AN ON-LINE ‘SURROGATE PROBLEM’ METHODOLOGY FOR STOCHASTIC DISCRETE RESOURCE ALLOCATION PROBLEMS[*]

Kagan Gokbayrak
Department of Manufacturing Engineering
Boston University
Boston, MA 02215
kgokbayr@bu.edu

Christos G. Cassandras
Department of Manufacturing Engineering
Boston University
Boston, MA 02215
cgc@bu.edu

Abstract

We consider stochastic discrete optimization problems where the decision variables are non-negative integers. We propose and analyze an *on-line* control scheme which transforms the problem into a “surrogate” continuous optimization problem and proceeds to solve the latter using standard gradient-based approaches while simultaneously updating both actual and surrogate system states. It is shown that the solution of the original problem is recovered as an element of the discrete state neighborhood of the optimal surrogate state. For the special case of separable cost functions, we show that this methodology becomes particularly efficient. Finally, convergence of the proposed algorithm is established under standard technical conditions and numerical results are included in the paper to illustrate the fast convergence properties of this approach.

1 Introduction

We consider stochastic discrete optimization problems where the decision variables are non-negative integers. In the context of resource allocation, in particular, classic problems of this type include: buffer allocation in queueing models of manufacturing systems or communication

[*] This work was supported in part by the National Science Foundation under Grants EEC-95-27422 and ACI-98-73339, by AFOSR under contract F49620-98-1-0387, by the Air Force Research Laboratory under contract F30602-99-C-0057 and by EPRI/ARO under contract WO8333-03.

networks [1],[2],[3], where a fixed number of buffers (the integer-valued decision variables) must be allocated over a fixed number of servers to optimize some performance metric; and the transmission scheduling problem in radio networks [4],[5], where a fixed number of time slots forming a "frame" must be allocated over several nodes. In the context of Discrete Event Systems (DES), integer-valued control variables have proven to be very common (e.g., as threshold parameters in many control policies), making the issue of optimizing over such variables of particular interest.

In the problems we consider in this paper, let $r \in Z_+^N$ be the decision vector or "state". In general, there is a set of feasible states denoted by A_d such that $r \in A_d$ represents a constraint. In a typical resource allocation setting, r_i denotes the number of resources that user i is assigned subject to a capacity constraint of the form

$$A_d = \left\{ r : \sum_{i=1}^{N} r_i = K \right\} \tag{1}$$

where K is the total number of *identical* resources and N is the number of users.

In a stochastic setting, let $L_d(r, \omega)$ be the cost incurred over a specific sample path ω when the state is r and $J_d(r) = E[L_d(r)]$ be the expected cost of the system operating under r. The sample space is $\Omega = [0,1]^\infty$, that is, $\omega \in \Omega$ is a sequence of random numbers from $[0,1]$ used to generate a sample path of the system. To be more specific, the cost functions are defined as $L_d : A_d \times \Omega \to \mathsf{R}$ and $J_d : A_d \to \mathsf{R}$, and the expectation is defined with respect to a probability space $(\Omega, \Im, P)$ where $\Im$ is an appropriately defined σ-field on Ω and P is a conveniently chosen probability measure. In the sequel, 'ω' will be dropped from $L_d(r, \omega)$ and, unless otherwise noted, all costs will be over the same sample path. Then, the discrete optimization problem we are interested in is the determination of $r^* \in A_d$ such that

$$J_d(r^*) = \min_{r \in A_d} J_d(r) = \min_{r \in A_d} E[L_d(r)] \tag{2}$$

In general, this is a notoriously hard stochastic integer programming problem. Even in a deterministic setting, where we may set $J_d(r) = L_d(r)$, this class of problems is NP-hard (see [6] [7] and references therein). In some cases, depending upon the form of the objective function $J_d(r)$ (e.g., separability, convexity), efficient algorithms based on finite-stage dynamic programming or generalized Lagrange relaxation methods are known (see [7] for a comprehensive discussion on aspects of deterministic resource allocation algorithms). Alternatively, if no a priori information is known about the structure of the problem, then some form of a search algorithm is employed (e.g., Simulated Annealing [8], Genetic Algorithms [9]). When the system operates in a stochastic environment (e.g., in a resource allocation setting users request resources at random time instants or hold a particular resource for a random period of time) and no closed-form expression for $E[L_d(r)]$ is available, the problem is further complicated by the need to estimate $E[L_d(r)]$. This generally requires Monte Carlo simulation or direct measurements made on the actual system.

While the area of stochastic optimization over *continuous* decision spaces is rich and usually involves gradient-based techniques as in several well-known stochastic approximation algorithms [10],[11], the literature in the area of *discrete* stochastic optimization is relatively limited. Most known approaches are based on some form of random search, with the added difficulty of having to estimate the cost function at every step. Such algorithms have been recently proposed by Yan and Mukai [12], Gong et al [13], Shi and Olafsson [14]. Another recent contribution to this area involves the *ordinal* optimization approach presented in [15]. For a class of resource allocation problems of the form (1)-(2), an approach of this type was used in [16] to develop a specific algorithm that moves one resource unit from the least sensitive user to the most sensitive, thus preserving feasibility at every iteration step. This algorithm was shown to converge in probability (and *a.s.* under certain added conditions [17]). Among other features, this approach is intended to exploit the fact that ordinal estimates are particularly robust with respect to estimation noise compared to cardinal estimates (see also [18]). The implication is that convergence of such algorithms is substantially faster. Even though the approach in [16] yields a fast resource allocation algorithm, it is still constrained to iterate so that every step involves the transfer of no more than a single resource from one user to some other user. One can expect, however, that much faster improvements can be realized in a scheme allowed to reallocate multiple resources from users whose cost-sensitivities are small to users whose sensitivities are much larger. This is precisely the rationale of most gradient-based continuous optimization schemes, where the gradient is a measure of this sensitivity.

With this motivation in mind, it is reasonable to pose the following question: Is it possible to transform a *discrete* optimization problem as in (2) into a “surrogate” *continuous* optimization problem, proceed to solve the latter using standard gradient-based approaches, and finally transform its solution into a solution of the original problem? Moreover, is it possible to design this process for *on-line* operation? That is, at every iteration step in the solution of the surrogate continuous optimization problem, is it possible to immediately transform the surrogate continuous state into a feasible discrete state r? This is crucial, since whatever information is used to drive the process (e.g., sensitivity estimates) can only be obtained from a sample path of the *actual* system operating under r.

This idea was used in [19] to solve the classical stochastic discrete resource allocation problem where there are K identical resources to be allocated over N user classes. Auxiliary control variables were introduced to denote the probability θ_i that any one resource is allocated to user i. A particular continuous optimization problem over $\theta = [\theta_1 \dots \theta_N]'$ was then solved and its solution was transformed into a solution of the original discrete optimization problem. In this paper, based on the ideas we introduced in [20], we propose and analyze a different and more general approach. In particular, we transform the original discrete set A_d into a continuous set over which a “surrogate” optimization problem is defined and subsequently solved. As in earlier work in [16], [19] and unlike algorithms presented in [7], an important feature of our approach is that every state r in the optimization process remains feasible, so that our scheme can be used *on line* to adjust the decision vector as operating conditions (e.g., system parameters) change over time. Thus, at every step of the continuous optimization process, the continuous state obtained is mapped back into a feasible discrete state; based on a realization under this feasible state, new sensitivity estimates are obtained that drive the surrogate problem to yield the next continuous state. The proposed scheme, therefore, involves an interplay of sensitivity-driven iterations and continuous-to-discrete state transformations. The key issue then is to show that when (and if) an optimal allocation is obtained in the continuous state space, the transformed discrete state is in fact r^* in (2).

The rest of the paper is organized as follows. In Section 2 we give an overview of our basic approach. In Section 3 we present the key results enabling us to transform a stochastic discrete resource allocation problem into a "surrogate" continuous resource allocation problem. In Section 4 we discuss the construction of appropriate "surrogate" cost functions for our approach and the evaluation of their gradients. Section 5 discusses how to recover the solution of the original problem from that of the "surrogate" problem. In Section 6 we describe how our approach generalizes to optimization problems with arbitrary feasible sets. Section 7 presents the detailed optimization algorithm and analyzes its convergence properties. Some numerical examples and applications are presented in Section 8.

2 Basic approach for on-line control

In the sequel, we shall adopt the following notational conventions. We shall use subscripts to indicate components of a vector (e.g., r_i is the ith component of r). We shall use superscripts to index vectors belonging to a particular set (e.g., r^j is the jth vector of the same form as r within a subset of A_d that contains such vectors). Finally, we reserve the index n as a subscript that denotes iteration steps and not vector components (e.g., r_n is the value of r at the nth step of an iterative scheme, not the nth component of r).

The expected cost function $J_d(r)$ is generally nonlinear in r, a vector of integer-valued decision variables, therefore (2) is a nonlinear integer programming problem. One common method for the solution of this problem is to relax the integer constraint on all r_i so that they can be regarded as continuous (real-valued) variables and then apply standard optimization techniques such as gradient-based algorithms.

Let the "relaxed" set A_c be the convex hull of the original constraint set A_d and define $L_c : A_c \times \Omega \to \mathsf{R}$ be the cost function over a specific sample path. As before let us drop 'ω' from $L_c(\rho, \omega)$ and agree on that unless otherwise noted all costs will be over the same sample path. The resulting "surrogate" problem then becomes: Find ρ^* that minimizes the "surrogate" expected cost function $J_c : A_c \to \mathsf{R}$ over the continuous set A_c, i.e.,

$$J_c(\rho^*) = \min_{\rho \in A_c} J_c(\rho) = \min_{\rho \in A_c} E[L_c(\rho)] \tag{3}$$

where $\rho \in \mathsf{R}_+^N$, is a real-valued state, and the expectation is defined on the same probability space $(\Omega, \Im, P)$ as described earlier. Assuming an optimal solution ρ^* can be determined, this state must then be mapped back into a discrete vector by some means (usually, some form of truncation). Even if the final outcome of this process can recover the actual r^* in (2), this approach is strictly limited to *off-line* analysis: When an iterative scheme is used to solve the problem in (3) (as is usually the case except for very simple problems of limited interest), a sequence of points $\{\rho_n\}$ is generated; these points are generally continuous states in A_c, hence they may be infeasible in the original discrete optimization problem. Moreover, if one has to estimate $E[L_c(\rho)]$ or $\frac{\partial E[L_c(\rho)]}{\partial \rho}$ through simulation, then a simulated model of the surrogate problem must be created, which is also not generally feasible. If, on the other hand, the only cost information available is through direct observation of sample paths of an actual system, then there is no obvious way to estimate $E[L_c(\rho)]$ or $\frac{\partial E[L_c(\rho)]}{\partial \rho}$, since this applies to the real-valued ρ, not the actual cost observable under integer-valued r.

In this paper we propose a different approach intended to operate *on line*. In particular, we still invoke a relaxation such as the one above, i.e., we formulate a surrogate continuous optimization problem with some state space $A_c \subset \mathsf{R}_+^N$ and $A_d \subset A_c$. However, at every step n of the iteration scheme involved in solving the problem, both the continuous and discrete states are simultaneously updated through a mapping of the form $r_n = f_n(\rho_n)$. This has two advantages: First, the cost of the original system is continuously adjusted (in contrast to an adjustment that would only be possible at the end of the surrogate minimization process); and second, it allows us to make use of information typically needed to obtain cost sensitivities from the actual operating system at every step of the process.

Before getting into details, we outline below the basic scheme we consider. Initially, we set the "surrogate system" state to be that of the actual system state, i.e.,

$$\rho_0 = r_0 \tag{4}$$

Subsequently, at the nth step of the process, let $H_n(r_n, \omega_n)$ denote an estimate of the sensitivity of the cost $J_c(\rho_n)$ with respect to ρ_n obtained over a sample path ω_n of the actual system operating under allocation r_n; details regarding this sensitivity estimate will be provided later in the paper. Two sequential operations are then performed at the nth step:

1. The continuous state ρ_n is updated through

$$\rho_{n+1} = \pi_{n+1}[\rho_n - \eta_n H_n(r_n, \omega_n)] \tag{5}$$

   where $\pi_{n+1} : \mathsf{R}^N \to A_c$ is a projection function so that $\rho_{n+1} \in A_c$ and η_n is a "step size" parameter.

