Optimal Control of General Dynamic Matching Systems

Mohammadreza Nazari
Alexander L. Stolyar

Department of Industrial and Systems Engineering, Lehigh University, USA

ISE Technical Report 16T-007

Optimal Control of General Dynamic Matching Systems

Mohammadreza Nazari
Lehigh University
200 West Packer Ave., Room 357
Bethlehem, PA 18015
mon314@lehigh.edu

Alexander L. Stolyar
Lehigh University
200 West Packer Ave., Room 484
Bethlehem, PA 18015
stolyar@lehigh.edu

Abstract

We consider a matching system with random arrivals of items of multiple types. The items wait in queues, one queue per each type, until they are matched with other items; after a matching is complete, the associated items leave the system. There exists a finite number of possible matchings, each producing a certain amount of "reward".

In this paper, we propose an optimal matching policy in the sense that it asymptotically maximizes the long-term average matching reward, while keeping the queues stable. This algorithm is constructed by applying an extended version of the greedy primal-dual (GPD) algorithm to a virtual system (with possibly negative queues). The proposed algorithm is real-time, it does not require any knowledge of the arrival rates; at any time it uses a simple rule, based on the current state of virtual queues.

Keywords: Dynamic Matching, Optimal Control, Greedy Primal-Dual Algorithm, EGPD Algorithm, Virtual Queues, Stability

1 Introduction

We consider a dynamic matching system with random arrivals. Items of different types arrive in the system according to a stochastic process and wait in their dedicated queues to be matched with items of other types. There exists a finite number of "possible matchings", each being a certain subset of item types. After each matching, the matched items leave the systems and a certain amount of reward is generated. The objective is to maximize the long-term average rewards, subject to the constraint that the queues of currently unmatched items remain stochastically stable. In this paper, we propose a dynamic matching policy and prove its asymptotic optimality. (In fact, the policy works for a more general objective, being a concave function of the long-term rates at which different matchings are used.)

The analysis of *static* matching has a large literature, cf. [9]. The *dynamic* version of the model has attracted a lot of attention recently, due to new applications such as Internet advertising [11], where the problem is to find appropriate matchings between the ad slots and the advertisers. Web portals as places for business and personal interactions is another important application; the problem in these portals (such as dating websites, employment portals, online games) is to match people with similar interests [3]. Matching problems also arise in systems with random arrival of customers and servers; e.g. in taxi allocation, where matched "items" are passengers and taxis [8]. (See, e.g., [4,6] for further examples.)

A special case of the matching system is where customers and servers are randomly arriving in the system and each server can be matched with one customer from a certain subset. This model, also known as the (stochastic) bipartite matching system, was initially studied by Caldentey et al. [6]. Majority of the previous research was focused on finding the stationary distribution [1, 2] and stability issues [3, 5, 10]. Bušić et al. [5] established the necessary and sufficient conditions for stability. They consider a bipartite matching model where by symmetry, one customer-server pair arrives at each time and analyze stability of various matching policies. One possible objective is to minimize the holding cost. Gurvich and Ward [7] study the problem of minimizing finite-horizon cumulative holding costs for a matching model. The problem of minimizing the long-term average holding cost for the bipartite matching system is studied by Bušić and Meyn [4].

They have shown that with known arrival rates (and some other conditions on the problem structure), a threshold-type policy is asymptotically optimal.

Similar to [7], we consider the matching problem in a more general setting in that there is no customer-server classification between the item types and the matchings may include items of multiple types (not necessarily two). Figure 1.1 shows an example of a matching system with 4 item types. The arrivals of each type $i \in \{1, \cdots, 4\}$ follow a Poisson process with rate α_i. There exist 3 possible matchings; e.g. $\langle 1, 2\rangle$ is a matching which matches one item of type 1 with one item of type 2. $\langle 2, 3, 4\rangle$ is another matching which matches one item of types 2, 3 and 4. A matching can only be applied if all contributing items are present in the system; and if it is applied, the contributing item instantaneously leave the system.

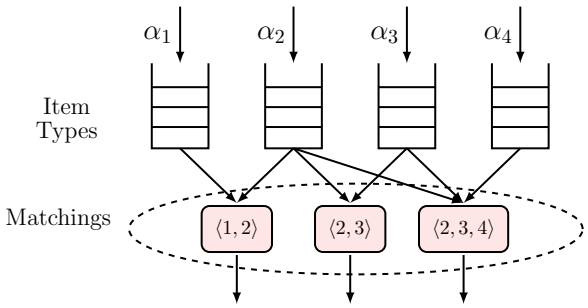


Figure 1.1: An example of the matching model

In this paper, we show that the reward-maximizing optimal control of the matching model can be obtained by putting it into a typical queueing network framework. Our algorithm uses a virtual system for making control decisions. Specifically, "in parallel" to the actual (physical) system, we will consider a virtual system, where any matching can be applied at any time, and the queues are allowed to be negative; the matchings in the virtual system are controlled by (an extended version of) the Greedy Primal-Dual (GPD) algorithm [12], which maximizes a queueing network utility subject to stability of the queues. Negative queues in the virtual system can be interpreted as the shortage of physical items of the corresponding types. The GPD algorithm in [12] does *not* allow negative queues, so it is insufficient for the control of our virtual system. That is why we introduce and study an extended version of it, which does allow negative queues. (The approach of using a virtual system to control the original one has been used before, e.g. in [13], but the virtual system employed in this paper is substantially different, primarily because it allows negative queues.)

The objective of EGPD algorithm is the average reward maximization; the theoretical results of this paper are for that objective. However, in many practical applications the objective is more general, namely it is to maximize "profit", which is the average reward minus the holding cost. In addition to the main results of this paper (on the average reward maximization), we will discuss heuristic approaches to improve the average profit by tuning EGPD parameters.

This paper is organized as follows. In Section 3, we introduce the Extended Greedy Primal-Dual (EGPD) algorithm for a general network and prove that this algorithm is asymptotically optimal. In Section 4, we formally define the matching problem and the virtual system. By embedding the virtual system into the more general framework of Section 3, we obtain an asymptotically optimal reward-maximizing dynamic algorithm for the matching problem. In Section 5, we informally discuss how average profit maximization can be addressed by tuning EGPD parameters. Finally, we evaluate the performance of the algorithm via numerical experiments in Section 6.

2 Basic Notation

We denote by $\mathbb{R}$, $\mathbb{R}_+$ and $\mathbb{R}_-$ the set of real, real non-negative and real non-positive numbers, respectively. $\mathbb{R}^N$, $\mathbb{R}^N_+$ and $\mathbb{R}^N_-$ are the corresponding N-dimensional vector spaces. A vector $x \in \mathbb{R}^N$ is often written as

$x = (x_n, n \in \mathcal{N})$, where $\mathcal{N} = \{1, 2, \cdots, N\}$. For two vectors $x, y \in \mathbb{R}^N$,

$$x \cdot y = \sum_{n=1}^{N} x_n y_n$$

is the scalar (dot) product. The standard Euclidean norm of x is denoted by $\|x\| = \sqrt{x \cdot x}$. The distance between point x and set $V \subseteq \mathbb{R}^N$ is denoted by $\rho(x, V) = \inf_{y \in V} \|x - y\|$.

For a vector function $f : \mathbb{R}_+ \to \mathbb{R}^N$ and a set $V \subseteq \mathbb{R}^N$, the convergence $f(t) \to V$ means that $\rho(f(t), V) \to 0$ as $t \to \infty$.

For differentiable functions $f : \mathbb{R} \to \mathbb{R}$ and $g : \mathbb{R}^N \to \mathbb{R}$, we use $f'(t)$ (or $(d/dt)f(t)$) to denote the derivative with respect to t and $\nabla g(x) = ((\partial/\partial x_n)g(x), n \in \mathcal{N})$ is the gradient of g at $x \in \mathbb{R}^N$.

For a set V and a real-valued function $g(v)$, $v \in V$,

$$\operatorname*{argmax}_{v \in V} g(v)$$

denotes a subset of vectors $v \in V$ which maximizes $g(v)$.

