Mathematical Programming 75 (1996) 137-176

Sample-path optimization of convex stochastic performance functions

Erica L. Plambeck [1], Bor-Ruey Fu [2], Stephen M. Robinson *, Rajan Suri [2]

Department of Industrial Engineering, University of Wisconsin–Madison, 1513 University Avenue, Madison, WI 53706-1539, USA

Received September 1993; revised May 1994

Abstract

In this paper we propose a method for optimizing convex performance functions in stochastic systems. These functions can include expected performance in static systems and steady-state performance in discrete-event dynamic systems; they may be nonsmooth. The method is closely related to retrospective simulation optimization; it appears to overcome some limitations of stochastic approximation, which is often applied to such problems. We explain the method and give computational results for two classes of problems: tandem production lines with up to 50 machines, and stochastic PERT (Program Evaluation and Review Technique) problems with up to 70 nodes and 110 arcs.

Keywords: Stochastic optimization; Steady-state performance; Expected performance; Discrete event systems; Nonsmooth optimization; Stochastic convexity; Sample-path optimization

* Corresponding author. Sponsored by the National Science Foundation under grant number CCR-9109345, by the Air Force Systems Command, USAF, under grant numbers F49620-93-1-0068 and F49620-95-1-0222, by the U.S. Army Research Office under grant number DAAL03-92-G-0408, and by the U.S. Army Space and Strategic Defense Command under contract number DASG60-91-C-0144. The U.S. Government has certain rights in this material, and is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon.

[1] Sponsored by a Wisconsin/Hilldale Research Award, by the U.S. Army Space and Strategic Defense Command under contract number DASG60-91-C-0144, and by the Air Force Systems Command, USAF, under grant number F49620-93-1-0068.

[2] Sponsored by the National Science Foundation under grant number DDM-9201813.

Published by Elsevier Science B.V.
PII S0025-5610(96)00010-X

1. Introduction

In this paper we propose a method for optimizing convex, possibly nonsmooth, performance functions in certain stochastic systems. The criterion to be optimized can be an expected value in a static system, or a steady-state performance function in a dynamic system, to which average performance along a sample path of the system will converge with probability one. An example of a steady-state performance function is the long-run throughput of a production line. We consider systems for which such functions exist and in addition obey certain restrictions that will be explained below.

It is well known that for many stochastic systems such performance functions cannot be expressed analytically. Therefore, people use Monte Carlo simulation to evaluate them. When it is necessary also to optimize performance with respect to some set of parameters, then the function values obtained by simulation must generally be supplemented by (approximate) gradient evaluations [13,24,25]. Several methods for obtaining approximations to gradients in dynamic systems are commonly used: these include for example infinitesimal perturbation analysis (IPA) [11,18,19,40], and methods using likelihood ratios (LR) or score functions, for a treatment of which see [12,13,35].

Given a method for producing approximate gradients, people needing to optimize performance functions have often used some variant of the method of stochastic approximation [32] to locate an optimizer. More recently, the method of single-run optimization (SRO) was proposed in [27] and studied in [44] and [45] as an attempt to improve the efficiency of stochastic approximation; in this method, instead of making one or more simulation runs to obtain gradients, and then making a gradient step in the parameter space, one makes gradient steps at intervals during a single simulation run, using approximate gradients obtained from IPA. The key point here is that the simulation does not have to be restarted, with a new warmup period, after each step.

However, both the classical stochastic approximation method and the SRO variant have certain drawbacks. For one thing, inequality constraints – even linear inequalities – present severe difficulties since the underlying gradient descent method must then be modified in some ad hoc manner so that the sequence of parameter values remains feasible; typically this is done by projecting the new point obtained from the iteration into the feasible set in some manner. This difficulty does not appear with linear equation constraints because one can reduce such a problem to an unconstrained problem in fewer variables by an appropriate affine transformation.

In addition, if the function being optimized is nondifferentiable, then the stochastic approximation technique reduces to a variant of subgradient optimization [3,15]. That method is known not only to be slow, but also to suffer from other drawbacks, such as the lack of a good stopping criterion and the difficulty in enforcing feasibility already mentioned above. This concern about nondifferentiability is not just theoretical; Suri and Fu [43] observed nondifferentiability of steady-state throughput in tandem production lines, and results of Shapiro and Wardi [39] show that steady-state functions in discrete event dynamic systems can be nondifferentiable when the sample performance functions are convex and the event time distributions contain atoms.

For nondifferentiable, or even discontinuous, objective functions several authors have presented and analyzed methods of the stochastic quasigradient (SQG) class; see for example [4] and references therein. These methods can be thought of as pushing the philosophy of SRO to an extreme in that they obtain a "quasigradient" estimate, often from a single sample realization, move in the parameter space using that estimate, obtain another estimate, make another move, and so on. It is possible to obtain the quasigradients by using function values only, or by other means that avoid the necessity for techniques like IPA or LR.

To our knowledge there is not much documented numerical experience in applying SQG methods to realistic problems, though a code has been made available [6], but there are indications that the technique can be slow (see for example the small illustrative numerical computation given in [4]). In any case, the two other drawbacks mentioned above – difficulty in enforcing feasibility, and the lack of a good stopping criterion – also apply to SQG methods.

The method we propose, which in its pure form we call *sample-path optimization*, appears to avoid these two difficulties. It exploits the fact that the function we wish to optimize is the limit, along almost every sample path, of a sequence of approximating functions (outputs of simulation runs of increasing lengths, all using the same random number stream(s)). The basic idea is simply to go out far enough along the sample path to have a good estimate of the limit function, and then to optimize the resulting *deterministic* function by the most efficient methods available, taking the result as an estimate of an optimizer of the limit function. This is closely related to the method of retrospective optimization proposed by Healy and Schruben [14], but it differs in that we do not suggest making multiple runs for the purpose of averaging or constructing a distribution; rather, we use a single sample point and a relatively long run.

This conceptual method is analyzed in [33], which also contains additional references to the literature. The algorithm is shown there to converge with probability one under two hypotheses: first, that the sequence of approximating functions *epiconverges* to the limit function; second, that the limit function almost surely has a nonempty, compact set of minimizers. It is further shown there that the epiconvergence property holds under certain assumptions that are convenient for application in practice. We explain these assumptions further in Subsection 2.1 below.

Our objective in this paper is to show that variants of this method can be applied successfully to large, hard problems in which the objective functions may be nonsmooth. To accomplish this we used a version of the bundle method, with some adaptations that we explain later. We also adapted the basic sample-path optimization idea by retaining certain information when increasing the simulation run length. Although not theoretically supported, this procedure conserved information that in some cases was very costly.

This implementation of sample-path optimization has, in our view, at least two advantages as compared to more conventional methods. First, it deals directly with the issue of nondifferentiability, and in the process it develops an effective stopping criterion in the form of the ϵ-subgradient that is evaluated and tested as part of the algorithm. Second, it has no difficulty with linear constraints of any kind (inequality and/or equality),

because these can simply be incorporated into the quadratic programming problem that determines the step in the parameter space.

We proposed this approach for the tandem manufacturing line problem in [31], but with little detail except for a specification of the algorithm and a report of some computational results. In this paper, by contrast, we attempt to give a careful treatment of several issues that were not discussed in [31] because of limited space. We also give much more extensive computational results, and we point out some areas that need further research. Related methods were described in [7,8,16,17] with computational results for solving stochastic linear programs with recourse.

The remainder of this paper is organized in four main sections followed by six appendices containing additional detail, operational or theoretical, supporting the main body of the paper. References conclude the paper. Of the main sections, Section 2 describes the method that we suggest, including some of the details of its implementation, while Sections 3 and 4 contain our numerical results.

In Section 3 we give examples of optimization of tandem manufacturing lines, with respect to machine cycle times, for lines with up to 50 machines. The cycle times were required to satisfy various linear constraints, including binding inequality constraints. We do not know of any reports in the published literature covering optimization of lines of this size, with or without constraints.

In Section 4 we present results for optimization of stochastic PERT problems with up to 70 nodes and 110 arcs, with respect to parameters appearing in the probability distributions of the activity lengths. Again, we do not know of any published numerical work in whichs stochastic networks of this size have been successfully optimized.

Section 5 briefly summarizes the work presented, and comments on issues that need further research.

2. Modified sample-path optimization

In this section we describe the algorithm that we suggest, as well as some implementation issues. In Subsection 2.1 we describe the underlying algorithm and some expedient modifications that we made to it for better performance on the problems presented in Sections 3 and 4. We explain in Subsection 2.2 the bundle-type algorithm that we used for the computations. We show there how we extended the method's basic stopping criterion to take account of linear constraints whose presence was not reflected in the subgradient estimates obtained from the simulations. We also give more detail about one of the modifications just mentioned.

2.1. Sample-path optimization and its applicability

This subsection explains the algorithm that we employed for the stochastic optimization calculations presented in Sections 3 and 4. As explained earlier, the underlying method is simply to optimize the deterministic function defined by a simulation run

having fixed length and using a single sample path (implemented by the method of common random numbers).

An abstract form of our problem is the following: suppose we are given an extended-real-valued stochastic process $\{ S_n(x) \mid n \geqslant 1 \}$, depending on a parameter $x \in \mathbb{R}^k$. For $n \geqslant 1$ and $x \in X$ the $S_n(x)$ are defined on a common probability space $(\Psi, \mathcal{F}, P)$, and we denote a sample path of the process by $\{ S_n(\psi, x) \mid n \geqslant 1 \}$ for $x \in X$. The $S_n(x)$ could, for example, represent averages derived from simulation runs of length n; we are interested in what happens as $n \to \infty$.

This setup covers both cases with which we are concerned in this paper. In the static case, we repetitively simulate a single event (the realization of a PERT network and the subsequent longest-path calculation) and average the results; under weak assumptions the strong law of large numbers tells us that these averages converge almost surely to the expected value in which we are interested. In the dynamic case, we simulate the operation of a manufacturing line for periods of increasing length, and assume that the system satisfies conditions sufficient for the resulting sample average throughput to converge almost surely to a steady-state throughput.