2. The newly determined state of the surrogate system, ρ_{n+1}, is transformed into an actual feasible discrete state of the original system through

$$r_{n+1} = f_{n+1}(\rho_{n+1}) \tag{6}$$

   where $f_{n+1} : A_c \to A_d$ is a mapping of feasible continuous states to feasible discrete states which must be appropriately selected as will be discussed later.

One can recognize in (5) the form of a stochastic approximation algorithm (e.g., [21]) that generates a sequence $\{\rho_n\}$ aimed at solving (3). However, there is an additional operation (6) for generating a sequence $\{r_n\}$ which we would like to see converge to r^* in (2). It is important to note that $\{r_n\}$ corresponds to feasible realizable states based on which one can evaluate estimates $H_n(r_n, \omega_n)$ from observable data, i.e., a sample path of the actual system under r_n (not the surrogate state ρ_n). We can therefore see that this scheme is intended to combine the advantages of a stochastic approximation type of algorithm with the ability to obtain sensitivity estimates with respect to discrete decision variables. In particular, sensitivity estimation methods for discrete parameters based on Perturbation Analysis (PA) and Concurrent Estimation [22],[23] are ideally suited to meet this objective.

Before addressing the issue of obtaining estimates $H_n(r_n, \omega_n)$ necessary for the optimization scheme described above to work, there are two other crucial issues that form the cornerstones of the proposed approach. First, the selection of the mapping f_{n+1} in (6) must be specified. Second, a surrogate cost function $L_c(\rho, \omega)$ must be identified and its relationship to the actual cost $L_d(r, \omega)$ must be made explicit. These issues for a stochastic discrete resource allocation problem of the form (1)-(2) are discussed next, in Sections 3 and 4 respectively.

3 Continuous-to-discrete state transformations

In order to facilitate the presentation of the main concepts in our approach, we shall limit ourselves to the discrete feasible set A_d of the form (1), which corresponds to a large class of resource allocation problems subject to a total capacity constraint. We would like to stress the fact that our approach is more generally applicable to optimization problems with certain feasible set structures. We will return to the issue of generality of the approach in Section 6.

We define A_c as the convex hull of A_d:

$$A_c = \left\{ \rho : \sum_{i=1}^{N} \rho_i = K \right\} \tag{7}$$

Given a vector $\rho \in \mathsf{R}_+^N$, we begin by specifying a set F_ρ of mappings $f(\rho)$ as in (6). To do so, first define

$$I_\rho = \{ i \,|\, \rho_i \in \mathsf{Z}_+ \} \tag{8}$$

to be the set of components of ρ (i.e., user indices) that are strictly integer. Next, define

$$f_i^+(\rho) = \begin{cases} \rho_i & \text{if } i \in I_\rho \\ \lceil \rho_i \rceil & \text{otherwise} \end{cases} \quad \text{and } f_i^-(\rho) = \begin{cases} \rho_i & \text{if } i \in I_\rho \\ \lfloor \rho_i \rfloor & \text{otherwise} \end{cases} \tag{9}$$

where, for any $x \in \mathsf{R}$, $\lceil x \rceil$ and $\lfloor x \rfloor$ denote the ceiling (smallest integer $\geq x$) and floor (largest integer $\leq x$) of x respectively. Then, let

$$F_\rho = \{ f \,|\, f : A_c \to A_d, \forall i \; f_i(\rho) \in \{ f_i^+(\rho),\; f_i^-(\rho) \} \} \tag{10}$$

Note that for all $f \in F_\rho$ and $r \in A_d$, $f(r) = r$. The purpose of $f \in F_\rho$ is to transform some continuous state vector $\rho \in A_c$ into a "neighboring" discrete state vector $r \in A_d$ obtained by seeking $\lceil \rho_i \rceil$ or $\lfloor \rho_i \rfloor$ for each component $i = 1, \ldots, N$. With this definition of continuous-to-discrete state transformations, we can now define $\mathcal{N}(\rho)$, the set of all feasible neighboring discrete states of some $\rho \in A_c$:

Definition 1 *The set of all feasible discrete neighboring states of $\rho \in A_c$ is*

$$\mathcal{N}(\rho) = \{ r \,|\, r = f(\rho) \textit{ for some } f \in F_\rho \} \tag{11}$$

A more explicit and convenient characterization of the set $\mathcal{N}(\rho)$ is often possible by defining the *residual* vector $\tilde{\rho} \in [0,1)^N$ of the continuous state ρ, given by

$$\tilde{\rho} = \rho - \lfloor \rho \rfloor \tag{12}$$

where $\lfloor \rho \rfloor$ is the vector whose components are $\lfloor \rho \rfloor_i = \lfloor \rho_i \rfloor$.

For the case of (7), we set

$$m_\rho = \sum_{i=1}^{N} \tilde{\rho}_i = \sum_{i=1}^{N} (\rho_i - \lfloor \rho_i \rfloor) = K - \sum_{i=1}^{N} \lfloor \rho_i \rfloor \tag{13}$$

and note that $m_\rho \in \mathsf{Z}_+$ is an integer with the following convenient interpretation: If all users are assigned $\lfloor \rho_i \rfloor$ resources, then m_ρ is the total *residual resource capacity* to be allocated. Recalling the definition of the set I_ρ in (8), let $q = |I_\rho|$ be the number of components of ρ with strictly integer values. Then,

$$m_\rho \in \{0, \ldots, N - q - 1\} \tag{14}$$

Therefore, F_ρ may be interpreted as a set of mappings that allocate m_ρ residual resources over all $i \notin I_\rho$ in addition to a fixed integer $\lfloor \rho_i \rfloor$ already assigned to i. Let us then define

$$\tilde{r}^j(\rho) = [\tilde{r}_1^j, ..., \tilde{r}_N^j]', \quad \text{with } \tilde{r}_i^j \in \{0, 1\}$$

to be the jth discrete residual vector corresponding to some given ρ which satisfies

$$\sum_{i=1}^{N} \tilde{r}_i^j = m_\rho \text{ and } \tilde{r}_i^j = 0 \text{ for } i \in I_\rho$$

Thus, $\tilde{r}^j(\rho)$ is an N-dimensional vector with components 0 or 1 summing up to m_ρ. It is easy to see that the number of such distinct vectors, and hence the cardinality of the set $\mathcal{N}(\rho)$, is

$$|\mathcal{N}(\rho)| = \binom{N-q}{m_\rho} \tag{15}$$

It follows that we can write, for all $f^j \in F_\rho$,

$$f^j(\rho) = \lfloor \rho \rfloor + \tilde{r}^j(\rho) \tag{16}$$

In order to clarify the notation, we consider the following example, which we will use throughout our analysis:

Example: Consider the allocation problem of $K = 20$ resources over $N = 4$ users, and let $\rho = [1.9, 9.1, 6.1, 2.9]'$. The feasible set is

$$A_c = \left\{ \rho : \sum_{i=1}^{N} \rho_i = 20 \right\} \tag{17}$$

Since $\lfloor \rho \rfloor = [1, 9, 6, 2]'$, we have $\tilde{\rho} = [0.9, 0.1, 0.1, 0.9]'$ and $m_\rho = \sum_{i=1}^{N} \tilde{\rho}_i = 2$. Note that since ρ does not have any integer components, I_ρ is empty and $q = 0$.

The residual discrete vectors and the corresponding discrete neighbors, using (16), are

$$\begin{aligned}
\tilde{r}^1(\rho) &= [1,1,0,0]' \Rightarrow f^1(\rho) = [2,10,6,2]', & \tilde{r}^2(\rho) = [1,0,1,0]' \Rightarrow f^2(\rho) = [2,9,7,2]' \\
\tilde{r}^3(\rho) &= [1,0,0,1]' \Rightarrow f^3(\rho) = [2,9,6,3]', & \tilde{r}^4(\rho) = [0,1,1,0]' \Rightarrow f^4(\rho) = [1,10,7,2]' \\
\tilde{r}^5(\rho) &= [0,1,0,1]' \Rightarrow f^5(\rho) = [1,10,6,3]', & \tilde{r}^6(\rho) = [0,0,1,1]' \Rightarrow f^6(\rho) = [1,9,7,3]'
\end{aligned}$$

Therefore,

$$\mathcal{N}(\rho) = \{[2,10,6,2]', [2,9,7,2]', [2,9,6,3]', [1,10,7,2]', [1,10,6,3]', [1,9,7,3]'\}$$

Note that $|\mathcal{N}(\rho)| = \binom{N-q}{m_\rho} = \binom{4}{2} = 6$.

The following theorem is the main result of this section and it establishes the fact that any $\rho \in A_c$ can be expressed as a convex combination of points $r \in \mathcal{N}(\rho)$.

Theorem 3.1 *Any $\rho \in A_c$ is a convex combination of its discrete feasible neighboring states, i.e., there exists a vector α such that*

$$\rho = \sum_{j=1}^{M} \alpha_j r^j, \qquad \textit{with} \sum_{j=1}^{M} \alpha_j = 1, \;\; \alpha_j \geq 0 \quad \textit{for all } j = 1,..,M$$

where $M = |\mathcal{N}(\rho)|$ and $r^j \in \mathcal{N}(\rho)$, $j = 1, \ldots, M$.

Proof. Consider the residual vectors $\tilde{\rho} = \rho - \lfloor \rho \rfloor$ with $\sum_{i=1}^{N} \tilde{\rho}_i = m_\rho \leq N$, in which case the feasible neighboring state set $\mathcal{N}(\tilde{\rho})$ consists of the vectors

$$\tilde{r}^j = r^j - \lfloor \rho \rfloor, \quad j = 1, ..., M$$

Let e be an M-dimensional vector with all entries equal to 1. Then, consider the Linear Program (LP)

$$\min_{\alpha} e'\alpha \qquad \text{s.t.} \;\; \tilde{\mathbf{R}}\alpha = \tilde{\rho}, \;\; \alpha \geq 0 \tag{18}$$

where $\tilde{\mathbf{R}} = [\tilde{r}^1, ..., \tilde{r}^M]$ is the $N \times M$ matrix with all discrete feasible states in $\mathcal{N}(\tilde{\rho})$ as columns. The dual LP problem is

$$\max_{\gamma} \gamma'\tilde{\rho} \qquad \text{s.t.} \;\; \gamma'\tilde{\mathbf{R}} \leq e'$$

Note that in this dual problem, the constraint requires the sum of any m_ρ components (whose indices are not in I_ρ) of the dual vector γ to be less than or equal to 1. Since all components of $\tilde{\rho}$ are nonnegative, the dual optimal vector γ^* must make the constraint active, i.e., $\gamma^{*\prime}\tilde{\mathbf{R}} = e'$; otherwise, increasing any component of γ whose index is not in I_ρ increases the value of $\gamma'\tilde{\rho}$. It is then easy to verify that $\gamma'\tilde{\mathbf{R}} = e'$ yields an N-dimensional dual optimal vector

$$\gamma_i^* = \begin{cases} \frac{1}{m_\rho} & i \notin I_\rho \\ 0 & \text{otherwise} \end{cases}$$

Therefore, the optimal cost of the dual problem is $(\gamma^*)'\,\tilde{\rho} = 1$ which is equal to the optimal cost of the primal problem, i.e., $e'\alpha = \sum_{j=1}^{M} \alpha_j = 1$. Hence, the primal problem is feasible and there exists a solution α such that

$$\tilde{\rho} = \tilde{\mathbf{R}}\alpha = \sum_{j=1}^{M} \alpha_j \tilde{r}^j, \quad \alpha \geq 0, \quad \sum_{j=1}^{M} \alpha_j = 1$$

Finally, returning to the original vector ρ, we get

$$\begin{aligned} \rho &= \tilde{\rho} + \lfloor \rho \rfloor = \sum_{j=1}^{M} \alpha_j \tilde{r}^j + \sum_{j=1}^{M} \alpha_j \lfloor \rho \rfloor \\ &= \sum_{j=1}^{M} \alpha_j (\tilde{r}^j + \lfloor \rho \rfloor) = \sum_{j=1}^{M} \alpha_j r^j \end{aligned} \tag{19}$$

which completes the proof. ■

Remark. Note that since $\tilde{\rho} = \tilde{\mathbf{R}}\alpha = \sum_{j=1}^{M} \tilde{R}_j \alpha_j$, it follows that

$$\sum_{i=1}^{N} \tilde{\rho}_i = \sum_{i=1}^{N} \sum_{j=1}^{M} \tilde{R}_{ij} \alpha_j = m_\rho$$

Therefore, $\sum_{j=1}^{M} m_\rho \alpha_j = m_\rho$, which implies that $\sum_{j=1}^{M} \alpha_j = 1$, i.e., all feasible points in (18) are optimal.