For $\xi, \eta \in \mathbb{R}$ and $\gamma \in \mathbb{R}_+$, we denote $\xi \wedge \eta = \min\{\xi, \eta\}$. Let $[\xi]_\gamma^+ = \xi$ if $\gamma > 0$ and $[\xi]_\gamma^+ = \max\{\xi, 0\}$ if $\gamma = 0$.

Abbreviation *a.e.* means *almost everywhere* with respect to Lebesgue measure.

3 A General Network Model and EGPD Algorithm

In this section, we introduce the "*Extended Greedy Primal-Dual*" (EGPD) algorithm for a general network model, which includes the matching system as a special case. This algorithm is a generalization of the GPD algorithm of [12] in the sense that there exists an additional set of network nodes, called "*free*" nodes. There is no constraint on the queue lengths of the free nodes; i.e. they can be either positive or negative. First, we formally define the model and the underlying optimization problem in Sections 3.1-3.3. Then, we define the EGPD algorithm in Section 3.4. We show that the "fluid scaled" version of the process converges (under some conditions on its parameters) to a random process with sample paths being what we define as EGPD-trajectories (Section 3.5). Finally, we prove asymptotic optimality of the algorithm in Section 3.6. [1]

3.1 The Model

Consider a network consisting of a finite set of nodes $\mathcal{N} = \{1, 2, \cdots, N\}$, $N \geq 1$. The nodes are of two different types: N_1 "*constrained*" nodes form the set $\mathcal{N}^c = \{1, 2, \cdots, N_1\}$ and $N_2 = N - N_1$ "*free*" nodes form $\mathcal{N}^f = \{N_1 + 1, N_1 + 2, \cdots, N\}$. (Either $\mathcal{N}^c$ or $\mathcal{N}^f$ is allowed to be empty.) There is a queue associated with each node, where we denote by $Q_n(t)$ the queue length of node $n \in \mathcal{N}$ at time t and we will denote $Q(t) = (Q_n(t), n \in \mathcal{N})$. The queue length of node $n \in \mathcal{N}^c$ is always *non-negative*, but node $n \in \mathcal{N}^f$ can have queue length of any sign.

The system operates in discrete time $t = 1, 2, \cdots$. (By convention, we identify an integer time t with unit time interval [t,t+1), which is usually referred as time slot t.) A finite number of "*controls*" is available, where we denote by K the set of controls. With activation of control $k \in K$ at time t, the following occurs sequentially:

(i) A random (bounded) non-negative integer number of items enters each node $n \in \mathcal{N}$. The vector of these quantities is equal in distribution to random vector $\lambda(k) = (\lambda_n(k), n \in \mathcal{N})$.

(ii) A certain (non-random) integer number $\mu_n(k) \geq 0$ of items is removed from queue $Q_n(t)$ and leaves the network. Queues in constrained nodes cannot go below zero, so if $Q_n(t) \leq \mu_n(k)$ for $n \in \mathcal{N}^c$, the entire content of $Q_n(t)$ will be removed.

[1] A reader interested only in the application for EGPD algorithm in the dynamic matching model may skip the analysis in Sections 3.5-3.6.

According to steps *(i)* and *(ii)*, the queue update rules for constrained and free nodes are defined as follows:

$$Q_n(t+1) = Q_n(t) + \lambda_n(k,t) - [(Q_n(t) + \lambda_n(k,t)) \wedge \mu_n(k)], \quad n \in \mathcal{N}^c \tag{3.1}$$

$$Q_n(t+1) = Q_n(t) + \lambda_n(k,t) - \mu_n(k), \quad n \in \mathcal{N}^f \tag{3.2}$$

where the sets of random vectors $\{\lambda(k,t),\ k \in K\}$ corresponding to different t are mutually independent and $\lambda(k,t)$ is equal in distribution to $\lambda(k)$.

3.2 System Rate Region

For each $k \in K$ and time t, consider random vector $b(k,t) = (b_n(k,t), n \in \mathcal{N})$ equal in distribution to $\lambda(k) - \mu(k)$. Indeed, $b(k,t)$ is equal to random vector of queue increments $Q(t+1) - Q(t)$ provided that control k is chosen at time t and $Q_n(t) \geq \mu_n(k)$ for all $n \in \mathcal{N}^c$. We call $b(k,t)$ the "*nominal increments*" of queues upon control k at time t. Let $k(t)$ denote the control chosen at time t by a given control policy. Informally speaking, the finite-dimensional convex compact rate region $V \subset \mathbb{R}^N$ is defined as the set of all possible long-term average values of $b(k(t),t)$, which can be induced by different control policies.

Formal definition of the rate region is as follows. For each $k \in K$, denote by $\bar{b}(k) = \mathbb{E}[\lambda(k) - \mu(k)]$ the drift of queue lengths upon control k. Suppose a probability distribution $\phi = (\phi_k, k \in K)$ (with $\phi_k \geq 0$ and $\sum_{k \in K} \phi_k = 1$) is fixed and consider the vector

$$v(\phi) = \sum_{k \in K} \phi_k \bar{b}(k). \tag{3.3}$$

If we interpret ϕ_k as the long-term average fraction of time slots when control k is chosen from the set of controls K, then $v(\phi)$ corresponds to the vector of long-term average drifts of $Q(t)$, assuming that the queues in the constrained nodes never hit zero. Then the system rate region V is defined as the set of all possible vectors $v(\phi)$ corresponding to all possible ϕ.

3.3 Underlying Optimization Problem

Consider an open convex set $\tilde{V} \subseteq \mathbb{R}^N$ such that $\tilde{V} \supseteq V$. Consider a concave continuously differentiable "*utility*" function $H : \tilde{V} \to \mathbb{R}$ and the following optimization problem:

$$\begin{aligned} \max_{v \in V} \quad & H(v) \\ s.t. \quad & v_n \in \mathbb{R}_-, \ \forall\, n \in \mathcal{N}^c \\ & v_n = 0, \quad \forall\, n \in \mathcal{N}^f, \end{aligned} \tag{3.4}$$

in which case we denote by $V^* \subseteq V$ the set of its optimal solutions.

The dual to optimization problem (3.4) is

$$\min_{(y_n \in \mathbb{R}_+, n \in \mathcal{N}^c), (y_n \in \mathbb{R}, n \in \mathcal{N}^f)} \left(\max_{v \in V} (H(v) - y \cdot v) \right), \tag{3.5}$$

and we denote by Q^* the closed convex set of optimal solutions $q^* \in \mathbb{R}_+^{N_1} \times \mathbb{R}^{N_2}$ of problem (3.5).

Assumption 3.1. *Optimization problem (3.4) has non-empty feasible rate region, i.e.*

$$\{v \in V : v_n \in \mathbb{R}_-, \forall n \in \mathcal{N}^c \text{ and } v_n = 0, \forall n \in \mathcal{N}^f\} \neq \varnothing. \tag{3.6}$$

In Section 3.4, we will introduce an algorithm, which is asymptotically optimal under the following stronger assumption:

Assumption 3.2. *For any subset $\bar{\mathcal{N}}^f \subseteq \mathcal{N}^f$, there exists $v \in V$ such that $v_n > 0$ for $n \in \bar{\mathcal{N}}^f$ and $v_n < 0$ for $n \notin \bar{\mathcal{N}}^f$.*

Assumption 3.2 means that there always exists a control policy which provides, simultaneously, a *strictly negative average drift* to all the constrained node queues and *non-zero average drifts* toward zero for all free node queues.

Note that under assumption 3.2, set Q^* is compact. Indeed, the optimal value of the problem (3.4) is equal to

$$H(v^*) = \max_{v \in V} (H(v) - q^* \cdot v) \tag{3.7}$$

for any $v^* \in V^*$ and any $q^* \in Q^*$. Set Q^* must be bounded, because otherwise, from assumption 3.2, there exist $v \in V$ such that $v_n < 0$ for all nodes with $q_n \geq 0$, and $v_n > 0$ for all nodes with $q_n < 0$. Then we can arbitrarily increase RHS of (3.7) by choosing $q^* \in Q^*$ with large $|q_n^*|$.

