STOCHASTIC APPROXIMATION SCHEMES FOR A LOAD BALANCING PROBLEM

Rahul Simha and James F. Kurose

COINS Technical Report 89-10

Stochastic Approximation Schemes for a Load Balancing Problem [1]

RAHUL SIMHA & JAMES F. KUROSE

Department of Computer and Information Science
University of Massachusetts
Amherst, Mass. 01003

Abstract

We consider a special class of load balancing problems: a centralized job scheduler and several processors, together with a probabilistic scheduling policy that the scheduler uses to direct an incoming job to one the processors. We examine the case that no assumptions are made about the form of the arrival and service distributions and that direct on-line sensitivity measurements are made in the system. These measurements are typically noisy and thus optimizations algorithms used to produce optimal scheduling policies must be redesigned to handle the randomness introduced by these system measurements. Two stochastic approximation schemes, one scheme from the literature and the other introduced here, are used to augment an optimization algorithm in order to minimize, on-line, the mean system time of a job with respect to the scheduling policy. The second scheme, sampling-controlled iteration, provides a method of dealing with biasedness in estimators, a property present in several relevant estimation procedures. Our main result is that these iterative mechanisms converge to the optimal policy for this formulation of the problem. A comparison between the two schemes is made using simulation results for a system of six processors and additional results are presented in the case that the arrival process may be assumed Poisson. In addition, we present a technique for the case that the feasibility constraints are unknown and must themselves be estimated from system measurements.

[1] This work was supported in part by the Office of Naval Research under contract number N00014-87-K-0796

1. Introduction

The general problem of load balancing in a computer system has been the subject of widespread study for several years [24]. In this paper, we consider a restricted class of load balancing problems [6,19,26,29,42], in which jobs arrive at a central job scheduler at a rate λ to be served at any one of K processors. A job requires an average service time of $\frac{1}{\mu_i}$ on processor i. The central job scheduler dispatches an incoming job to processor queue i with probability x_i and once a job is assigned to a particular queue it remains there until served. The service discipline is assumed to be first-come-first-served. Stated simply, the objective is to find the optimal scheduling policy, i.e., the vector of probabilities $X^* = (x_1^*, ..., x_K^*)$ that minimizes the expected completion time of a job.

In this paper, we are concerned with systems in which little is known about the stochastic nature of the job arrival process or job service requirements and in which *measurements* must be made to obtain even the basic parameters λ and μ_i. We identify two parts to the problem of finding the optimal policy, namely estimation and optimization, and examine several approaches towards solving this problem. It is concluded that a scheme which uniformly integrates these two processes is desirable and in this case, care must be taken to ensure that random errors encountered in estimation do not greatly impede the optimization process. In this paper, we focus on applying stochastic approximation techniques to the problem of load balancing. Our contribution is to present an optimization algorithm and to augment it into a stochastic approximation (optimization) procedure through the use of two schemes for interleaving an iterative optimization algorithm with the process of estimation. One scheme is well-established and is commonly called stochastic approximation. In this paper, we refer to this scheme as *stepsize-controlled iteration*; we also introduce an alternative iteration technique which we refer to as *sampling-controlled iteration*. This second scheme, as we will demonstrate, has the added advantage of removing bias from estimators, the importance of this being that several commonly used estimators associated with queueing systems are known to be biased [18]. One of the main results of this paper is that we establish the theoretical convergence of the resulting algorithms. We present simulation results that compare the convergence behavior of the two schemes and, furthermore, also consider the situation in which the feasibility constraints are unknown and must themselves be estimated from the system. Finally, in the case that certain assumptions may be made about the arrival process, we discuss the use of alternative estimation procedures.

In the next section we define the load balancing problem examined in this paper. In Section 3, we discuss several past approaches towards solving the problem and motivate the use of stochastic approximation. Section 4 contains a presentation of an optimization algorithm and the two schemes for augmenting the algorithm into a stochastic approximation procedure. The main theoretical results are also presented in this section. In Section 5, we examine simulation results, consider the estimation of unknown constraints and discuss alternative estimation procedures. Finally, in Section 6, we present our concluding remarks and outline possible future work.

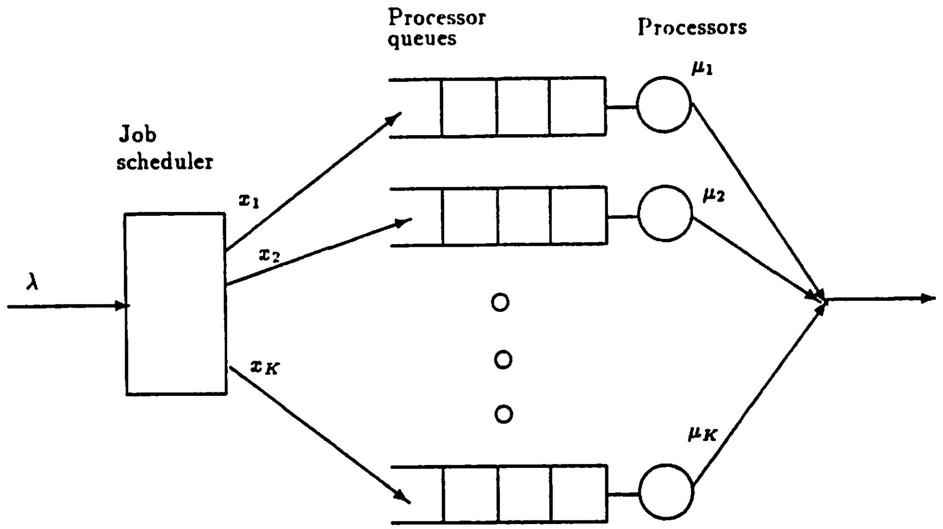

Figure 1: A centralized load balancing problem

2. Problem Definition

In this section we present a formal definition of the load balancing problem under consideration. We also introduce the notation used in the remainder of the paper. Consider the system in figure 1. Jobs arrive at a central job scheduler which assigns each job to a processor queue according to a probabilistic policy. The processors serve their job queues in a first-come-first-served fashion and jobs that complete execution leave the system. We further characterize the system as follows:

- *Arrival process.* The interarrival times of the jobs arriving at the scheduler are independent random samples from an unknown distribution A with finite mean and variance. We denote the mean by $\frac{1}{\lambda}$ where λ is the unknown arrival rate.

- *Service requirements.* A job served at processor i, $1 \leq i \leq K$, requires a period of time that is taken independently from an unknown distribution B_i with finite mean and variance. As with the arrival rate, the mean of this distribution is also unknown and, in the usual manner, is written as $\frac{1}{\mu_i}$. The collection of parameters relevant to the distributions A and B_i, which includes λ and μ_i, is denoted as Θ.

- *Scheduling (load balancing) policy.* The central job scheduler sends an incoming job to the i^{th} processor queue with probability x_i. Thus, $\sum_{i=1}^{K} x_i = 1$. The vector $X = (x_i, ..., x_K)$ is called the scheduling policy. Once a job is assigned to a particular queue, it remains there until it is served by the corresponding processor.

- *Performance metric.* The performance metric we consider in this paper is the mean completion time of a job, that is, the time between a job's arrival at the centralized dispatcher and its completion at one of the K processors. The completion time is also commonly referred to as the delay or response time for a job. We assume the existence of a continuous mean delay function $D(x_1, ..., x_K, \Theta)$ that is convex in $x_1, ..., x_K$ for every set of parameters Θ. We remark here that the methods presented in this paper are not bound to this choice of a performance metric. Provided that appropriate estimators are used, the algorithms and theoretical results shown here apply to any metric.

We are now in a position to formally state the problem:

$$\begin{aligned} \min_{x_1,...,x_K} \quad & D(x_1, ...x_K, \Theta) \\ s.t. \quad & \sum_{i=1}^{K} x_i = 1 \\ & \forall i: \ x_i \geq 0 \\ & \forall i: \ \lambda x_i < \mu_i \qquad ..... \quad (P_1) \end{aligned}$$

Let X^* be the solution to the above problem and define the set S of policies such that for each policy $X \in S$, the constraints above are satisfied. We assume that an initial policy $X_0 \in S$ is known. Note that the set S, being an intersection of the three convex sets that specify feasibility, is itself convex. Also note that the second inequality constraint contains λ and μ_i which have been assumed to be unknown. Initially, we consider the simplified case that $\lambda x_i < \mu_i$ is satisfied for all $x_i < 1$, i.e., no queue can be overloaded. Later, in section 5, we consider more difficult problem of *estimating* the constraint $\lambda x_i < \mu_i$ concurrently with the execution of the optimization procedure. In the next section we examine several general approaches towards determining the optimal control policy X^*.

3. Approaches to Policy Optimization

Figure 2 shows a tree categorizing various solution paradigms to this load balancing problem. These may be broadly divided into two categories: methods based on models of the system and those based on direct stochastic approximation, which is the approach adopted in this paper. We argue that former are sensitive to assumptions made about the arrival and service distributions and, thus, consider the latter method, which does not involve solving any such model. Below we

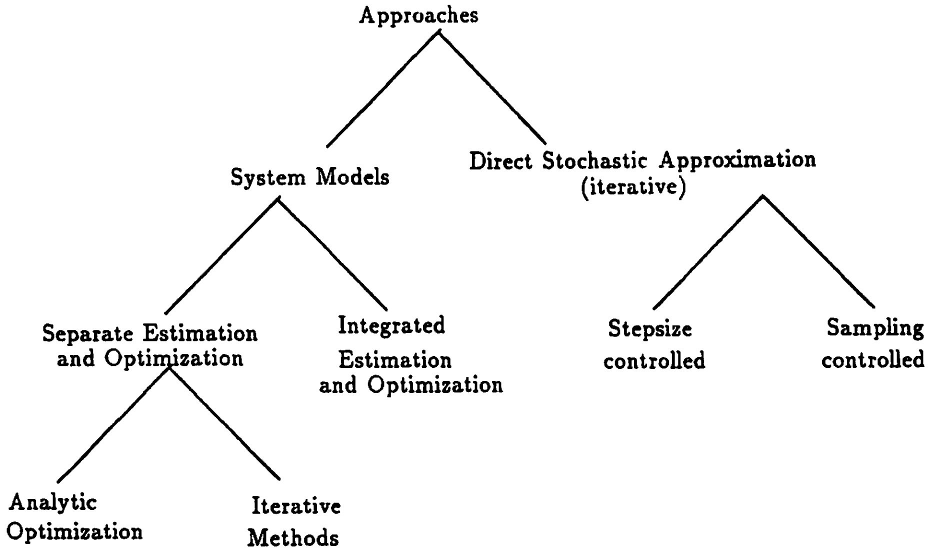
Approaches
System Models
Direct Stochastic Approximation (iterative)
Separate Estimation and Optimization
Integrated Estimation and Optimization
Stepsize controlled
Sampling controlled
Analytic Optimization
Iterative Methods

Figure 2: Approaches to solving problem P_1

discuss some of the modeling approaches in detail and motivate the use of stochastic approximation methods.