Theorem 3.1 asserts that every $\rho \in A_c$ belongs to *conv*$(\mathcal{N}(\rho))$, the convex hull of the feasible neighboring state set $\mathcal{N}(\rho)$ defined in (11). We can further limit the set of states over which such a convex combination can be formed as follows.

Corollary 3.1 *Any $\rho \in A_c$ is a convex combination of at most $N-q$ discrete feasible neighboring states, i.e., there exists a vector α such that*

$$\rho = \sum_{j=1}^{N-q} \alpha_j r^j \quad \textit{with} \quad \sum_{j=1}^{N-q} \alpha_j = 1, \quad \alpha_j \geq 0 \textit{ for all } j = 1, .., N \tag{20}$$

where $r^j \in \mathcal{N}(\rho)$, $j = 1, \ldots, |\mathcal{N}(\rho)|$, $q = |I_\rho|$.

Proof. The feasibility constraint (7) and the constraints

$$r_i^j = \rho_i \text{ for } i \in I_\rho \text{ and } j = 1, \ldots, |\mathcal{N}(\rho)|$$

reduce the dimensionality of our space from R^N to R^{N-q-1}. By Caratheodory's fundamental theorem [24], every point in the convex hull of a set S in R^N is a convex combination of $N+1$ or fewer points of S. Using this theorem, we conclude that ρ can be written as a convex combination of at most $N-q$ discrete feasible neighbors r^j. ■

Based on this corollary, we define a subset of $\mathcal{N}(\rho)$ in (11) as follows:

Definition 2 *$\mathcal{N}_{N-q}(\rho)$ is a subset of $\mathcal{N}(\rho)$ that contains $N-q$ (with $q=|I_\rho|$) linearly independent discrete neighboring states whose convex hull includes ρ.*

The existence of this set is guaranteed by the previous corollary and it plays a crucial role in our approach, because the mapping $f_n(\rho_n)$ that we select in (6) will be an element of $\mathcal{N}_{N-q}(\rho_n)$. Therefore, it is important to be able to identify $N-q$ elements of $\mathcal{N}(\rho)$ that satisfy (20), and hence determine $\mathcal{N}_{N-q}(\rho_n)$. The Simplex Method (e.g., see [24]) of Linear Programming (LP) provides one way to accomplish this as follows. We define $\mathbf{R}=[r^1 \, ...r^M]$ as the $N \times M$ matrix with all discrete feasible states in $\mathcal{N}(\rho)$ as columns and solve the following LP problem

$$\min_s e's \qquad \text{s.t.} \quad [\mathbf{R}\ \mathbf{I}] \begin{bmatrix} \alpha \\ s \end{bmatrix} = \rho, \quad \alpha, s \geq 0 \tag{21}$$

with the initial basic feasible solution $[\alpha' \ s']_0 = [0 \ \rho']$ to start the Simplex. Since we know (by Theorem 3.1) that $\mathbf{R}\alpha=\rho$, $\alpha \geq 0$ has a solution, we can drive s components to zero. The nonzero components of the resulting basic feasible solution are associated with vectors $r \in \mathcal{N}(\rho)$ and they are linearly independent by the Simplex Method. The set $\mathcal{N}_{N-q}(\rho)$ will be formed by these vectors. Note that $\mathcal{N}_{N-q}(\rho)$ is not unique because different column and row selections are possible in the Simplex Method.

Given this "reduced" set of discrete feasible neighbors of ρ, $\mathcal{N}_{N-q}(\rho)$, we restrict our original set of continuous-to-discrete transformations F_ρ in (10) to

$$\mathcal{F}_\rho = \{f : f(\rho) \in \mathcal{N}_{N-q}(\rho)\}$$

Note that when the continuous state is ρ_n, the continuous-to-discrete mapping f_n will be an element of $\mathcal{F}_{\rho_n}$. In order to clarify the new notation let us continue with the previous example:

Example (Continued): Recall that $q=|I_\rho|$ was zero, therefore $\mathcal{N}_{N-q}(\rho)=\mathcal{N}_4(\rho)$ will have 4 elements from the previously obtained set

$$\mathcal{N}(\rho) = \{[2,10,6,2]', [2,9,7,2]', [2,9,6,3]', [1,10,7,2]', [1,10,6,3]', [1,9,7,3]'\}$$

To determine a set $\mathcal{N}_4(\rho)$, we can write ρ as

$$\rho = [1.9, 9.1, 6.1, 2.9]' = 0.1[2,10,6,2]' + 0.8[2,9,6,3]' + 0.1[1,9,7,3]'$$

which is a convex combination of only three elements of $\mathcal{N}(\rho)$. Since $[2,9,7,2]'$ is linearly independent of these three discrete neighbors, $\mathcal{N}_4(\rho)$ can be selected to be

$$\mathcal{N}_4(\rho) = \{[2,10,6,2]', [2,9,6,3]', [1,9,7,3]', [2,9,7,2]'\} \tag{22}$$

Alternatively,

$$\rho = 0.1[2,9,7,2]' + 0.8[2,9,6,3]' + 0.1[1,10,6,3]'$$

and $[1,9,7,3]'$ is linearly independent of these three discrete neighbors, so another possible $\mathcal{N}_4(\rho)$ is

$$\mathcal{N}_4(\rho) = \{[1,10,6,3]', [2,9,6,3]', [1,9,7,3]', [2,9,7,2]'\} \tag{23}$$

It should be clear that $\mathcal{N}_{N-q}(\rho)$ *may not be unique.*

4 Construction of surrogate cost functions and their gradients

Since our approach is based on iterating over the continuous state $\rho \in A_c$, yet drive the iteration process with information involving $L_d(r)$ obtained from a sample path under r, we must establish a relationship between $L_d(r)$ and $L_c(\rho)$. The choice of $L_c(\rho)$ is rather flexible and may depend on information pertaining to a specific model and the nature of the given cost $L_d(r)$.

As seen in the previous section, it is possible that some components of ρ are integers, in which case the set I_ρ is non-empty and we have $q = |I_\rho| > 0$. In order to avoid the technical complications due to such integer components, let us agree that whenever this is the case we will perturb these components to obtain a new state $\hat{\rho}$ such that $I_{\hat{\rho}} = \emptyset$. In what follows, we will assume that all states ρ either have $I_\rho = \emptyset$ or have already been perturbed and relabeled ρ so that $I_\rho = \emptyset$. Since q is going to be zero, we will rename $\mathcal{N}_{N-q}(\rho)$ as $\mathcal{N}_N(\rho)$.

We shall select a surrogate cost function $L_c(\rho)$ to satisfy the following two conditions:

(C1): *Consistency*: $L_c(r) = L_d(r)$ for all $r \in A_d$.

(C2): *Piecewise Linearity:* $L_c(\rho)$ is a linear function of ρ over $conv(\mathcal{N}_N(\rho))$.

Consistency is an obvious requirement for $L_c(\rho)$. Piecewise linearity is chosen for convenience, since manipulating linear functions over $conv(\mathcal{N}_N(\rho))$ simplifies analysis, as will become clear in the sequel.

Given some state $\rho \in A_c$ and cost functions $L_d(r^j)$ for all $r^j \in \mathcal{N}_N(\rho)$, it follows from **(C2)** and (20) in Corollary 3.1 that we can write

$$L_c(\rho) = \sum_{j=1}^{N} \alpha_j L_c(r^j)$$

with $\sum_{j=1}^{N} \alpha_j = 1$, $\alpha_j \geq 0$ for all $j = 1, .., N$. Moreover, by **(C1)**, we have

$$L_c(\rho) = \sum_{j=1}^{N} \alpha_j L_d(r^j) \tag{24}$$

that is, $L_c(\rho)$ is a convex combination of the costs of N discrete feasible neighbors. Note that $L_c(\rho)$ depends on $\mathcal{N}_N(\rho)$, which may not be unique therefore, $L_c(\rho)$ may not be unique.

Next, if we are to successfully use the iterative scheme described by (5)-(6), we need information of the form $H_n(r_n, \omega_n)$ following the nth step of the on-line optimization process. Typically, this information is contained in an estimate of the gradient. Our next objective, therefore, is to seek the sample gradient $\nabla L_c(\rho)$ expressed in terms of directly observable sample path data.

4.1 Gradient evaluation

Since $L_c(\rho)$ is a linear function on the convex hull defined by the N discrete neighbors in (24), one can write

$$L_c(\rho) = \sum_{i=1}^{N} \beta_i \rho_i + \beta_0 \tag{25}$$

for some $\beta_i \in \mathsf{R}$, $i = 0, ..., N$. Moreover, due to the linearity of $L_c(\rho)$ in $conv(\mathcal{N}_N(\rho))$, we have

$$\beta_i = \frac{\partial L_c}{\partial \rho_i}, \ i = 1, ..., N \tag{26}$$

Note that the β_i values depend on the selection of $\mathcal{N}_N(\rho)$ which, as already pointed out, may not be unique.

For any discrete feasible neighboring state $r^j \in \mathcal{N}_N(\rho)$, one can use (25) and **(C1)** to write

$$L_d(r^j) = \sum_{i=1}^{N} \beta_i r_i^j + \beta_0, \ j = 1, \ldots, N \tag{27}$$

Letting

$$\nabla L_c(\rho)' = [\beta_1, \ldots, \beta_N]$$

be the gradient of $L_c(\rho)$, our objective is to obtain an expression for $\beta_1, \ldots, \beta_N$ (not β_0) in terms of $L_d(r^j)$. Note that $L_d(r^j)$ are costs that can be evaluated at feasible states $r^j \in \mathcal{N}_N(\rho)$. These may be obtained by direct simulation; however, they are not available if a system is operating on line under one of these states, say r^1. This is where techniques such as Concurrent Estimation and Perturbation Analysis mentioned earlier can be used to facilitate this task.

To obtain expressions for $\beta_1, \ldots, \beta_N$ in terms of $L_d(r^j)$, let r^1 be the current state of the system (without loss of generality), and define

$$\delta_i^{j,1} = r_i^j - r_i^1 = \begin{cases} -1 & \text{if } r_i^1 > r_i^j \\ 1 & \text{if } r_i^1 < r_i^j \\ 0 & \text{otherwise} \end{cases} \tag{28}$$

and

$$\Delta L_{j,1} = L_d(r^j) - L_d(r^1) \tag{29}$$

for all $i = 1, \ldots, N$ and $j = 2, \ldots, N$. Using (27), the last equation can be rewritten as

$$\Delta L_{j,1} = \sum_{i=1}^{N} \beta_i (r_i^j - r_i^1) = \sum_{i=1}^{N} \beta_i \delta_i^{j,1} \tag{30}$$

If all $L_d(r^j)$ in (29) are observable, then (30) provides $N - 1$ linearly independent equations for the N variables $\beta_1, \ldots, \beta_N$. An additional equation is obtained as follows. Recall (5) with $H_n(r_n, \omega_n) = \nabla L_c(\rho_n)$:

$$\rho_{n+1} = \pi_{n+1}[\rho_n - \eta_n \nabla L_c(\rho_n)]$$

and let

$$\bar{\rho}_{n+1} = \rho_n - \eta_n \nabla L_c(\rho_n)$$

be an "intermediate" state prior to applying the projection π_{n+1}. In order to force $\bar{\rho}_{n+1}$ to satisfy the total capacity constraint (7), i.e.,

$$\sum_{i=1}^{N} \left(\bar{\rho}_{n+1}\right)_i = \sum_{i=1}^{N} (\rho_n)_i - \eta_n \sum_{i=1}^{N} \frac{\partial L_c(\rho_n)}{\rho_i} = K$$

we require that

$$\sum_{i=1}^{N} \frac{\partial L_c(\rho_n)}{\rho_i} = \sum_{i=1}^{N} \beta_i = 0 \tag{31}$$

Note that this additional equation is particularly convenient because it allows us to use an identity mapping for the projection π_{n+1}, i.e., $\rho_{n+1} = \bar{\rho}_{n+1} \in A_c$, provided that η_n is selected small enough to maintain the additional component constraints $0 \leq \left(\bar{\rho}_{n+1}\right)_i \leq K$ for all $i = 1, ..., N$.