The problem that we are going to address is as follows. Let u denote the long-term average value of $b(k(t), t)$ under a given dynamic control policy, that is, a policy of choosing $k(t)$ depending on system state. We are interested in finding a dynamic control policy such that when optimization problem (3.4) is feasible, i.e. assumption 3.1 holds, the corresponding u is close to V^*, while the system queues remain stochastically stable.

3.4 Extended Greedy Primal-Dual Algorithm

Consider the following control policy:

Algorithm 1 EGPD algorithm for the general network model

At time $t = 1, 2, \cdots$, choose a control

$$k(t) \in \underset{k \in K}{\operatorname{argmax}} [\nabla H(X(t)) - \beta Q(t)] \cdot \bar{b}(k), \tag{3.8}$$

where $\beta > 0$ is a small parameter. Here $X(t)$ is the running average of $b(k(t), t)$, updated as follows:

$$X(t+1) = (1 - \beta) X(t) + \beta\, b(k(t), t) \tag{3.9}$$

and $Q(t)$ is updated according to (3.1) and (3.2).

The initial condition is $X(0) \in \tilde{V}$. Note that such initial condition and update rule (3.9) imply that $X(t) \in \tilde{V}$ for all $t \geq 0$. Hence the system evolution is well-defined for all $t \geq 0$, since the gradient and argmax in (3.8) are well-defined.

3.5 Asymptotic Regime and Fluid Limit

We define *EGPD-trajectory* as a pair of absolutely continuous functions $(x, q) = ((x(t), t \geq 0), (q(t), t \geq 0))$, each taking values in $\mathbb{R}^N$ and satisfying the following conditions:

(i) For all $t \geq 0$,

$$x(t) \in \tilde{V} \tag{3.10}$$

and for almost all $t \geq 0$,

$$x'(t) = v(t) - x(t) \tag{3.11}$$

where

$$v(t) \in \underset{v \in V}{\operatorname{argmax}} [\nabla H(x(t)) - q(t)] \cdot v. \tag{3.12}$$

(ii) We have

$$q_n(0) \geq 0,\ n \in \mathcal{N}^c \tag{3.13}$$

$$q_n(t) \geq 0,\ \forall t \geq 0,\ n \in \mathcal{N}^c \tag{3.14}$$

$$q_n'(t) = [v_n(t)]^+_{q_n(t)},\quad a.e.\ in\ t \geq 0,\ n \in \mathcal{N}^c \tag{3.15}$$

$$q_n'(t) = v_n(t),\quad a.e.\ in\ t \geq 0,\ n \in \mathcal{N}^f \tag{3.16}$$

Functions $x(t)$ and $q(t)$ are dynamically changing primal and dual variables, respectively, for problems (3.4) and (3.5), which arise as asymptotic limits of the fluid scaled version of the process as described next.

Consider a sequence of processes (X^β, Q^β), indexed by a parameter β, where $\beta \downarrow 0$ along a sequence $\mathcal{B} = \{\beta_j\}_{j=1}^\infty$ with $\beta_j > 0$ for all j. The initial state $(X^\beta(0), Q^\beta(0)) \in \tilde{V}$ is fixed for each $\beta \in \mathcal{B}$. (The processes and variables associated with a fixed parameter β will be supplied by superscript β.)

We need to augment the definition of the process. Let us assume $X^\beta(t)$ and $Q^\beta(t)$ are functions defined on $t \in \mathbb{R}_+$ and constant within each time slot $[l, l+1)$, $l = 0, 1, 2, \cdots$. Thus for each β, consider the (continuous-time) process $Z^\beta = (X^\beta, Q^\beta)$, where

$$X^\beta = (X^\beta(t) = (X_n^\beta(t), n \in \mathcal{N}), t \geq 0), \tag{3.17}$$

$$Q^\beta = (Q^\beta(t) = (Q_n^\beta(t), n \in \mathcal{N}), t \geq 0). \tag{3.18}$$

For each β,

$$z^\beta = (x^\beta, q^\beta) \tag{3.19}$$

is the fluid scaled version of process Z^β, obtained by

$$x^\beta = X^\beta(t/\beta), \quad q^\beta = \beta Q^\beta(t/\beta). \tag{3.20}$$

The following theorem is straightforward modification of Theorem 3 in [12], which we present without proof.

Theorem 3.3. *Consider a sequence of process $\{z^\beta\}$ with $\beta \downarrow 0$ along set $\mathcal{B}$. Each process is considered as a random element in the Skorohod space of RCLL ("right continuous with left limits") functions. Assume that $z^\beta(0) \to z(0)$, where $z(0)$ is a fixed vector in $\mathbb{R}^{2N}$ such that $X(0) \in \tilde{V}$. Then, the sequence $\{z^\beta\}$ is relatively compact and any weak limit of this sequence (i.e a process obtained as the weak limit of a subsequence of $\{z^\beta\}$) is a process with sample paths z being EGPD-trajectories (with initial state $z(0)$) with probability 1.*

3.6 Global Attraction Result

The following theorem is the main result of this section which shows the convergence of EGPD-trajectories to some point in the saddle set $V^* \times Q^*$.

Theorem 3.4. *Under assumption 3.1, the following holds:*

(i) For any EGPD-trajectory (x, q), as $t \to \infty$,

$$x(t) \to V^*, \tag{3.21}$$

$$q(t) \to q^*, \text{ for some } q^* \in Q^*. \tag{3.22}$$

(ii) Let compact subsets $V^\square \subset \tilde{V}$ and $Q^\square \subset \mathbb{R}_+^{N_1} \times \mathbb{R}^{N_2}$ be fixed. Then, the convergence

$$(x(t), q(t)) \to V^* \times Q^* \text{ as } t \to \infty \tag{3.23}$$

of EGPD-trajectories is uniform with respect to initial conditions $(x(0), q(0)) \in V^\square \times Q^\square$.

The proof of this theorem is similar to that of Theorem 2 in [12]. To save space, we will only give a proof of (3.21), and only for the special case when $x(0) \in V$ and $H(\cdot)$ is *strictly* concave. Consider a fixed EGPD-trajectory (x, q). The property

$$\rho(x(t), V) \leq \rho(x(0), V) e^{-t}, \; t \geq 0 \tag{3.24}$$

holds regardless of Assumptions 3.1 or 3.2 (cf. Lemma 20 in [12]). This shows that entire trajectory $(x(t), t \geq 0)$ is contained within V. This fact implies that $\nabla H(x(t))$ is uniformly bounded for all $t \geq 0$.

A time point $t \geq 0$ is called "*regular*" if conditions (3.10)-(3.12) are satisfied and proper derivatives $x'(t)$, $q'(t)$ and $f'(t)$ exist. Almost all t are regular.

Let us introduce the following function:

$$F(v, y) = H(v) - \frac{1}{2} \sum_{n \in \mathcal{N}} y_n^2, \quad v \in \tilde{V}, \quad y_n \in \mathbb{R}_+ \text{ for } n \in \mathcal{N}^c, \quad y_n \in \mathbb{R} \text{ for } n \in \mathcal{N}^f.$$

Lemma 3.5. *Trajectory $(q(t), t \geq 0)$ is uniformly bounded; i.e.*

$$\sup_{t\geq 0} \|q(t)\| < \infty \tag{3.25}$$

Proof. According to Assumption 3.2, we observe that the following holds for some fixed number $\delta > 0$. For any (regular) $t \geq 0$, there exists $\xi = (\xi_n, n \in \mathcal{N}) \in V$ such that for any n, $|\xi_n| \geq \delta$, $\xi_n > 0$ if $q_n < 0$, and $\xi_n < 0$ if $q_n \geq 0$. Pick such ξ for every regular t. Then we have:

$$\begin{aligned}
\frac{d}{dt}F(x(t), q(t)) &= [\nabla H(x(t)) - q(t)] \cdot v(t) - \nabla H(x(t)) \cdot x(t) \\
&\geq [\nabla H(x(t)) - q(t)] \cdot \xi - \nabla H(x(t)) \cdot x(t) \\
&= -\sum_{n\in\mathcal{N}} \xi_n q_n(t) + \nabla H(x(t)) \cdot (\xi - x(t)) \\
&\geq \delta \sum_{n\in\mathcal{N}} |q_n(t)| + \nabla H(x(t)) \cdot (\xi - x(t))
\end{aligned} \tag{3.26}$$