The pure form of sample-path optimization would fix a large n and a sample point ψ and optimize the deterministic function $S_n(\psi, \cdot)$. As we point out below, we made some expedient modifications to this method in our actual implementation. However, it is of interest to ask what the convergence properties of the basic method are. That question is dealt with elsewhere [33], where it is shown that under two assumptions the method converges almost surely. This result holds both for exact minimization of S_n and for ϵ-minimization.

The two assumptions just mentioned, expressed in the present notation, are first that with probability one the $S_n(\cdot)$ are lower semicontinuous proper functions that epiconverge to a (possibly random) function $S_\infty(\cdot)$, and second that with probability one $S_\infty(\cdot)$ is lower semicontinuous and proper, and for some positive ϵ_0 the level set $\{x \mid S_\infty(x) \leqslant (\inf S_\infty) + \epsilon_0\}$ is nonempty and compact. For a good elementary treatment of epiconvergence see [21], and for more detail see [1].

The first of the above assumptions (epiconvergence) is rarely given explicitly. We give here two assumptions that are more convenient for practical application, and outline the result [33, Proposition 2.5], showing that they imply epiconvergence.

The assumptions deal with random functions Y_n defined on a relatively open convex subset A of $\mathbb{R}^k$. In our applications A will be an open convex subset of the nonnegative orthant $\mathbb{R}^k_+$. For the tandem production lines this subset is just the interior of $\mathbb{R}^k_+$, whereas for the PERT networks it is the open set G discussed in Appendix E.

Assumption 1. For each n, Y_n is with probability one a finite convex function on A.

Assumption 2. There is a countable dense subset Θ of A such that for each $x \in \Theta$, with probability one the sequence $\{Y_n(x)\}$ converges to a finite limit $Y_\infty(x)$.

Now suppose that D is a nonempty closed subset of A. Proposition 2.5 of [33] shows that under Assumptions 1 and 2, with probability one the following hold:

(1) The limit $Y_\infty(x)$ exists for *each* $x \in A$ (not only for $x \in \Theta$).
(2) Y_∞ is a finite convex function on A.
(3) If for $1 \leqslant n \leqslant \infty$ we define functions X_n on $\mathbb{R}^k$ to be Y_n on D and $+\infty$ off D, then the X_n are lower semicontinuous and proper, and as $n \to \infty$ the X_n epiconverge to X_∞.

In this paper our sample-path functions S_n play the role of the X_n in the above result, since they are restrictions of the Y_n to a closed subset D of A. In our work D is a polyhedral convex set, whose definition includes the linear equation and inequality constraints imposed on the variables. For the PERT problems these constraints included explicit positive lower bounds on all variables. For the tandem production lines the definition of D may include implicit positive lower bounds on the cycle times, small enough to be inactive at the optimizer, if these are needed to produce a closed set.

Assumption 1 is the property of *strong stochastic convexity* introduced by Shanthikumar and Yao [38]. Assumption 2 always holds, in the case of the PERT problems of Section 4, because the strong law of large numbers applies. It holds in the case of the tandem production lines of Section 3 if there is a limiting distribution (steady state). We do not go into any detail here about conditions for existence of a steady state in such lines; see, for example, [30] or [10]. However, we shall discuss the other assumption in the context of our problem types.

In Appendix E we provide analysis to show that Assumption 1 (strong stochastic convexity) holds for the PERT problems analyzed in Section 4. We also show there how to determine the exact form of the subdifferential that is needed for the computational work.

The situation is somewhat more complicated for the tandem production lines treated in Section 3. The objective in these problems is to maximize the steady-state throughput, the mean amount of production per unit time by the last machine in the line. Recall that tandem production lines can be either *continuous* or *discrete*: in the continuous case (e.g. chemical production), the product flowing through the system is a continuous fluid as opposed to separate workpieces in the discrete case (e.g. automobile assembly). We shall refer to these as CT and DT lines respectively.

Shanthikumar and Yao [37, Section IV.B] proved that in a given sample path of a DT line with manufacturing blocking, the departure time D_n of the nth unit of product from the system is a convex function of parameters in the distributions of the external interarrival times and the machine service times, provided that these times themselves are convex functions of those parameters. Related results are in [20]. This proof was extended to lines with unreliable machines by Fu [5] (Theorem 5.2 for manufacturing blocking, and Theorem 5.4 for communication blocking). Fu's results are stated in

terms of flow rates (the reciprocals of cycle times); however, his technique of proof is to establish convexity of the departure time as a function of cycle times and then to use a version of the theorem on convexity of composite functions [34, Theorem 5.1] to prove that the departure time is also convex as a function of flow rates. The result we need here is the convexity with respect to cycle times; this establishes Assumption 1 for DT lines.

However, for the computations of Section 3 we did not use DT simulations, but approximations to these by CT simulations. Studies of such approximations, and of the errors introduced by using them, appear in [5,41–43]. For example, Suri and Fu [43] concluded that for fairly small lines (up to 6 machines), the throughput values obtained from a CT approximation were very close to those of the DT line being approximated (relative errors ranging from 0.0% to −2.1%), and for an extensive study of 132 15-machine lines the relative error of throughput was under 4% in 90% of the cases studied. Further, the CT simulations had a substantial time advantage over DT simulations: in a third of the 132 cases, the ratio of the DT simulation time to the CT simulation time was at least 10, whereas in 6 of the cases the ratio was more than 60.

In view of these findings, since we wished to optimize lines of large size – for which we thought the time advantage of the CT simulation would probably be even larger than for the lines studied in [43] – we decided to approximate the DT lines by CT lines. To establish the convexity of departure times in such lines, we can use another result of Fu [5]. He begins with a DT line with communication blocking, and then derives from it a sequence of DT lines with communication blocking, constructed so that in the mth such line the size of a unit of product is 2^{-m} times that in the original line. The ith buffer in the mth line holds 2^m times as many products as does the ith buffer in the original line (that is, the total product volume is the same in both buffers). The machine failure and repair distributions are the same.

Let us choose one of the machines and, for a fixed nonnegative integer q, denote the earliest time at which q units of volume (that is, q products in the original line, or $2^m q$ products in the mth line) have departed that machine by D_m, where m may be any nonnegative integer. The original line corresponds to the choice $m = 0$. Fu proves in Theorem 5.6 of [5] that $\lim_{m\to\infty} D_m = D_{CT}$, where D_{CT} is the earliest time at which q units of volume have departed the chosen machine in a CT line having the same buffer sizes, mean volumes to failure, mean times to repair, and flow rates. If we define a vector $c \in \operatorname{int}\mathbb{R}^k$ by letting c_i be the processing time that the ith machine requires to complete one volumetric unit of product (that is, one unit of product in the CT line, or 2^m units in the mth DT line), then we can write $D_m = D_m(c)$ for $0 \leqslant m < \infty$, and $D_{CT} = D_{CT}(c)$; these are then functions defined on $\operatorname{int}\mathbb{R}^k$. As stated in the discussion above, Fu proves in Theorem 5.4 of [5] that for $0 \leqslant m < \infty$ the functions $D_m(c)$ are convex. As the pointwise limit of convex functions is convex, we conclude that $D_{CT}(c)$ is also convex. Accordingly, Assumption 1 applies to the CT line models, with cycle times as variables, that we used in our work.

It is important to note that the property of strong stochastic convexity depends on the choice of machine failure model. It is argued in [42] that a natural failure model, for

CT lines that are approximations to DT lines, is one where the *quantity produced* by a machine since the last failure determines its next failure (as opposed to failures' being determined by the *time of operation* of the machine since the last failure). The failure model used in our CT line simulation is therefore based on the quantity produced by each machine.

In our computations we modified the basic sample-path optimization method by occasionally increasing the simulation run length n, while carrying over certain information (cutting planes; see Subsection 2.2 below) from the shorter simulation runs. The effect of this practice was to introduce additional constraints on the decision variables. We did this in the interest of efficient use of information, since in large problems the computation of these cutting planes is extremely expensive. In cases where these additional constraints proved to be too restrictive we used a heuristic procedure for relaxing them. This procedure is described in Subsection 2.2.3.

In this section we have explained the conceptual method of sample-path optimization and have exhibited assumptions under which the method has been shown to converge with probability one. We have also noted that in our computations we employed expedients to improve the speed of computation, which amounted to alterations of the basic method. These expedients consisted of (1) using CT approximations to the underlying DT lines being modeled in Section 3, and (2) retaining some "old" cutting planes after increasing the length of a simulation run. Our actual computations should therefore be regarded as results of a variant of the basic conceptual method.

2.2. *The bundle/trust-region method*

As our objective functions S_n were in general nonsmooth, we chose to use a nonsmooth convex minimization algorithm of the bundle class: specifically, the bundle/trust-region (BTR) method of Schramm and Zowe [36,51]. A very similar algorithm is described and justified in [22]. In the first part of this section we give a brief description of the method; then we provide three subsections dealing with, respectively, the stopping criterion, modifications to take account of additional linear constraints, and the carryover of old cutting planes when increasing the length of a simulation run.

The basic idea of the algorithm is to use subgradients of the objective function to construct a cutting-plane approximation, which is then regularized by adding a quadratic function in order to control the proximity of the next iterate to the current one. This results in a quadratic programming problem of the form

$$\min\{\, v + (2t)^{-1}\|d\|^2 \mid v \geqslant \langle y_i^*, d\rangle - \alpha_i, \quad i \in \mathcal{B}\,\}, \tag{1}$$

where the y_i^* are subgradients and $\mathcal{B}$ is the set of indices included in the current bundle. The regularization parameter t controls the size of the step d from the current iterate, say x_n, to the next iterate x_{n+1}. A somewhat simplified description of the procedure is that one applies standard stepsize tests to the candidate step, resulting either in acceptance of that step (a *serious step*) or rejection (a *null step*); in either case a new subgradient and function value are computed. This enlarges the bundle to be used for the next

calculation, in which a different value of t may also be used. To avoid uncontrolled increase of the bundle size, there is provision for periodic restarts with a smaller bundle. A pseudocode summary of the algorithm is in Appendix A.