3.1 Approaches based on system models

In the first and most prevalent approach, assumptions are made about the distributions A and B_i of the arrival and service processes respectively [26,29] and a queueing model is formulated for the system in figure 1. If this model is analytically tractable, an expression for the mean delay function $D(X, \Theta)$ can be constructed and then solved for X^*. For example, if we assume that both the interarrival and service distributions are exponential then it is possible to show that (e.g. [6,29]):

$$D(x_1, ..., x_K, \Theta) = \sum_{i=1}^{K} \frac{x_i}{(\mu_i - \lambda x_i)}$$

We note that, in many cases, however, assumptions made about the distributions A and B_i make it difficult to obtain an expression for $D(X, \Theta)$. Nonetheless, tractable models are important and useful and accordingly, we first consider two approaches towards solving problem P_1 given that an expression for D is available.

As mentioned earlier, the parameters of the distributions are often not known *a priori* and hence, must be estimated using measurements taken from the system. For example, in the case of the $M/M/1$ type example above, the arrival rate, λ, and the service rates, μ_i must be estimated. In the first of the two approaches discussed here, the processes of estimation and optimization are temporally separated. The relevant parameters are first estimated in order to completely specify the function $D(X, \Theta)$ and $D(X, \Theta)$ is then minimized using some convenient optimization technique. For example in the $M/M/1$ case above, estimates $\hat{\mu}_1, ..., \hat{\mu}_K$ and $\hat{\lambda}$ are obtained after a finite period of observation in the system and then, using these estimates, the delay function $D(x_1, ..., x_K, \hat{\lambda}, \hat{\mu}_1, ..., \hat{\mu}_K)$ is minimized. We use the notation $\hat{\Theta}$ to denote an estimate of Θ.

We note that a closed-form analytic solution of the minimization problem is typically very difficult [29] (for solutions when the distributions A and B_i are assumed to be exponential see [6] and also the $O(K \log K)$ algorithm in [42]) and, therefore, in order to find $\min_{X \in S} D(X, \hat{\Theta})$, we focus on iterative methods of optimization instead. Typically, an initial, feasible policy $X_0 \in S$ is chosen and at each step (iteration) of the algorithm, a new vector is generated:

$$X^{(n+1)} = Q(X^{(n)}, \hat{\Theta})$$

where Q represents the procedure by which $X^{(n+1)}$ is computed by the algorithm. In this manner, one may expect $X^{(n)}$ to converge to a feasible policy. In the case that the estimates of the parameters are exact, i.e., $\hat{\lambda} = \lambda$ and $\hat{\mu_i} = \mu_i$, it is possible to show that for several algorithms (see [2,8,20,35,39,43] for example):

$$\lim_{n \to \infty} X^{(n)} = X^*$$

i.e. that these algorithms converge to the optimal policy for problem P_1. Note that since the optimization phase begins only after the estimation is complete, the quality of the estimate $\hat{\Theta}$ is independent of n, the iteration number of the algorithm.

If the assumptions made about the arrival and service distributions are reasonable then this approach of separate estimation followed by iterative optimization may be appropriate since one hopes that the two functions, $D(X, \Theta)$ and $D(X, \hat{\Theta})$ are approximately equal and thus $\lim_{n \to \infty} X^{(n)} \approx X^*$. However, we observe two major disadvantages with this method. First, note that during the estimation phase, some scheduling policy must be in operation in order to make the measurements. If this policy is suboptimal, then the resulting performance of the system *during* the estimation phase may be poor. The second disadvantage, which from a theoretical standpoint is the more serious one, is that if the estimates are not accurate, we might not obtain convergence close to the optimal policy, i.e.,

$$\lim_{n \to \infty} X^{(n)} \not\approx X^*$$

(even worse, it is possible to converge to an infeasible policy, i.e., $\lim_{n \to \infty} X^{(n)} \notin S$). It is interesting to observe that the two disadvantages described above have diametrically opposed remedies. In order to limit the use of a poor scheduling policy during the estimation phase. the period of estimation should be kept small. However, this may result in highly inaccurate estimates and algorithmic convergence to an unacceptably suboptimal policy. Similarly, if the estimates are required to be substantially accurate then one would require a protracted estimation phase in which the system operates under a suboptimal policy. Since the estimation phase must halt in some finite time, one is faced with the question of judging *when* Θ and $\hat{\Theta}$, which is being estimated, are close enough so that the two delay functions $D(X, \Theta)$ and $D(X, \hat{\Theta})$ are approximately equal.

In order to overcome the two problems discussed above, we are motivated to interleave the estimation and optimization phases into a single integrated process. Consider the example of $M/M/1$ processor queues and an iterative optimization algorithm, Q. Further, let $\hat{\mu}_i^{(n)}$ be an estimate of μ_i using all the measurements up to the time at which the n^{th} algorithmic iteration of Q is executed ($\hat{\lambda}^{(n)}$ is similarly defined). For example, $\hat{\mu}_i^{(n)}$ may be taken as the inverse of the average of all the service time samples obtained at queue i between the time the system first started operation and n^{th} iteration. Then define each iterative step as [5]:

$$X^{(n+1)} = Q(X^{(n)}, \hat{\lambda}^{(n)}, \hat{\mu}_1^{(n)}, ..., \hat{\mu}_K^{(n)})$$

Intuitively, the optimization algorithm proceeds using all the information gathered by the concurrently executing estimation process. As n increases, we may expect that the estimates $\hat{\lambda}^{(n)}$ and $\hat{\mu}_i^{(n)}$ get increasingly accurate and thus later iterations tend to use better information. Theoretically, it is possible to show [5] that, under certain regularity conditions, $X^{(n)}$ converges to X^*. More formally,

$$\lim_{n \to \infty} \left| X^{(n)} - X^* \right| = 0 \;\; a.s.$$

where the notation *a.s.* is used to denote convergence *almost surely* [34].

We note that the disadvantages of separate estimation and optimization are overcome to some extent here. Since several iterative schemes gradually improve the objective function with each successive iteration, it may be reasonable to expect some improvement as the scheduling policy iteratively changes as a result of the iterative optimization process. Furthermore, from a theoretical standpoint, it is possible to prove eventual convergence to the optimum.

3.2 Direct stochastic approximation

The approaches discussed above require that assumptions be made about the nature of the distributions A and B_i and further, that an expression for the mean delay $D(X, \Theta)$ be obtained in terms of the parameters, Θ, and the scheduling policy vector, X. This expression is used in the algorithm Q where, typically [2,20], partial derivatives, $\frac{\partial D}{\partial x_i}$, are used in the computation at each iteration. As discussed in the above section, if the modeling assumptions are valid then the above iterative method is a reasonable option. However, in some cases, even if the assumptions are valid, the resulting model can be intractable and therefore, it may not be easy to obtain an expression for $D(X, \Theta)$.

We are thus interested in the case in which no modeling assumptions are made about the distributions A and B_i or that, even if they are valid, the corresponding queueing problem is not amenable to analytic solution. We restrict our attention to a situation in which no model of the system is solved and, instead, derivative information is directly estimated from the system. In this paper, we focus on a particular iterative mechanism, Q (described below), and present two schemes by which this mechanism may be augmented into a combined estimation-optimization procedure.

Stochastic approximation refers to an algorithmic technique that is used to handle noise or randomness in data. Since its inception [33], several iterative optimization algorithms have been augmented to stochastic approximation procedures and many new procedures have been proposed [21,44]. The technique provides a method based on strong theoretical foundations in which noise in the data is systematically masked out in the process of iterating towards a solution. This procedure has found applications in a large number of research and applied problems [44]. In computer science, the stochastic approximation technique has been applied in the area of multiaccess communications [9], learning [23] and, in addition, to centralized load balancing [19].

There are several differences between our work and the research of [19]. We apply the standard stochastic approximation technique to a different optimization algorithm, and more importantly, introduce a new stochastic approximation technique for this optimization algorithm. This new technique, sampling-controlled iteration, has the desirable property of being able to handle bias in the class of estimators useful here, most of which are, in fact, known to be biased. In addition, we consider the problem of unknown feasibility constraints. There are other differences [19] as well: in [19] the distributions A and B_i are all taken to be of the exponential type and the minimization of the mean delay function, D, is reformulated into a problem of minimizing weighted least squares of processor idle times. An idle time estimator is introduced and convergence to the optimum is demonstrated. In contrast, we assume very little about the nature of the arrival and service distributions and use a different estimator, the Perturbation Analysis estimator [14,15] for the derivative of the delay with respect to the scheduling policy. We note that this estimator together with others [32] have been used in problems concerning load balancing [31] and routing [4].

In the next section we present an iterative optimization algorithm together with two stochastic approximation schemes and we discuss their convergence properties.

4. Two Stochastic Approximation Schemes

For the reasons discussed in earlier sections, namely better performance during estimation and eventual convergence, we have observed that it is important to interleave the processes of estimation and optimization. In this section we describe an iterative procedure in which successive algorithmic computations are separated by periods of estimation. At each computational step n, the current scheduling policy, $X^{(n)}$, is updated to $X^{(n+1)}$ using the measurements accrued in the observation period between the current time and the time at which $X^{(n)}$ was computed.