Since $\nabla H(x(t))$ and $x(t)$ are uniformly bounded and according to (3.26), we see that $(d/dt)F(x(t), q(t)) \geq \epsilon_1 > 0$ as long as $\|q(t)\| \geq C_1 > 0$, for some fixed constants ϵ_1 and C_1. This implies (since $H(x(t))$ is uniformly bounded) that $(d/dt)F(x(t), q(t)) \geq \epsilon_2 > 0$ as long as $F(x(t), q(t)) \leq C_2$, for some fixed constants ϵ_2 and C_2. This in turn implies that $F(x(t), q(t))$ is uniformly bounded below and as a result, $q(t)$ is uniformly bounded. □

Lemma 3.6. *For any EGPD-trajectory, at any regular time $t \geq 0$,*

$$\frac{d}{dt}F(x(t), q(t)) = \nabla H(x(t)) \cdot (v(t) - x(t)) - q(t) \cdot v(t) \tag{3.27}$$

and

$$v(t) \in \operatorname*{argmax}_{v\in V} \nabla H(x(t)) \cdot (v - x(t)) - q(t) \cdot v \tag{3.28}$$

Furthermore, if assumption 3.1 holds,

$$\frac{d}{dt}F(x(t), q(t)) \geq \nabla H(x(t)) \cdot (v^* - x(t)) \geq H(v^*) - H(x(t)). \tag{3.29}$$

Proof. Noting $q_n'(t) = v_n(t)$ and $v_n^* = 0$, for any $n \in \mathcal{N}^f$, every step of the proof is analogous to that of Lemma 3 in [12], so the proof of this lemma will not be provided here. □

Select an arbitrary point $q^* \in Q^*$ and associate it with the following function

$$F^*(v, y) = H^*(v) - \frac{1}{2}\sum_{n\in\mathcal{N}} (y_n - q_n^*)^2, \quad v \in \tilde{V}, \quad y_n \in \mathbb{R}_+ \text{ for } n \in \mathcal{N}^c, \quad y_n \in \mathbb{R} \text{ for } n \in \mathcal{N}^f,$$

where

$$H^*(v) = H(v) - q^* \cdot v$$

is the Lagrangian of problem (3.4) with the dual variable equal to $q^* \in Q^*$. Having strictly concave $H(\cdot)$ implies that $H^*(\cdot)$ is also a strictly concave function and

$$v^* = \operatorname*{argmax}_{v\in V} H^*(v) \tag{3.30}$$

is the unique optimal solution.

Lemma 3.7. *Consider $F^*(\cdot,\cdot)$ associated with an arbitrary $q^* \in Q^*$. Then for all (regular) $t \geq 0$,*

$$\frac{d}{dt}F^*(x(t),q(t)) \geq [\nabla H(x(t)) - q^*] \cdot (v(t) - x(t)) - (q(t) - q^*) \cdot v(t) \tag{3.31}$$

and

$$x(t) \in V \text{ implies } \frac{d}{dt}F^*(x(t),q(t)) \geq 0. \tag{3.32}$$

Proof. The proof is analogous to that of Lemma 5 in [12]. The only difference is the existence of free nodes, where we can easily validate this Lemma by using $q_n'(t) = v_n(t)$ and $v_n^* = 0$ for any $n \in \mathcal{N}^f$. □

Proof of Theorem 3.4. The convergence result 3.21 follows from an inequality that we first derive. For any (regular) $t \geq 0$,

$$\begin{aligned} &\tfrac{d}{dt}F^*(x(t),q(t)) \geq (\nabla H(x(t)) - q^*) \cdot (v(t) - x(t)) - (q(t) - q^*) \cdot v(t) \\ &= \nabla H^*(x(t)) \cdot (v^* - x(t)) - (q(t) - q^*) \cdot v^* + (\nabla H(x(t)) - q(t)) \cdot (v(t) - v^*) \qquad (3.33) \\ &= B_1(t) + B_2(t) + B_3(t), \qquad (3.34) \end{aligned}$$

where $B_i(t)$, $i \in \{1,2,3\}$ is the ith term in the RHS of (3.33). Since $x(t) \in V$ and v^* is maximizing $H^*(\cdot)$ over the compact set V, then we have

$$B_1(t) \geq H^*(v^*) - H^*(x(t)) \geq 0. \tag{3.35}$$

Thus, for any $\epsilon_1 > 0$, there exist sufficiently small $\epsilon_2 > 0$ such that

$$B_1(t) \geq \epsilon_2 \text{ as long as } \|x(t) - v^*\| \geq \epsilon_1. \tag{3.36}$$

Moreover,

$$B_2(t) = -(q(t) - q^*) \cdot v^* = -q(t) \cdot v^* = -\sum_{n \in \mathcal{N}^c} q_n(t)\, v_n^* \geq 0, \tag{3.37}$$

and

$$B_3(t) = (\nabla H(x(t)) - q(t)) \cdot (v(t) - v^*) \geq 0, \tag{3.38}$$

because $v(t)$ maximizes $\nabla H(x(t)) - q(t)) \cdot v$ over all $v \in V$.

Non-negativity of $B_1(\cdot)$, $B_2(\cdot)$ and $B_3(\cdot)$ along with Lipschitz continuity of $x(t)$ show that $\|x(t) - v^*\|$ must converge to zero, because otherwise $\int_0^\infty (d/dt)F^*(x(t),q(t)) = \infty$. (This is impossible, since $F^*(x(t),q(t))$ is a uniformly bounded function.) This proves (3.21). □

4 Optimal Control of the Matching System

The outline of this section is as follows: first, we formally define the "*physical*" matching system in Section 4.1. A key constraint of the physical system is that there should be sufficient number of physical items available in order to make a particular matching. Our proposed control algorithm will be based on the corresponding "virtual" system, where any matching can be used at any time, regardless of the physical items availability. Virtual system and its evolution along the physical system is studied in Sections 4.2 and 4.3.

Our main goal in this section is to find a dynamic matching algorithm which maximizes a concave utility function of the long-term average rewards due to different matchings, while keeping the system queues stochastically stable. By embedding the virtual system into the general model of Section 3, we obtain a matching algorithm in Section 4.5 which is a special case of the EGPD algorithm. Thereafter the asymptotic optimality of the proposed matching algorithm follows from that of the EGPD algorithm. Notice that this algorithm will only make the matching decisions based on the queues of the virtual system.

4.1 Definition of the (Physical) Matching System

Consider a matching system with I "*item types*" forming set $\mathcal{I} = \{1, \cdots, I\}$. The items of each type are randomly arriving to the system. For simplicity of the analysis, we let the arrivals of each type $i \in \mathcal{I}$ follow Poisson process with rate α_i. (In fact, it would suffice that the arrival process satisfies a functional strong law of large numbers. This allows, for example, dependence between arrivals of different types.)

Suppose set $\mathcal{J}$ is formed by J possible "*matchings*". We denote by $\mathcal{I}(j)$, the set of item types required for matching j. Similarly, we denote by $\mathcal{J}(i) \subseteq \mathcal{J}$, the set of possible matchings that type i items can join as a part of a matching. Let $\mu(j) = (\mu_i(j), i \in \mathcal{I})$, where $\mu_i(j)$ is the required number of type i items for activation of matching $j \in \mathcal{J}$. To simplify exposition, let us assume that $\mu_i(j) = 1$ for any $i \in \mathcal{I}(j)$; this assumption is not essential. (Clearly, $\mu_i(j) = 0$ for $i \notin \mathcal{I}(j)$.)