Note that since the step $x_{n+1} - x_n$ is determined by (1), one can easily incorporate additional linear inequality or equality constraints on x_{n+1} by simply adjoining them to the other constraints of (1). We think that this is an important advantage of using a technique (such as the bundle method) that employs constraints in computing a step, rather than modifying an unconstrained method, such as stochastic approximation, by projecting iterates onto the feasible set. We discuss this advantage further in Appendix F.

2.2.1. *Stopping criterion*

The stopping criterion for this method is the generation of a γ-subgradient x_γ^* of the objective function f at the current iterate x_n, with $\|x_\gamma^*\| \leqslant \gamma$, where γ is a preset positive tolerance. To say that x_γ^* is a γ-subgradient of f at x_n means that for each $x \in \mathbb{R}^N$ we have

$$f(x) \geqslant f(x_n) + \langle x_\gamma^*, x - x_n \rangle - \gamma.$$

If x_γ^* were exactly zero then we should have a γ-minimizer of f, but here that is not the case. Therefore we comment briefly on why this criterion is a reasonable one to use.

Suppose that f is a closed convex function, and assume further that its level sets are compact; this is a regularity assumption on f, without which one can perturb f by an arbitrarily small amount to produce a function having no minimizer. This assumption is equivalent to requiring that the origin be in the interior of dom f^*, the effective domain of the conjugate function. For each nonnegative ϵ let X_ϵ be the set of all x for which there exists $x^* \in \partial_\epsilon f(x)$ with $\|x^*\| \leqslant \epsilon$. Clearly X_0 is the set of minimizers of f, and our stopping criterion amounts to requiring that the current iterate x_n lie in X_γ for our preselected positive γ. In our numerical results in Sections 3 and 4, we refer to points of X_γ as *γ-solutions*. Also, it is clear that the sets X_ϵ are nested, and that their intersection over all positive ϵ is X_0.

Our regularity condition ensures that $0 \in \operatorname{int} \operatorname{dom} f^*$, so let β be a positive number such that the closed Euclidean ball βB about the origin with radius β lies in the interior of dom f^*. f^* is then continuous on that ball; let its minimum and maximum there be μ and M respectively. Then the indicator $I_{\beta B}$ of the ball majorizes $f^* - M$, so we have

$$\beta\|\cdot\| = I_{\beta B}^* \leqslant (f^* - M)^* = f + M.$$

It follows that for each x, $f(x) \geqslant \beta\|x\| - M$.

Now let $\gamma < \beta$. A point x_γ^* is a γ-subgradient of f at a point x exactly when for each z we have $f(z) \geqslant f(x) + \langle x_\gamma^*, z - x \rangle - \gamma$; this is the same as saying that

$$f(x) + f^*(x_\gamma^*) - \langle x_\gamma^*, x \rangle \leqslant \gamma. \tag{2}$$

Suppose that actually $x \in X_\gamma$: that is, we have in addition $\|x_\gamma^*\| \leqslant \gamma$. Then using (2) we find that

$$\begin{aligned}\gamma &\geqslant f(x) + f^*(x_\gamma^*) - \langle x_\gamma^*, x\rangle \\ &\geqslant (\beta\|x\| - M) + \mu - \gamma\|x\| = (\beta - \gamma)\|x\| - (M - \mu).\end{aligned}$$

It follows that X_γ is contained in the ball about the origin of radius $(\beta - \gamma)^{-1}[\gamma + (M - \mu)]$. In particular, the X_γ are compact sets for all sufficiently small γ.

We now claim that for any preassigned positive ϵ, there exists a positive δ such that whenever $\gamma < \delta$ we have $e[X_\gamma, X_0] < \epsilon$, where for two subsets Γ and Δ of $\mathbb{R}^N$ we define the *excess* of Γ with respect to Δ to be

$$e[\Gamma, \Delta] = \sup_{\gamma\in\Gamma} \inf_{\delta\in\Delta} \|\gamma - \delta\|.$$

Note that if $e[\Gamma, \Delta]$ is small, then each point of Γ is close to some point of Δ, but not necessarily *vice versa*.

Indeed, if the above claim were not true we could find a positive ϵ and a sequence $\{\gamma_n\}$ of positive numbers converging to zero, with a sequence of points $\{x_n\}$ such that for each n, $x_n \in X_{\gamma_n}$ and $e[x_n, X_0] \geqslant \epsilon$. The X_{γ_n} are nested and compact, so a subsequence of the $\{x_n\}$ converges to some x_0, and by our remarks above this x_0 must belong to X_0. But this contradicts the assumption that $e[x_n, X_0] \geqslant \epsilon$ for every n.

In this sense, the bundle method with the above stopping criterion is a priori convergent. However, it is not generally feasible to determine such a δ explicitly for a given ϵ. In practice, one just sets the tolerance γ at some number thought to be reasonable and computes a γ-solution, sometimes checking by repeating the computation with a smaller value of γ. This is what we did in the computations of Sections 3 and 4.

2.2.2. *Additional linear constraints*

We next describe the modifications that have to be made to a bundle method when additional linear constraints are adjoined to the problem. We consider the problem of minimizing a closed proper convex function f, and to avoid complications in the convergence analysis of the bundle method we suppose that the effective domain of f includes an open set G large enough to contain all of the points generated in the course of the computation. As we pointed out in Subsection 2.1 above, this assumption holds for the problems we considered. We shall minimize f subject to the additional constraints $Ax \leqslant a$, where A is $m \times n$ and $a \in \mathbb{R}^m$; these constraints define the set F of Subsection 2.1, and we have $F \subset G$. Let $h = f + I_F$, where I_F is the convex indicator of F (zero on F and $+\infty$ elsewhere); then our problem is to minimize h.

Solution of the quadratic programming problem (1) under the additional constraints $A(x_n + d) \leqslant a$ yields primal solutions v and d, and dual solutions (Lagrange multipliers) λ and μ, satisfying the first-order necessary optimality conditions:

$$t^{-1}d + \sum_{i\in\mathcal{B}} \lambda_i y_i^* + \mu A = 0, \qquad \lambda \geqslant 0, \quad \mu \geqslant 0, \quad \sum_{i\in\mathcal{B}} \lambda_i = 1,$$
$$\sum_{i\in\mathcal{B}} \lambda_i(\alpha_i - \langle y_i^*, d\rangle + v) = 0, \quad \langle \mu, a - A(x_n + d)\rangle = 0. \tag{3}$$

Now define

$$s^* = \sum_{i\in\mathcal{B}} \lambda_i y_i^* + \mu A, \qquad \sigma = \sum_{i\in\mathcal{B}} \lambda_i \alpha_i + \langle \mu, a - Ax_n\rangle,$$

and recall that in methods of the bundle class α_i is the "linearization error" defined by $\alpha_i = f(x_n) - f(y_i) - \langle y_i^*, x_n - y_i\rangle$. Under the assumptions that $x_n \in F$ and that f is convex we have $\sigma \geqslant 0$. We shall show that s^* is a σ-subgradient of h at x_n, and therefore that an appropriate stopping criterion to use is that both s^* and σ be small. This is the criterion we used for the computational work of Sections 3 and 4.

We have to show that for each $z \in \mathbb{R}^N$, $h(z) \geqslant h(x_n) + \langle s^*, z - x_n\rangle - \sigma$. This is clearly true whenever $z \notin F$, so assume $z \in F$. As $y_i^* \in \partial f(y_i)$ we have for each $i \in \mathcal{B}$

$$h(z) = f(z) \geqslant f(y_i) + \langle y_i^*, z - y_i\rangle = h(x_n) + \langle y_i^*, z - x_n\rangle - \alpha_i, \tag{4}$$

in which we used the definition of α_i and the fact that $x_n \in F$. Multiplying (4) by λ_i and summing over all $i \in \mathcal{B}$ yields

$$\begin{aligned} h(z) &\geqslant h(x_n) + \langle s^*, z - x_n\rangle - \sum_{i\in\mathcal{B}} \lambda_i \alpha_i - \langle \mu, A(z - x_n)\rangle \\ &\geqslant h(x_n) + \langle s^*, z - x_n\rangle - \sigma, \end{aligned}$$

where the last line follows from the fact that

$$\langle \mu, A(z - x_n)\rangle = \langle \mu, Az - a\rangle + \langle \mu, a - Ax_n\rangle \leqslant \langle \mu, a - Ax_n\rangle.$$

Therefore $s^* \in \partial_\sigma h(x_n)$, as required.

Note that if the set of constraints among $Ax \leqslant a$ that are active at x_n contains the set active at $x_n + d$, then whenever $\mu_j > 0$ we have $[A(x_n + d) - a]_j = 0$ from (3) and therefore by assumption $[Ax_n - a]_j = 0$. In this case σ reduces to the convex combination $\alpha := \sum_{i\in\mathcal{B}} \lambda_i \alpha_i$ ordinarily used in the bundle method. This happens in particular when the constraints $Ax \leqslant a$ represent linear equalities. In the general case, $\sigma \geqslant \alpha$.

2.2.3. Carryover of cutting-plane information

This subsection discusses a modification that we made to the sample-path optimization algorithm to conserve information previously computed. Recall that the bundle constraints shown in (1) are of the form $v \geqslant \langle y_i^*, d\rangle - \alpha_i$, where $\alpha_i = f(x_n) - f(y_i) - \langle y_i^*, x_n - y_i\rangle$; here f is the objective function, x_n is the current iterate and y_i is the point at which the bundle subgradient y_i^* was taken. As f is assumed to be convex, we see that the "linearization error" α_i should be nonnegative. Much of the theory behind the bundle method relies on this nonnegativity.