The combination of (30) and (31) provides N equations used to determine unique $\beta_1, \ldots, \beta_N$. Specifically, define the $(N-1)$-dimensional vector

$$\Delta L' = \left[\Delta L_{2,1}, \ldots, \Delta L_{N,1}\right]$$

whose components were defined in (29), and the $(N-1) \times N$ matrix

$$\mathbf{\Delta R} = \begin{bmatrix} \delta_1^{2,1} & \cdots & \delta_N^{2,1} \\ \vdots & & \vdots \\ \delta_1^{N,1} & \cdots & \delta_N^{N,1} \end{bmatrix}$$

whose rows are the vectors $\delta^{j,1} = \left[\delta_1^{j,1}, \cdots, \delta_N^{j,1}\right]'$ as defined in (28). Therefore, given r^1 and some $\mathcal{N}_N(\rho)$, $\mathbf{\Delta R}$ is fixed. We then get from (30) and (31):

$$\begin{bmatrix} \mathbf{\Delta R} \\ e' \end{bmatrix} \nabla L_c(\rho) = \begin{bmatrix} \Delta L \\ 0 \end{bmatrix}$$

where e, as before, is an N-dimensional vector with all entries equal to 1. It follows that

$$\nabla L_c(\rho) = \begin{bmatrix} \mathbf{\Delta R} \\ e' \end{bmatrix}^{-1} \begin{bmatrix} \Delta L \\ 0 \end{bmatrix} \tag{32}$$

Therefore, $\nabla L_c(\rho)$, the sample gradient to be used as an estimate of $\nabla J_c(\rho)$ in (3), is obtained through the N costs $L_d(r^1), \ldots, L_d(r^N)$. The sample path at our disposal corresponds to one of the state vectors, which we have taken to be $r^1 \in \mathcal{N}_N(\rho)$, so that $L_d(r^1)$ is observable; the remaining $N-1$ costs therefore need to be obtained by some other means. One possibility is to perform $N-1$ simulations, one for each of these states. This, however, is not attractive for an on-line methodology. Fortunately, there exist several techniques based on Perturbation Analysis (PA) [22],[23] or Concurrent Estimation [1], which are ideally suited for this purpose; that is, based on observations of a sample path under r^i, one can evaluate $L_d(r^j)$ for states $r^j \neq r^i$

with limited extra effort. The efficiency of these techniques depends on the nature of the system and cost function. In the next section, we will examine a specific class of problems where the evaluation of $\nabla L_c(\rho)$ in (32) is greatly simplified and standard PA techniques may be used to evaluate neighboring state costs with minimal effort.

Example (Continued): Consider the previous allocation problem of $K = 20$ resources over $N = 4$ users so as to minimize the cost function

$$J_d(r) = -\prod_{i=1}^{N} r_i$$

Assume that the current continuous state is $\rho = [1.9, 9.1, 6.1, 2.9]'$ and the discrete state obtained through some mapping $f(\rho) \in \mathcal{F}_\rho$ is $r = [2, 9, 6, 3]'$. We previously obtained two different possible $\mathcal{N}_4(\rho)$ sets. Let us use the one in (22):

$$\mathcal{N}_4(\rho) = \{[2, 10, 6, 2]', [2, 9, 6, 3]', [1, 9, 7, 3]', [2, 9, 7, 2]'\}$$

with $\alpha = [0.1, 0.8, 0.1, 0.0]$. The continuous cost function $J_c(\rho)$ is defined as

$$J_c(\rho) = \sum_{j=1}^{N} \alpha_j J_d(r^j) = -(24 + 259.2 + 18.9) = -302.1$$

and using (32):

$$\begin{aligned}
\nabla J_c(\rho) &= \begin{bmatrix} \mathbf{\Delta R} \\ e' \end{bmatrix}^{-1} \begin{bmatrix} \Delta J \\ 0 \end{bmatrix} \\
&= \begin{bmatrix} 0 & 1 & 0 & -1 \\ -1 & 0 & 1 & 0 \\ 0 & 0 & 1 & -1 \\ 1 & 1 & 1 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 84 \\ 135 \\ 72 \\ 0 \end{bmatrix} = \begin{bmatrix} -86.25 \\ 60.75 \\ 48.75 \\ -23.25 \end{bmatrix}
\end{aligned} \tag{33}$$

If, on the other hand, we use the set in (23):

$$\mathcal{N}_4(\rho) = \{[1, 10, 6, 3]', [2, 9, 6, 3]', [1, 9, 7, 3]', [2, 9, 7, 2]'\}$$

with $\alpha = [0.1, 0.8, 0.0, 0.1]$ then,

$$J_c(\rho) = \sum_{j=1}^{N} \alpha_j J_d(r^j) = -302.4$$

and

$$\begin{aligned}
\nabla J_c(\rho) &= \begin{bmatrix} \mathbf{\Delta R} \\ e' \end{bmatrix}^{-1} \begin{bmatrix} \Delta J \\ 0 \end{bmatrix} \\
&= \begin{bmatrix} -1 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & 0 & 1 & -1 \\ 1 & 1 & 1 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 144 \\ 135 \\ 72 \\ 0 \end{bmatrix} = \begin{bmatrix} -85.5 \\ 58.5 \\ 49.5 \\ -22.5 \end{bmatrix}
\end{aligned} \tag{34}$$

This example illustrates the fact that $\nabla L_c(\rho)$ is generally *not unique*, since it depends on the selected (not necessarily unique) set $\mathcal{N}_N(\rho)$, and serves to highlight the flexibility inherent in our approach. As we shall see in Section 5, regardless of the choice of $\mathcal{N}_N(\rho)$ throughout each step of the iterative process (5)-(6), the optimal state r^* is recovered from the state ρ^*, the solution of the surrogate problem (3).

Finally, note that the properties of $\nabla L_c(\rho)$ in (32) as an estimator of $\nabla J_c(\rho)$ reduce to the basic stochastic properties of the system itself: Since $\nabla L_c(\rho)$ is simply a linear combination of costs $L_d(r^1), \ldots, L_d(r^N)$, $\nabla L_c(\rho)$ is a strongly consistent estimator of $\nabla J_c(\rho)$ as long as standard ergodicity conditions are satisfied.

4.2 Separable cost functions

Suppose that the cost function, $L_d(\cdot)$, is *separable* in the sense that it is a sum of component costs each dependent on its local state only, i.e., let

$$L_d(r) = \sum_{i=1}^{N} L_{d,i}(r_i) \tag{35}$$

In this case, our approach is significantly simplified and is easily applicable to *arbitrary* constraint sets as we will see in Theorem 4.1. We begin with a technical lemma, which holds regardless of the nature of $L_d(\cdot)$.

Lemma 4.1 *If $\rho = \sum_{j=1}^{N} \alpha_j r^j$, where $r_i^j \in \{\lfloor \rho_i \rfloor, \lceil \rho_i \rceil\}$ and $\sum_{j=1}^{N} \alpha_j = 1$, then the ith component of the residual vector of ρ, can be written as*

$$\tilde{\rho}_i = \rho_i - \lfloor \rho_i \rfloor = \sum_{\substack{j=1 \\ r_i^j = \lfloor \rho_i \rfloor + 1}}^{N} \alpha_j$$

Proof. Since $\rho = \sum_{j=1}^{N} \alpha_j r^j$, we have $\rho_i = \sum_{j=1}^{N} \alpha_j r_i^j$ and it follows that

$$\tilde{\rho}_i = \rho_i - \lfloor \rho_i \rfloor = \sum_{j=1}^{N} \alpha_j (r_i^j - \lfloor \rho_i \rfloor)$$

Since $r_i^j \in \{\lfloor \rho_i \rfloor, \lceil \rho_i \rceil\}$, we get

$$\tilde{\rho}_i = \sum_{\substack{j=1 \\ r_i^j = \lfloor \rho_i \rfloor}}^{N} \alpha_j (r_i^j - \lfloor \rho_i \rfloor) + \sum_{\substack{j=1 \\ r_i^j = \lfloor \rho_i \rfloor + 1}}^{N} \alpha_j (r_i^j - \lfloor \rho_i \rfloor) = \sum_{\substack{j=1 \\ r_i^j = \lfloor \rho_i \rfloor + 1}}^{N} \alpha_j$$

which completes the proof. ■

Note that in the lemma a state r^j is not required to be feasible and the existence of such states is still guaranteed by Theorem 3.1 and Corollary 3.1. Thus, expressing ρ in the form $\rho = \sum_{j=1}^{N} \alpha_j r^j$ with $\sum_{j=1}^{N} \alpha_j = 1$ is always possible. Using this lemma, one can prove the following:

Theorem 4.1 *If $L_d(\cdot)$, is separable and $\rho = \sum_{j=1}^{N} \alpha_j r^j$ where $r_i^j \in \{\lfloor \rho_i \rfloor, \lceil \rho_i \rceil\}$ and $\sum_{j=1}^{N} \alpha_j = 1$, then*

$$\frac{\partial L_c}{\partial \rho_i} = L_{d,i}(\lfloor \rho_i \rfloor + 1) - L_{d,i}(\lfloor \rho_i \rfloor) \tag{36}$$

and

$$L_c(\rho) = L_d(r) + \sum_{i=1}^{N} (\rho_i - r_i)[L_{d,i}(\lfloor \rho_i \rfloor + 1) - L_{d,i}(\lfloor \rho_i \rfloor)] \tag{37}$$

for any discrete feasible neighbor r.

Proof. Suppose that the current feasible state is r when the surrogate state is ρ. Using (24) and (35),

$$\begin{aligned} L_c(\rho) &= \sum_{j=1}^{N} \alpha_j L_d(r^j) = \sum_{j=1}^{N} \sum_{i=1}^{N} \alpha_j L_{d,i}(r_i^j) \\ &= \sum_{j=1}^{N} \sum_{i=1}^{N} \alpha_j [L_{d,i}(r_i^j) - L_{d,i}(\lfloor \rho_i \rfloor)] + \sum_{j=1}^{N} \sum_{i=1}^{N} \alpha_j L_{d,i}(\lfloor \rho_i \rfloor) \\ &= \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_j [L_{d,i}(r_i^j) - L_{d,i}(\lfloor \rho_i \rfloor)] + \sum_{i=1}^{N} L_{d,i}(\lfloor \rho_i \rfloor) \end{aligned}$$

Since $r_i^j \in \{\lfloor \rho_i \rfloor, \lceil \rho_i \rceil\}$,

$$\begin{aligned} L_c(\rho) &= \sum_{i=1}^{N} \sum_{\substack{j=1 \\ r_i^j = \lfloor \rho_i \rfloor + 1}}^{N} \alpha_j [L_{d,i}(\lfloor \rho_i \rfloor + 1) - L_{d,i}(\lfloor \rho_i \rfloor)] + \sum_{i=1}^{N} \sum_{\substack{j=1 \\ r_i^j = \lfloor \rho_i \rfloor}}^{N} \alpha_j [L_{d,i}(\lfloor \rho_i \rfloor) - L_{d,i}(\lfloor \rho_i \rfloor)] \\ &\quad + \sum_{i=1}^{N} L_{d,i}(\lfloor \rho_i \rfloor) \\ &= \sum_{i=1}^{N} \left(\sum_{\substack{j=1 \\ r_i^j = \lfloor \rho_i \rfloor + 1}}^{N} \alpha_j \right) [L_{d,i}(\lfloor \rho_i \rfloor + 1) - L_{d,i}(\lfloor \rho_i \rfloor)] + \sum_{i=1}^{N} L_{d,i}(\lfloor \rho_i \rfloor) \end{aligned}$$

By Lemma 4.1,

$$L_c(\rho) = \sum_{i=1}^{N} \tilde{\rho}_i [L_{d,i}(\lfloor \rho_i \rfloor + 1) - L_{d,i}(\lfloor \rho_i \rfloor)] + \sum_{i=1}^{N} L_{d,i}(\lfloor \rho_i \rfloor)$$

Adding and subtracting the cost at the current feasible point we get

$$L_c(\rho) = \sum_{i=1}^{N} \tilde{\rho}_i [L_{d,i}(\lfloor \rho_i \rfloor + 1) - L_{d,i}(\lfloor \rho_i \rfloor)] + \sum_{i=1}^{N} L_{d,i}(\lfloor \rho_i \rfloor) - L_d(r) + L_d(r)$$

¿From (35) it follows that

$$L_c(\rho) = L_d(r) + \sum_{i=1}^{N} [\tilde{\rho}_i L_{d,i}(\lfloor \rho_i \rfloor + 1) - \tilde{\rho}_i L_{d,i}(\lfloor \rho_i \rfloor) + L_{d,i}(\lfloor \rho_i \rfloor) - L_{d,i}(r_i)]$$