Without loss of generality, we do assume that the matching decisions are made only at the times of item arrivals into the system. As a result, we can consider the process as a discrete-time process of the system states at the arrival times, which from now on are indexed by $t = 1, 2, \cdots$.

There is a queue $Q_i(t)$ associated with item type $i \in \mathcal{I}$, which is formed by type i items waiting to be matched. At any given time t, any matching $j \in \mathcal{J}$ can be activated subject to the constraint that all the required items must be available in the system. With activation of matching $j \in \mathcal{J}$,

(i) Certain (real-valued) reward w_j is generated;

(ii) Number $\mu_i(j)$ of items is removed from the queues of the corresponding types i.

Let u_j be the long-term average reward generated by matching j, under a given control policy. We are interested in finding a dynamic matching policy, which maximizes a concave utility function $H(u_1, \cdots, u_J)$ subject to the constraint that all $Q_i(t)$ remain stochastically stable.

4.2 Virtual Matching System

We will propose a matching control algorithm in Section 4.5, which utilizes the state of a *virtual* matching system (instead of the physical system) for making matching decisions. The virtual matching system is defined as follows.

Consider a system with the same item types, set of matchings and arrival flows as in the physical system. It is only different in the sense that any matching can be activated at any time and the queues of the virtual system can be negative, as well as positive. We wish to emphasize that any matching can be activated in the virtual system, regardless of the state of physical system. The activated matchings in the virtual system become *actual* matchings either immediately, or later in time, depending on availability of physical items. The virtual matchings, until they become actual ones, are called "*incomplete*" matchings. Any incomplete matching will wait in a queue Q_0 to be "*complete*" with the future arrivals in FCFS order.

Denote by $\hat{Q}(t) = (\hat{Q}_i(t), i \in \mathcal{I})$ the vector of queue lengths in the virtual system. Positive $\hat{Q}_i(t)$ is the number of type i items in the physical system which are not associated with incomplete matchings, while the negative value shows the shortage of type i items for completing current incomplete matchings.

Figure 4.1 illustrates a physical matching system with two item types and one possible matching and its corresponding virtual system:

In this example, $\hat{Q}_2 = -1$ shows the shortage of one type 2 item for completion of incomplete matching $\langle 1, 2\rangle$, and $\hat{Q}_1 = 1$ is the number physical items that is not yet assigned to any incomplete matching.

4.3 Dynamics of the Physical and Virtual Matching Systems

Let us denote by $\lambda(t) = (\lambda_i(t), i \in \mathcal{I})$, the random vector of arrivals at time t such that only one of the components is 1 with probability $\alpha_i/(\sum_{i\in\mathcal{I}} \alpha_i)$, and all others are 0. Upon an arrival at time t, the following occurs sequentially:

(i) Both $Q(t)$ and $\hat{Q}(t)$ will be increased by $\lambda(t)$.

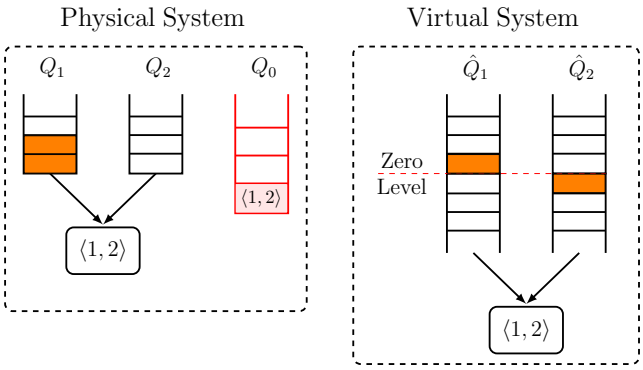


Figure 4.1: An example of the physical and virtual matching systems

(ii) If $Q(t) \geq \mu(j)$ holds for one of the incomplete matchings j, chosen in the FCFS order, then it will become actual matching. Upon completion, this matching j will be removed from Q_0 and $\mu(j)$ items will also be disposed from $Q(t)$.

(iii) A new matching is chosen in the virtual system. If it is matching j, then the virtual queues are updated as $\hat{Q}(t+1) = \hat{Q}(t) - \mu(j)$.

According to steps *(i)* and *(iii)* above, with activation of matching $j \in \mathcal{J}$ at time t, the queues of the virtual system are updated by the following rule:

$$\hat{Q}(t+1) = \hat{Q}(t) + \lambda(t) - \mu(j). \tag{4.1}$$

4.4 Mapping of the Virtual Matching System into EGPD Framework

Mapping the virtual system into the more general model of Section 3 is straightforward. Suppose the item types $\mathcal{I}$ are modelled as free nodes and let matchings $\mathcal{J}$ be controls K. Consider a slightly enhanced virtual system with one *constrained* node per each matching $j \in \mathcal{J}$. (These additional nodes are the *utility* nodes in the terminology of GPD algorithm [12]). From this point on, for convenience of the notations, we replace the set of indices of item types $\mathcal{I}$ with $\{J+1, \cdots J+I\}$ and denote by $\mathcal{I}^c = \{1, \cdots, J\}$ the set of all constrained nodes.

Recall that by matching j, certain amount of reward w_j is produced. Without loss of generality, we can assume that all $w_j < 0$. (Otherwise, we could replace each w_j with $w_j - c$ for some large $c > 0$, and replace $H(u_1, \ldots, u_J)$ with $H(u_1 + c, \ldots, u_J + c)$). Then by convention, let $\mu_j(j) = -w_j$ for $j \in \mathcal{I}^c$, $\mu_i(j) = 0$ for $i \in \mathcal{I}^c \setminus \{j\}$ and $\lambda_i(t) = 0$ for any $i \in \mathcal{I}^c$ at all t.

For any $i \in \mathcal{I}^c$, using (3.1) with an arbitrary initial value of $Q_i(0)$, the queue length will decrease until it hits 0 and then it will remain at 0. This in turn implies the stability of the queues associated with constrained nodes, so let us further assume that for constrained nodes $Q_i(0) = 0$.

Note that the queues of the physical system are stochastically stable as long as the virtual ones are.

For matching j, consider random vector $b(j,t)$ equal to $(\lambda_i(t) - \mu_i(j),\ i \in \mathcal{I}^c \cup \mathcal{I})$ and let $\bar{b}(j) = \mathbb{E}b(j(t), t)$. Note that, for constrained node $i \in \mathcal{I}^c$, $b_i(j,t) = b_i(j) = \bar{b}_i(j) \leq 0$ for all t. Clearly $\bar{b}_j(j) = w_j$ and $\bar{b}_i(j) = 0$ for $i \in \mathcal{I}^c \setminus \{j\}$.

Consider the rate region $V \subset \mathbb{R}^{J+I}$ as it is defined in Section 3.2 and let $\tilde{V}$ be a fixed open convex set containing V. Suppose $H(v)$ be a continuously differentiable concave utility function defined on $\tilde{V}$, which depends only on $(v_1, \cdots, v_J)$, and not on $(v_{J+1}, \cdots, v_{J+I})$.

Let u denote the long-term average value of $b(j(t), t)$ under a given matching policy. In the next section, we

present a specialization of the EGPD algorithm, which asymptotically solves optimization problem

$$\begin{array}{ll} \max\limits_{v\in V} & H(v) \\ s.t. & v_i = 0\ , \forall\, i \in \mathcal{I} \end{array} \tag{4.2}$$

in the sense that it dynamically decides about the activation of matchings such that vector u gets close to V^*, as $t \to \infty$. (Notice that for $v^* \in V^*$, we automatically have $v_i^* \le 0$ for any $i \in \mathcal{I}^c$.)

4.5 EGPD Algorithm for The Matching Problem

The specialization of EGPD algorithm to the matching problem is as follows.