For a system satisfying strong stochastic convexity (Assumption 1 above) this nonnegativity should hold when we carry out the sample-path optimization procedure, since for fixed ψ and n, $S_n(\psi, x)$ is a deterministic convex function. We ensure that we use the same sample point ψ by operating the simulation with common random numbers, and the IPA algorithm [5,42] used for the continuous tandem production line problems in Section 3 provides an exact gradient for the sample function. Therefore the bundle algorithm will not produce a negative linearization error.

However, the use of a fixed n is not necessarily efficient. To shorten the computation time we would like to begin with a short simulation run and then increase the simulation effort to provide more accurate information in the later stages of the computation. A negative linearization error frequently occurs immediately after such an increase in simulation effort due to the change in the sample function being observed. We now describe the expedient that we employed to continue the computation in such a case; in the description we use the notation f_{old} for the objective function corresponding to the previous (shorter) simulation run, and f_{new} for that corresponding to the current (longer) run.

To recover from a negative linearization error one could certainly discard the bundle elements corresponding to f_{old}. However, doing so would waste much of the effort already expended to produce the current cutting plane model of the objective function. An alternate heuristic procedure was suggested to us by A. Ruszczyński: namely, instead of discarding a bundle element, lower the corresponding cutting plane by adding a correcting value to the linearization error α_i in (1).

Such a correcting value evidently has to be at least as large as $|\alpha_i|$, where i is the index of the constraint in which the negative linearization error occurred. In fact, if the negative linearization error occurs directly after a serious step the correction should be strictly larger than $|\alpha_i|$, for the following reason. The stopping criterion for the method involves the creation of a small ϵ-subgradient of f with a small ϵ, where the value ϵ is a convex combination of the α_j using Lagrange multipliers of (1) (which will be nonnegative and will sum to 1). After a serious step is taken, all α_j corresponding to indices j of constraints active in the last quadratic programming solution will be equal. Thus, if a negative α_i is found after a serious step and if a correction of $|\alpha_i|$ is added, these values become zero. This, in turn, makes it appear that an ϵ-subgradient with $\epsilon = 0$ has been found. If this ϵ-subgradient happens to be small, premature termination follows. Even if the ϵ-subgradient is not sufficiently small, the zero value of ϵ forces the algorithm remain at the current point by generating a step d of zero.

Therefore, for the problems in Sections 3 and 4, if one or more negative linearization errors occur after a serious step we add a correcting value of $2|\min\{\alpha_i \mid i \in \mathcal{B}\}|$ to all α_j corresponding to f_{old}. In such a case the function value of the current iterate x_n has been estimated via f_{new}; hence, negative linearization errors do not occur in bundle elements corresponding to f_{new}.

On the other hand, when a negative linearization error occurs directly after a null step the function value at the current iterate x_n has been estimated via f_{old}, and the new bundle element responsible for the negative linearization error at x_n corresponds to f_{new}.

In such a case we re-estimate the function value at x_n by f_{new} and add the correcting value $f_{\text{new}}(x_n) - f_{\text{old}}(x_n)$ to all α_j in the current bundle.

In our computational results in Sections 3 and 4, we refer to solutions calculated on the basis of both old and new cutting planes as *approximate γ-solutions*.

This subsection has explained how we modified the underlying method to conserve computational effort by carrying over some of the bundle information already computed. With this, the description of our implementation of the BTR method is complete. The next section begins the description of our numerical results.

3. Numerical results: tandem manufacturing lines

This section presents the results of applying the algorithm described in Section 2 to the optimization of tandem manufacturing lines containing unreliable machines. The optimization is with respect to the machine cycle times, which are required to satisfy various linear constraints including, in some cases, binding inequality constraints. In Subsection 3.1 we explain the structure and functioning of the lines. As part of our numerical experimentation was to compare our proposed method with an existing method (SRO), we describe that competing method in Appendix B. Finally, in Subsection 3.2 we present numerical results, both for comparison of our method with SRO and for solution of larger problems that SRO could not handle.

3.1. Description of the tandem production line problem

This subsection describes the manufacturing optimization problem that we first addressed in [31]. In a tandem production line processing machines are arranged in series. The product, whether discrete or continuous, arrives from an external source and starts its processing at the first machine. After being processed, it goes to the second machine, and so on, in order, until it is processed by the last machine, after which it departs from the system. Machines may have different processing rates. In a DT line, the time it takes a machine to process one unit of product is called the *cycle time*. In a CT line the natural description for processing rate of a machine is the *flow rate*, which is the maximum amount of product that a machine can process in unit time. It is clear that for a continuous machine with the same processing rate as a discrete machine, the flow rate of the former must be the reciprocal of the cycle time of the latter. Since, in our manufacturing problems, the decision variables to be optimized (and which appear in some of the constraints) are the cycle times of machines in a DT line, we will typically use the term "cycle time" in the problem specifications below. However, since we used CT lines in the simulations, it should be understood that any cycle time shown represents the reciprocal of the flow rate of the corresponding machine in the CT line simulation.

Between each pair of machines in a tandem line is a buffer of a specified size. While a machine is processing it may fail, and once failed, take some time to be repaired; both these occurrences are characterized by specified random variables. If a buffer gets full

(due to failure, or just slow speed, of a downstream machine) then upstream machines cannot process product at their normal rate. In a DT line upstream machines must stop processing until space is available in the next buffer, while in a CT line upstream machines may either stop processing or else just be forced to process at a rate less than their flow rate, specifically the rate of the bottleneck machine downstream which is constricting the flow. Similarly, if a buffer is empty, then downstream machines have to stop processing or slow down. All these characteristics result in complex dynamics of a tandem production line, which makes it difficult to analyze these lines and predict their performance. The main performance measure for a tandem line is its *throughput*, which is the amount of production completed by the last machine in unit time. Since the throughput is random, typically one is concerned with *steady-state throughput*, and that will be the performance measure for our problems as well. Specifically, our goal will be to adjust the cycle times (subject to certain constraints) to maximize the steady-state throughput.

A typical assumption used in determining the steady-state throughput of such lines is that an unlimited supply of raw material is available to the first machine, and the last machine has unlimited space for output of its production. We will also use this assumption in our simulations. We further assume that for each machine in the line, the volume of product processed to failure, and the time required for repair after a failure, are exponentially distributed random variables (the exponential distribution is convenient, but in no way necessary here). Under these assumptions, analytic expressions for steady-state throughput of two-machine CT lines are available in [9]. Also, analytic expressions for steady-state throughput for two- and three-machine DT lines have been derived under various assumptions (see the extensive bibliography in [42]). However, no such results exist for either DT or CT lines of the length considered here. Therefore they must be treated by computer simulation models. A detailed description of our simulation algorithm is in [42].

Suri and Fu [43] demonstrated that throughput in a discrete or a continuous tandem line can be nondifferentiable as a function of cycle times at points where two or more machines' cycle times are equal, and in studying these lines, we have found empirically that optimizers often have equal cycle times. Therefore, nondifferentiability appears to be a common feature in this class of problems.

3.2. *Test examples and computational results*

In this subsection we present the results of computational experiments that we performed using bundle-based stochastic optimization (BSO) and single run optimization (SRO). The objective is to minimize the reciprocal of steady-state throughput as a function of cycle times for the continuous tandem production line. The simulation used to compute objective function and gradient estimates is described in detail in [5,42]. We applied both SRO and BSO to optimize a 50-machine line, a 15-machine line, and two 2-machine continuous tandem lines, all subject to linear equality constraints. The time-consuming job for both algorithms is the computation of function and gradient

estimates via simulation. Therefore, we compared the solutions obtained using BSO and SRO with equal simulation budgets. The random number streams for the simulation were consistent for BSO and SRO. Appendix B gives the parameter settings used for the SRO method in the cases reported in this section. In addition, we used BSO to optimize the 15-machine problem and 50-machine problem subject to additional linear inequality constraints. In these cases we did not not fix the simulation budget. Instead the algorithm terminated with a γ-solution.

The specifications for the 2-machine tandem lines appear in Table 1. We minimized the reciprocal of throughput with respect to cycle times, subject to the constraint that the sum of cycle times is constant. Observe that the optimal solution to Case 1 is a nonsmooth homogeneous point (that is, a point at which two or more cycle times are equal).

Cases 1 and 2 of the 2-machine CT line problem were solved by BSO and SRO from two different initial points, denoted "a" and "b." The random number streams for the simulation differed in the "a" and "b" cases. Both methods were restricted to a total simulation budget of 1,000,000 units of simulated product volume. The BSO method used 500 units of simulation warmup volume and 99,500 units of run volume per function and gradient evaluation.

For the 2-machine problem an analytic optimal solution is available [9]. Therefore, we checked the accuracy of our results by calculating the Euclidean distance of the computed minimizer from the optimal solution. These values are labeled "Error" in Table 2. We also evaluated the reciprocal of throughput at the initial points and at the BSO and SRO solution points. These objective values are also shown in Table 2.

Since analytic solutions are not available for problems with more than two machines, we also verified all solutions by checking that the reduced gradient (that is, an objective

Table 1
Specifications for 2-machine, CT line problems

Case	Mean volume to failure		Mean time to repair		Buffer capacity	Optimal cycle times	
1	100	100	10.0	10.0	4	0.50000	0.50000
2	60	119	2.0	1.2	3	0.49707	0.50293

Table 2
Results for 2-machine CT line problems

	Initial values		BSO solution				SRO solution			
Case	Cycle times	Objective value	Objective value	ν	Norm of gradient	Error	Objective value	ν	Norm of gradient	Error
1a	0.2 0.8	0.97202	0.68956	2.0E−1	7.5E−1	1.4E−15	0.68958	2.0E−1	7.5E−1	6.7E−5
1b	0.9 0.1	1.06928	0.68963	2.0E−1	7.5E−1	1.8E−9	0.68966	2.0E−1	7.5E−1	9.8E−5
2a	0.2 0.8	0.82035	0.53741	8.3E−3	7.0E−1	1.8E−4	0.53750	3.6E−2	7.0E−1	8.2E−4
2b	0.9 0.1	0.93441	0.53741	7.7E−3	7.0E−1	1.7E−4	0.53741	5.9E−4	7.0E−1	7.1E−6

gradient reduced by an appropriate linear combination of the constraint gradients) was close to zero. For this purpose, we generated an IPA gradient estimate at each solution using a simulation warmup volume of 1,000,000 and a run volume of 10,000,000 and calculated an approximate reduced gradient by:

$$\min_{\lambda} \|f'(x) - A^*\lambda\|_\infty = \min_{\lambda,\nu}\{\nu \mid -\nu\mathbf{1} \leqslant f'(x) - A^*\lambda \leqslant \nu\mathbf{1}\},$$

where $f'(x)$ is the gradient estimate, A is the constraint coefficient matrix, and $\mathbf{1}$ is a vector with each component equal to 1. A small optimal value of ν should correspond, in a reasonably well conditioned problem with a differentiable objective function, to a near-optimal solution x.