Since r is a feasible neighbor,

$$\begin{aligned} L_c(\rho) &= L_d(r) + \sum_{\substack{i=1 \\ r_i = \lfloor \rho_i \rfloor + 1}}^{N} (\tilde{\rho}_i - 1)[L_{d,i}(\lfloor \rho_i \rfloor + 1) - L_{d,i}(\lfloor \rho_i \rfloor)] \\ &\quad + \sum_{\substack{i=1 \\ r_i = \lfloor \rho_i \rfloor}}^{N} \tilde{\rho}_i [L_{d,i}(\lfloor \rho_i \rfloor + 1) - L_{d,i}(\lfloor \rho_i \rfloor)] \end{aligned}$$

Observe that $\tilde{\rho}_i - 1 = \rho_i - (\lfloor \rho_i \rfloor + 1) = \rho_i - r_i$ when $r_i = \lfloor \rho_i \rfloor + 1$. Similarly, when $r_i = \lfloor \rho_i \rfloor$ we get $\tilde{\rho}_i = \rho_i - \lfloor \rho_i \rfloor = \rho_i - r_i$. Thus,

$$\begin{aligned} L_c(\rho) &= L_d(r) + \sum_{\substack{i=1 \\ r_i = \lfloor \rho_i \rfloor + 1}}^{N} (\rho_i - r_i)[L_{d,i}(\lfloor \rho_i \rfloor + 1) - L_{d,i}(\lfloor \rho_i \rfloor)] \\ &\quad + \sum_{\substack{i=1 \\ r_i = \lfloor \rho_i \rfloor}}^{N} (\rho_i - r_i)[L_{d,i}(\lfloor \rho_i \rfloor + 1) - L_{d,i}(\lfloor \rho_i \rfloor)] \\ &= L_d(r) + \sum_{i=1}^{N} (\rho_i - r_i)[L_{d,i}(\lfloor \rho_i \rfloor + 1) - L_{d,i}(\lfloor \rho_i \rfloor)] \end{aligned}$$

Therefore, (36) immediately follows:

$$\frac{\partial L_c}{\partial \rho_i} = L_{d,i}(\lfloor \rho_i \rfloor + 1) - L_{d,i}(\lfloor \rho_i \rfloor)$$

■

Note that the theorem is proven for arbitrary feasible sets, not necessarily of the form (7).

In order to evaluate $\frac{\partial L_c}{\partial \rho_i}$ while operating at a discrete neighbor r, one will need to evaluate the effect of decreasing or increasing the number of resources allocated to user i. There is a number of PA techniques developed precisely for this type of sensitivity estimation problem; for example, estimating the sensitivity of packet loss in a radio network with respect to adding/removing a transmission slot available to the ith user [4],[25]. In [26] a PA technique is used together with our methodology to solve a call admission problem (with a separable cost function) over a communication network where there are capacity constraints on each node, but there is no total capacity constraint for the network. This set of constraints define a different type of feasible set than the one considered so far in this paper.

5 Recovery of optimal discrete states

Our ultimate goal remains the solution of (2), that is the determination of $r^* \in A_d$ that solves this optimization problem. Our approach is to solve (3) by iterating on $\rho \in A_c$ and, at each step, transforming ρ through some $f \in \mathcal{F}_\rho$. The connection between ρ and $r = f(\rho)$ for each step is therefore crucial, as is the relationship between ρ^* and $f(\rho^*)$ when and if this iterative process comes to an end identifying a solution ρ^* to the surrogate problem (3). The following theorem identifies a key property of feasible neighboring states of an *optimal* surrogate state ρ^*.

Theorem 5.1 *Let ρ^* minimize $L_c(\rho)$ over A_c. Then, there exists a discrete feasible neighboring state $r^* \in \mathcal{N}_N(\rho^*)$ which minimizes $L_d(r)$ over A_d and satisfies $L_d(r^*) = L_c(\rho^*)$*

Proof. We begin by noting that $A_d \subset A_c$ and $L_c(r) = L_d(r)$ for any $r \in A_d$. Therefore, if

$$L_d(r^*) = \min_{r \in A_d} L_d(r) \tag{38}$$

then

$$L_d(r^*) \geq L_c(\rho^*) \tag{39}$$

The optimal surrogate state ρ^* satisfies

$$L_c(\rho^*) = \min_{\rho \in A_c} L_c(\rho) = \sum_{j=1}^{N} \alpha_j L_c(r^j) = \sum_{j=1}^{N} \alpha_j L_d(r^j)$$

where we have used (24), to write $L_c(\rho)$ as a convex combination of the costs of N feasible discrete neighboring states $r^j \in \mathcal{N}_N(\rho)$. Thus, $\sum_{j=1}^{N} \alpha_j = 1$, $\alpha_j \geq 0$ for $j = 1, .., N$.

Let $s \in \{1, .., N\}$ be such that

$$L_d(r^s) = \min_{j=1,..,N} L_d(r^j) \tag{40}$$

Then,

$$L_c(\rho^*) = \sum_{j=1}^{N} \alpha_j L_d(r^j) \geq \sum_{j=1}^{N} \alpha_j L_d(r^s) = L_d(r^s)$$

In view of (39), we then get

$$L_d(r^s) \leq L_c(\rho^*) \leq L_d(r^*)$$

and since $L_d(r^*) \leq L_d(r^s)$ from (38), it also follows that

$$L_d(r^*) = L_c(\rho^*) = L_d(r^s)$$

that is, r^s is optimal over A_d. Finally, since r^s is one of N discrete feasible neighboring states of ρ^*, we have $r^s = f(\rho^*)$ for some $f \in \mathcal{F}_{\rho^*}$. ■

Example (Continued): Let us return to our earlier allocation problem of $K = 20$ resources over $N = 4$. Using our iterative procedure (5)-(6) with a gradient obtained through either (33) or (34), if an appropriate step size sequence is selected, the continuous state ρ converges to a point that satisfies Theorem 5.1 and has the optimal allocation $[5,5,5,5]'$ in its neighborhood. To illustrate this, using (33) and $\eta_n = \frac{0.05}{n+1}$, the states evolve as follows:

Iteration n	ρ'_n	r'_n	**Cost**
0	$[1.9, 9.1, 6.1, 2.9]$	$[2, 9, 6, 3]$	-324
1	$[6.2125, 6.0625, 3.6625, 4.0625]$	$[6, 6, 4, 4]$	-576
2	$[5.5375, 5.3875, 4.7875, 4.2875]$	$[6, 5, 5, 4]$	-600
3	$[5.3292, 5.1792, 4.9958, 4.4958]$	$[5, 5, 5, 5]$	-625

6 Generalization to arbitrary feasible sets

We considered above optimization problems over a feasible set determined by the total capacity constraint (7), but stressed the fact that our method is applicable to optimization problems with certain feasible set structures. Specifically, this method is applicable to any feasible set as long as any feasible surrogate state can be written as a convex combination of its feasible discrete neighbors. In this case, using Caratheodory's fundamental theorem [24] one can form a 'reduced' discrete neighbor set (e.g., $\mathcal{N}_N(\rho)$ for the total capacity constraint case). One element of this set will be the operating point for the discrete system and Concurrent Estimation [1] (or related techniques) will yield cost estimates for the other elements. Given the points and their cost estimates, a subset of a hyperplane will be formed on which sensitivity can be estimated. Then, the surrogate state will be updated as in (5). During the update, feasibility will be preserved by choosing an appropriate projection mapping π, which typically maps to the closest point on the feasible set. This approach has been applied, e.g., to the 'lot sizing' problem common in manufacturing systems [27], and is illustrated in the next example.

Example: Consider the following Nonlinear Integer Programming Problem:

$$\min_{x,y \in \mathsf{Z}} J = \max((x-2)^2, (y-1)^2) \tag{41}$$

Note that this problem has a unique optimal point at $(2,1)$. The feasible set A_d is Z^2, therefore, the relaxed set A_c will be R^2. All points in R^2 can be written as a convex combination of the discrete neighbors. In particular, one can show that each surrogate state ρ will fall within a 'square' formed by discrete neighbors, and by Caratheodory's Theorem, three of those vertices will form a triangle that includes ρ (this argument extends to $A_c = \mathsf{R}^N$ where $N+1$ neighbors will suffice.)

Let us assume that the current surrogate state is $\rho_n = (2.5, 3.1)$. One can write

$$(2.5, 3.1) = 0.4(2,3) + 0.5(3,3) + 0.1(2,4)$$

therefore, the three neighbors are determined. The costs at these points are $J(2,3) = 4$, $J(3,3) = 4$, and $J(2,4) = 9$. In Figure 1, costs of some feasible points are plotted and the triangular segment formed by these three neighbors is shown. All points on this plane segment satisfy

$$J = \beta_0 + \nabla J \bullet \rho$$

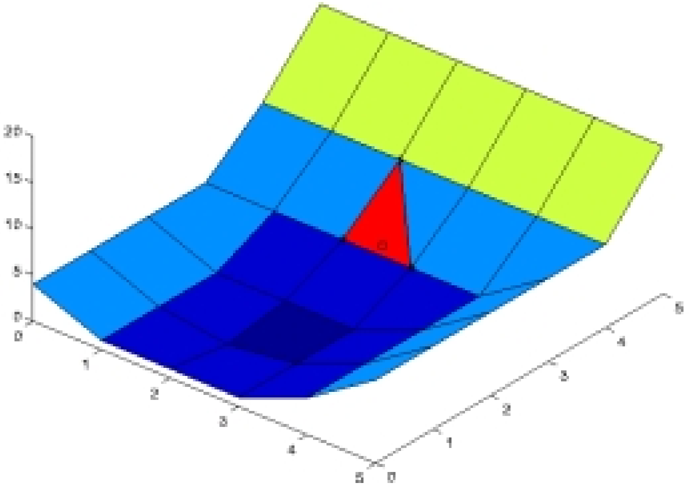

Figure 1: Plane segment for the nth iteration

If the operating point is $(2,3)$, then the gradient is obtained from $\Delta J = \nabla J \cdot \Delta\rho$ with $\frac{\partial J}{\partial \rho_1} = 4 - 4$ and $\frac{\partial J}{\partial \rho_2} = 9 - 4$, and we get

$$\nabla J = \left[\begin{array}{c} 0 \\ 5 \end{array} \right]$$

Once the sensitivity information is acquired, the gradient-descent method with projection updates both the surrogate and the discrete states.

Finally, if the cost function at hand is separable, the method is simplified, as we saw in Section 4.2, in that we do not need to form the ‘reduced’ set. Instead, a feasible discrete neighbor is found and we only need to estimate sensitivities so as to use Theorem 4.1 directly. Then, the continuous state is updated as in (5). During the update, feasibility is preserved by choosing an appropriate projection π, which typically maps to the closest point on the feasible set.

7 Optimization Algorithm

Summarizing the results of the previous sections and combining them with the basic scheme described by (5)-(6), we obtain the following optimization algorithm for the solution of the basic problem in (2):

- Initialize $\rho_0 = r_0$
- For any iteration $n = 0, 1, \ldots$:
    1. Perturb ρ_n so that $I_{\rho_n} = \emptyset$.
    2. Determine $\mathcal{N}(\rho_n)$ [using (9)-(11)].

3. Determine $\mathcal{N}_N(\rho_n)$ [using the Simplex Method as described in the last part of Section 3]; recall that this set is generally not unique.
4. Select $f_n \in \mathcal{F}_{\rho_n}$ such that $r_n = f_n(\rho_n) = \arg\min_{r \in \mathcal{N}_N(\rho_n)} \|r - \rho_n\|$.
5. Operate at r_n to collect $L_d(r^i)$ for all $r^i \in \mathcal{N}_N(\rho_n)$ [using Concurrent Estimation or some form of Perturbation Analysis; or, if feasible, through off-line simulation].
6. Evaluate $\nabla L_c(\rho_n)$ [using (32)].
7. Update the continuous state: $\rho_{n+1} = \pi_{n+1}[\rho_n - \eta_n \nabla L_c(\rho_n)]$.
8. If some stopping condition is not satisfied, repeat steps for $n+1$. Else, set $\rho^* = \rho_{n+1}$.

- Obtain r^* as one of the neighboring feasible states in the set $\mathcal{N}_N(\rho^*)$.

The last step to obtain r^* follows from Theorem 5.1, where it was shown that if the solution ρ^* of the surrogate problem is obtained, then the solution r^* of the original problem is easily obtained as one of the neighboring feasible states in the set $\mathcal{N}_N(\rho^*)$. Also, note that the choice of $f_n \in \mathcal{F}_{\rho_n}$ in step 4 is arbitrary; the specific choice shown above is made for convenience and has the interpretation of being the "nearest" neighbor in a Euclidean sense.

