 *The optimal policy assigns the shortest jobs to the server with the smallest backlog, the next size interval to the server with the second smallest backlog, and so on.*

PROOF. We prove this by contradiction. Suppose $u_i > u_j$ and the optimal policy routes jobs of size $(x, x+\delta)$ to server i and jobs of size $(y, y+\delta')$ to server j, where $x+\delta < y$. We choose δ and δ' so that

$$\mathrm{E}[X \cdot \mathcal{I}(x \leq X \leq x+\delta)] = \mathrm{E}[X \cdot \mathcal{I}(y \leq X \leq y+\delta')],$$

i.e., the loads contributed by the two streams of jobs are equal. Let $\lambda_x = \lambda P(x \leq X \leq x+\delta)$ and $\lambda_y = \lambda P(y \leq X \leq y+\delta')$, so $\eta := \lambda_x - \lambda_y > 0$. The cost rate due to these subflows of jobs is

$$c_{ij} = n(\lambda_x u_i + \lambda_y u_j).$$

Interchanging the corresponding job flows reduces the cost rate:

$$c_{ji} = n(\lambda_x u_i + \lambda_y u_j) = n(\lambda_y u_i + \lambda_x u_j + \eta(u_i - u_j)) < c_{ij},$$

which contradicts the optimality of the proposed policy. □

Corollary 1 (Structure of optimal policies) *The optimal dispatching policy can be characterized by $n-1$ appropriately chosen* state-dependent *thresholds $h_i(\mathbf{u})$. Therefore, the optimal policy from any initial point $\mathbf{u}$ is determined by a path from that point to an absorbing state.*

4.1 The Value Function in Terms of Paths

Note that there is a one-to-one correspondence between a policy π starting in some state $\mathbf{u}$ and a path $\mathbf{r}(s)$ from $\mathbf{u}$ to $(0, \ldots, 0)$, where s is the curve parameter. By time reversal, we can also consider paths from the origin to $\mathbf{u}$. We shall consider only paths $\mathbf{r}(s) = (r_1(s), \ldots, r_n(s))$ that are admissible by an appropriately chosen control policy. Then $\mathbf{r}(0) = (0, \ldots, 0)$, $\mathbf{r}(s_1) = \mathbf{u}$, and $\mathbf{r}'(s)$ defines the direction of the path at point s, which must coincide with the direction of the drainage rates $\boldsymbol{\nu} = (\nu_1, ..., \nu_n)$, $\boldsymbol{\nu}(s) = A(s)\,\mathbf{r}'(s)$, where the scalar function $A(s)$ adjusts the "speed" at curve point s to the actual drift. Before the path reaches an absorbing state (while $n_0 < n^*$), the total drift is $\nu_1 + \ldots \nu_n =$

$1-\rho$, giving $A(s) = (1-\rho)/(r_1'(s) + \ldots + r_n'(s))$, where $r_1'(s)+\ldots+r_n'(s) > 0$ as long as $\rho < 1$. Once the path reaches an absorbing state, $r_i'(s) = 0$ if $u_i = 0$, and $r_i'(s) = 1/n$ if $u_i > 0$. Then,

$$\boldsymbol{\nu}(s) = \frac{1-\rho}{r_1'(s) + \ldots + r_n'(s)} \left(r_1'(s), \quad \ldots \quad , r_n'(s)\right).$$

Thus, the (Euclidean) length of the drainage vector $||\boldsymbol{\nu}(s)||$ along the path $\mathbf{r}(s)$ is

$$||\boldsymbol{\nu}(s)|| = \frac{1-\rho}{r_1'(s) + \ldots + r_n'(s)} ||\mathbf{r}'(s)||.$$

Note that $\nu_i(s) < 0$ if server i is receiving more work than it can process (at the given time).

Lemma 1 (Value function) *The total accumulated cost along a work-conserving path $\mathbf{r}(s)$ is*

$$v_{\mathbf{r}}(\mathbf{u}) = \frac{1}{1-\rho} \int_0^{s_1} c(s)(r_1'(s) + \ldots + r_n'(s))\, ds, \qquad (2)$$

where $c(s)$ is the cost rate (per unit time) that depends on the respective control actions according to (1).

As $\mathbf{r}'(s)$ defines λ_i and ρ_i, the cost rate $c(s)$ depends solely on $\mathbf{r}(s)$ and $\mathbf{r}'(s)$, and therefore (2) can be evaluated (at least numerically if not in closed form) for any given path $\mathbf{r}(s)$.