However, in a nonsmooth problem one would not necessarily expect such a reduced gradient to be small, because one might find large subgradients close to, or even at, the minimizer. We see this illustrated in the nonsmooth Case 1, where ν is not small even though the error in the BSO solution is of the order of 10^{-15}. On the other hand, in the (barely) smooth Case 2, ν is fairly small at the computed solution.

The results shown in Table 2 indicate that BSO is competitive with the SRO method for small, equality-constrained CT line problems. We expected the gradient-based SRO method to have some difficulty in converging to the homogeneous optimal point in Case 1 because of nonsmoothness, but it successfully solved the problem. Apparently the Robbins–Monro sequence $\{a_n\}$, described in Appendix B, damped the gradient values sufficiently.

We also applied BSO and SRO to optimize a 15-machine line subject to six linear equality constraints on the cycle times, and a 50-machine line subject to five linear equality constraints. The 15-machine problem specified in Table 3 is motivated by machining system research at Ford Motor Company [49] and was also considered in [31]. Note that in Table 3 "Buffer capacity" refers to the buffer preceding the machine. The system characteristics for the 50-machine line are similar, with mean operating volumes to failure ranging between 80 and 120, mean times to repair between 4 and 10, and buffer capacities all equal to 10. A detailed description of the 50-machine CT line in Case 4 is in Appendix C.

In Case 3, the 15-machine CT line problem, the cycle times $(c_1, \ldots, c_{15})$ are subject to the following constraints:

$$c_1 + c_2 + c_3 = 0.966, \quad c_4 + c_5 = 0.6, \quad c_6 + c_7 + c_8 = 0.933,$$

$$c_9 + c_{10} = 0.6, \quad c_{11} + c_{12} + c_{13} = 1.008, \quad c_{14} + c_{15} = 0.6.$$

Table 3
Specifications for Case 3, 15-machine CT line

Machine	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Mean volume to failure	100	90	100	90	90	100	90	90	90	90	120	100	90	90	90
Mean time to repair	10.0	4.5	6.0	4.5	4.5	5.0	5.4	5.4	4.5	5.4	6.0	8.0	1.8	4.5	4.5
Buffer capacity		10	10	10	10	10	10	10	10	10	10	10	10	10	10

In Case 4, the 50-machine CT line problem, the cycle times $(c_1, \ldots, c_{50})$ are subject to the following constraints:

$$\sum_{n=1}^{10} c_n = 7.43, \quad \sum_{n=11}^{20} c_n = 4.09, \quad \sum_{n=21}^{30} c_n = 6.24,$$

$$\sum_{n=31}^{40} c_n = 5.71, \quad \sum_{n=41}^{50} c_n = 5.17.$$

The linear equality constraints in Cases 3 and 4 represent the total work to be done by machines of a given capability. We need to allocate the specific amount of work to be done by each individual machine in order to maximize throughput.

We allowed a total computer simulation budget of 1,000,000 units of run volume for solving Case 3 and 10,000,000 units for Case 4. Each problem was solved by the BSO method and by the SRO method from two different initial points, denoted "a" and "b." The BSO method used 5,000 units of simulation warmup and 35,000 units of run volume per function and gradient evaluation for the 15-machine problem, and for the 50-machine problem, 100,000 units of warmup volume and 100,000 units of run volume.

We computed approximate reduced gradient values ν to check the accuracy of the BSO and SRO results. These values appear in Table 4. As in Cases 1 and 2 above, the values ν were calculated from an IPA gradient estimate generated by simulation warmup 1,000,000 and run volume 10,000,000. We also evaluated the Euclidean distance of each computed minimizer from an "optimal" BSO γ-solution with small tolerance γ and large simulation warmup and run volume. These values are labeled "Error" in Table 4. For Case 3 we used warmup volume 50,000 and run volume 350,000 per function and gradient evaluation to compute the BSO γ-solution with tolerance $\gamma = 0.0001$. This point is labeled '"Optimal" cycle times' in Table 5. For Case 4 we used warmup volume 1,000,000 and run volume 1,000,000 per function and gradient evaluation to compute the BSO γ-solution with tolerance $\gamma = 0.001$. This point is specified in Appendix C. The objective values at these γ-solutions are labeled "Optimal" in Table 4. All objective values presented in this section are estimated using a simulation with warmup volume 1,000,000 and run volume 10,000,000.

In all cases the value of the approximate reduced gradient ν is small relative to the norm of the gradient, indicating that the solution nearly satisfies an optimality condition. Furthermore, the BSO method consistently found a solution with smaller error than did the SRO method. These results indicate that BSO is at least competitive with SRO for large CT line problems with linear equality constraints.

We also used our bundle-based method to optimize the 15-machine and 50-machine problems subject to additional linear inequality constraints. Because of the necessity for projection or other ad hoc techniques, we did not apply the SRO method to the inequality

Table 4
Results for Case 3 (15 machines) and Case 4 (50 machines)

Case	Initial objective value	"Optimal" objective value	BSO solution				SRO solution			
			Objective value	Error	ν	Norm of gradient	Objective value	Error	ν	Norm of gradient
3a	0.9224	0.6097	0.6110	2.6E−2	5.1E−2	2.3E−1	0.6104	4.6E−2	3.7E−2	2.3E−1
3b	0.6512	0.6097	0.6105	1.5E−2	3.7E−2	2.2E−1	0.6104	2.0E−2	3.8E−2	2.3E−1
4a	1.1922	1.0509	1.0516	2.2E−1	3.7E−2	2.4E−1	1.0607	9.4E−1	3.8E−2	2.0E−1
4b	1.2243	1.0509	1.0514	5.2E−1	5.4E−2	2.5E−1	1.0645	1.3E0	2.9E−2	1.8E−1

constrained problems. The added constraints on the 15-machine problem were:

$$c_1 \geqslant 0.33, \quad c_5 \geqslant 0.31, \quad c_6 \geqslant 0.40, \quad c_6 + c_8 \geqslant 0.7, \quad c_{15} \geqslant 0.35, \tag{5}$$

and

$$c_i \geqslant 0.05, \qquad i = 1, \ldots, 15.$$

The added constraints on the 50-machine problem were:

$$c_1 \geqslant 0.85, \quad c_{21} \geqslant 0.67, \quad c_{31} \geqslant 0.66, \quad c_{10} + c_{20} + c_{30} + c_{40} + c_{50} \geqslant 3.30, \tag{6}$$

and

$$c_{11} \geqslant 0.35, \quad c_{41} \geqslant 0.41, \quad c_1 + c_{50} \geqslant 0.99, \qquad c_i \geqslant 0.05 \quad i = 1, \ldots, 50.$$

These constraints were designed to perturb the solution slightly from the observed optimal solutions for Cases 3 and 4.

To optimize the inequality constrained 15-machine CT line, we initiated the BSO algorithm from a feasible point with objective value 0.9224. Using a simulation warmup of 5,000 units and a run volume of 35,000 units per function and gradient evaluation, we observed convergence to a γ-solution with tolerance $\gamma = 0.01$ in a total simulation volume of 520,000 units. At this solution the value of the objective function was 0.6256. In order to verify this solution we continued with the simulation warmup and run volume increased by a factor of 10 to 50,000 and 350,000 units per function and gradient evaluation. We observed convergence to an approximate γ-solution with tolerance $\gamma = 0.001$ in an additional simulation volume of 4,400,000 units. At this point, denoted '"Optimal" cycle times *' in Table 5, the value of the objective function is 0.6242, and the constraints shown in (5) are active, while all others are inactive. The Euclidean distance between the γ-solution with tolerance $\gamma = 0.01$ and the approximate γ-solution with tolerance $\gamma = 0.001$ is 0.042.

To optimize the inequality constrained 50-machine CT line we initiated the BSO algorithm from a feasible point with objective value 1.2217. Using a simulation warmup of 100,000 units and a run volume of 100,000 units per function and gradient evaluation, we observed convergence to a γ-solution with tolerance $\gamma = 0.05$ in a total simulation volume of 1,600,000 units. At this point the objective function value is 1.0679. In

Table 5
Case 3, 15-machine CT line, approximate optimal cycle time solutions

Machine	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
"Optimal" cycle times	.322	.322	.322	.300	.300	.311	.311	.311	.300	.300	.327	.327	.353	.301	.299
"Optimal" cycle times *	.330	.330	.306	.290	.310	.400	.233	.300	.301	.299	.327	.327	.355	.250	.350

* Inequality constrained case.

order to verify this solution we continued with the simulation warmup and run volume increased by a factor of 10 to 1,000,000 and 1,000,000 units per function and gradient evaluation. We observed convergence to an approximate γ-solution with tolerance $\gamma = 0.005$ in an additional simulation volume of 52,000,000 units. At this point, specified in Appendix C, the objective function value is 1.0537, and the constraints in (6) are active, while all others are inactive. The Euclidean distance between the γ-solution with tolerance $\gamma = 0.05$ and the approximate γ-solution with tolerance $\gamma = 0.005$ is 0.705.

4. Numerical results: stochastic PERT problems

In this section we report the results of applying our algorithm to stochastic PERT (Program Evaluation and Review Technique) problems. We describe the problems in Subsection 4.1 and report the numerical results in Subsection 4.2. Appendix E justifies the method that we used to compute the subgradient estimates.