The computational procedure at each step uses estimates of the partial derivatives, $\frac{\partial D}{\partial x_i}$, in updating the scheduling policy. We use the Perturbation Analysis technique [14,15] in order to obtain these estimates from the queues. In this paper, we assume that this technique can be implemented in the system and only briefly outline the procedure for obtaining derivatives; for further details the reader is referred to [14,15]. The main idea behind Perturbation Analysis is simple: by tracking each incoming job through the system, it is not only possible to collect statistics such as the average completion time, but also to collect, in parallel, the statistics that would correspond to a hypothetical increase in the arrival rate. For example, consider processor queue i and let $D_i(r_i)$ denote the delay at that queue corresponding to some input rate r_i. By *simultaneously* obtaining estimates of $D_i(r_i)$ and $D_i(r_i + \Delta r_i)$ at input rates r_i and $r_i + \Delta r_i$, one obtains an estimate of the derivative as $\frac{D_i(r_i+\Delta r_i)-D_i(r_i)}{\Delta r_i}$. Clearly, it is easy to obtain estimates of $D_i(r_i)$ when the queue is operating with an arrival rate of r_i. What is not obvious is that the Perturbation Analysis technique allows one to obtain estimates of $D_i(r_i + \Delta r_i)$ simultaneously with those of $D_i(r_i)$ while incurring little extra overhead [14,15].

In the load balancing problem discussed in this paper, if D_i represents the expected delay at queue i, then $D = \sum_{i=1}^{K} x_i D_i$. Therefore,

$$\frac{\partial D}{\partial x_i} = x_i \frac{\partial D_i}{\partial x_i} + D_i \tag{1}$$

If we take $r_i = \lambda x_i$, then

$$\frac{\partial D_i}{\partial x_i} = \frac{\partial D_i}{\partial r_i} \lambda \tag{2}$$

Now, as discussed above, the Perturbation Analysis technique provides a means of obtaining estimates of $\frac{\partial D_i}{\partial r_i}$ and thus, by additionally using estimates for λ and D_i, we may, through equations (1) and (2), get an estimate for $\frac{\partial D}{\partial x_i}$. We introduce now some of the notation used in describing the computation at each step:

t - denotes time. Therefore, $t \geq 0$; we assume that the system is started at time $t = 0$.

n - the iteration number of the algorithm.

τ_n - the random time at which the n^{th} iteration of the algorithm starts execution or, equivalently, the end of the n^{th} measurement interval. We assume that the computation itself, which uses the measurements accrued in this interval, requires negligible time and therefore, its contribution to the duration of the interval may be ignored.

$M_i^{(n)}$ - the (random) number of samples obtained of $\frac{\partial D}{\partial x_i}$ since τ_{n-1}, i.e., since the $(n-1)^{th}$ iteration. We assume that each estimation of the derivative is performed over a busy period for that queue and that the measurement obtained from a single busy period constitutes a sample.

$L^{(n)}$ - a sequence of positive integers to be described later.

$f_i(x_i)$ - for convenience of description, we use the notation $f_i(x_i)$ to denote the partial derivative, $\frac{\partial D}{\partial x_i}$.

$\hat{f}_{iL^{(n)}}^{(n)}(x_i)$ - an estimate of $\frac{\partial D}{\partial x_i}$ constructed from $M_i^{(n)}$ samples of $\frac{\partial D}{\partial x_i}$ since τ_{n-1}. These estimates may be very noisy (possessing a high variance) when $M_i^{(n)}$ is small, but are nonetheless used in the schemes to be described below.

q_{min} - a small positive number arbitrarily close to but not equal to zero.

Given the above definitions, we can now more formally define τ_n as

$$\tau_n = \inf_{t > \tau_{n-1}} \{\forall i : M_i^{(n)} \geq L^{(n)}\}$$

Thus, τ_n is the time at which at least $L^{(n)}$ samples have been obtained from each queue since the last iteration.

In order to discuss the updating algorithm and the stochastic approximation schemes that we present, it is necessary to distinguish between the algorithm and the stochastic approximation techniques that are incorporated into the algorithm handling the noisy estimates, $\hat{f}_{iL^{(n)}}^{(n)}$. Below, we present the combined procedures that result from applying the techniques to a particular optimization algorithm. The optimization algorithm itself is based on general gradient projection methods [2,35] and was first used in the context of microeconomics as an alternative to the pricing mechanism [10]. Since then it was introduced into computer science literature in the area of general resource allocation by [13]. In [13], a framework is introduced for distributed resource allocation mechanisms and mappings are described that permit decentralized execution with different communication structures. We note that the construction of the feasibility set, $A^{(n)}$ (described below), in [10] was found to be incorrect. This was corrected for in [20] and in addition, in [20], the use second derivative information and constant-order communication was considered. We also note that related gradient-based approaches have been used in several routing and load balancing problems [2,4,24,31] and also in the area of learning automata [36].

We now describe the computational procedure at each step. Note that an algorithm for updating the policy, $X^{(n)}$, at each step may be described by the equation: $X^{(n+1)} = X^{(n)} + a_n \Delta X^{(n)}$, where a_n is a scaling factor or *stepsize*. Thus, expressions for $\Delta x_i^{(n)}$ are sufficient to completely describe the computation at each step:

$$\Delta x_i^{(n)} = -\left(\hat{f}^{(n)}_{iL^{(n)}} - \frac{1}{|A^{(n)}|} \sum_{j \in A^{(n)}} \hat{f}^{(n)}_{jL^{(n)}} \right) \tag{3}$$

where $\Delta x_i^{(n)} = 0, \forall i \notin A^{(n)}$ and $A^{(n)}$ is the set described below:

```
/* Algorithm for computing the set $A^{(n)}$ */
(i)    For all $i$, sort $\hat{f}^{(n)}_{iL^{(n)}}$
(ii)   Set $E = \{m | \hat{f}^{(n)}_{mL^{(n)}} = \min_i \hat{f}^{(n)}_{iL^{(n)}}\}$.
(iii)  Do step (iv) for each $j, j \notin E$ in ascending order of $\hat{f}^{(n)}_{jL^{(n)}}$.
(iv)   If $x_i^{(n)} + a_n \Delta x_i^{(n)} \geq q_{min}$ as a result of
       the reallocation defined by equation 3 above with $A^{(n)} = E \cup \{j\}$,
       then set $E = E \cup j$. The use of $q_{min}$ is discussed below.
(v)    Set $A^{(n)} = E$
```

The following intuitive explanation of equation (3) may be given. The probability of sending a job to a particular processor i is increased ($\Delta x_i > 0$) if the estimated *marginal* expected completion time is less than that of the estimated average (over all processors) of the marginal expected completion time. In contrast, if this estimated marginal completion time for processor i is higher than the measured average, then the probability of sending a job to that processor is decreased ($\Delta x_i < 0$). The set $A^{(n)}$ is used to maintain the feasibility constraints, $x_i^{(n)} \geq 0$. Note that if we were to allow $x_i^{(n)} = 0$ then no jobs would be sent to that queue and hence for $m > n$ we would not obtain any samples from the i^{th} queue. To ensure that $x_i^{(n)}$ remains positive, we thus force $x_i^{(n)} \geq q_{min}$ and add this as a constraint to problem P_1. Furthermore, it is assumed that the feasibility constraint, $\lambda x_i < \mu_i$, is satisfied whenever $x_i < 1$. Later, in section 5, we consider the more general problem of arbitrary processor utilizations.

Due to the noise in the estimates, $\hat{f}^{(n)}_{iL^{(n)}}$, the update algorithm described above would not ordinarily converge to the optimum policy, X^*, unless some care is taken to mask out the effects of the noise. We now present two such methods and thereby, describe how the basic optimization algorithm is augmented into a stochastic approximation procedure. The first of these schemes is based on the standard stochastic approximation technique [21,33] and we call this *stepsize-controlled iteration*. The second technique, introduced here, is called *sampling-controlled iteration*. The two schemes are presented implicitly by stating the assumptions required for convergence. Thus, the first scheme is captured in statement $A1$ of the assumptions for Theorem 1 whereas the second one is described in statement $A4$ below.

Stepsize-controlled iteration:

$A1$. The stepsizes satisfy the following conditions: $\forall n : a_n > 0$, $\lim_{n\to\infty} a_n = 0$ and $\lim_{n\to\infty} \sum_{i=1}^{n} a_i = \infty$. Furthermore, $\forall n : L^{(n)} = L$ where $L > 1$.

$A2$. We assume that a strong law of large numbers exists for the random variables $\Delta x_i^{(n)}$ in the sense that if $\forall n : E[\Delta x_i^{(n)}] > \delta$ for some $\delta > 0$ then $\lim_{n\to\infty} \sum_{j=1}^{n} \Delta x_i^{(j)} = \infty$ *a.s.*. Note that if the queues have a regenerative property when the queue is empty then successive random samples of $\Delta x_i^{(n)}$ are independent samples and thus, we may apply the standard form of the strong law.

$A3$. The estimates $\hat{f}_{iL^{(n)}}^{(n)}$ are independent, unbiased estimates of $\frac{\partial D}{\partial x_i}$ with finite variance.

THEOREM 1: Under assumptions stated in the problem definition and $A1 - A3$, we have

$$\lim_{n\to\infty} X^{(n)} = X^{*} \quad a.s.$$

PROOF: (see appendix).

Sampling-controlled iteration:

$A4$. The sequence of integers $L^{(n)}$ satisfies the following conditions: $L^{(0)} > 1$ and $\forall n : L^{(n+1)} > L^{(n)}$. In addition, $\forall n : a_n = a$ where $a > 0$.