Algorithm 2 EGPD algorithm for the Matching Problem

At each time $t = 1, 2, \cdots$, activate matching

$$j(t) \in \operatorname*{argmax}_{j\in\mathcal{J}} \left[(\partial H(X(t))/\partial x_j)\ w_j + \sum_{i\in\mathcal{I}} \beta \hat{Q}_i(t)\ \mu_i(j) \right], \tag{4.3}$$

where running averages $X_i(t)$ of the values $b_i(j(t))$ for constrained nodes are updated as follows:

$$X_{j(t)}(t+1) = (1-\beta)X_{j(t)}(t) + \beta\, w_{j(t)}, \tag{4.4}$$

$$X_i(t+1) = (1-\beta)X_i(t), \qquad i \neq j(t), \tag{4.5}$$

with $\beta > 0$ being a small parameter, and $\hat{Q}_i(t)$ is updated according to rule (4.1) for all $i \in \mathcal{I}$.

Remark 4.1. Feasibility of the underlying optimization problem. Note that the underlying optimization problem may or may not be feasible. It is always feasible, if for every type i there is a matching consisting of a single item i, and if an "empty" matching $\langle\varnothing\rangle$ is also available. By choosing a single matching, one item of the associated type will be removed from the system at a certain (possibly negative) reward. Matching $\langle\varnothing\rangle$ means "no change" and it has zero reward.

Remark 4.2. If we augment the original matching system by single matchings and $\langle\varnothing\rangle$ and there exist some positive arrival rates of items of each type, Assumption 3.2 is easily seen to hold. This assumption is sufficient to establish asymptotic optimality of the EGPD algorithm.

Remark 4.3. Assumption that the items arrive into the system one at a time is not essential, and the arrivals may enter the system in batches consisting of items of one or several types. In this case, our algorithm can be applied repeatedly at the times of arrival, until it keeps activating "non-empty" matchings.

Remark 4.4. Probabilistic matching. The EGPD algorithm easily generalizes to the case when a matching j completes not with certainty, but with some probability $p_j \le 1$. This requires that the probabilities p_j are known; if they are not, they could be estimated dynamically.

5 EGPD in a System with both Matching Rewards and Holding Costs

EGPD algorithm is proved to be asymptotically optimal for the reward maximization problem. In practical systems, the objective may be more general, namely maximizing the average "profit" defined as average reward minus average queue holding cost. We now informally discuss how EGPD can be used to achieve better profit in the system (even though it is not specifically designed for that).

For the purposes of the discussion below, we assume linear holding costs with rate vector $c = (c_i,\ i \in \mathcal{I})$; that is the average holding cost over interval $[0, T]$ is

$$\frac{1}{T}\int_0^T c \cdot Q(t)dt. \tag{5.1}$$

Suppose the arrival rates scaled up by a factor $r > 0$. This simply speeds up the process r times, so that the average reward increases r times, while the holding cost remains (roughly) same. Thus for systems with "high" arrival rates, the rewards dominate the profit objective and we expect the average profit obtained by the EGPD algorithm to be "close" to the optimal one.

When the average values of reward and holding cost are on the same scale, the EGPD parameter settings can be used to control the tradeoff between them, thus potentially improving the average profit. We now discuss the effects of changing parameter β and scaling of queue lengths by additional parameters γ_i.

Effect of parameter β. In order for $\beta\hat{Q}$ to be "close" to some $q^* \in Q^*$, parameter β should be small. Then $|\hat{Q}_i(t)|$, $i \in \mathcal{I}$ would be large (of the order of $1/\beta$). To see how this affects the holding cost, consider two cases:

(i) If $\hat{Q}_i(t) \geq 0$ for some $i \in \mathcal{I}$, then $Q_i(t)$ will also be large (of the order of at least $1/\beta$) since the inequality $Q_i(t) \geq \hat{Q}_i(t)$ holds for all $i \in \mathcal{I}$ at all t.

(ii) $\hat{Q}_i(t) < 0$ has an indirect impact on the holding cost. In particular, large $|\hat{Q}_i(t)|$ in this case would imply more incomplete matchings. This subsequently results in a higher holding cost.

Therefore, the smaller parameter β, the more matching reward is gained by using the EGPD algorithm, but at the higher holding cost. On the contrary, large values of β reduces the "precision" of the algorithm in terms of reward maximization, while it generally decreases the queues and the holding cost. Therefore, the value of parameter β should be chosen, very informally speaking, "as large as possible, but not larger".

Effect of additional queue scaling. Consider arbitrary positive weights γ_i, $i \in \mathcal{I}$. All the results for the EGPD algorithm hold if we use more general rule

$$j(t) \in \operatorname*{argmax}_{j \in \mathcal{J}} \left[(\partial H(X(t))/\partial x_j)\, w_j + \sum_{i \in \mathcal{I}} \beta\, \gamma_i\, \hat{Q}_i(t)\, \mu_i(j) \right]. \tag{5.2}$$

instead of (4.3). In this case, it is the weighted vector $(\gamma_i \hat{Q}_i(t),\, i \in \mathcal{I})$ (not $\hat{Q}(t)$ itself) that is getting close to an optimal point q^*. This property might be used to reduce the holding cost by giving higher weights to more "expensive" queues, thus making them relatively smaller.

Finally, there is a flexibility in choosing which incomplete matching to complete; here we may pick the incomplete matchings with higher associated holding cost to be completed first.

6 Simulation Results

In this section, we evaluate the performance of EGPD algorithm via simulation results. We present two sample experiments. We start with an experiment to illustrate the behavior and performance of the EGPD algorithm for a simple matching system. The second experiment is constructed to study the average profit under the EGPD algorithm and compare it with that of the the existing algorithms in the literature.

6.1 Experiment 1: Average Reward Maximization

We carry out our first simulation experiment for the matching system example introduced in Section 1. To ensure feasibility, we extend the set of possible matchings as follows (See Remark 4.1):

$$\mathcal{J} = \{\langle\varnothing\rangle, \langle 1\rangle, \langle 2\rangle, \langle 3\rangle, \langle 4\rangle, \langle 1,2\rangle, \langle 2,3\rangle, \langle 2,3,4\rangle\}.$$

We have a linear utility function, namely the average reward (i.e., the sum of average rewards due to different matchings). The vector of arrivals rates is $\alpha = (1.2, 1.5, 2, 0.8)$. Reward vector is $w = (0, -1, -1, 1, 2, 5, 4, 7)$ where its jth component corresponds to jth element of set $\mathcal{J}$. We use parameter $\beta = 0.01$. The vector of holding cost rates is $c = (1, 2, 3, 4)$.

We will apply EGPD algorithm, in which case rule (4.3) in the algorithm simplifies to

$$j(t) \in \underset{j \in \mathcal{J}}{\operatorname{argmax}} \left(w_j + \sum_{i \in \mathcal{I}} \beta \hat{Q}_i \mu_i(j) \right). \tag{6.1}$$

A. Performance of EGPD algorithm. Figure 6.1 shows the queue trajectories of the virtual and physical systems under the EGPD algorithm. We observe that all queues are quickly "converging". Nearly all type 2 and 4 items are matched right after they arrive the system, while there exist around 100 items of types 1 and 3.

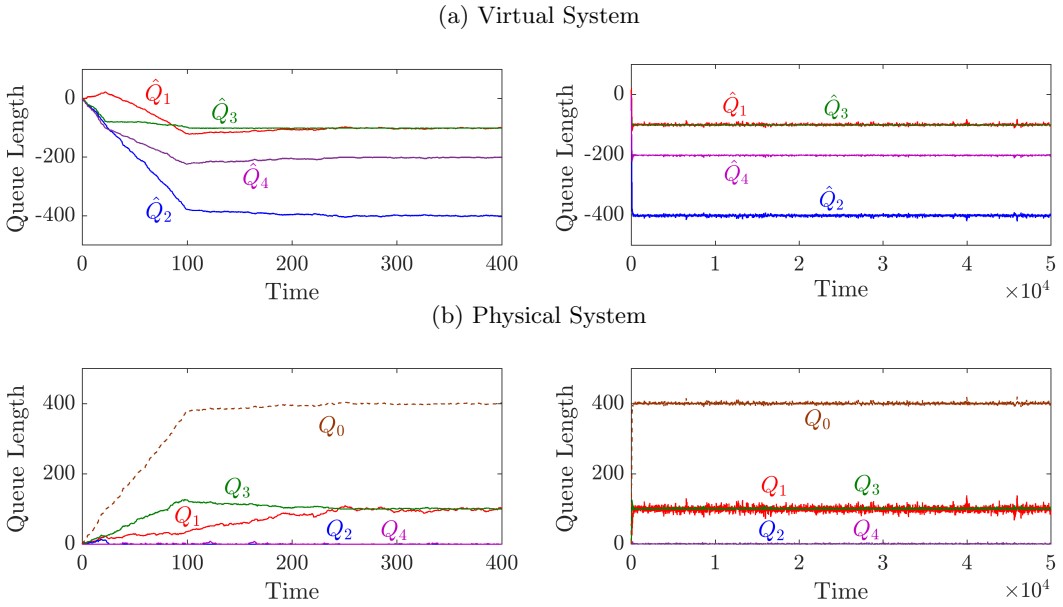


Figure 6.1: Queue trajectories of the virtual and physical systems under EGPD algorithm.