4.2 Scale-free paths

Let us consider a family of "scale-free" paths that are obtained by *scaling* a reference path $\mathbf{r}_0(s)$,

$$\mathbf{r}(s) = \alpha \mathbf{r}_0(s),$$

where $\mathbf{r}_0(s)$ is some fixed path to $\mathbf{u} = \mathbf{r}_0(s_1)$ and $\alpha > 0$ is a free scaling parameter. Substituting $\mathbf{r}(s)$ into (2) reveals a quadratic relationship between the two value functions:

Lemma 2 *The value function of $\mathbf{r}(s) = \alpha \mathbf{r}_0(s)$ is*

$$v_{\mathbf{r}}(\alpha \mathbf{u}) = \alpha^2 v_{\mathbf{r}_0}(\mathbf{u}).$$

Lemma 2 gives the following two important structural properties of the optimal policy.

Corollary 2 *Scale-free paths are optimal.*

PROOF. By contradiction, suppose $\mathbf{r}(s)$ and $\tilde{\mathbf{r}}(s)$ are optimal paths for two states $\mathbf{u}$ and $\tilde{\mathbf{u}} = \alpha \mathbf{u}$ such that $v(\mathbf{u}) \neq \alpha^2 v(\mathbf{u}_0)$, so $\tilde{\mathbf{r}}(s) \neq \alpha\, \mathbf{r}(s)$. Then either $\mathbf{r}(s)$ or $\tilde{\mathbf{r}}(s)$ cannot be optimal. First, if $v(\alpha \mathbf{u}) > \alpha^2 v(\mathbf{u})$, then $\tilde{\mathbf{r}}(s) = \alpha \mathbf{r}(s)$ has lower cost than $\mathbf{r}(s)$. The other case is similar. □

Corollary 3 *The value function for any scale-free path scales quadratically in the Euclidean distance $|\mathbf{u}|$,*

$$v_{\mathbf{r}}(\mathbf{u}) = |\mathbf{u}|^2 w(\theta),$$

where $w(\theta)$ is the value function at unit distance in the direction θ in n-dimensional space.

Due to the scaling property, we need only determine $w(\theta)$ in an $n-1$ dimensional surface to obtain the optimal paths for every state in the n-dimensional space.

5. OPTIMAL PATHS WITH TWO SERVERS

For $n = 2$ servers, both with service rate $1/2$, we assume that $u_1 \geq u_2$, so that u_1 corresponds to the horizontal axis. Now it is also convenient to consider an alternate state representation, (x, y), where $x = u_1 + u_2$ is the total backlog and $y = u_1 - u_2$ is the imbalance. From the scale-free property (Corollary 2) the optimal path depends only on the relative imbalance, either captured by θ, the angle for the point (u_1, u_2), or the relative queue difference, $\hat{y} = y/x$. This means that if, for some point (x_0, y_0) with relative imbalance $\hat{y}_0$, the optimal $\hat{y}_0' = 0$, i.e., the relative imbalance should not change, then this will be true until the system empties. We call such a path a straight to the origin (STO) path. Because our paths are continuous, the path will visit a point with a given $\hat{y}$ either exactly once, or will maintain $\hat{y}$ on an STO path. Indeed, we have the following corollary.

Corollary 4 (Monotonicity of optimal paths) *For two servers, the optimal fluid path is monotonic in $\hat{y}$. That is, from any initial point (x, y), the optimal path will always increase or decrease the relative imbalance until the system empties or until some $\hat{y}_0$, after which it will follow an STO path.*

A consequence is that if the optimal policy empties queue 2 before queue 1, queue 2 will remain empty, even if $\rho > 1/2$. Similarly, if the optimal path moves to perfectly balanced queues, they will remain balanced.