$A5$. We assume that the estimates $\hat{f}_{iL^{(n)}}^{(n)}$ are strongly consistent in the following sense:

$$\lim_{n\to\infty} \hat{f}_{iL^{(n)}}^{(n)}(x_i) = \frac{\partial D}{\partial x_i} \quad a.s.$$

$A6$. Since $\hat{f}_{iL^{(n)}}^{(n)}$ is strongly consistent, $Pr[\lim_{n\to\infty} \hat{f}_{iL^{(n)}}^{(n)}(x_i) \to f_i(x_i)] = 1$ for fixed X. Consider those sample paths for which $\hat{f}_{iL^{(n)}}^{(n)}(x_i) \to f_i(x_i)$. Then, for a given such sample path, $\forall \epsilon > 0, \exists M$ such that $\forall n > M$,

$$\left| \hat{f}_{iL^{(n)}}^{(n)}(x_i) - f_i(x_i) \right| < \epsilon$$

We assume that for each x_i and each such sample path, there exists an open neighbourhood of x_i, denoted by $N_\epsilon(x_i)$, such that $\forall n > M$ and $\forall x \in N_\epsilon(x_i)$,

$$\left| \hat{f}_{iL^{(n)}}^{(n)}(x) - f_i(x) \right| < \epsilon$$

We demonstrate that this assumption is not overly restrictive by showing that additive white noise satisfies this condition (and thus, the work of [5] falls within this framework). Consider the following model:

$$\hat{f}^{(n)}_{iL(n)}(x_i) = f_i(x_i) + \xi_n$$

where ξ_n is the measurement noise such that $E[\xi_n] = 0$ and $\xi_n \to 0$ almost surely. Then, for a given sample path on which the estimate converges, choose M as above and select M' such that $\forall n > M' : |\xi_n| < \epsilon$. Then, $\forall n > \max(M, M')$ and $\forall x$,

$$\left|\hat{f}^{(n)}_{iL(n)}(x) - f_i(x)\right| < \epsilon$$

Intuitively, this assumption states that the underlying model of randomness is such that for a particular sample path, the accuracy of $\hat{f}^{(n)}_{iL(n)}(x_i)$ is approximately the same for values of x in a small neighbourhood of x_i using the same sample path. Stated differently, the rate at which noise is removed in estimating $f_i(x_i)$ through measurements of $f_i(x_i)$ along a particular sample path is approximately the same as that for $f_i(x_i + \delta)$ for small enough δ.

THEOREM 2: Under the conditions described in earlier sections and $A4 - A6$, there exists a stepsize $a > 0$ such that

$$\lim_{n\to\infty} X^{(n)} = X^* \quad a.s.$$

PROOF: (see appendix).

A few comments are in order here. First, most of the known estimators associated with queueing systems such as the derivative estimators [14,32,41], the estimator in [19] and the estimator of equation (4) in this paper are known to be biased [18]. The bias is usually due to the correlation present in a single cycle of a regenerative system [18] or due to small sample bias (for example, the estimator in [41]). In these cases, assumption $A3$ is not satisfied. However, we note that, in using stepsize-controlled iteration, there will still be convergence to a stable, albeit possibly suboptimal policy. In this case one may hope that the bias is small enough not to matter or that there is some cancelling effect due to symmetry (such as when the service distributions are identical).

Secondly, we note that sampling-controlled iteration does not require that the estimators be unbiased, only that they be strongly consistent. It is known that the likelihood estimator of [32] and formula-based estimators, such as the one in equation (4) in this paper, are strongly consistent. We note that the Infinitesimal Perturbation Analysis estimator, described in [41], is strongly consistent for the $M/G/1$ queue.

Thirdly, we do not know whether assumption $A6$ is satisfied by any of the known estimators and instead, make the following intuitive appeal. The arrival and service distributions may be assumed independent of the scheduling policy and therefore, the degree of noise present along a particular sample path for fixed X should remain independent of X. Thus, for that sample path, if we change x_i to $x_i + \delta$, then the rate at which noise is filtered out in estimating $f_i(x_i)$ should be approximately the same as that for $f_i(x_i + \delta)$ for small enough δ. We have certainly observed this to be true whenever measured in our simulation system. Note also that a third algorithm scheme results from taking stepsizes according to $A1$ and the number of samples according to $A4$. For this case assumption $A5$ is sufficient to prove convergence. However, we believe that convergence will be too slow for practical use.

Finally, we note that, for experimental results in this paper, we use the Finite Perturbation Analysis estimator [14] instead of Infinitesimal Perturbation Analysis [41]. Theoretically, Finite Perturbation Analysis, is only a heuristic from which an estimate can only be made arbitrarily close but not equal to the true derivative. However, we find that it is simple to implement, requires no assumption to be made about the distributions and, in our simulation results, appears to be the best estimator for the load balancing problem among those studied here.

5. Experimental Results and Discussion

In the sections above we have seen that there are several approaches towards solving problem P_1 and we have focused, in particular, on two stochastic approximation schemes that augment a proven optimization algorithm. In this section, we present some simulation results using these schemes in a system consisting of six processors. In addition, we consider the problem of unknown inequality constraints We note here that in addition to the Perturbation Analysis estimator considered in the previous sections, there exist several other estimators that might possibly be used, provided that certain assumptions may be made about the arrival distribution. We thus also introduce, in this section, a new estimator based on measurements of delays and also examine its use in our load balancing problem.

5.1 Comparison of the Two Schemes

For our simulation results, we considered a homogeneous system of six processors and took the interarrival and service distributions to be exponential with parameters $\lambda = \mu_1 =, ..., \mu_K = 2$ (Thus, $x_1^* =, ..., x_K^* = \frac{1}{6}$ and $D(X^*, \Theta) = 0.6$). The initial vector, $X^{(0)}$ was taken to be $X^{(0)} = 0.3, 0.2, 0.1, 0.1, 0.15, 0.15$. Next, we took the perturbation constant to be $\Delta r_i = 0.001$. Each point plotted in each curve is the result of an average taken from 50 simulation runs. The confidence intervals, computed from tables of Student-t distributions, were found to be less than one-hundreth of each y-axis unit and are omitted from the graphs.

Figure 3 shows a plot of $D(X^{(n)}, \Theta)$ at various values of n, i.e., a plot of the expected job delay for various policies generated through the progress of the algorithm. In order to study the progress

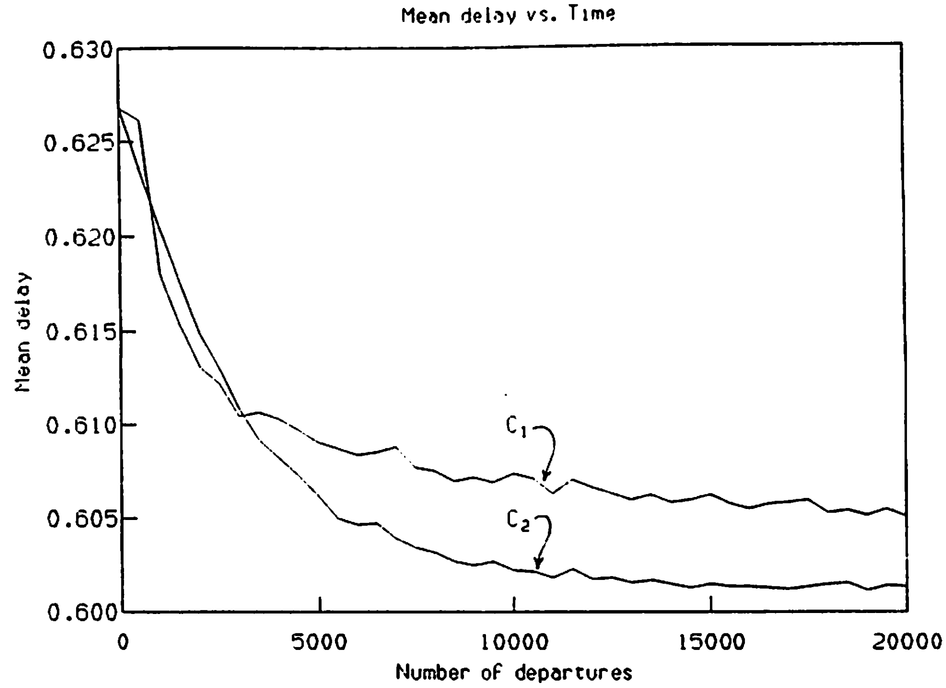

Figure 3: Convergence profiles

made through time, we plot the performance metric after every 500 departures from the entire system. The two curves plotted in figure 3 may be characterized as follows:

- C_1: Stepsize-controlled iteration (standard stochastic approximation) with $a_n = 0.1/n$, $a_0 = 0.1$, $L^{(n)} = L = 50$. Although in standard stochastic approximation, L is usually taken to be unity, we have observed that it is possible to obtain slightly better performance with higher values of L and, certainly, for a given choice of system parameters, there is likely to be an optimal value. A search for an optimal value of L is extremely time consuming and since $L = 50$ performs at least as well as $L = 1$ and our primary purpose was to make a comparison between sampling-controlled iteration and standard stochastic approximation, we settled on the chosen value of L.

- C_2: The second stochastic approximation scheme, sampling-controlled iteration, with $a_n = a = 0.01$, $L^{(n)} = L^{(n-1)} + 10$ and $L^{(0)} = 10$. As with the above scheme, the performance here also depends on the choice of a and the recurrence relation for $L^{(n)}$. We mention that the stepsize a was chosen after a 4-iteration binary search in the interval $[0.005, 0.1]$ and, in lieu of the cost of searching for an optimal recurrence relation, the choice of 10 entirely arbitrary.

We observe, from figure 3, that sampling-controlled iteration results in poor performance in the initial stages of the procedure and then rapidly gains on standard stochastic approximation, with clearly better performance towards the middle and later stages.