The rates at which matchings are performed under EGPD algorithm are provided in table 6.1, which shows that these rates are close to the optimal ones. Therefore, as we expected, this algorithm yields near optimal performance for small β.

Table 6.1: Matching rates: Optimal vs. EGPD. (Runtime=30000)

Method	Matchings						
	$\langle 1 \rangle$	$\langle 2 \rangle$	$\langle 3 \rangle$	$\langle 4 \rangle$	$\langle 1,2 \rangle$	$\langle 2,3 \rangle$	$\langle 2,3,4 \rangle$
EGPD	0	0	1.69345	0.4829	1.1924	0	0.31075
Optimal	0	0	1.70005	0.49995	1.2001	0	0.29975

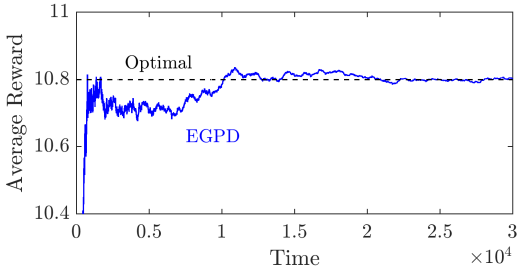


Figure 6.2: Average matching reward under the EGPD algorithm.

Figure 6.2 demonstrates the average matching reward per unit time. We have calculated the optimal average reward (by solving the linear program) which is equal to 10.8, and plotted it on the figure. As clear from the graph, the running average reward under EGPD algorithm is getting very close to optimal objective value and this convergence is rather "fast".

B. Changes in arrival process. An important robustness issue is how EGPD algorithm responds to the changes in the arrival process. In the following experiment, the arrival rates are changed to be $\alpha = (1.8, 0.8, 1.4, 1)$ at time 2000. This change leads to different optimal matching rates and thus different optimal value. Here we use $\beta = 0.1$ in order to decrease the adaptation time to the changes in input rates. Figure 6.3 shows the queue trajectories of the virtual and physical systems. We observe that EGPD automatically adapts to the new arrival rates and reaches the new "right" queue lengths, without using any a priori information on this change.

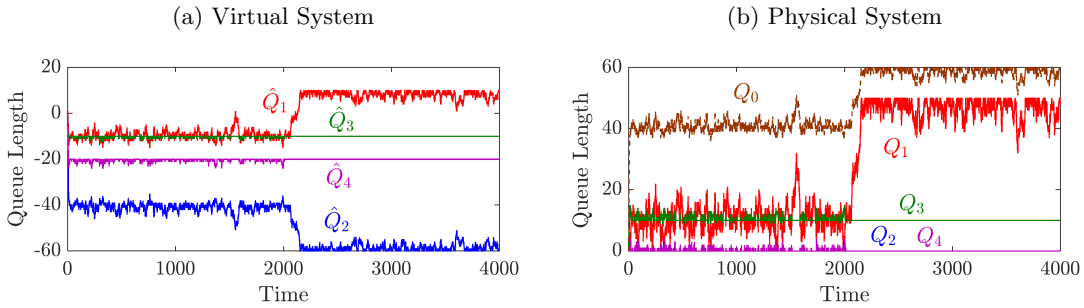


Figure 6.3: Adaptation to the changes in arrival rates.

C. Effect of parameter β on average profit. Figure 6.4 shows how the average profit of the EGPD algorithm changes for different values of parameter β. As β increases, the average reward is getting far from the optimal one and the average holding cost decreases. We observe that the average profit increases and then decreases for larger values of β. We conjecture a similar behavior to hold in general. With this regard, we observe that by fixing appropriate β (for example, it is around 2 in this example), the average profit obtained from the EGPD algorithm will be increased.

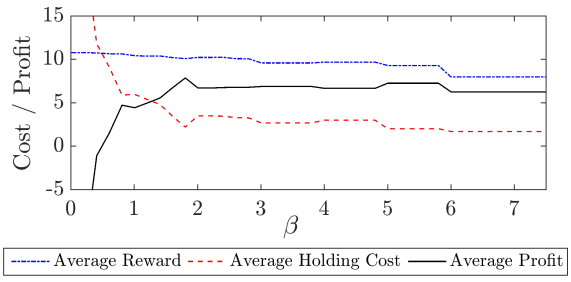


Figure 6.4: Average profit for different values of β.

6.2 Experiment 2: Average Profit in Bipartite Matching System

In this experiment, we construct a matching system in order to study the average profit. Figure 6.5 depicts a bipartite matching system with 8 item types $\mathcal{I} = \{1, 2, 3, 4, 1', 2', 3', 4'\}$. The arrival graph is on the left, where each edge shows a possible arrival pair, and the plot in the right hand side is the matching graph with edges representing the possible matchings.

We consider the process in discrete time $t = 1, 2, \cdots$. An i.i.d. arrival process is chosen in a way to satisfy necessary and sufficient conditions for stability as well as three conditions (A1)-(A3) in [4]. In particular, at each time t, a paired arrival enters the system with some probability, as it is specified in table 6.2.

Figure 6.5: Illustration of the matching system.

Table 6.2: Probability of having a paired arrival at a given time t.

Arrival pairs	(1,1′)	(1,2′)	(2,1′)	(2,2′)	(3,4′)	(4,3′)	(4,4′)
Probability	0.166	0.083	0.087	0.083	0.166	0.322	0.083

Carrying items from period t to $(t+1)$ has linear holding cost $c \cdot Q(t)$ with $c = (1, 2, 3, 4, 4, 3, 2, 1)$. We consider average reward maximization (linear utility function) problem. Table 6.3 shows the amount of reward associated with each matching.

Table 6.3: Reward associated with each matching.

Matching pairs	$\langle 1, 3' \rangle$	$\langle 1, 4' \rangle$	$\langle 2, 3' \rangle$	$\langle 2, 4' \rangle$	$\langle 3, 1' \rangle$	$\langle 3, 2' \rangle$	$\langle 3, 3' \rangle$	$\langle 4, 3' \rangle$	$\langle 4, 4' \rangle$
Reward ($\times 400$)	5	1	1	5	1	1	5	1	1

EGPD, MaxWeight and h-MWT policies (such as in [4]) are implemented for this particular example. In addition to the EGPD algorithm, we have also implemented a weighted version of this algorithm. Specifically, we select weights

$$\gamma_i = \begin{cases} c_i & \text{if } \hat{Q}_i(t) \geq 0 \\ \max\limits_{i \in \mathcal{I}} c_i & \text{if } \hat{Q}_i(t) < 0 \end{cases}, i \in \mathcal{I} \tag{6.2}$$

and then use rule (5.2) in the EGPD algorithm.