4.1. The stochastic PERT optimization problem

The PERT technique, advanced in 1959 [26], is used to estimate the expected duration for a project defined as a set of activities which consume time and resources and are subject to temporal precedence relationships. In practice, most activities can be finished in shorter or longer time periods by increasing or decreasing the resources such as funding, labor, or machinery available to them. Typically, reducing the duration of an activity entails an additional cost for resources. We are concerned with the trade-off between completion time and cost for a project with random variable activity lengths.

This type of problem has been addressed before. In [48] Wallace examines various methods for bounding the expected completion time and cost performance function for a stochastic PERT network from below. An algorithm is presented in [50] to minimize a linear combination of expected completion time and cost when each activity length has a deterministic component based on resource investment and a discrete random variable component that is independent of resource investment. This method is based on the L-shaped decomposition method by Van Slyke and Wets [47]. In [50] Wollmer suggests that "sampling may be a viable alternative" when the random variable distributions of activity times are not discrete. However, although the idea of using Monte Carlo

simulation to analyze stochastic PERT networks was proposed as early as 1963 [46], to our knowledge no computational experiments using Monte Carlo simulation to optimize expected time and cost performance functions have been published until now.

We apply bundle-based stochastic optimization to find optimal parameter settings for stochastic PERT problems in which activity length distribution parameters may be perturbed at some cost. These parameters are subject to linear constraints representing limits on resource availability and on activity duration. We consider problems of two types: the first has uniformly distributed activity lengths, and the problem is to minimize expected completion time and cost with respect to the means in the uniform probability distributions. The choice of the uniform distribution is motivated by problems in which the expected value of an activity length may be reduced at some cost, but the variance is fixed. In the second type of problem, each activity length has a triangular probability distribution characterized by the parameters a, b, and c which represent the minimum, maximum and mode, respectively. The triangular distribution is often used in place of a beta distribution [28], partly because it requires less computer time for random variable generation. We optimize a cost function in the second type of problem with respect to a common factor x in the distribution parameters (which thus became ax, bx, and cx). This setup is motivated by problems in which both the mean and variance of activity length may be reduced at some cost.

4.2. *Numerical results: stochastic PERT*

This subsection presents numerical results for four stochastic PERT problems. In these problems we took the objective (to be minimized) to be the sum of two components: (1) the expected project completion time, and (2) a cost function that was a sum of terms of the form $k_i z_i^{-1}$, where the z_i were parameters in the underlying distributions. This choice of objective form expresses a tradeoff of increased cost against decreased project length.

To evaluate the cost function and its subgradient, we repeatedly simulated a set of activity lengths using a combined multiplicative linear congruential random number generator [23], then solved the resulting longest-path network problem using the Bellman–Ford algorithm [29, Section I.3] with an obvious modification to find longest instead of shortest paths. This calculation provided the completion time and a subgradient. Mean subgradient and completion time values were then combined with the corresponding cost components to provide objective values and subgradients for the bundle algorithm.

We first considered two PERT problems, T1 and U1, characterized by an activity-on-arc network with 7 nodes and 11 arcs. The incidence matrix of this network was:

Table 6
Triangular distribution parameters and objective cost coefficients for problem T1

Activity	1	2	3	4	5	6	7	8	9	10	11
a_i	10.0	2.0	4.0	12.0	3.0	10.1	7.3	4.9	11.1	3.5	4.9
c_i	11.1	2.2	4.9	12.4	3.5	10.3	7.9	5.1	11.3	3.6	5.1
b_i	12.3	3.0	5.2	13.0	4.1	10.5	8.2	5.5	11.9	3.8	5.5
k_i	0.6	1.0	0.4	0.8	1.4	0.2	1.6	1.8	1.2	0.6	0.4

$$\begin{bmatrix} -1 & -1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ +1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & +1 & 0 & 0 & 0 & -1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & +1 & 0 & 0 & 0 & +1 & +1 & 0 & -1 & 0 \\ 0 & 0 & 0 & +1 & 0 & 0 & 0 & -1 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & +1 & +1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & +1 & +1 & +1 \end{bmatrix}$$

In problem T1 the activity (arc) lengths have triangular distributions with minimum, maximum, and mode equal to $a_i x_i$, $b_i x_i$, and $c_i x_i$ respectively. The decision variables are the factors x_i, and these were required to satisfy the following additional linear constraints:

$$x_2 + x_7 \geqslant 1.8, \quad x_3 \geqslant 0.8, \quad x_9 \geqslant 0.6, \quad x_i \geqslant 0.5, \quad i = 1, \ldots, 11.$$

The objective function to be minimized is the sum of expected project completion time and $\sum_{i=1}^{11} k_i x_i^{-1}$. The parameter values, a_i, b_i and c_i, and the objective cost coefficients, k_i, are given in Table 6.

To solve T1 we initiated the BSO algorithm from the point $x_i = 1$, $i = 1, \ldots, 11$, which has an objective value of 36.1039. We used 5,000 network solutions per function and subgradient calculation to achieve convergence to a γ-solution with tolerance $\gamma = 0.01$. To validate this solution we continued the BSO algorithm with the number of network solutions per function and subgradient evaluation increased by a factor of 5 to 25,000, and observed convergence to an approximate γ-solution with tolerance $\gamma = 0.001$. The γ-solutions for $\gamma = 0.01$ and $\gamma = 0.001$, the corresponding objective function values, and the total number of network solutions required are shown in Table 7. In order to obtain increased accuracy, the objective function values at the initial point (given above) and at the computed solutions (given in Table 7) were estimated using 50,000 network solutions.

Problem U1 has the same network configuration as T1 with 7 nodes and 11 arcs, but the activity (arc) length distribution in U1 is uniform. Our objective is of the same form as before, except that the mean activity lengths μ_i, $i = 1, \ldots, 11$ replace the factors x_i. The costs k_i are specified in Table 8, which also shows the spreads

Table 7
Results for 7-node, 11-arc, triangular distribution problem T1

Tolerance γ	Network solutions	Objective value	Solution					
0.010	360,000	27.1949	0.500	1.338	2.534	0.512	0.864	0.500
			1.217	1.255	0.600	0.676	1.668	
0.001	1,335,000	27.1944	0.500	1.307	2.546	0.510	0.861	0.501
			1.227	1.265	0.600	0.666	1.675	

(lengths of supports) of the uniform distributions for activity lengths. The variables μ_i, $i = 1, \ldots, 11$ are subject to the following constraints:

$$\mu_4 + \mu_8 \geqslant 9, \qquad \mu_i \geqslant 2, \quad i = 1, \ldots, 11.$$

We initiated the BSO algorithm from the point $\mu_i = 5$, $i = 1, \ldots, 11$, which has objective value 25.6. We used 5,000 network solutions per function and subgradient estimation to achieve convergence to a γ-solution with tolerance $\gamma = 0.01$. To validate this solution we continued the BSO algorithm with the number of network solutions increased by a factor of 5 to 25,000, and observed convergence to an approximate γ-solution with tolerance $\gamma = 0.001$. The γ-solutions for $\gamma = 0.01$ and $\gamma = 0.001$, the corresponding objective function values, and the total number of network solutions required are contained in Table 9. Note that all objective function values presented in this section were estimated using 50,000 network solutions.

We also applied BSO to problems T2 and U2 characterized by an activity-on-arc network with 70 nodes and 110 arcs. This network is specified in Appendix D. In problem T2 the activity lengths have triangular distributions. As for the previous problem with triangular distributions, the decision variables are factors x_i in the minimum $(a_i x_i)$, maximum $(b_i x_i)$ and mode $(c_i x_i)$ of the activity length distribution. We chose the values

Table 8
Spreads of uniform activity distributions and objective cost coefficients for problem U1

Activity	1	2	3	4	5	6	7	8	9	10	11
Spread	0.2	0.3	0.6	0.7	0.1	0.2	0.4	0.6	0.1	0.4	0.5
k_i	3.0	5.0	4.0	6.0	1.0	7.0	3.0	5.0	9.0	6.0	3.0

Table 9
Results for 7-node, 11-arc, uniform distribution problem U1

Tolerance γ	Network solutions	Objective value	Solution					
0.01	125,000	24.134	5.023	3.411	8.775	4.527	2.929	4.592
			5.513	4.473	4.472	3.432	7.714	
0.001	850,000	24.133	5.052	3.416	8.812	4.487	2.908	4.613
			5.504	4.513	4.487	3.460	7.749	

of a_i, c_i and b_i randomly from $[2.0, 2.2]$, $[2.2, 2.4]$ and $[2.4, 2.6]$, respectively. The objective function is the sum of expected project completion time and $\sum_{i=1}^{110} k_i x_i^{-1}$; here the k_i were chosen from the interval $[0, 3]$. Parameter values a_i, b_i, c_i, and costs k_i are specified in Appendix D. The decision variables are required to satisfy the following constraints:

$$x_{26} + x_{27} \geqslant 2.0, \quad x_{31} + x_{44} \geqslant 1.9, \quad x_{36} + x_{50} + x_{88} \leqslant 3,$$

$$x_{58} + x_{61} + x_{101} \leqslant 3, \quad x_i \geqslant 0.6, \quad i = 1, \ldots, 110.$$

In problem U2 each activity length has a uniform distribution with spread, s_i, in the interval $[0, 1]$. As for the previous problem with uniform distributions, the decision variables μ_i are the mean activity lengths. The objective function is the sum of the expected project completion time and $\sum_{i=1}^{110} k_i \mu_i^{-1}$, with k_i in the interval $[1, 10]$. Parameter values s_i and costs k_i are specified in Appendix D. The decision variables are subject to the following constraints:

$$\mu_{26} + \mu_{27} \geqslant 29, \quad \mu_{31} + \mu_{44} \geqslant 19, \quad \mu_{50} + \mu_{88} \geqslant 6, \quad \mu_{58} + \mu_{61} \leqslant 20,$$

$$\mu_1 \geqslant 5, \quad \mu_6 \geqslant 5, \quad \mu_8 \geqslant 5, \quad \mu_{27} \geqslant 6, \quad \mu_{37} \geqslant 12, \quad \mu_{40} \geqslant 3,$$

$$\mu_{94} \geqslant 6, \quad \mu_{95} \geqslant 26, \quad \mu_i \geqslant 2, \qquad i = 1, \ldots, 110.$$