We are not only concerned with the mean performance of a stochastic approximation scheme but also with the variance with which successive policies are produced in the course of algorithmic

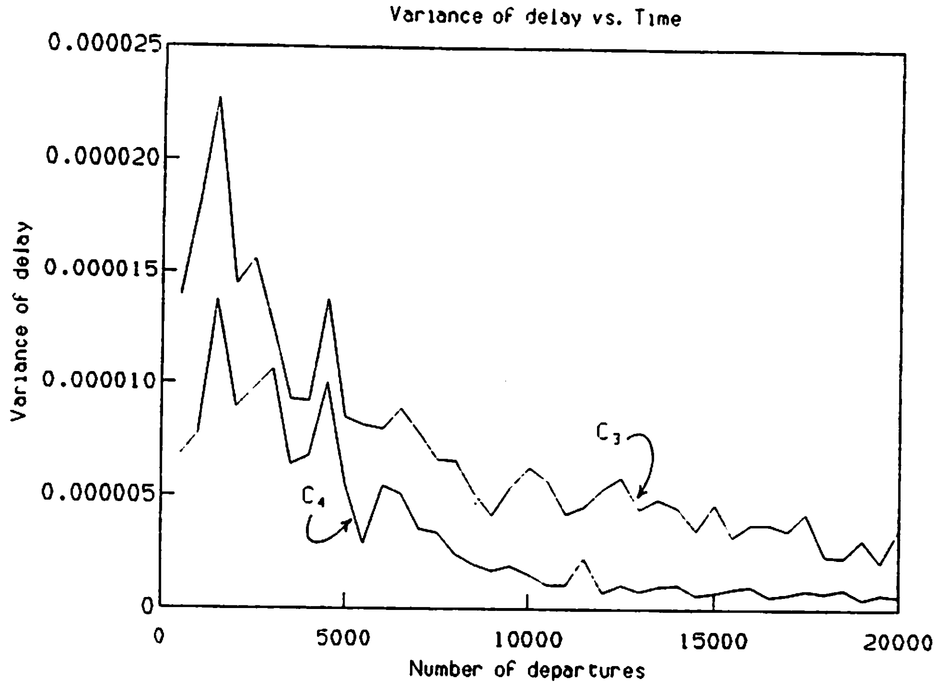

Figure 4: Variance in performance

iteration. That is, the curves in figure 3 reveal an average trend but do not indicate how an arbitrary single run might perform in comparison with the mean performance. We attempt to capture this variability in performance in figure 4 which plots the variances of mean delay for the policies produced by the two stochastic approximation schemes:

- C_3: Stepsize-controlled iteration with $a_n = 0.1/n$, $a_0 = 0.1$ and $L = 50$.
- C_4: Sampling-controlled iteration with $a_n = a = 0.01$, $L^{(n)} = L^{(n-1)} + 10$ and $L^{(0)} = 10$.

We observe that the variance in policies produced by stepsize-controlled iteration is significanly higher than in the policies produced by sampling-controlled iteration. While we cannot state conclusively that sampling-controlled iteration is better than stepsize-controlled iteration, we seek to provide an intuitive explanation of the better performance here and attempt to chararacterize situations in which sampling-controlled iteration may be preferred over the standard method.

In order to explain better the performance of sampling-controlled iteration, consider the simple case in which the inequality constraints are not active at the optimum. We use the notation $d(X^{(n)})$ to denote the *spread* of the marginal delays: let

$$d(X^{(n)}) = \max_{i,j} \left| f_i(x_i^{(n)}) - f_j(x_j^{(n)}) \right|$$

Then, as the algorithm converges, $d(X^{(n)}) \to 0$, implying that the derivatives get closer and closer and thereby resulting in the convergence of the policy to a Kuhn-Tucker point. Now, each algorithmic step represented by equation (3) uses precisely the above spread of derivatives to achieve movement of the current policy in the direction of the optimum. However, noise in the estimates of the derivatives can obscure the true derivative values and, hence, as $d(X^{(n)})$ becomes smaller, the effect of the noise is potentially more damaging. Thus, as $d(X^{(n)})$ diminishes in magnitude, it becomes necessary to obtain more refined estimates of the derivatives and also to remove small sample bias in order to distinguish them within a ball of radius $d(X^{(n)})$. This, then, is the principle behind sampling-controlled iteration.

For large n, and hence large $L^{(n)}$, the estimates get progressively more accurate and so we may expect this scheme to perform better than standard stochastic approximation. Similar observations have been made by other research efforts in stochastic approximation [3,16] as well as in other areas [1]. In [3], a fixed number of iterations was considered and the optimal selection of stepsizes and sample sizes was studied in order to minimize mean-squared-error for the particular case of the Robbins-Munro algorithm [33]. We note, however, that stepsize-controlled iteration is still well-suited to the problem examined here and may be the better scheme for other problems.

In the early stages of our research, it was thought that the general principle above might be extended by computing confidence intervals about the estimates of the derivatives and then determining the sampling period, $L^{(n)}$, so that the spread of derivatives could be distinguished with high probability. In this manner one would sample only as much as necessary. However, in order to estimate confidence intervals, estimates of the variance are required and we find that the noise content in samples of the variance is much higher than in derivative samples. Therefore, the use of confidence information resulted in poorer performance and, in some cases involving fixed stepsizes, loss of convergence.

5.2 Unknown Feasibility Constraints

We have thus far only treated the case in which the constraint $\lambda x_i^{(n)} < \mu_i$ is automatically satisfied when $x_i^{(n)} < 1$. However, it is possible that in certain systems, $\frac{\mu_i}{\lambda} < 1$ for some i and thus, ensuring that $x_i^{(n)} < 1$ does not guarantee that $\lambda x_i^{(n)} < \mu_i$. Since λ and μ_i are unknown, there are several problems to be considered when $x_i^{(n)}$ is inadvertently made greater (through algorithmic iteration) than $\frac{\mu_i}{\lambda}$. Firstly, the i^{th} queue gets overloaded and hence queue instability occurs. As a result, the current busy period may never end and (since derivative estimation is done over several busy periods) may jeopardize the very execution of the algorithm. Secondly, in the case that $\lambda x_i^{(n)} < \mu_i$, nothing is known about the properties of the estimators themselves, i.e., whether they even produce any meaningful results in the circumstances when a queue is overloaded.

We now present a method by which the above problems may be overcome. This involves changing both the optimization algorithm as well as the stochastic approximation procedure . We refer the reader to [37] for details and instead, present only an intuitive explantion and simulation

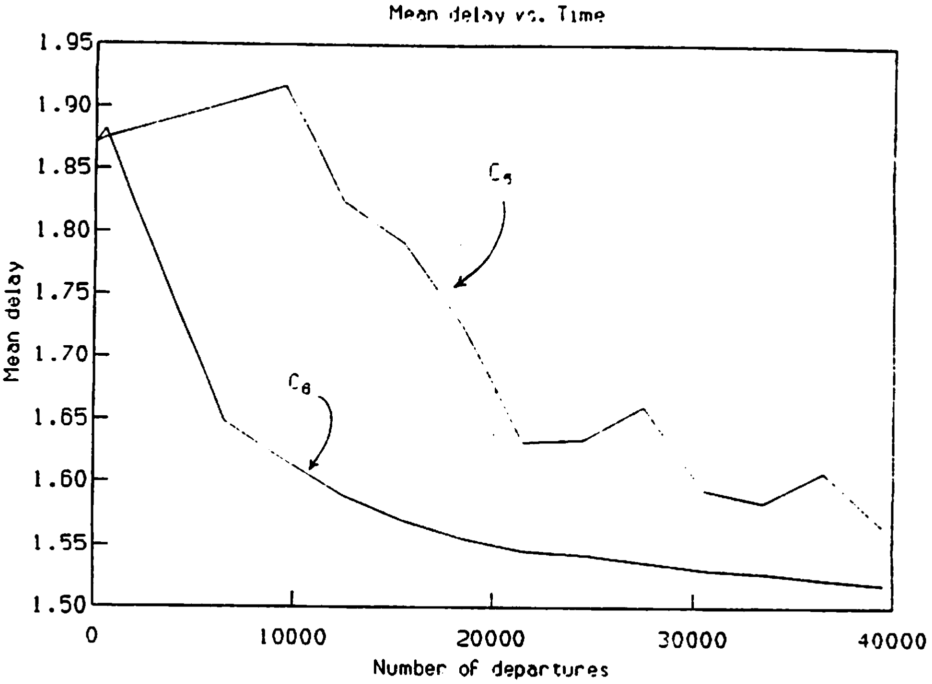

Figure 5: Unknown Feasibility Constraints

results here. Let $\mu_i^{(n)}$ and $\lambda^{(n)}$ be estimates of μ_i and λ as defined in section 3, i.e., $\mu_i^{(n)}$ is an estimate constructed from observing all service time samples from $t = 0$ upto τ_n, the time of the n^{th} iteration ($\lambda^{(n)}$ is similarly defined). Define $b_i^{(n)} = \frac{\mu_i^{(n)}}{\lambda^{(n)}}$, the current estimate of the upper constraint on x_i before the n^{th} iteration. The algorithm is modified [37] so that at each iteration $0 < x_i^{(n)} < b_i^{(n)} - \zeta$ for some small $\zeta > 0$. If ζ is chosen so that $\forall i : x_i^* < \frac{\mu_i}{\lambda} - \zeta$ then it can be shown [37] that $\lim_{n\to\infty} X^{(n)} \to X^*$ *a.s.*. Furthermore, if at any iteration $\exists i : x_i^{(n)} > b_i^{(n)}$ then a new policy, $X'^{(n)}$, is chosen (using a *feasibility operator* [37]) in which $\forall i : x_i^{(n)} < b_i^{(n)}$ and such that the distance $\left|X'^{(n)} - X^{(n)}\right|$ is as small as possible.

We may offer the following intuitive explanation. We assume that $b_i^{(n)}$ is a strongly consistent estimator of $\frac{\mu_i}{\lambda}$. Consider a sample path on which both $\hat{f}_{iL(n)}^{(n)}$ and $b_i^{(n)}$ converge; these occur with probability one [37]. Then, for this sample path, $\exists N$ such that $\forall n > N : \left|b_i^{(n)} - \frac{\mu_i}{\lambda}\right| < \frac{\zeta}{2}$. Thus, $\forall n > N : x_i^{(n)} < b_i^{(n)}$ and so $\hat{f}_{iL(n)}^{(n)}$ will provide consistent estimates for all $n > N$. The earlier convergence results in this paper may then be modified to prove that $X^{(n)} \to X^*$ *a.s.* [37].