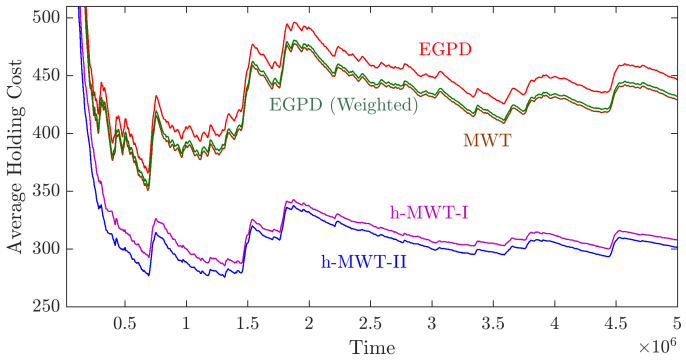
The cost-weighted MaxWeight policy chooses a matching

$$j(t) \in \operatorname*{argmax}_{j \in \mathcal{J}} \sum_{i \in \mathcal{I}} c_i \; Q_i(t) \, \mu_i(j). \tag{6.3}$$

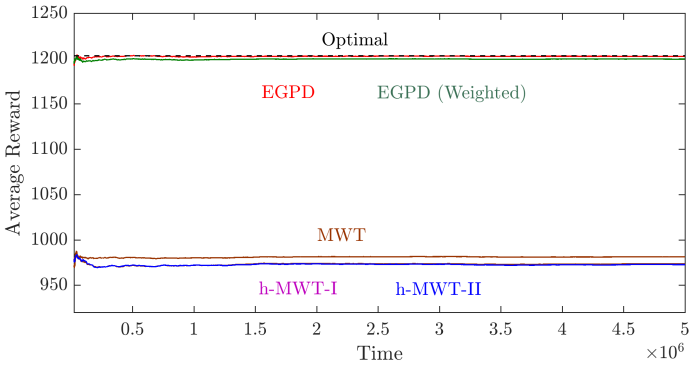
The h-MWT policy [4] is such that matching $\langle 3, 1' \rangle$ and $\langle 3, 2' \rangle$ are allowed only when $Q_4(t) - Q_{1'}(t) - Q_{2'}(t)$ is below some threshold value, while when above this value, we will perform matchings by cost-weighted MaxWeight policy. We have implemented this policy in two different versions: in h-MWT-I, we use the approximate threshold value (as specified in [4], equation (17)) and, in h-MWT-II, threshold value which gives the minimum holding cost is considered. Through simulations, we observe that the best threshold value is around -75 for this problem.

We would like to emphasize that h-MWT solves minimum average cost problem, while EGPD is optimal for the reward maximization problem; however, EGPD can be controlled in a way to have lower cost. (See the discussion on Section 5.) Our main goal in this experiment is to study the behavior of the EGPD algorithm in achieving higher average profit and compare it with that of MaxWeight and h-MWT policies.

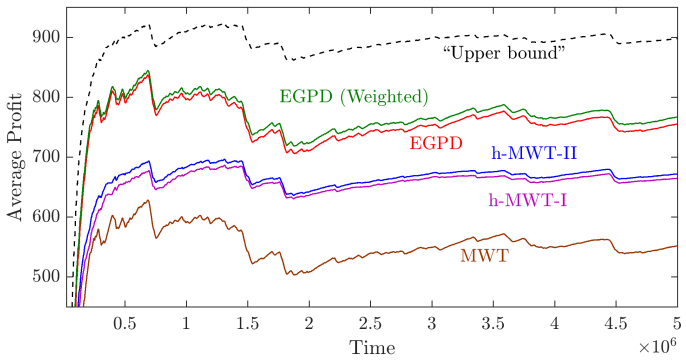
Comparison is performed on the basis of average values of holding cost, reward and profit. The detailed simulation results are presented in Figure 6.6. We see fast convergence behavior to optimality as in the previous experiment, but we set runtime 5×10^6 for a reliable comparison. The plots on this figure are regarding the case when parameter β in EGPD and Weighted-EGPD are set to be 150 and 100, respectively. In part (a), we observe that the average holding cost under the h-MWT-II policy is the lowest. Also the holding cost imposed by the Weighted-EGPD algorithm can be close to that of the cost weighted MaxWeight algorithm, but we conjecture that it typically will be worse since EGPD only uses the information of virtual system which does not recover the actual state of the physical system. Part (b) depicts the average reward of the system, where both EGPD and Weighted-EGPD obtain higher (close to optimal) reward compared



(a) Average holding cost of the system under different algorithms



(b) Average reward obtained by different matching policies



(c) Average profit of the system under different matching policies

Figure 6.6: Comparison of h-MWT-I (with approximate threshold value), h-MWT-II (with best threshold value from simulations), (Cost-Weighted) MaxWeight, EGPD and Weighted-EGPD algorithms.

to the other two algorithms. The optimal level for this problem is found by solving the underlying linear optimization problem. The average profit levels for different policies are shown in part (c), which is, in fact, drawn from the results of parts (a) and (b). Specifically, the average profit of a particular algorithm equals to its average reward obtained from part (b) minus its average holding cost as in part (a). For this problem, observe that the Weighted-EGPD performs the best among the implemented algorithms. In this figure, we have also shown the "upper bound" on average profit, which is defined to be the average reward of EGPD algorithm minus holding cost of the h-MWT policy. Roughly speaking, it is natural to expect that no policy can attain higher profit than this "upper bound" (up to the fact that both EGPD with fixed β and h-MWT do not achieve the optimal values for the corresponding problems exactly).

We want to emphasize that the results of this experiment does not mean that Weighted-EGPD is always the best algorithm. For instance, if the average holding cost dominates the average reward (in average profit maximization problem), then it is reasonable to use h-MWT which is specifically designed for holding cost minimization.

7 Conclusion

In this paper, we have proposed an approach for optimal dynamic control of general matching systems. The central idea is using a virtual matching system allowing negative queues, which facilitates the design of a policy. This puts the problem into a queueing network control framework, to which an extended version of the GPD algorithm, called EGPD, can be applied. The approach is very generic, not restricted to a special class of matching problems, such as bipartite customer-server matching.

Although the EGPD algorithm that we develop has the average reward maximization as its objective, the parameter setting can be used to achieve good performance in terms of the more general matching-reward-minus-holding-cost objective.

The algorithm is also very robust in the sense that it does not require the knowledge of input rates. Simulations demonstrate good performance of the algorithm.

References

[1] Ivo Adan and Gideon Weiss. Exact fcfs matching rates for two infinite multitype sequences. *Operations research*, 60(2):475–489, 2012.

[2] Ivo Adan, Ana Bušić, Jean Mairesse, and Gideon Weiss. Reversibility and further properties of fcfs infinite bipartite matching. *arXiv preprint arXiv:1507.05939*, 2015.

[3] Burak Büke and Hanyi Chen. Stabilizing policies for probabilistic matching systems. *Queueing Systems*, 80(1-2):35–69, 2015.

[4] Ana Bušić and Sean Meyn. Optimization of dynamic matching models. *arXiv preprint arXiv:1411.1044*, 2014.

[5] Ana Bušić, Varun Gupta, and Jean Mairesse. Stability of the bipartite matching model. *ACM SIGMETRICS Performance Evaluation Review*, 38(2):6–8, 2010.

[6] René Caldentey, Edward H. Kaplan, and Gideon Weiss. Fcfs infinite bipartite matching of servers and customers. *Advances in Applied Probability*, 41(3):695–730, 2009.

[7] Itai Gurvich and Amy Ward. On the dynamic control of matching queues. *Stochastic Systems*, 4(2): 479–523, 2014.

[8] BRK Kashyap. The double-ended queue with bulk service and limited waiting space. *Operations Research*, 14(5):822–834, 1966.

[9] László Lovász and Michael D Plummer. *Matching theory*, volume 367. American Mathematical Soc., 2009.

[10] Jean Mairesse and Pascal Moyal. Stability of the stochastic matching model. *arXiv preprint arXiv:1404.6677*, 2014.

[11] Aranyak Mehta. Online matching and ad allocation. *Theoretical Computer Science*, 8(4):265–368, 2012.

[12] Alexander L Stolyar. Maximizing queueing network utility subject to stability: Greedy primal-dual algorithm. *Queueing Systems*, 50(4):401–457, 2005.

[13] Alexander L Stolyar and Tolga Tezcan. Control of systems with flexible multi-server pools: a shadow routing approach. *Queueing Systems*, 66(1):1–51, 2010.