For each point (x, y) any path is characterized by $y' = dy/dx$, which in turn, for an optimal path, is controlled by a single job-size threshold $h = h(x, y) = h(u_1, u_2)$: jobs shorter than h are assigned to queue 2 and the rest to queue 1. Let $g(h)$ denote the load due to particles (jobs) smaller than h, and recalling that $\lambda = \rho$, we have

$$g(h) := \rho \int_0^h z\, f(z)\, dz = \rho\, \mathrm{P}\{X \leq h\} \cdot E[X | X \leq h].$$

Then, while $u_2 > 0$ ($y < x$), u_2 changes at rate $g(h) - 1/2$, u_1 changes at rate $\rho - g(h) - 1/2$, x changes at rate $\rho - 1$, and

$$y' = \frac{(\rho - g(h) - 1/2) - (g(h) - 1/2)}{\rho - 1} = \frac{2g(h) - \rho}{1 - \rho},$$

so $g(h) = (\rho + (1-\rho)y')/2$. Because $0 \leq g(h) \leq \rho$, we have, for $y < x$, that $|y'| = |y'(x)| \leq \rho/(1-\rho)$. That is, as the load increases there is more room to maneuver in terms of changing the path direction, including having the ability to increase (temporarily) the load at one server if $\rho > 1/2$.

To capture total costs in terms of the control y', let $\Phi(y')$ denote the fraction of jobs forwarded to queue 2,

$$\Phi(y') := F\left[g^{-1}\left(\frac{\rho + (1-\rho)y'}{2}\right)\right].$$

The cost rate is then

$$2\rho(u_2\, \Phi(y') + u_1\, (1 - \Phi(y'))) = \rho(x + (1 - 2\Phi(y'))y),$$

and the total cost for an arbitrary path $y' = y'(x)$ (and $y(x)$) from initial state (x_1, y_1) is

$$v(x_1, y_1) = \frac{\rho}{1-\rho} \left[x_1^2/2 + \int_0^{x_1} (1 - 2\Phi(y'))y\, dx\right].$$

Because x_1^2 is independent of the path, an optimal policy minimizes

$$T = \int_0^{x_1} (1 - 2\Phi(y')y)\, dx.$$

As observed in Corollary 4, the optimal $\hat{y}$ will be monotone until some $\hat{y}_0 = y_0/x_0$, after which it will be constant until $(x, y) = (0, 0)$. For such an STO path, $y' = \hat{y}_0$ and

$$T_{\mathrm{STO}}(x_0, y_0) = x_0^2 \hat{y}_0 (1 - 2\Phi(\hat{y}_0)).$$

When $\hat{y} = 1$ ($u_2 = 0$), $\Phi(\hat{y}) = \Phi(1) = F(g^{-1}(1/2))$, and $T_{\mathrm{STO}}(x_0, x_0) = x_1^2(1 - 2\Phi(1))$. If $u_2 = 0$ and $\rho < 1/2$, then $\Phi(1) = 1$ and $T_{\mathrm{STO}}(x_0, x_0) = 0$.

Because F and g are nondecreasing, so is Φ, and $\Phi(\hat{y}) \geq \Phi(0) > 1/2$. Taking the derivative we have the following.

Proposition 2 *The cost of an STO path, from any point (x_0, y_0) directly to the origin, is decreasing in $\hat{y}_0 = y_0/x_0$, for any job-size distribution.*

We conjecture that the optimal $\hat{y}' \geq 0$ (the relative imbalance should increase for the optimal path). This is supported numerically as well as by the proposition above.

Let us now consider the optimal policy when $\rho \to 1$. Define the MWL (most work left) policy as the policy that routes all jobs (fluid) to the long queue until the short queue is empty; then short jobs are routed to the short queue such that the rate of fluid to the short queue is $\max\{\rho, 1/2\}$. It is not hard to derive the following, where $\bar{\Phi}(x) = 1 - \Phi(x)$.

Proposition 3 *The MWL value function, $v_{MWL}(\mathbf{u})$, is*

$$2\rho[2u_1u_2 + (2\rho - 1)u_2^2 + \frac{\rho\bar{\Phi}(1)}{1-\rho}(u_1 + u_2(2\rho - 1))^2]$$

and MWL is asymptotically optimal, with

$$(1 - \rho)v_{MWL}(\mathbf{u}) \to 2\rho\bar{\Phi}(1)(u_1 + u_2)^2 = 2\rho\bar{\Phi}(1)x^2.$$

The first two terms in the MWL value function correspond to the cost to empty the short queue, and the last term (the dominating term in heavy traffic) is the cost for the STO path that then empties the long queue once the short queue is empty.

For the original stochastic dispatching system with identical servers, the mean waiting time is $\mathrm{E}[W] = \mathrm{E}[v(X, 0)]$ [33], i.e., $(1-\rho)\mathrm{E}[W] \to 2\rho(1-\Phi(1))\mathrm{E}\left[X^2\right]$. This agrees with the exact analysis of the asymptotically optimal policy for the original stochastic system [19], suggesting that the far more tractable fluid value function may be a good approximation for the original value function even when $\rho < 1$.