Figure 5 shows a graph that compares stepsize and sampling-controlled iteration for the six-processor system described above with both iteration schemes modified as outlined above (see [37] for details). In this case the service and arrival parameters were taken to be $\mu_i = 2$ and $\lambda = 8$ and thus we have the constraint $x_i < 0.25$. The initial policy was taken to be $X^{(0)} =$ $(0.2, 0.2, 0.18, 0.18, 0.12, 0.12)$. We note, however, that the initial policy may be arbitrary and not

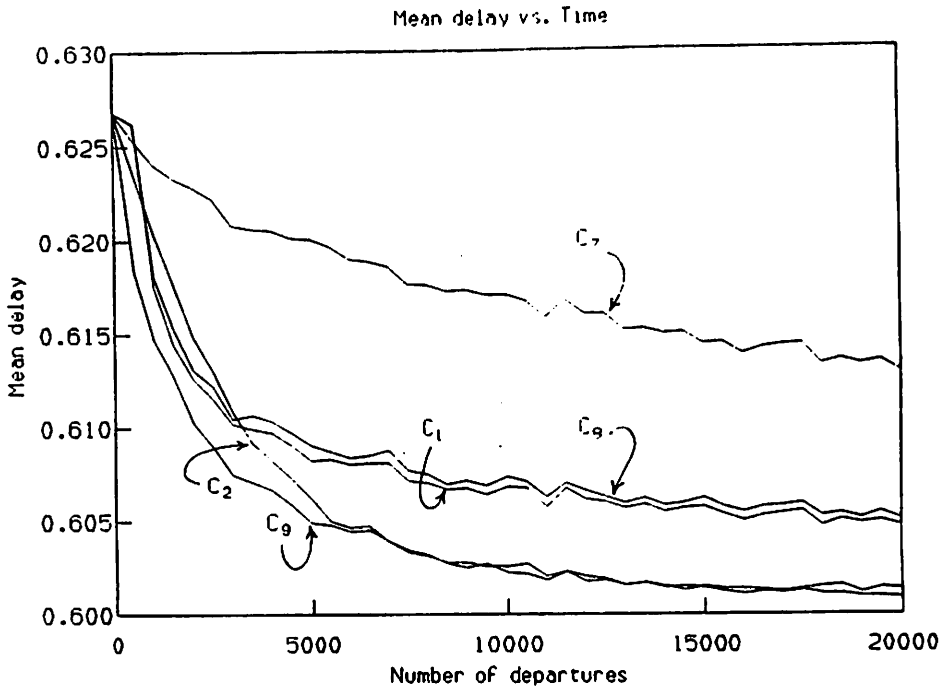

Figure 6: Other estimators

even feasible [37].

- C_5: Stepsize-controlled iteration with $a_n = 0.1/n$, $a_0 = 0.1$ and $L = 50$.
- C_6: Sampling-controlled iteration with $a_n = a = 0.001$, $L^{(n)} = L^{(n-1)} + 10$ and $L^{(0)} = 10$.

We observe that in this case the algorithm under sampling-controlled iteration results in a better convergence profile.

5.3 Other Estimators

We now turn our attention to the case that assumptions may be made about the distributions A and B_i. When A is exponentially distributed, i.e., the arrivals form a Poisson process, a likelihood-ratio estimator of the type in [11,32] may be used. Our experience with the simulated system described above shows that, in general, the perturbation analysis estimator results in better performance. For example, figure 6 shows the convergence profile for the following parameters:

- C_7: Sampling-controlled iteration with $a_n = a = 0.01$, $L^{(n)} = L^{(n-1)} + 10$ and $L^{(0)} = 10$ using the likelihood estimator in [11,31].

We observe that, in this case, using the perturbation analysis estimator (curves C_1 and C_2) results in better performance.

We focus now on yet another estimator that also requires the Poisson arrival assumption, but uses estimates of the service times, the mean delays and processor utilization (rather than direct derivative estimates) in order to estimate the derivatives. Let $\hat{D}_i^{(n)}$, $\hat{\mu}_i^{(n)}$ and $\hat{\rho}_i^{(n)}$ be estimates of the delays, service times and processor utilizations at the i^{th} processor queue during a period $T^{(n)}$. The first two estimators may be taken, for example, as the average of all delays and service times respectively. For $\hat{\rho}_i^{(n)}$, we may observe the total busy time, $B^{(n)}$ during $T^{(n)}$ and let $\hat{\rho}_i^{(n)} = B^{(n)}/T^{(n)}$. Using the expression for the mean delay of an $M/G/1$ queue [17], the following estimator, $f_{iT^{(n)}}^{(n)}$ of f_i can be constructed:

$$f_{iT^{(n)}}^{(n)} = \frac{1}{\hat{\mu}_i^{(n)}} + \frac{\beta\hat{\mu}_i^{(n)}}{c}\left(\hat{D}_i^{(n)} - \frac{1}{\hat{\mu}_i^{(n)}}\right)^2 + 2\beta\left(\hat{D}_i^{(n)} - \frac{1}{\hat{\mu}_i^{(n)}}\right)^2$$

$$+ \frac{c\,(1-\beta)}{\hat{\mu}_i^{(n)}}\left(\frac{\hat{\rho}_i^{(n)}}{1-\hat{\rho}_i^{(n)}}\right)^2 + \frac{2c\,(1-\beta)}{\hat{\mu}_i^{(n)}}\left(\frac{\hat{\rho}_i^{(n)}}{1-\hat{\rho}_i^{(n)}}\right)^2 \qquad (4)$$

The constant, c, represents $\frac{1+C_v^2}{2}$, where C_v^2 is the coefficient of variation of the service time and $\beta \in [0,1]$ is a weighting factor. Assuming that $\hat{D}_i^{(n)}$, $\hat{\mu}_i^{(n)}$ and $\hat{\rho}_i^{(n)}$ are strongly consistent, then $f_{iT^{(n)}}^{(n)}$ is easily seen to be a strongly consistent estimator for f_i.

We now consider that case that $c = 1$ and therefore, each queue is an $M/M/1$ system. In figure 6, we plot the convergence profiles for C_1 and C_2, which use the perturbation analysis estimator, together with the following curves that use the estimator of equation (4) (with $\beta = 0.5$):

- C_8: Stepsize-controlled iteration with $a_n = 0.1/n$, $a_0 = 0.1$ and $L^{(n)} = L = 50$ using the estimator in equation (4).

- C_9: Sampling-controlled iteration with $a_n = a = 0.05$, $L^{(n)} = L^{(n-1)} + 10$ and $L_0 = 10$ using the estimator in equation (4).

We note that with both stochastic approximation schemes, the estimator, $f_{iT^{(n)}}^{(n)}$, of equation (4) results in slightly better performance. Our reason for introducing this estimator is for cases in which the service rates of the processors are known. In this case, if we take $\beta = 1$ for example, we obtain an estimator that uses only response time estimates $\hat{D}_i^{(n)}$. This is useful when the centralized scheduler maintains a timer and the individual processors do not possess timing or derivative estimation capabilities. Initial results such as those in figure 6 indicate that such estimators may be useful and efficient in these contexts.

Finally, for the special case of our simulation system, in which each queue is an $M/M/1$ queue, we consider the use of an estimator based on the expression for the mean delay for this queue. The derivative of the mean delay with respect to each x_i can be expressed in terms of the parameters λ

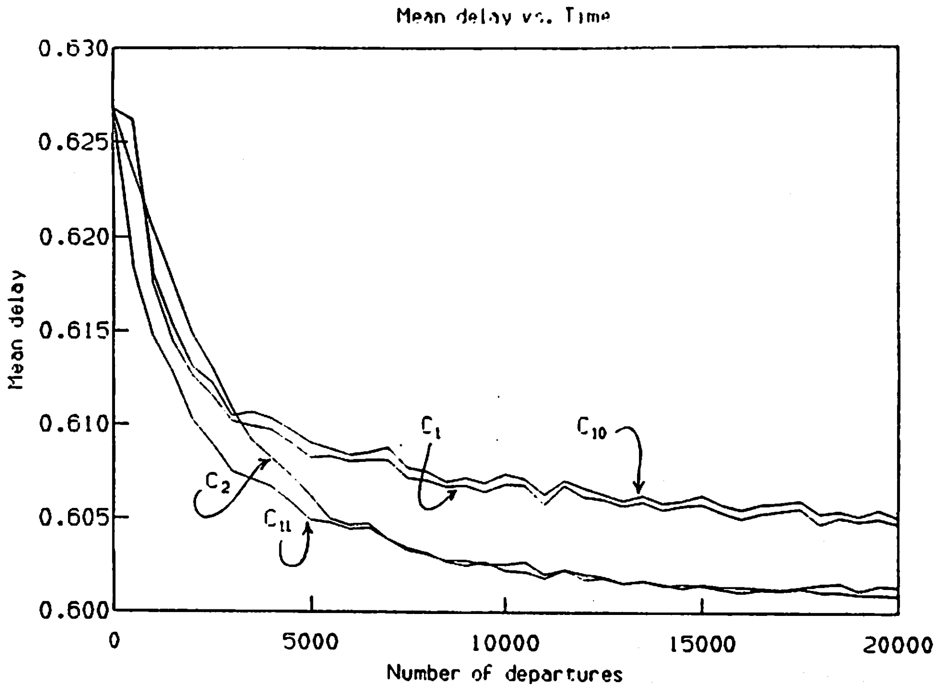

Figure 7: MM1-formula based estimator

and μ_i and, therefore, an estimator for the derivative can be constructed using the estimates $\hat{\lambda}^{(n)}$ and $\hat{\mu}_i^{(n)}$ described in section 3:

$$\hat{f}_i^{(n)}(x_i) = \frac{\hat{\mu}_i^{(n)}}{\left(\hat{\mu}_i^{(n)} - \hat{\lambda}^{(n)} x_i\right)^2} \tag{5}$$

Figure 7 shows the results in comparing the above estimator in equation (5) with the Perturbation Analysis estimator of previous sections. We plot the convergence profiles C_1 and C_2 together with the following curves:

- C_{10}: Stepsize-controlled iteration with $a_n = 0.1/n$, $a_0 = 0.1$ and $L^{(n)} = L = 50$ using the estimator in equation (5).
- C_{11}: Sampling-controlled iteration with $a_n = a = 0.05$, $L^{(n)} = L^{(n-1)} + 10$ and $L_0 = 10$ using the estimator in equation (5).