Note that for separable cost functions, steps 1-8 can be replaced by

1. Perturb ρ_n so that $I_{\rho_n} = \emptyset$.
2. Select f_n such that $r_n = f_n(\rho_n) = \arg\min_{r \in \mathcal{A}_d} \|r - \rho_n\|$.
3. Operate at r_n to evaluate $\nabla L_c(\rho_n)$ using Perturbation Analysis and (36).
4. Update the continuous state: $\rho_{n+1} = \pi_{n+1}[\rho_n - \eta_n \nabla L_c(\rho_n)]$.
5. If some stopping condition is not satisfied, repeat steps for $n+1$. Else, set $\rho^* = \rho_{n+1}$.

7.1 Convergence analysis

Since the solution r^* of the original problem can be obtained from ρ^*, we now concentrate on the convergence analysis of the algorithm above, and specifically the iteration in step 7. For unconstrained optimization problems, this iteration is of the general form

$$\begin{aligned} \rho_{n+1} &= \rho_n - \eta_n H(r_n, \omega) \\ &= \rho_n - \eta_n H(f_n(\rho_n), \omega) \\ &= \rho_n - \eta_n [h(\rho_n) + \varepsilon_{n+1}] \end{aligned} \tag{42}$$

where $h(\rho_n)$ is the cost sensitivity when the state is $\rho_n \in A_c$, $H(f_n(\rho_n), \omega)$ is the estimated sensitivity obtained from a sample path ω under a discrete feasible state $r_n = f_n(\rho_n)$, and $\varepsilon_{n+1} = H(r_n, \omega) - h(\rho_n)$ is the estimation error. Theorems dealing with the convergence of stochastic algorithms in such form have appeared in the literature see, for example, [21],[11],[28],[29], [30], [31]. These theorems give conditions for a.s. convergence of $\{\rho_n\}$ to some ρ^*, where

$h(\rho^*) = 0$. In our formulation, the cost sensitivity is given by the N-dimensional vector $h(\rho_n) = [h_1(\rho_n), ..., h_N(\rho_n)]$ with $h : A_c \to \mathsf{R}$, and the estimated cost sensitivity is $H(r_n, \omega) = [H_1(r_n, \omega), ..., H_N(r_n, \omega)]$, where

$$h_i(\rho_n) = \frac{\partial J_c(\rho)}{\partial \rho_i}|_{\rho=\rho_n} \tag{43}$$

$$H_i(r_n, \omega) = H_i(f_n(\rho_n), \omega) = \frac{\partial L_c(\rho)}{\partial \rho_i}|_{\rho=\rho_n} \tag{44}$$

Our objective is to find ρ^* that minimizes the cost function $J_c(\cdot)$. Note that cost functions J_c and L_c are piecewise linear, hence $h_i(\rho_n)$ and $H_i(f_n(\rho_n), \omega)$ are not defined at the integer points. Also, note that h and H are lower semicontinuous functions.

In order to deal with arbitrary feasible sets where ρ may be constrained, the recursive process takes the form

$$\rho_{n+1} = \pi_{n+1}\left[\rho_n - \eta_n[h(\rho_n) + \varepsilon_{n+1}]\right] \tag{45}$$

where the projection $\pi_{n+1}[\cdot]$ is such that

$$\pi_{n+1}[\bar{\rho}] = \rho_{n+1} = \arg\min_{\rho \in A_c} \|\bar{\rho} - \rho\| \tag{46}$$

Convergence of the projected algorithm has also been considered in the literature (e.g., [32],[33]). In this paper we shall follow a very similar line of proof based on results from martingale convergence arguments, a method that seems to have originated with Gladyshev [28]. This requires an appropriately decreasing step size sequence; alternatively, one may pursue an approach using a finite step size but increasing estimation intervals (e.g., [34],[35]).

Let $\{\Im_n\}$ represent the information available up to the time of the nth iteration, so that $\{\Im_n\}$ is an increasing sequence of σ-algebras to which $\{\varepsilon_n\}$, $\{\eta_n\}$ and $\{\rho_n\}$ are adapted. We let $\|.\|$ denote the standard Euclidean norm on R^N. The assumptions that we will need for the convergence theorem are the standard ones found in the literature (e.g., [32], [33]) except for **(H1)** and they are as follows:

Assumptions on $h(\rho)$:

(H1) There exists a unique optimal $\rho^* \in \overset{\circ}{A}_c$, where $\overset{\circ}{A}_c$ denotes the interior of A_c, which satisfies the following conditions:

- For $i = 1, \ldots, N$, $h_i(\rho) \leq 0$ when $\rho_i \leq \rho_i^*$ and $h_i(\rho) \geq 0$ when $\rho_i \geq \rho_i^*$.
- If $\rho \neq \rho^*$ then there exists at least one i such that $\rho_i \neq \rho_i^*$ and $h_i(\rho) \neq 0$.

(H2) $\sup_{\rho \in A_c} \|h(\rho)\| < \infty$

Assumptions on $\{\eta_n\}$:

(A1) $\sum_{n=1}^{\infty} \eta_n = \infty$ a.s.

(A2) $\sum_{n=1}^{\infty} \eta_n^2 < \infty$ a.s.

Assumptions on $\{\varepsilon_n\}$:

(E1) $\sum_{i=1}^{\infty} \eta_i \left\| E(\varepsilon_{i+1} | \Im_i) \right\| < \infty$ a.s.

(E2) For all n, $E(\left\|\varepsilon_{n+1}\right\|^2 | \Im_n) \leq \sigma_n^2$ for some sequence of random variables $\{\sigma_n\}$ adapted to $\{\Im_n\}$ such that $\sum_{n=1}^{\infty} \sigma_n^2 \eta_n^2 < \infty$ a.s.

Assumptions **(A1)-(A2)** are easily satisfied by choosing an appropriate sequence $\{\eta_n\}$. Assumptions **(E1)-(E2)** depend on the properties of the estimator used for $H(r_n, \omega)$. As we have seen in Section 4, the estimators used in our approach are based on finite differences and, therefore, are characterized by properties such as unbiasedness and consistency under mild conditions on the cost function and the stochastic properties of the underlying system (i.e., ergodicity). Thus, these assumptions are not restrictive. Regarding **(H1)-(H2)**, the latter is not restrictive, as the choice of A_c typically guarantees it; the former is required to guarantee uniqueness and relaxing it leads to possible convergence to a local, rather than global, optimum.

The following lemma due to Neveu [36] is instrumental in the proof of the main convergence result:

Lemma 7.1 *Let* $\{T_n\}, \{\alpha_n\}, \{\beta_n\}$ *be sequences of nonnegative random variables adapted to an increasing sequence of* σ*-algebras* $\{\Im_n\}$ *such that*

$$E(T_{n+1} | \Im_n) \leq T_n - \alpha_n + \beta_n \tag{47}$$

If $\sum_{n=1}^{\infty} \beta_n < \infty$ *a.s., then* T_n *converges a.s. to a finite random variable* T*, and* $\sum_{n=1}^{\infty} \alpha_n < \infty$ *a.s.*

Proof. See [36], p.33.

Theorem 7.1 (Convergence) *Assume* (H1),(H2),(A1),(A2),(E1),(E2) *are satisfied. Suppose there exists a twice-differentiable nonnegative function* $U : \mathsf{R}^N \to \mathsf{R}$ *such that*

(U1) *For all* $\rho \in A_c$, $\nabla U_i(\rho) h_i(\rho) \geq 0$

(U2) *For* $\rho \in A_c$, $\nabla U_i(\rho) = 0$ *iff* $\rho_i = \rho_i^*$, *where* ρ^**is the unique minimizer of* U *on* A_c.

(U3) *For all* n *and all* $\rho \in \mathsf{R}^N$, $U(\pi_{n+1}(\rho)) \leq U(\rho)$

(U4) *For all* $\rho \in \mathsf{R}^N$, $\left\|\nabla^2 U(\rho)\right\| \leq \nu$, *for some* $\nu \in \mathsf{R}$.

Let $\{\rho_n\}$ *be defined by (45), with* $\rho_0 \in A_c$ *(initial condition) arbitrary. Then* $\rho_n \to \rho^*$ *as* $n \to \infty$ *with probability 1.*

Proof. The proof follows the same lines as similar results in the literature, e.g., [32], and is included in the Appendix.

Remark. Conditions **(U1)** and **(U2)** may be viewed as Lyapunov conditions, and U can be thought of as a Lyapunov function.

The last remaining step is to identify a function U serving our purposes.

Proposition 7.2 *Under assumption* (H1)*, the function* $U(\rho) = \|\rho - \rho^*\|^2$ *satisfies assumptions* (U1)-(U4)*.*

Proof. Note that $U(\cdot)$ is continuous, twice differentiable and nonnegative. Clearly assumptions **(U2)** and **(U4)** are satisfied with $\nu = 2$. It has also been shown in [31] that the projection mapping defined in (46) is continuous and nonexpansive, i.e.,

$$\|\pi[\rho] - \rho^*\| = \|\pi[\rho] - \pi[\rho^*]\| \leq \|\rho - \rho^*\| \text{ for all } \rho \in \mathsf{R}^N$$

therefore, **(U3)** is satisfied. By assumption **(H1)**, $(\rho_i - \rho_i^*)h_i(\rho) \geq 0$ for all $\rho \in A_c$ and $i = 1, ..., N$ which satisfies assumption **(U1)**. ■

8 Numerical Examples and Applications

We first illustrate our approach by means of a simple deterministic example, followed by a more challenging stochastic optimization application for a classic problem in manufacturing systems.

Example 1: Consider an allocation problem of $K = 20$ resources over $N = 4$ users so as to minimize the convex cost function $J_d(r)$ defined as

$$J_d(r) = \left\|r - [4, 5, 3, 8]'\right\|^2$$

Suppose the current state is $\rho_n = [1.9, 9.1, 6.1, 2.9]'$. Following the eight steps shown in the algorithm of the previous section, we have:

1. All components of ρ_n are real numbers, so no perturbation to avoid integer values is involved.

2. Obtain the set of feasible neighboring states $\mathcal{N}(\rho_n)$: Since $\lfloor \rho_n \rfloor = [1, 9, 6, 2]'$, we have $\tilde{\rho}_n = [0.9, 0.1, 0.1, 0.9]'$ so that the residual capacity is $m_{\rho_n} = 2$, and we get

$$\mathcal{N}(\rho_n) = \{[2, 10, 6, 2]', [2, 9, 7, 2]', [2, 9, 6, 3]', [1, 10, 7, 2]', [1, 10, 6, 3]', [1, 9, 7, 3]'\}$$

3. Obtain a subset $\mathcal{N}_N(\rho_n)$ with $N = 4$ linearly independent discrete neighboring states whose convex hull includes ρ_n. The Simplex method gives

$$\mathcal{N}_N(\rho_n) = \{[2, 10, 6, 2]', [2, 9, 6, 3]', [1, 9, 7, 3]', [2, 9, 7, 2]'\}$$

Note that $\rho_n = 0.1[2, 10, 6, 2]' + 0.8[2, 9, 6, 3]' + 0.1[1, 9, 7, 3]'$. The last vector, $[2, 9, 7, 2]'$, is linearly independent of the other three and is included in the set $\mathcal{N}_N(\rho_n)$ to ensure the desired cardinality $N = 4$.

4. Choose $r_n = f_n(\rho_n) = \arg\min_{r\in\mathcal{N}_N(\rho_n)} \|r - \rho_n\| = [2,9,6,3]'$.

5. Evaluate the cost functions for r_n and for the remaining three states in $\mathcal{N}_N(\rho_n)$:

$$J_d(r_n) = 54, \quad J_d([2,10,6,2]') = 74, \quad J_d([1,9,7,3]') = 66, \quad J_d([2,9,7,2]') = 72$$

6. Evaluate the gradient of the cost at ρ_n using (32):

$$\begin{aligned}\nabla J_c(\rho_n) &= \begin{bmatrix} \mathbf{\Delta R} \\ e' \end{bmatrix}^{-1} \begin{bmatrix} \Delta J \\ 0 \end{bmatrix} \\ &= \begin{bmatrix} 0 & 1 & 0 & -1 \\ -1 & 0 & 1 & 0 \\ 0 & 0 & 1 & -1 \\ 1 & 1 & 1 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 20 \\ 12 \\ 18 \\ 0 \end{bmatrix} = \begin{bmatrix} -5 \\ 9 \\ 7 \\ -11 \end{bmatrix}\end{aligned}$$

7. Update the surrogate state: $\rho_{n+1} = \rho_n - \eta_n \nabla J_c(\rho_n)$.

8. If the stopping condition is not satisfied, go to step 1 and repeat with ρ_{n+1} replacing ρ_n.

Letting $n = 0$ and using a step size sequence $\eta_n = 0.5/(n+1)$, the following table shows the evolution of the algorithm for the first few steps. Note that the optimal allocation $[4,5,3,8]$ is reached after a single step.