We now consider the effect of job size variability on the dispatching cost. Intuitively, since the policy uses job-size information, we expect more variability to decrease costs, which is in fact the case. Recall that for two random variables X and Y, X is more variable than Y in the convex sense, $X \geq_{cx} Y$, if $Ef(X) \geq Ef(Y)$ for all convex functions f. The proof of the following is in the appendix.

Proposition 4 *$X \geq_{cx} Y \Rightarrow$ the cost is lower for any path when X is a random job size rather than Y, so the optimal cost is also lower.*

Now let us consider the extreme case, where all jobs have the same size, $X \equiv 1$. This is equivalent to allowing an arbitrary job-size distribution but job sizes are not observed or are not used in the dispatching algorithm. In this case, we can show that any work-conserving dispatching policy is optimal: the proof is in the appendix.

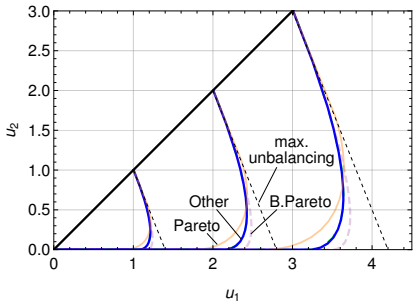


Figure 1: Optimal paths with exponential, uniform, and two Pareto distributions when $\rho = 0.7$.

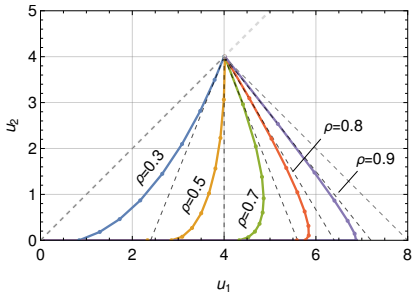


Figure 2: Optimal paths depending on ρ for exponential job sizes.

Proposition 5 *All size-unaware work-conserving dispatching policies are equally good.*

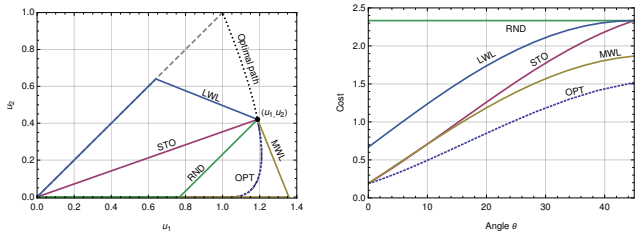
6. NUMERICAL EXAMPLES

6.1 Optimal fluid paths

Let us first study how the shape of the job size distribution affects the optimal trajectories. Figure 1 shows the optimal paths for exponential, uniform, and both bounded and normal Pareto distributions. The bounded Pareto distribution with $\alpha = 1$ is truncated to $[1/66, 6]$, so that the mean is approximately one, and the variance is approximately 2. The variance for the Pareto is infinite.

The scaling property is clearly visible, and the difference in job-size distributions matters more when the relative imbalance is larger. It also seems advantageous to empty one queue at maximal speed when backlogs are identical regardless of the job-size distribution. (The feasibility conditions are indicated by dashed lines on the graph.) The trajectories are consistent with our conjecture that the relative imbalance always increases along any optimal trajectory to the origin (or until one queue empties).

Figure 2 shows the optimal paths with exponentially distributed job sizes at different loads. As ρ increases, the optimal strategy unbalances the queues more, consistent with our heavy-traffic result.



(a) Paths for different policies (b) Cost at unit distance

Figure 3: Heuristic and optimal paths and costs with exponential job sizes and ρ= 0.7.

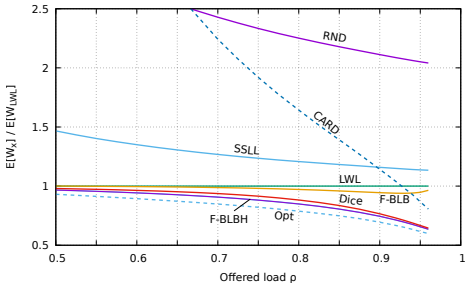


Figure 4: Performance with different heuristic policies relative to LWL as a function of ρ.

6.2 Heuristic policies

Next, we compare some heuristic routing policies with the optimal dispatching policy for fluid flow with exponential job sizes and $\rho = 0.7$.