We observe that, although the estimator (5) uses information collected since $t = 0$ and appears to be the natural choice for an $M/M/1$-based system, the resulting performance is no better than when Perturbation Analysis is used (curves C_1 and C_2). Interestingly, these results also indicate that the two formula-based estimators, (4) and (5) result in almost identical performance.

6. Conclusions

In this paper, we have defined and examined a load balancing problem with centralized scheduling. Our concerns focused on systems in which little is known about the nature of the arrival or service distributions and each associated parameter must be estimated. Several approaches were discussed and it was argued that a procedure interleaving both estimation and optimization was desirable. Two stochastic approximation schemes, the standard method as well as one introduced here, were presented to augment an optimization algorithm. The convergence of both these schemes was established and simulation results were presented that provided a basis for comparison. It was found that the second scheme, sampling-controlled iteration, in addition to its capability of removing biasedness in estimators, resulted in better performance than standard stochastic approximation. Additional results were presented for estimators in the case that the arrival process could be assumed Poisson.

We note that there has recently been considerable interest in the use of learning automata to solve non-linear optimization problems such as call routing [28,38] and scheduling [27]. The similarities between the control problems of call routing and centralized load balancing suggest the use of these automata for problem P_1. It has been shown [28,38], for example, that certain automata algorithms tend to equalize penalty probabilities. In problem P_1, this would correspond, loosely, to equalizing mean delays across the processors. However, this does not necessarily imply convergence to the optimum and our experience with simulating these algorithms has shown that even when convergence is theoretically possible, it is extremely slow.

For future work, we observe that it would be important to study the behavior of stochastic approximation procedures in the presence of non-stationary environments. In this case, recent results on weak convergence methods [22] suggest that small but fixed stepsizes may be used to track a moving environment. Next, in the case that no estimators for the derivatives are available, Kiefer-Wolfowitz methods [21,44], which are often used in such situations, need to be examined more carefully for the problem of load balancing. Finally, we note that, in a distributed system, it is possible to have several interacting schedulers and thus issues such as uniqueness of optimality and convergence must be addressed.

Acknowledgements:

The authors would like to thank Prof. Don Towsley of the Department of Computer Science at the University of Massachusetts for several insightful comments and valuable discussions. We also would like to thank Prof. Chris Cassandras of the Department of Electrical Engineering and Prof. Paul Dupuis of the Department of Mathematics and Statistics at University of Massachusetts for their useful feedback regarding this paper.

REFERENCES

[1] A.G.Barto and M.I.Jordan, "Gradient Following Without Back-propogation in Layered Networks", *Proc. IEEE First Annual Conf. on Neural Networks*, San Diego, California, 1987.

[2] D.Bertsekas, E.Gafni and R.Gallager, "Second Derivative Algorithms for Minimum Delay Distributed Routing in Networks," *IEEE Trans. on Communications*, Vol COM-32, No. 12, pp. 911-919, August 1984.

[3] H.D.Block, "Estimates of Error for Two Modifications of the Robbins-Munro Stochastic Approximation Process", *Ann. Math. Stat.*, Vol. 28, pp. 1003-1010.

[4] C.Cassandras, V.Abidi and D.Towsley, "Distributed Routing with On-line Marginal Delay Estimation", *Proc. 8th Int. Conf. on Distributed Computing Systems*, San Jose, 1988.

[5] F.Chang and L.Wu, "An Optimal Adaptive Routing Algorithm", *IEEE Trans. on Automatic Control*, pp. 690-700, Vol. AC-31, No. 8, Aug 1986.

[6] Y-C.Chow and W.H.Kohler, "Models for Dynamic Load Balancing in a Heterogeneous Multiple Processor System", *IEEE Trans. on Computers*, Vol. C-28, No. 5., May 1979, pp. 354-361.

[7] A.Dvoretzky, "On Stochastic Approximation," *Proc. 1959 Berkeley Symp. Math. Stat. and Prob.*, Vol 1, 1959.

[8] R.G.Gallager, "A Minimum Delay Routing Algorithm Using Distributed Computation," pp. 73-85, *IEEE Trans. on Communications*, Vol COM-25, 1977.

[9] B.Hajek, "Stochastic Approximation Methods for Decentralized Control of Multiaccess Communications", *IEEE Trans. on Communications*, pp. 176-184, Vol. IT-31, No.2, March 1985.

[10] G.Heal, "Planning Without Prices," *Rev. of Econ Studies*, Vol 36, pp. 346-362, 1969.

[11] P.Heidelberger and D.Towsley, "Sensitivity Analysis from Sample Paths Using Likelihoods", *COINS Technical Report TR87-131*, University of Massachusetts, Amherst, MA.

[12] P.Heidelberger, "Limitations of Infinitesimal Perturbation Analysis", *IBM Research Report RC 11891*, IBM Thomas J. Watson Research Center, Yorktown Heights, NY.

[13] Y.C.Ho, L.Servi and R.Suri, "A Class of Center-Free Resource Allocation Algorithms," *Large Scale Systems*, Vol 1, pp. 51-62, 1980.

[14] Y.C.Ho, X.R.Cao and C.Cassandras, "Finite and Infinitesimal Perturbation Analysis of Queueing Networks", *Automatica*, 19, (4), pp. 439-445.

[15] Y.C.Ho and C.Cassandras, "A New Approach to the Analysis of Discrete Event Systems", *Automatica*, Vol. 19, No. 2, pp. 149-167.

[16] R.L.Kasyap, C.C.Blaydon and K.S.Fu, "Stochastic Approximation", in J.M.Mendel and K.S.Fu, editors, *Adaptive, Learning and Pattern Recognition Systems: Theory and Applications*, Academic Press, New York, 1970.

[17] L.Kleinrock, "Queueing Systems, Vol. I: Theory", *Wiley*, 1975.

[18] H.Kobayashi, "Modeling and Analysis: An Introduction to System Performance and Evaluation Methodology", *Addison-Wesley*, Reading, Mass., 1978, Chapter 4.

[19] A.Kumar and F.Bonomi, "Adaptive Load Balancing in a Multiprocessor System with a Central Job Scheduler", *Proc. 2nd Int. Workshop on Applied Mathematics and Performance/Reliability Models of Computer Communication Systems*, Univ. of Rome II, pp. 173-188, 1987.

[20] J.Kurose and R.Simha, "A Microeconomic Approach to Optimal Resource Allocation in Distributed Computer Systems," *IEEE Trans. on Computers*, to appear.

[21] H.J.Kushner and D.S.Clark, "Stochastic Approximation Methods for Constrained and Unconstrained Systems," *Springer Verlag*, New York, 1978.

[22] H.J.Kushner, "Approximations and Weak Convergence Methods for Random Processes", *M.I.T. Press*, Cambridge, 1984.

[23] S.Lakshmivarahan, "Learning Algorithms: Theory and Applications", *Springer-Verlag*, New York.

[24] K.J.Lee, "Load Balancing in Distributed Computer Systems", Ph.D. Dissertation, Univ. of Massachusetts, Amherst.

[25] L.Ljung, "Analysis of Recursive Stochastic Algorithms," *IEEE Trans. on Automatic Control*, Vol AC-20, 1977, pp. 551-575.

[26] P.V.McGregor and R.R.Boorstyn, "Optimal Load Sharing in a Computer Network", *Proc. of ICC*, 1975.

[27] R.Mirchandaney and J.A.Stankovic, "Using Stochastic Learning Automata for Job Scheduling in Distributed Processing Systems", *J. Parallel and Distributed Computing*, Vol. 3, 1986, 527-552.

[28] O.V.Nedzelnitsky,Jr. and K.S.Narendra, "Nonstationary Models of Learning Automata Routing in Data Communication Networks," *IEEE Trans. on Systems, Man & Cybernetics*, Vol SMC-17, No. 6, Nov. 1987.

[29] L.M.Ni and K.Hwang, "Optimal Load Balancing in a Multiple Processor System With Many Job Classes", *IEEE Trans. on Software Engineering*, pp. 491-496, Vol. SE-11, No. 5, May 1985.

[30] W.R.Parzinski and P.W.Zipse, "Introduction to Mathematical Analysis", *McGraw-Hill*, 1982.

[31] S.Pulidas, D.Towsley and J.Stankovic, "Design of Efficient Parameter Estimators for Decentralized Load Balancing Policies", *Proc. 8th Int. Conf. on Distributed Computing Systems*, San Jose, 1988.

[32] M.I.Reiman and A.Weiss, "Sensitivity Analysis for Simulations Via Likelihood Ratios", *Operations Research*, to appear.

[33] H.Robbins and S.Munro, "A Stochastic Approximation Method," *Ann. Math. Stat.*, Vol 22, 1951, pp. 400-407

[34] V.K.Rohatgi, "An Introduction to Probability Theory and Mathematical Statistics", *Wiley*, 1976.

[35] D.Russell, "Optimization Theory," *W.A.Benjamin Inc.*, New York, 1970.

[36] R.Simha and J.F.Kurose, "Relative Reward Algorithms for Learning Automata", *IEEE Trans. on Systems, Man & Cybernetics*, to appear.

[37] R.Simha and J.F.Kurose, "Stochastic Approximation Schemes for a Resource Allocation Problem with Unknown Constraints", *COINS Technical Report TR 89-11*, University of Massachusetts, Amherst, MA, 1989.

[38] P.R.Srikantakumar and K.S.Narendra, "A Learning Model for Routing in Telephone Networks," *SIAM J. Control & Optimization*, Vol 20, No. 1, Jan 1982.