STEP	ρ'	r'	$J_c(\rho)$	$J(r)$
0	$[1.900, 9.100, 6.100, 2.900]$	$[2,9,6,3]$	56.84	54
1	$[4.400, 4.600, 2.600, 8.400]$	$[4,5,3,8]$	0.6400	0
2	$[4.150, 4.850, 2.850, 8.150]$	$[4,5,3,8]$	0.0900	0
3	$[3.980, 5.020, 3.020, 7.980]$	$[4,5,3,8]$	0.0016	0
4	$[4.105, 4.895, 2.895, 8.105]$	$[4,5,3,8]$	0.0441	0
5	$[4.005, 4.995, 2.995, 8.005]$	$[4,5,3,8]$	0.0001	0

Example 2: Consider a kanban-based manufacturing system where 15 kanban (resources) are allocated to 3 servers (users) in series. The objective is to find the optimal allocation r^* that minimizes the average cycle time, defined as the time between two job completions at the last server (this is equivalent to a throughput maximization problem). The arrival process is Poisson with rate $\lambda = 1.6$. The service times of the servers are exponentially distributed with rates $\mu_1 = 2.0$, $\mu_2 = 1.6$, $\mu_3 = 3.0$. We use the decreasing step size sequence $\eta_n = 100/(n+1)$ and constant observation intervals. The system is started with an initial allocation $r_0 = [3,5,7]'$ and the algorithm performs as follows:

Number of jobs completed	ρ'	r'	Average Cycle Time
0	$[2.80, 4.90, 7.30]$	$[3, 5, 7]$	0.830316
100	$[7.50, 2.57, 4.93]$	$[7, 3, 5]$	0.737110
200	$[7.18, 3.35, 4.47]$	$[7, 3, 5]$	0.749643
300	$[6.92, 3.89, 4.19]$	$[7, 4, 4]$	0.661367
400	$[6.79, 4.41, 3.80]$	$[7, 4, 4]$	0.811934
500	$[6.79, 4.48, 3.73]$	$[7, 4, 4]$	0.737795
600	$[6.70, 4.50, 3.80]$	$[7, 4, 4]$	0.666858
700	$[6.71, 4.76, 3.53]$	$[7, 5, 3]$	0.770734
800	$[6.66, 4.78, 3.56]$	$[7, 5, 3]$	0.736424

In later iterations, the system stays at the allocation $[7, 5, 3]$ which is determined to be the optimal allocation using brute-force simulation. A similar kanban allocation problem was considered in [37] where an algorithm that incrementally adjusts the allocation one resource at a time was used. As seen in the results here, however, the use of gradient information allowed us to adjust the initial allocation $[3, 5, 7]$ to $[7, 3, 5]$ without having to go through several intermediate steps, thus substantially speeding up the optimization process.

9 Conclusions

In this paper we presented a methodology for solving stochastic discrete optimization problems where the decision variables are non-negative integers. The discrete optimization problem was transformed into a “surrogate” continuous optimization problem which was solved using gradient-based techniques. It was shown that the solution of the original problem is readily recovered as an element of the discrete state neighborhood of the optimal surrogate state. Convergence of the surrogate problem was also established under standard technical conditions. A key contribution of the methodology is its *on-line* control nature, based on actual data from the underlying system. One can therefore see that this approach is intended to combine the advantages of a stochastic approximation type of algorithm with the ability to obtain sensitivity estimates with respect to discrete decision variables. This combination leads to very fast convergence to the optimal point, as illustrated in Section 8. It appears, therefore, feasible to apply this approach to problems with local extreme points by developing a procedure that allows the algorithm to operate from multiple initial states in an effort to determine a global optimum.

Although our analysis was carried out for a class of resource allocation problems with a capacity constraint of the form (7), this methodology may be applied to other types of constraints (equivalently, feasible sets) as long as any feasible surrogate state can be written as a convex combination of its feasible discrete neighbors. Unconstrained problems, for instance, are easily handled, as illustrated in Section 6. Unfortunately, it is not always trivial to determine if the method is applicable to a feasible set structure. Identifying exactly what the limitations of the proposed methodology are is currently under investigation.

It is also worth stressing the fact that in the case of separable cost functions, treated in Section 4.2, this approach is greatly simplified and requires limited sensitivity estimation that is generally easily obtained through standard methods such as Perturbation Analysis and Concurrent Estimation [22],[23].

References

[1] C. G. Cassandras and C. G. Panayiotou, "Concurrent sample path analysis of discrete event systems," *Journal of Discrete Event Dynamic Systems: Theory and Applications*, vol. 9,2, pp. 171–195, 1999.

[2] Y. Ho, A. Eyler, and D. Chien, "A gradient technique for general buffer storage design in a serial production line," *International Journal of Production Research*, vol. 17, pp. 557–580, 1979.

[3] H. Yan, X. Zhou, and G. Yin, "Finding optimal number of kanbans in a manufacturing system via stochastic approximation and perturbation analysis," *Proc. of 11th Intl. Conf. on Analysis and Optimization of Systems*, pp. 572–578, 1994.

[4] C. G. Cassandras and V. Julka, "Scheduling policies using marked/phantom slot algorithms," *Queueing Systems: Theory and Applications*, vol. 20, pp. 207–254, 1995.

[5] C. M. Barnhart, J. E. Wieselthier, and A. Ephremides, "Admission control policies for multihop wireless networks," *Wireless Networks*, vol. 1, no. 4, pp. 373–387, 1995.

[6] R. Parker and R. Rardin, *Discrete Optimization.* Inc, Boston: Academic Press, 1988.

[7] T. Ibaraki and N. Katoh, *Resource Allocation Problems: Algorithmic Approaches.* MIT Press, 1988.

[8] E. Aarts and J. Korst, *Simulated Annealing and Boltzmann Machines.* Wiley, 1989.

[9] J. Holland, *Adaptation in Natural and Artificial Systems.* University of Michigan Press, 1975.

[10] J. Kiefer and J. Wolfowitz, "Stochastic estimation of the maximum of a regression function," *Annals of Mathematical Statistics*, vol. 23, pp. 462–466, 1952.

[11] H. Robbins and S. Monro, "A stochastic approximation method," *Annals of Mathematical Statistics*, vol. 22, pp. 400–407, 1951.

[12] D. Yan and H. Mukai, "Stochastic discrete optimization," *SIAM Journal on Control and Optimization*, vol. 30, 1992.

[13] W. B. Gong, Y. C. Ho, and W. Zhai, "Stochastic comparison algorithm for discrete optimization with estimation," *Proc. of 31st IEEE Conf. on Decision and Control*, pp. 795–800, 1992.

[14] L. Shi and S. Olafsson, "Nested partitions method for global optimization," *Operations Research*, 1999. To appear.

[15] Y. C. Ho, R. S. Sreenivas, and P. Vakili, "Ordinal optimization in DEDS," *J. of Discrete Event Dynamic Systems: Theory and Applications*, vol. 2, pp. 61–88, 1992.

[16] C. G. Cassandras, L. Dai, and C. G. Panayiotou, "Ordinal optimization for deterministic and stochastic resource allocation.," *IEEE Trans. Automatic Control*, vol. 43, no. 7, pp. 881–900, 1998.

[17] L. Dai, C. G. Cassandras, and C. G. Panayiotou, "On the convergence rate of ordinal optimization for a class of stochastic discrete resource allocation problems," *IEEE Trans. Automatic Control*, 1999. To appear.

[18] L. Dai, "Convergence properties of ordinal comparison in the simulation of discrete event dynamic systems," *J. of Optimization Theory and Applications*, vol. 91, pp. 363–388, 1996.

[19] C. G. Cassandras and V. Julka, "A new approach for some combinatorially hard stochastic optimization problems," *Proc. of 31st Annual Allerton Conference on Communication, Control, and Computing*, pp. 667–676, 1993.

[20] K. Gokbayrak and C. G. Cassandras, "Stochastic discrete optimization using a surrogate problem methodology," in *Proceedings of 38th IEEE Conf. On Decision and Control*, pp. 1779–1784, Dec. 1999.

[21] H. Kushner and D. Clark, *Stochastic Approximation for Constrained and Unconstrained Systems.* Springer-Verlag, 1978.

[22] Y. Ho and X. Cao, *Perturbation Analysis of Discrete Event Dynamic Systems.* Kluwer Academic Publishers, 1991.

[23] C. G. Cassandras, *Discrete Event Systems: Modeling and Performance Analysis.* Irwin Publ., 1993.

[24] D. Bertsimas and J. N. Tsitsiklis, *Introduction to Linear Optimization.* Belmont, Massachusetts: Athena Scientific, 1997.

[25] J. Wieselthier, C. Barnhart, and A. Ephremides, "Standard clock simulation and ordinal optimization applied to admission control in integrated communication networks," *Journal of Discrete Event Dynamic Systems*, vol. 5, pp. 243–279, 1995.

[26] K. Gokbayrak and C. G. Cassandras, "Adaptive call admission control in circuit switched networks," Submitted for publication.

[27] C. G. Cassandras and R. Yu, "A 'surrogate problem' approach for lot size optimization in manufacturing systems," *Proc. of 2000 American Control Conference*, 2000. To appear.

[28] E. G. Gladyshev, "On stochastic approximation," *Theo. Prob. Appl.*, vol. 10, pp. 275–278, 1965.

[29] L. Ljung, "Analysis of recursive stochastic algorithms," *IEEE Tr. Automat. Control*, vol. 22, pp. 551–575, 1977.

[30] M. Metivier and P. Priouret, "Applications of a kushner and clark lemma to general classes of stochastic algorithms," *IEEE Tr. Inform. Theory*, vol. 30, pp. 140–151, 1984.

[31] D. P. Bertsekas, *Nonlinear Programming.* Belmont, Massachusetts: Athena Scientific, 1995.

[32] E. K. P. Chong and P. J. Ramadge, "Convergence of recursive optimization algorithms using ipa derivative estimates," *Journal of Discrete Event Dynamic Systems: Theory and Applications*, vol. 1, pp. 339–372, 1992.

[33] H. Kushner and G. Yin, *Stochastic Approximation Algorithms and Applications.* Springer-Verlag New York, Inc., 1997.

[34] H. J. Kushner and F. J. V'azquez-Abad, "Stochastic approximation methods for systems of interest over an infinite horizon," *to appear in SIAM J. on Control and Optimization.*

[35] F. Vázquez-Abad, C. Cassandras, and V. Julka, "Centralized and decentralized asynchronous optimization of stochastic discrete event systems," *IEEE Trans. Automatic Control*, 1995. To appear.

[36] J. Neveu, *Discrete Parameter Martingales.* 1975.

[37] C. G. Panayiotou and C. G. Cassandras, "Optimization of kanban-based manufacturing systems," *Automatica*, vol. 35, pp. 1521–1533, September 1999.

APPENDIX

Proof of Theorem 7.1: The proof consists of five steps as follows.