To solve T2 we initiated the BSO algorithm from the point $x_i = 1$, $i = 1, \ldots, 110$ with objective value 200.3, using 1,000 network solutions per function and subgradient evaluation. In a total of 51,000 network solutions we observed convergence to a γ-solution with tolerance $\gamma = 0.5$. At this solution the value of the objective function was 119.7. In order to verify this solution we continued with the number of network solutions increased by a factor of 5 to 5,000 per function and subgradient evaluation. After an additional 390,000 network solutions we observed convergence to an approximate γ-solution with tolerance $\gamma = 0.05$ and objective value 119.0. The Euclidean distance between the 0.5-solution and the 0.05-solution, which appears in Appendix D, was 7.8. The following constraints were active at both solutions:

$$x_2 \geqslant 0.6, \quad x_{71} \geqslant 0.6, \quad x_{88} \geqslant 0.6,$$

$$x_{36} + x_{50} + x_{88} \leqslant 3, \quad x_{58} + x_{61} + x_{101} \leqslant 3.$$

To solve U2 we initiated the BSO algorithm from a feasible point with objective value 252.0, using 5,000 network solutions per function and subgradient evaluation. We observed convergence to a γ-solution with tolerance $\gamma = 0.5$ in a total of 195,000 network solutions. At this solution the value of the objective function is 144.3. In order to verify this solution we continued with the number of network solutions increased by a factor of 5 to 25,000 per function and subgradient evaluation. We observed convergence to an approximate γ-solution with tolerance $\gamma = 0.05$ in an additional 850,000 network solutions. At the $\gamma = 0.05$ solution the value of the objective function is 142.8. The

Euclidean distance between the 0.5-solution and the 0.05-solution, which appears in Appendix D, is 15.1. The following constraints are active at both solutions:

$$\mu_1 \geqslant 5, \quad \mu_6 \geqslant 5, \quad \mu_8 \geqslant 5, \quad \mu_{40} \geqslant 3,$$

$$\mu_{46} \geqslant 2, \quad \mu_{89} \geqslant 2, \quad \mu_{94} \geqslant 6, \quad \mu_{95} \geqslant 26.$$

5. Conclusion

This paper has proposed a method, related to retrospective optimization, for optimizing performance functions in certain stochastic systems, with respect to parameters appearing in the underlying probability distributions of the systems. These performance functions may be nonsmooth, and the optimization may be made subject to linear constraints of both equality and inequality type. The essential requirement for applying this method is that the function to be minimized be a limit of functions that are convex, with respect to the parameters, along almost every sample path of the system. We have explained in Sections 2, 3, and 4 respectively the mathematical foundations of the method and the results of its application to continuous tandem production lines (Section 3) and to stochastic PERT problems (Section 4). In these numerical studies the method appeared to give good results on problems that by current standards would be considered very difficult to solve.

Appendix A. Summary of the bundle-based stochastic optimization algorithm

This appendix contains a pseudocode summary of the Bundle-Based Stochastic Optimization algorithm used to solve the continuous tandem production line problems in Section 3 and stochastic PERT problems in Section 4. The algorithm is derived from bundle/trust-region code written by Helga Schramm. For further information see [36,51].

Step 0: INITIALIZATION

Given an initial feasible point $x \in \mathbb{R}^N$ and parameters:

$k_{\max} \geqslant 3$ (*maximum number of bundle elements for the quadratic program*)

$\gamma > 0$ (*tolerance used in stopping criterion*)

$f_{\min}$ (*rough approximation of function minimum used to scale the quadratic program*)

comment We use $k_{\max} = 200$ in our experiments. In the CT-line problems of Section 3, $f_{\min}$ takes the value of the largest ratio of "mean time to repair" to "mean volume to failure" for a machine in the line. In the stochastic PERT problems of Section 4 we use $f_{\min} = 0$.)

estimate $f(x)$ and $y_1^* \in \partial f(x)$ via simulation

(*use consistent random number stream*)

initialize $k \leftarrow 0$, $n \leftarrow 1$, $\alpha_{k+1} \leftarrow 0$, $\sigma_k \leftarrow 0$, $s_k^* \leftarrow y_1^*$,
$t_{\max} \leftarrow \max\{\, 100(f(x) - f_{\min}),\ 1 \,\}$,
$t \leftarrow \min\{\, t_{\max},\ \max\{\, (100\|y_1^*\|)^{-1}(f(x) - f_{\min}),\ \|y_1^*\|^{-1} \,\}\}$

Step 1: BUNDLE UPDATE
add α_{k+1} and y_{k+1}^* to the bundle: $k \leftarrow k+1$
reset (QP) scale variables: $t_l \leftarrow 0$, $t_u \leftarrow t_{\max}$

Step 2: QUADRATIC PROGRAM
$\min\{\, v + (2t)^{-1}\|d\|^2 \mid A(x+d) \leqslant a, \quad v \geqslant \langle y_i^*, d\rangle - \alpha_i \quad i = 1, \ldots, k\}$ (QP)
comment In the quadratic programming problem (QP) the y_i^* are subgradients, α_i are cutting plane linearization errors at x, and k is the number of elements in the current bundle.
solve (QP) for primal solutions v and d, and for dual solutions λ and μ corresponding to the cutting plane constraints $v \geqslant \langle y_i^*, d\rangle - \alpha_i$ and the additional linear constraints $A(x+d) \leqslant a$, respectively
compute $s_k^* = \sum_{i=1}^k \lambda_i y_i^* + \mu A$ and $\sigma_k = \sum_{i=1}^k \lambda_i \alpha_i + \langle \mu, a - Ax\rangle$
if $[\|s_k^*\| \leqslant \gamma$ **and** $\sigma_k \leqslant \gamma]$
stop (*x is approximate γ-solution*)
if the simulation computational budget is exhausted
stop (*x is approximate solution*)
else go to

Step 3: DESCENT TEST.
Step 3: DESCENT TEST
comment To shorten the total computation time, we may choose to begin with a short simulation run and then increase the simulation effort in the later stages of the computation, when s_k^* and σ_k are sufficiently small. This may cause one or more negative linearization errors. When the simulation effort is increased set $n \leftarrow k$.
estimate $f(x+d)$ and $y_{k+1}^* \in \partial f(x+d)$ via simulation.
(*use consistent random number stream*)
if $f(x+d) - f(x) < 0.1v$
$t_l \leftarrow t$
if $\langle y_{k+1}^*, d\rangle \geqslant 0.2$ **or** $t_u - t \leqslant 0.1$
go to Step 4a: SERIOUS STEP (*after a serious step the model will provide a different search direction or t is near its upper bound*)
if in the last iteration of Step 3, t was increased without a serious step but d was unchanged
go to Step 4a: SERIOUS STEP.
else (*attempt to increase the search step size in the quadratic program*)
if $t_u = t_{\max}$
$t \leftarrow \min\{\, 5t,\ t_{\max}\,\}$
else $t \leftarrow \min\{\, t_l + 0.5(t_u - t_l),\ \max\{5t,\ t_l + 0.25(t_u - t_l)\}\}$

go to Step 2: QUADRATIC PROGRAM

if $[f(x+d) - f(x) \geqslant 0.1v]$

(*the cutting plane model of f is not adequate or t is too large*)

$t_u \leftarrow t$.

if $t_l = 0$

if $f(x) + \langle y_{k+1}^*, d\rangle - f(x+d) \leqslant \max\{\sigma_k, \gamma\}$

go to Step 4b: NULL STEP

if $|f(x+d) - f(x)| \leqslant \|s_{k-1}^*\| + \sigma_{k-1}$

go to Step 4b: NULL STEP

else

$t \leftarrow t_l + 0.1(t_u - t_l)$

go to Step 2: QUADRATIC PROGRAM

else $[t_l > 0]$

(*in the last iteration of step 3, T was increased without a serious step*)

$t \leftarrow t_l + 0.5(t_u - t_l)$

go to Step 2: QUADRATIC PROGRAM.

Step 4a: SERIOUS STEP

if $f(x+d) - f(x) \leqslant 0.7v$ **or** this is the third consecutive serious step

(*without a null step or increase in t in Step 4a*)

$t \leftarrow \min\{\ 5t\ ,\ t_{\max}\ \}$

$x \leftarrow x + d$ (*accept candidate $x+d$*)

for $i = 1, \ldots, k$

$\alpha_i \leftarrow \left(\alpha_i + f(x+d) - f(x) - \langle y_i^*, d\rangle\right)$

$\alpha_{k+1} \leftarrow 0$ (*update the linearization errors*)

if $\alpha_i < 0$

(*negative linearization error caused by an increase in simulation effort in Step 3*)

$\mathrm{A} \leftarrow 2|\min\{\alpha_i |\ i \in 1, \ldots, n-1\}|$

for $i = 1, \ldots, n-1$

$\alpha_i \leftarrow \alpha_i + \mathrm{A}$

go to Step 5: BUNDLE RESET

Step 4b: NULL STEP

if this is the $(N+1)$st consecutive null step

(*without a serious step or decrease in t in Step 4b*)

$t \leftarrow 0.5t$

$\alpha_{k+1} \leftarrow f(x) - f(x+d) + \langle y_{k+1}^*, d\rangle$ (*update the linearization error*)

if $\alpha_{k+1} < 0$

(*negative linearization error caused by an increase in simulation effort in Step 3*)

re-estimate $f(x)$ and $y_{k+2}^* \in \partial f(x)$ via simulation

Δf is the increase in the estimate of $f(x)$ due to the increase in simulation effort in Step 3 ($\alpha_{k+1} < 0$ *implies* $\Delta f > 0$)

for $i = 1, \ldots, k+1$

$\alpha_i \leftarrow \alpha_i + \Delta f$
$\alpha_{k+2} \leftarrow 0$, $k \leftarrow k+1$
go to Step 5: BUNDLE RESET