Figure 3(a) depicts the paths for different heuristics and the optimal policy. The OPT path unbalances the workloads strongly, but not as much as MWL, which does it maximally. All of the chosen heuristics and the optimal policy have the scaling property, so the costs for the whole (u_1, u_2)-plane can be characterized by following an arc at unit distance. The resulting costs (value functions) are shown in Figure 3(b) as a function of the angle θ for points at unit distance, $||\mathbf{u}|| = 1$. The costs for all policies are increasing in θ (decreasing in the imbalance). When the fluid system starts with a large imbalance (small θ), RND and LWL have significantly larger costs than STO or MWL.

Let us next consider heuristics for the original stochastic dispatching system, using the value function of the fluid model as an approximation for the value function of the original system, and again assuming exponential job sizes.

We propose two heuristics that use the optimal fluid policy when backlogs are large, but adjusts to avoid idling when the short queue length is small. The F-BLB (Fluid with a buffer lower bound) policy uses the fluid optimal policy when the backlog of the short queue is greater than some threshold, or buffer lower bound, $u_2 > u_B$, and uses the LWL policy otherwise. Numerically we observe that $u_B = 3$ gives good results. The F-BLBH policy uses the fluid optimal policy when $u_2 > u_B$ and the arriving job size exceeds a threshold h_S. Jobs of size less than h_S are always sent to the short queue, and LWL is followed if $u_2 \leq u_B$. In the numerical examples, we use $(u_B, h_S) = (2, 1.5)$.

We also consider two sequential heuristics, DICE [18] and CARD [19]. DICE routes a job of size x to the queue with the least backlog if $u_2 + x < \tau$, where we use $\tau = 6$, i.e., the virtual buffer can accommodate six average-sized jobs. CARD uses two job-size thresholds to define "small," "medium," and "large" jobs, and it always routes small jobs to the short queue and large jobs to the long queue, and it routes medium jobs to the short (long) queue if the short queue length is below (above) some threshold. DICE generally tends to perform better than the sequential CARD heuristic [19], while requiring a single tuning parameter that works well for all loads, rather than CARD's three parameters that all need to be tuned for each load.

Finally, we consider the Short-to-Short and Long-to-Long (SSLL) heuristic. Jobs shorter than a threshold h_S are always routed to the shorter queue, and the rest go to the longer queue. We use the load balancing threshold, $h \approx 1.678$ (that is specific to job-size distribution). This is also known as (static) SITA with switch, which is an improvement on SITA, where short jobs are always sent to a given fixed queue, regardless of the relative backlogs.

Figure 4 shows the numerical results as a function of ρ for the heuristics and the optimal policy. The y-axis is scaled by the waiting time with LWL, as a base-line policy, i.e. the relative performance metric for policy π is the ratio,

$$\frac{\mathrm{E}[W_\pi]}{\mathrm{E}[W_{\mathrm{LWL}}]}.$$

The numerical results are based on average waiting time observed during relatively long simulation runs of 100-200 million arrivals. We can observe that F-BLB does a better job than LWL as load increases (LWL is optimal when the system is lightly loaded). CARD, while provably optimal as the load goes to 1, does not perform well at lower loads. On the other hand, the rather simple DICE and F-BLBH heuristics have near-optimal performance across all loads. We also note that if the initial loads are balanced, LWL is the individually optimal policy for strategic jobs that choose a queue upon arrival, so we can view the curve for the optimal policy in Figure 4 as a measure of the "cost of anarchy."

7. FUTURE WORK

Many of our fluid results hold for two servers. We would like to extend our results and heuristics to more than two servers. One possibility for a heurisitc approach, along the lines of the CARD heuristic, is to carefully control the fluid to the short queue, while dividing the remaining fluid to balance the loads of the other queues.

Heuristics based on the fluid model work well when there are a large number of jobs in all queues. We expect that they will work well for a model in which there are occasional large bursts of traffic, and queues have finite buffers, so that overflows are forced into shorter queues. We plan to explore the impact on the optimal fluid paths when queues have finite buffers, as well as the effect of bursty arrivals.