Step 1. Set up conditions that allow us to apply the previous lemma with $T_n = U(\rho_n)$. First, let

$$\bar{\rho}_{n+1} = \rho_n - \eta_n[h(\rho_n) + \varepsilon_{n+1}], \tag{48}$$

therefore $\rho_{n+1} = \pi_{n+1}[\bar{\rho}_{n+1}]$. Then, consider a Taylor expansion of $U(\bar{\rho}_{n+1})$ about ρ_n:

$$U(\bar{\rho}_{n+1}) = U(\rho_n) + \nabla U(\rho_n)'(\bar{\rho}_{n+1} - \rho_n) + \frac{1}{2}(\bar{\rho}_{n+1} - \rho_n)'\nabla^2 U(\zeta)(\bar{\rho}_{n+1} - \rho_n)$$

where $\zeta = \rho_n + \alpha(\bar{\rho}_{n+1} - \rho_n)$ and $0 \le \alpha \le 1$. Using (48) we get

$$\begin{aligned} U(\bar{\rho}_{n+1}) &= U(\rho_n) - \eta_n \nabla U(\rho_n)'[h(\rho_n) + \varepsilon_{n+1}] \\ &\quad + \frac{1}{2}\eta_n^2[h(\rho_n) + \varepsilon_{n+1}]'\nabla^2 U(\zeta)[h(\rho_n) + \varepsilon_{n+1}] \\ &= U(\rho_n) - \eta_n \nabla U(\rho_n)'h(\rho_n) - \eta_n \nabla U(\rho_n)'\varepsilon_{n+1} + \frac{\eta_n^2}{2}h'(\rho_n)\nabla^2 U(\zeta)h(\rho_n) \\ &\quad + \frac{\eta_n^2}{2}h'(\rho_n)\nabla^2 U(\zeta)\varepsilon_{n+1} + \frac{\eta_n^2}{2}\varepsilon'_{n+1}\nabla^2 U(\zeta)h(\rho_n) + \frac{\eta_n^2}{2}\varepsilon'_{n+1}\nabla^2 U(\zeta)\varepsilon_{n+1} \end{aligned}$$

Using the inequality $x'Py \le \|P\|\, x'y$ and assumption **(U4)**, as well as assumption **(H2)** with $\gamma = \sup_{\rho \in A_c}\|h(\rho)\| < \infty$, we have

$$U(\bar{\rho}_{n+1}) \le U(\rho_n) - \eta_n \nabla U(\rho_n)'h(\rho_n) - \eta_n \nabla U(\rho_n)'\varepsilon_{n+1}$$

$$
\begin{aligned}
& +\frac{\eta_n^2\nu}{2}\left\|h(\rho_n)\right\|^2+\eta_n^2\nu h'(\rho_n)\varepsilon_{n+1}+\frac{\eta_n^2\nu}{2}\left\|\varepsilon_{n+1}\right\|^2 \\
\leq\;& U(\rho_n)-\eta_n\nabla U(\rho_n)'h(\rho_n)+\frac{\eta_n^2\nu\gamma^2}{2} \\
& -\eta_n[\nabla U(\rho_n)'-\eta_n\nu h'(\rho_n)]\varepsilon_{n+1}+\frac{\eta_n^2\nu}{2}\left\|\varepsilon_{n+1}\right\|^2
\end{aligned}
$$

$U(\bar{\rho}_{n+1})$ is measurable and nonnegative, and taking conditional expectations of both sides with respect to $\Im_n$, we get

$$
\begin{aligned}
E[U(\bar{\rho}_{n+1})|\Im_n] \;\leq\;& U(\rho_n)-\eta_n\nabla U(\rho_n)'h(\rho_n)+\frac{\eta_n^2\nu\gamma^2}{2} \\
& -\eta_n[\nabla U(\rho_n)'-\eta_n\nu h'(\rho_n)]E(\varepsilon_{n+1}|\Im_n)+\frac{\eta_n^2\nu}{2}E(\left\|\varepsilon_{n+1}\right\|^2|\Im_n)
\end{aligned}
$$

Note that A_c is a closed and bounded subset of R^N, therefore compact. Using the continuity of $\nabla U(\rho)$, compactness of A_c, and **(U4)**, we have $\mu=\sup_{\rho\in A_c}\left\|\nabla U(\rho)\right\|<\infty$. Applying assumptions **(E2)** and **(H2)**,

$$
\begin{aligned}
E[U(\bar{\rho}_{n+1})|\Im_n] \;\leq\;& U(\rho_n)-\eta_n\nabla U(\rho_n)'h(\rho_n)+\frac{\eta_n^2\nu\gamma^2}{2} \\
& -\eta_n[\nabla U(\rho_n)'-\eta_n\nu h'(\rho_n)]E(\varepsilon_{n+1}|\Im_n)+\frac{\eta_n^2\nu\sigma_n^2}{2} \\
\leq\;& U(\rho_n)-\eta_n\nabla U(\rho_n)'h(\rho_n)+\frac{\eta_n^2\nu\gamma^2}{2}+\frac{\eta_n^2\nu\sigma_n^2}{2} \\
& +\eta_n^2\nu\gamma\left\|E(\varepsilon_{n+1}|\Im_n)\right\|-\eta_n\nabla U(\rho_n)'E(\varepsilon_{n+1}|\Im_n) \\
\leq\;& U(\rho_n)-\eta_n\nabla U(\rho_n)'h(\rho_n)+\frac{\eta_n^2\nu\gamma^2}{2}+\frac{\eta_n^2\nu\sigma_n^2}{2} \\
& +\eta_n^2\nu\gamma\left\|E(\varepsilon_{n+1}|\Im_n)\right\|+\eta_n\mu\left\|E(\varepsilon_{n+1}|\Im_n)\right\| \\
=\;& U(\rho_n)-\eta_n\nabla U(\rho_n)'h(\rho_n)+\phi_n\eta_n^2+\mu\eta_n\left\|E(\varepsilon_{n+1}|\Im_n)\right\| \qquad (49)
\end{aligned}
$$

where

$$
\phi_n=\frac{\nu}{2}(\gamma^2+\sigma_n^2+2\gamma\left\|E(\varepsilon_{n+1}|\Im_n)\right\|)
$$

is an $\Im_n$-measurable random variable. Note that the third inequality above follows from the fact that $-\eta_n\nabla U(\rho_n)'E(\varepsilon_{n+1}|\Im_n)\leq\eta_n\left\|\nabla U(\rho_n)\right\|\left\|E(\varepsilon_{n+1}|\Im_n)\right\|$. Next, by **(A2)**, **(E1)**, and **(E2)**,

$$
\begin{aligned}
\sum_{n=1}^{\infty}\phi_n\eta_n^2 \;=\;& \frac{\nu}{2}\sum_{n=1}^{\infty}(\gamma^2+\sigma_n^2+2\gamma\left\|E(\varepsilon_{n+1}|\Im_n)\right\|)\eta_n^2 \qquad (50) \\
=\;& \frac{\nu\gamma^2}{2}\sum_{n=1}^{\infty}\eta_n^2+\frac{\nu}{2}\sum_{n=1}^{\infty}\sigma_n^2\eta_n^2+\gamma\nu\sum_{n=1}^{\infty}\left\|E(\varepsilon_{n+1}|\Im_n)\right\|\eta_n^2<\infty \text{ a.s.}
\end{aligned}
$$

Since $\rho_{n+1}=\pi_{n+1}[\bar{\rho}_{n+1}]$, then by assumption **(U3)**,

$$
E[U(\rho_{n+1})|\Im_n]=E[U(\pi[\bar{\rho}_{n+1}])|\Im_n]\leq E[U(\bar{\rho}_{n+1})|\Im_n] \qquad (51)
$$

Step 2. Apply Lemma 7.1. Define a sequence of nonnegative (by assumption **(U1)**) random variables $\{\alpha_n\}$ by

$$
\alpha_n=\eta_n\nabla U(\rho_n)'h(\rho_n)
$$

Define another sequence of nonnegative random variables $\{\beta_n\}$ by

$$\beta_n = \phi_n \eta_n^2 + \mu \eta_n \left\| E(\varepsilon_{n+1} | \Im_n) \right\|$$

Therefore, using equations (49) and (51) we can write,

$$E(U(\rho_{n+1})|\Im_n) \leq U(\rho_n) - \alpha_n + \beta_n \tag{52}$$

Note that $\{U(\rho_n)\}, \{\alpha_n\}$ and $\{\beta_n\}$ are adapted to $\{\Im_n\}$ and using assumption **(E1)** and (50), $\sum_{n=1}^{\infty} \beta_n < \infty$ a.s. Applying Lemma 7.1, we get that $U(\rho_n) \to U^*$ a.s. for some finite random variable U^* and

$$\sum_{n=1}^{\infty} \alpha_n = \sum_{n=1}^{\infty} \eta_n \nabla U(\rho_n)' h(\rho_n) < \infty \text{ a.s.} \tag{53}$$

Step 3. Show that there is a subsequence $\{\rho_{k_n}\}$ such that $\{\nabla U(\rho_{k_n})' h(\rho_{k_n})\} \to 0$ a.s. Suppose there exist $\delta > 0$ and a finite M such that $P\{\omega : \forall j \geq M,\ \nabla U(\rho_j)' h(\rho_j) \geq \delta\} > 0$. In this case, with nonzero probability we have a sample path where

$$\begin{aligned}
\sum_{n=1}^{\infty} \alpha_n &= \sum_{n=1}^{\infty} \eta_n \nabla U(\rho_n)' h(\rho_n) \\
&= \sum_{n=1}^{M-1} \eta_n \nabla U(\rho_n)' h(\rho_n) + \sum_{n=M}^{\infty} \eta_n \nabla U(\rho_n)' h(\rho_n) \\
&\geq \sum_{n=1}^{M-1} \eta_n \nabla U(\rho_n)' h(\rho_n) + \delta \sum_{n=M}^{\infty} \eta_n = \infty
\end{aligned}$$

Note that the last equality follows from **(A1)**. This conclusion contradicts (53).

Step 4. Show that $\rho_{k_n} \to \rho^*$ a.s. Suppose that there exists $\delta > 0$ such that $\left\|\rho_{k_n} - \rho^*\right\| \geq \delta$ for infinitely many n with some nonzero probability. Consider the compact set $E = A_c \cap \{\rho : \|\rho - \rho^*\| \geq \delta\}$ and define $f(\rho) = \nabla U(\rho)' h(\rho)$. Since $f(\rho)$ is a lower semicontinuous function, there exists a ρ_E such that

$$f(\rho_E) \leq f(\rho) \text{ for all } \rho \in E \tag{54}$$

$$\begin{aligned}
f(\rho_E) &= \nabla U(\rho_E)' h(\rho_E) = \sum_{i=1}^{N} \nabla U_i(\rho_E) h_i(\rho_E) \\
&= \sum_{\substack{i=1 \\ \nabla U_i(\rho_E)=0}}^{N} \nabla U_i(\rho_E) h_i(\rho_E) + \sum_{\substack{i=1 \\ \nabla U_i(\rho_E)\neq 0}}^{N} \nabla U_i(\rho_E) h_i(\rho_E) \\
&= \sum_{\substack{i=1 \\ \nabla U_i(\rho_E)\neq 0}}^{N} \nabla U_i(\rho_E) h_i(\rho_E)
\end{aligned}$$

By assumption **(U1)**, each term in the summation is non-negative therefore $f(\rho_E) \geq 0$. Using assumption **(U2)**

$$f(\rho_E) = \sum_{\substack{i=1 \\ \nabla U_i(\rho_E) \neq 0}}^{N} \nabla U_i(\rho_E) h_i(\rho_E) = \sum_{\substack{i=1 \\ \rho_{E_i} \neq \rho_i^*}}^{N} \nabla U_i(\rho_E) h_i(\rho_E)$$

and by **(H1)**, at least one of the terms in the summation is non-zero, hence strictly positive by **(U1)**. Therefore,

$$0 < f(\rho_E) \leq f(\rho) \text{ for all } \rho \in E \tag{55}$$

Since we assumed that $\rho_{k_n} \in E$ infinitely often with nonzero probability, then $f(\rho_{k_n}) \geq f(\rho_E)$ infinitely often with nonzero probability. Using the same argument as in Step 3 (with $f(\rho_E) = \delta$) we arrive at a similar contradiction.

Step 5. Show that $\rho_n \to \rho^*$ a.s. As $\rho_{k_n} \to \rho^*$ a.s. and $U(\rho_n) \to U^*$ a.s. from steps 4 and 2 respectively, we have $U^* = U(\rho^*)$, the unique minimum of $U(\cdot)$. Suppose there exists $\delta > 0$ such that $\|\rho_n - \rho^*\| \geq \delta$ for infinitely many n with nonzero probability. Consider the compact set $E = A_c \cap \{\rho : \|\rho - \rho^*\| \geq \delta\}$. Then, since $U(\cdot)$ is continuous, there exists ρ_E such that for all $\rho \in E$, $U(\rho_E) \leq U(\rho)$. Note that $\rho_E \neq \rho^*$ and since ρ^* is the unique minimum of $U(\cdot)$, we have $U(\rho_E) > U(\rho^*)$. Since we assumed that $\rho_n \in E$ infinitely often with nonzero probability, $U(\rho_n) \geq U(\rho_E) > U(\rho^*)$ with nonzero probability which contradicts the fact that $U(\rho_n) \to U^*$. Therefore, $\rho_n \to \rho^*$ with probability one. This completes the proof. ■