[39] T.E.Stern, "A Class of Decentralized Routing Algorithms Using Relaxation", *IEEE Trans. on Communications*, pp. 1092-1102, Vol. COM-25, No. 10, Oct 1977.

[40] R.Suri and M.A.Zazanis, "Perturbation Analysis Gives Strongly Consistent Estimates for the M/G/1 Queue", *Technical Report*, Division of Applied Sciences, Harvard University, Cambridge, MA.

[41] R.Suri, "Perturbation Analysis: The State of the Art and Research Issues Explained Via the GI/G/1 Queue", *Proc. IEEE*, to appear.

[42] A.N.Tantawi and D.Towsley, "Optimal Static Load Balancing in Distributed Computer Systems", *J. ACM*, Vol. 32, No. 2, April 1985, pp. 445-465.

[43] J.N.Tsitsiklis and D.P.Bertsekas, "Distributed Asynchronous Optimal Routing in Data Networks," *Proc. of the 23rd Conf. on Decision & Control*, 1984.

[44] M.T.Wasan, "Stochastic Approximation", *Cambridge Univ. Press*, 1969.

Appendix

In this section we present the proofs for Theorem 1 and Theorem 2 of the paper. We note that while several general convergence theorems have been proved for stochastic approximation algorithms [7,21,25], we believe that the nature of the computation of the feasibility set $A^{(n)}$ precludes direct applicability of these theorems and instead, we prove our results directly. The difficulty in applying of these general theorems directly is that the *sorting* operation in the computation of $A^{(n)}$ causes $\Delta x_i^{(n)}$ to be discontinuous in X. Also, this projection of the gradient is not the minimum distance projection and therefore Theorem 5.3.1 in [21] is not directly applicable.

Given that X^* is the optimal policy and the assumptions made about the nature of the mean delay function D, we write the Kuhn-Tucker conditions at X^* as follows:

$$\begin{aligned} f_i(x_i^*) &= f_j\left(x_j^*\right) \;\; \forall i,j \in B \\ f_i(x_i^*) &> f_j\left(x_j^*\right) \;\; \forall i \notin B, j \in B \end{aligned}$$

for some set $B \subset \{1,...,K\}$. We note that given our restriction that $\sum_{i=1}^{K} x_i = 1$, it is easy to show that $\forall i \in B$, $x_i^* = q_{min}$. We are now in a position to prove the two theorems.

PROOF OF THEOREM 1: Consider $i \in B$. It is possible to show (lemma 2 in [36]) that $\exists N$ such that whenever $n > N$ and $x_i^{(n)} < x_i^* - \epsilon$, then $E[\Delta x_i^{(n)}] > \delta_\epsilon > 0$ (assumption $A3$ is required here). Intuitively, this is equivalent to asserting that when $x_i^{(n)} < x_i^* - \epsilon$ then the change at step n is strictly positive in expectation, i.e., there is a tendency to move towards x_i^*. Thus, since $\sum_{n=1}^{\infty} a_n = \infty$, the set of sample paths for which $x_i^{(n)}$ converges to some number other than x_i^* occurs with probability zero.

Hence, we now consider only the remaining possibility that $x_i^{(n)}$ does not converge at all. In this case, there exists a set of sample paths that occur with strictly positive probability such that $\exists \xi, \eta, i$ with $\eta > \xi$ and integers $n_1, n_2, n_3, ...$ such that $x_i^{(n_1)} < x_i^* - \eta$, $x_i^{(n_2)} > x_i^* - \xi$, $x_i^{(n_3)} < x_i^* - \eta$, $x_i^{(n_4)} > x^* - \xi, ...$, i.e., $x_i^{(n)}$ crosses between the $x^* - \xi$ and $x^* - \eta$ boundary infinitely often. If there were no such η and ξ then it would mean that $x_i^{(n)} \rightarrow x_i^*$.

Next, let $\gamma(1), \gamma(2), ...$ be the stopping times at which the $x^* - \eta$ boundary is crossed and $\delta(1), \delta(2), ...$ be the stopping times when the $x^* - \xi$ boundary is crossed. From this definition we see that $\gamma(1) = n_1, \gamma(2) = n_3, ...$ and $\delta(1) = n_2, \delta(2) = n_4, ...$ Since $a_n \rightarrow 0$ and $\Delta x_i^{(n)}$ is bounded in $[x_i^* - \eta, x_i^* - \xi]$, we must have $\delta(n) - \gamma(n) \rightarrow \infty$. This implies that $\lim_{n \to \infty} \sum_{m=\gamma(n)}^{\delta(n)} \Delta x_i^{(m)} \rightarrow -\infty$ which contradicts the strong law of large numbers for the independent random variables $\Delta x_i^{(n_m)}$ for which $E[\Delta x_i^{(n_m)}] > 0$ (assumption $A2$). Hence the measure of the set of sample paths on which these infinite crossings occur is zero and therefore $x_i^{(n)} \rightarrow x_i^*$.

Note that above we assumed that crossings occured below x_i^*, i.e., between $x_i^* - \eta$ and $x_i^* - \xi$. We could just as easily assumed that the crossings occured above, between $x_i^* + \xi$ and $x_i^* + \eta$ and a similar proof would go through □.

In order to prove Theorem 2, we first prove a lemma below. In the remainder of the appendix, we focus on sampling-controlled iteration.

LEMMA 1: We have assumed that the estimator $\hat{f}^{(n)}_{iL^{(n)}}$ is strongly consistent. We may now state that for a given sample path on which the estimate converges, $\forall \epsilon > 0$, $\exists M$ such that $\forall n > M : \forall x_i$,

$$\left| \hat{f}^{(n)}_{iL^{(n)}}(x_i) - f_i(x_i) \right| < \epsilon$$

PROOF: Consider a particular sample path on which the estimate converges. By the strong consistency assumption, $A5$, for each x in the domain of x_i choose M_x such that $\forall n > M_x$,

$$\left| \hat{f}^{(n)}_{iL^{(n)}}(x) - f_i(x) \right| < \epsilon$$

By $A6$, let $N(x, \epsilon)$ be the neighbourhood of each x such that $\forall n > M_x$ and $\forall y \in N(x, \epsilon)$,

$$\left| \hat{f}^{(n)}_{iL^{(n)}}(y) - f_i(y) \right| < \epsilon$$

Now, the neighbourhoods $\{\forall n : N(x, \epsilon)\}$ form an open covering of the domain of x ($q_{min} \leq x \leq 1 - Kq_{min}$), which is compact. Hence, by the Heine-Borel theorem [30], there is a finite subcover, $N(x_1, \epsilon), N(x_2, \epsilon), ..., N(x_p, \epsilon)$. Let $M = \max\{M_{x_1}, ..., M_{x_p}\}$. Then $\forall n > M$ it is true that $\forall x$,

$$\left| \hat{f}^{(n)}_{iL^{(n)}}(x) - f_i(x) \right| < \epsilon$$

and therefore we get our result □.

PROOF OF THEOREM 2: Consider the iteration for the probability associated with a particular queue. We may write the following

$$\begin{aligned}
x_i^{(n+1)} &= x_i^{(n)} + a_n \Delta x_i^{(n)} \\
&= x_i^{(n)} - a_n \left(\hat{f}^{(n)}_{iL^{(n)}} - \frac{1}{|A^{(n)}|} \sum_{j \in A^{(n)}} \hat{f}^{(n)}_{jL^{(n)}} \right) \\
&= x_i^{(n)} - a_n \left(f_i - \frac{1}{|A^{(n)}|} \sum_{j \in A^{(n)}} f_j \right) + \sigma_i^{(n)} \\
&= y_i^{(n+1)} + \sigma_i^{(n)}
\end{aligned}$$

where $\sigma_i^{(n)}$ is a noise term such that $\lim_{n\to\infty} \sigma_i^{(n)} = 0$ almost surely and $y_i^{(n+1)}$ is defined by the rest of the expression.

We now focus on a particular iteration n and the case that $d(X^{(n)}) > \epsilon$ for some $\epsilon > 0$. We then examine the difference (at the n^{th} iteration) between the hypothetical case that there is no noise in the measurement and the case that there is noise. Informally, the proof below consists of showing two facts: first, if there were no noise then the iteration would result in a monotonic decrease in objective function; secondly, the noise can be made small enough (by considering large n) so that there is a monotonic decrease even when noise is present.

First, consider the deterministic case when $\hat{f}_{iL(n)}^{(n)} = f_i$ and let

$$d(X^{(n)}) = \max_{i,j \in B} |f_i(x_i) - f_j(x_j)|$$

where B is defined as earlier, in the Kuhn-Tucker conditions. It was shown in [20] that $\forall \epsilon > 0$ there exists a (as defined in assumption $A4$) such that for $d(X^{(n)}) > \epsilon$, $\Delta D(X^{(n)}) < \delta_\epsilon < 0$, i.e.,

$$D(X^{(n+1)}) - D(X^{(n)}) < \delta_\epsilon$$

In [20], we proved that since $\Delta D < \delta_\epsilon$ and $\epsilon > 0$ is arbitrary, monotonic decreases in the delay function eventually result in convergence to the optimum. This means that in the deterministic case, $d(X^{(n)}) \to 0$, i.e., convergence to a Kuhn-Tucker point.

We return to the case of interest in this paper in which $\hat{f}_{iL(n)}^{(n)}$ is a random variable. Now, since D is continuous,

$$\left| D(X^{(n+1)}) - D(Y^{(n+1)}) \right| < \eta_\sigma$$

where $\eta_\sigma \to 0$ as $\sigma \to 0$ (since we consider only those sample paths on which $\hat{f}_{iL(n)}^{(n)}$ converges to f_i). From Lemma 1, we can then choose M large enough so that σ is small enough to ensure that $\eta_\sigma < |\delta_\epsilon|$. Then,

$$D(X^{(n+1)}) - D(X^{(n)}) < \eta_\sigma - |\delta_\epsilon|$$

resulting in a monotonic decrease in the mean delay function. Since $\epsilon > 0$ is arbitrary, it follows that $d(X^{(n)}) \to 0$ and hence $X^{(n)}$ converges to X^* □.