8. APPENDIX

8.1 Proof of Proposition 4

Consider two job-size distributions, $X \sim F_X$ and $Y \sim F_Y$, with similar subscripts for T, g, and g^{-1}. Because F, g, g^{-1},

and $z = z(y')$ are non-decreasing, we have

$$\begin{aligned} & g_X(h) \le g_Y(h) \quad \forall h \ge 0 \\ \Rightarrow \quad & g_X^{-1}(z) \ge g_Y^{-1}(z) \quad \forall z \in [0, \rho] \\ \Rightarrow \quad & \Phi_X(y') \ge \Phi_Y(y') \quad \forall y' \in [-\rho/(1-\rho), \rho/(1-\rho)] \\ \Rightarrow \quad & T_X \le T_Y . \end{aligned}$$

Also,

$$\begin{aligned} g_X(h) &= \rho \int_0^h x f_X(x) dx = \rho E[X|X \le h] P\{X \le h\} \\ &= \rho[1 - E[X|X > h] P\{X > h\}] \\ &= \rho[1 - E[(X-h)^+]]. \end{aligned}$$

Because $\rho[1 - (x-h)^+]$ is convex in x, the result follows.

8.2 Proof of Proposition 5

Here our admissible dispatching policies are the fraction of jobs, p, to be routed to server 2, depending on the backlogs, (u_1, u_2), and by work conserving, we assume that when $u_2 = 0$, the fraction of jobs routed to server 2 will be $p = 1/(2\lambda) = 1/(2\rho)$ (the maximal proportion to incur 0 costs, by making the load $= 1/2$). Given the fraction of jobs routed to server 2 is p,

$$y' = \frac{(2p-1)\rho}{1-\rho} \qquad \Rightarrow \qquad p = \frac{(1-\rho)y' + \rho}{2\rho},$$

and the cost rate is given by

$$c = 2\rho(pu_2 + (1-p)u_1) = 2\rho\left(\frac{x+y}{2} - py\right) = \rho x - (1-\rho)y'y$$

which is similar to our earlier cost rate, with p replacing Φ. The total cost rate is

$$v_{det}(x_1, y_1) = \frac{1}{1-\rho}\left[\rho x_1^2/2 - \int_0^{x_1} (1-\rho) y' y \, dx\right].$$

Thus, ignoring constants and x_1^2, which does not depend on the path, we obtain the following equivalent variational problem for the optimal path:

$$T = \int_0^{x_1} y' y \, dx = \max .$$

Integrating T by parts gives

$$\int_0^{x_1} y' y \, dx = \int_0^{y_1} y \, dy = \frac{y_1^2}{2},$$

i.e. *any* allowed (work-conserving) path is optimal.

9. REFERENCES

[1] W. Winston, "Optimality of the shortest line discipline," *Journal of applied probability*, vol. 14, no. 1, pp. 181–189, 1977.

[2] P. K. Johri, "Optimality of the shortest line discipline with state-dependent service rates," *European Journal of Operational Research*, vol. 41, no. 2, pp. 157–161, 1989.

[3] A. Ephremides, P. Varaiya, and J. Walrand, "A simple dynamic routing problem," *IEEE transactions on Automatic Control*, vol. 25, no. 4, pp. 690–693, 2003.

[4] S. Madni, M. Abd Latiff, M. Abdullahi, S. Abdulhamid, and M. Usman, "Performance comparison of heuristic algorithms for task scheduling in iaas cloud computing environment," *PLoS One*, vol. 12, no. 5, 2017.

[5] M. Tirmazi, A. Barker, N. Deng, M. Haque, Z. Qin, S. Hand, M. Harchol-Balter, and J. Wilkes, "Borg: the next generation," *Proceedings of the Fifteenth European Conference on Computer Systems, ACM*, 2020.

[6] E. Hyytiä, P. Jacko, and R. Righter, "Routing with too much information?" *Queueing Systems*, vol. 100, no. 3, pp. 441–443, 2022.

[7] R. R. Weber, "On the optimal assignment of customers to parallel servers," *Journal of Applied Probability*, vol. 15, no. 2, pp. 406–413, 1978.

[8] W. Winston, "Optimality of the shortest line discipline," *Journal of applied probability*, vol. 14, no. 1, pp. 181–189, 1977.

[9] D. Daley, "Certain optimality properties of the first-come first-served discipline for G/G/s queues," *Stochastic Processes and their Applications*, vol. 25, pp. 301–308, 1987.

[10] S. G. Foss, "Approximation of multichannel queueing systems," *Siberian Mathematical Journal*, vol. 21, no. 6, pp. 851–857, 1980.

[11] G. M. Koole, *On the optimality of FCFS for networks of multi-server queues.* Centre for Mathematics and Computer Science, 1992.

[12] O. T. Akgun, R. Righter, and R. Wolff, "Partial flexibility in routeing and scheduling," *Advances in Applied Probability*, vol. 45, no. 3, pp. 673–691, 2013.

[13] M. Harchol-Balter, M. Crovella, and C. Murta, "Task assignment in a distributed system: Improving performance by load unbalancing," in *Proceedings of SIGMETRICS*, vol. 98, 1998.

[14] H. Feng, V. Misra, and D. Rubenstein, "Optimal state-free, size-aware dispatching for heterogeneous M/G/-type systems," *Performance evaluation*, vol. 62, no. 1-4, pp. 475–492, 2005.

[15] A. Ephremides, P. Varaiya, and J. Walrand, "A simple dynamic routing problem," *IEEE transactions on Automatic Control*, vol. 25, no. 4, pp. 690–693, 1980.

[16] Z. Liu and D. Towsley, "Optimality of the round-robin routing policy," *Journal of applied probability*, vol. 31, no. 2, pp. 466–475, 1994.

[17] Z. Liu and R. Righter, "Optimal load balancing on distributed homogeneous unreliable processors," *Operations Research*, vol. 46, no. 4, pp. 563–573, 1998.

[18] E. Hyytiä and R. Righter, "On sequential dispatching policies," in *32nd International Telecommunication Networks and Application Conference (ITNAC'22)*, Wellington, New Zealand, Nov. 2022, pp. 1–6.

[19] R. Xie, I. Grosof, and Z. Scully, "Heavy-traffic optimal size-and state-aware dispatching," *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, vol. 8, no. 1, pp. 1–36, 2024.

[20] J. M. Harrison and A. Zeevi, "A method for staffing large call centers based on stochastic fluid models," *Manufacturing & Service Operations Management*, vol. 7, no. 1, pp. 20–36, 2005.

[21] W. Whitt, "Fluid models for multiserver queues with abandonments," *Operations research*, vol. 54, no. 1, pp. 37–54, 2006.

[22] ——, "Staffing a call center with uncertain arrival rate and absenteeism," *Production and operations management*, vol. 15, no. 1, pp. 88–102, 2006.

[23] R. Atar, C. Giat, and N. Shimkin, "The cμ/θ rule for many-server queues with abandonment," *Operations Research*, vol. 58, no. 5, pp. 1427–1439, 2010.

[24] Y. Chen and J. Dong, "Scheduling with service-time information: The power of two priority classes," *arXiv preprint arXiv:2105.10499*, 2021.

[25] N. Zychlinski, C. W. Chan, and J. Dong, "Managing queues with different resource requirements," *Operations Research*, vol. 71, no. 4, pp. 1387–1413, 2023.

[26] R. Talreja and W. Whitt, "Fluid models for overloaded multiclass many-server queueing systems with first-come, first-served routing," *Management Science*, vol. 54, no. 8, pp. 1513–1527, 2008.

[27] O. Perry and W. Whitt, "Responding to unexpected overloads in large-scale service systems," *Management Science*, vol. 55, no. 8, pp. 1353–1367, 2009.

[28] R. Stanojevic and R. Shorten, "Distributed dynamic speed scaling," in *2010 Proceedings IEEE INFOCOM*. IEEE, 2010, pp. 1–5.

[29] J. Huang, Y. Liu, R. Li, K. Li, J. An, Y. Bai, F. Yang, and G. Xie, "Optimal power allocation and load balancing for non-dedicated heterogeneous distributed embedded computing systems," *Journal of Parallel and Distributed Computing*, vol. 130, pp. 24–36, 2019.

[30] S. Hou, W. Ni, S. Chen, S. Zhao, B. Cheng, and J. Chen, "Real-time optimization of dynamic speed scaling for distributed data centers," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 3, pp. 2090–2103, 2020.

[31] E. Hyytiä, A. Penttinen, and S. Aalto, "Size-and state-aware dispatching problem with queue-specific job sizes," *European Journal of Operational Research*, vol. 217, no. 2, pp. 357–370, 2012.

[32] E. Hyytiä, R. Righter, and S. Aalto, "Task assignment in a heterogeneous server farm with switching delays and general energy-aware cost structure," *Performance Evaluation*, vol. 75, pp. 17–35, 2014.

[33] E. Hyytiä and R. Righter, "Towards the optimal dynamic size-aware dispatching," *Performance Evaluation*, vol. 164, p. 102396, 2024.