Step 5: BUNDLE RESET
if $k \geqslant k_{\max}$ (*reset the bundle*)
$j \leftarrow 0$
for $i = 1 \ldots k$ (*retain cutting planes that provide a good approximation to* f *at* x *or that correspond to an active constraint* $v \geqslant \langle y_i^*, d\rangle - \alpha_i$ *in the last solution of* (QP))
if $\lambda_i > 0$ **or** $\alpha_i = 0$
$j \leftarrow j+1$, $\lambda_j \leftarrow \lambda_i$, $\alpha_j \leftarrow \alpha_i$
if $j < k_{\max}$
$k \leftarrow j$
if $j \geqslant k_{\max}$ (*still too many cutting planes*)
$\alpha_1 \leftarrow \sum_{i=1}^{j} \lambda_i \alpha_i$, $y_1^* \leftarrow \sum_{i=1}^{j} \lambda_i y_i^*$
if $\alpha_i > 0$ for all $i = 1, \ldots, j$
$l \leftarrow j$
else $l \leftarrow \arg\max\{i \mid \alpha_i = 0,\ i \leqslant j\}$
$\alpha_2 \leftarrow \alpha_l$, $y_2^* \leftarrow y_l^*$, $k \leftarrow 2$
go to Step 1: BUNDLE UPDATE

Appendix B. Single run optimization (SRO)

This appendix describes single run optimization (SRO), a competitive stochastic approximation method that we implemented for comparison purposes. The fundamental concept of SRO is to apply a single run gradient estimator to a stochastic system in order to get sensitivity information while the system is evolving. The gradient estimate is used to improve the parameter settings while the system continues to evolve. By repeatedly producing a gradient estimate and updating the parameter settings the system may be optimized during a single experiment. SRO converges rapidly in comparison to other conventional simulation optimization algorithms because the simulation does not have to be restarted, with a new warmup period, after each gradient step. However, one disadvantage is that transient phenomena are introduced at each iteration. Another

Table B.1
SRO settings for CT line problems in Section 3.2

Case	a_0	Iterations	Run volume per iteration
1	1	1,000	1,000
2	1	1,000	1,000
3	1	100	10,000
4	10	100	100,000

disadvantage is that SRO, like other stochastic approximation methods, requires some ad hoc method, such as projection, to deal with linear inequality constraints. Empirical studies of single run optimization algorithms can be found in [25,44,45].

The SRO algorithm we use combines an adaptation of the stochastic approximation method of [32] with infinitesimal perturbation analysis (IPA) gradient estimates. Essentially, the simulation model of the CT line is optimized by iteratively estimating the gradient of throughput with respect to cycle times and then updating the cycle time values based on the gradient estimate, without restarting the simulation. The IPA gradient estimation algorithm is restarted after each change in the cycle times. For a complete discussion of the IPA algorithm and the CT line simulation model see [5,42].

The sequence of cycle times $\{c_i\}$ is generated by

$$c_{i+1} = c_i - a_i g_c(c_i),$$

where $a_i = i^{-1}a_0$ and $g_c(\cdot)$ is the IPA estimate of the gradient of the reciprocal of throughput, projected onto the feasible region. We terminate the SRO algorithm after a fixed simulation computational budget, and take the final set of cycle times in the sequence as the approximate optimizer. To achieve correct convergence, the SRO method requires tuning of the scale factor a_0 and the run volume per iteration (cycle time update). The values of a_0, number of iterations, and run volume per iteration used to solve the problems in Section 3.2 appear in Table B.1. Cases 1 and 2 are 2-machine CT line problems for which the total simulation budget is 1,000,000 volumetric units. Cases 3 and 4 are 15-machine and 50-machine line problems with simulation budgets of 1,000,000 and 10,000,000 volumetric units, respectively.

Appendix C. Specification and solution for the 50-machine CT line

This appendix provides a detailed description and approximate solutions for the 50-machine continuous tandem line problems considered in Section 3.2.

For each machine in the CT line Table C.1 specifies the following parameters: the machine number m, the mean volume to failure MVF, the mean time to repair MTR, and the buffer capacity BC. Note that buffer capacity refers to the buffer preceding the machine. Table C.1 also contains an approximate minimizer of the reciprocal of throughput for the 50-machine CT line subject to the following cycle time constraints:

$$\sum_{n=1}^{10} c_n = 7.43, \quad \sum_{n=11}^{20} c_n = 4.09, \quad \sum_{n=21}^{30} c_n = 6.24,$$

$$\sum_{n=31}^{40} c_n = 5.71, \quad \sum_{n=41}^{50} c_n = 5.17.$$

This approximate minimizer, denoted by CT in the table, is a BSO γ-solution with tolerance $\gamma = 0.001$ calculated with simulation warmup 1,000,000 and run volume 1,000,000 per function and subgradient estimation.

Table C.1
Parameters and approximate optimal cycle time solution for 50-machine CT line

m	MVF	MTR	BC	CT	CT*	m	MVF	MTR	BC	CT	CT*
1	116.65	4.82		0.808	0.850	26	102.58	5.67	10	0.605	0.609
2	101.44	8.79	10	0.747	0.753	27	98.44	9.85	10	0.614	0.596
3	101.86	7.02	10	0.747	0.750	28	109.22	5.89	10	0.630	0.624
4	112.07	6.91	10	0.746	0.752	29	117.36	8.27	10	0.623	0.625
5	112.42	9.69	10	0.727	0.722	30	96.46	8.55	10	0.622	0.633
6	116.08	9.11	10	0.729	0.724	31	110.98	6.80	10	0.538	0.660
7	97.69	9.75	10	0.729	0.715	32	118.29	5.12	10	0.535	0.601
8	95.06	7.25	10	0.730	0.723	33	90.25	6.39	10	0.528	0.393
9	115.51	6.59	10	0.733	0.722	34	103.57	6.87	10	0.537	0.443
10	99.05	5.29	10	0.734	0.719	35	88.05	8.15	10	0.532	0.378
11	111.69	9.18	10	0.245	0.353	36	113.83	5.77	10	0.616	0.633
12	101.34	9.82	10	0.242	0.320	37	84.58	4.06	10	0.637	0.622
13	115.08	5.16	10	0.286	0.355	38	87.14	6.55	10	0.628	0.651
14	98.38	9.22	10	0.290	0.357	39	113.94	4.27	10	0.628	0.663
15	82.09	4.51	10	0.419	0.463	40	118.64	9.38	10	0.529	0.665
16	101.59	5.71	10	0.435	0.470	41	89.88	5.22	10	0.371	0.483
17	86.58	7.87	10	0.430	0.316	42	108.54	6.36	10	0.492	0.492
18	105.86	4.15	10	0.566	0.421	43	101.22	9.33	10	0.365	0.327
19	104.92	5.49	10	0.575	0.413	44	95.23	4.60	10	0.611	0.262
20	81.22	4.19	10	0.601	0.622	45	96.02	4.75	10	0.649	0.504
21	96.88	4.94	10	0.668	0.670	46	103.29	7.48	10	0.482	0.695
22	95.89	4.27	10	0.668	0.668	47	114.63	4.91	10	0.674	0.372
23	99.93	8.33	10	0.606	0.602	48	98.60	4.34	10	0.449	0.696
24	87.48	9.83	10	0.603	0.605	49	87.42	8.47	10	0.373	0.678
25	100.32	6.45	10	0.602	0.607	50	93.68	6.51	10	0.706	0.661

*Inequality constrained case.

The final column of Table C.1, labeled CT*, is an approximate minimizer for the problem subject to additional linear inequality constraints:

$$c_1 \geqslant 0.85, \quad c_{21} \geqslant 0.67, \quad c_{31} \geqslant 0.66, \quad c_{10} + c_{20} + c_{30} + c_{40} + c_{50} \geqslant 3.30,$$

$$c_{11} \geqslant 0.35, \quad c_{41} \geqslant 0.41, \quad c_1 + c_{50} \geqslant 0.99, \quad c_i \geqslant 0.05 \qquad i = 1, \ldots, 50.$$

This point is an approximate BSO γ-solution with tolerance $\gamma = 0.005$ calculated with initial simulation warmup 100,000 and run volume 100,000, and with increased simulation warmup 1,000,000 and run volume 1,000,000 in the later stages of the computation, as described in Section 3.2.

Appendix D. Structure and solutions for stochastic PERT problems T2 and U2

This appendix contains a detailed description and approximate optimal solutions for the stochastic PERT problems T2 and U2 in Section 4.2.

The problems T2 and U2 are characterized by an activity-on-arc network with 70 nodes and 110 arcs. The network topology, which is the same for both, is specified in

Table D.1. Note that node 1 and node 70 are the distinguished nodes *Start* and *Finish*, respectively.

In problem T2 the activity (arc) lengths have triangular distributions. The decision variables $x_i, i = 1, \ldots, 110$ are factors in the minimum $(a_i x_i)$, maximum $(b_i x_i)$, and mode $(c_i x_i)$ of these triangular activity length distributions. The objective function is the sum of expected completion time and $\sum_{i=1}^{110} k_i x_i^{-1}$. The parameters a_i, b_i, c_i, and costs k_i appear in Tables D.2 and D.3, which also contain the approximate γ-solution with tolerance $\gamma = 0.05$ found by bundle-based stochastic optimization. The value of the objective function at this point is 119.0. Note that all objective values presented in this appendix were estimated using 50,000 network solutions.

In problem U2 each activity (arc) length has a uniform distribution with spread s_i. The decision variables $\mu_i, i = 1, \ldots, 110$ are the mean activity lengths. The objective function is the sum of expected completion time and $\sum_{i=1}^{110} k_i \mu_i^{-1}$. The parameters k_i and s_i appear in Table D.4, which also contains the approximate γ-solution with tolerance $\gamma = 0.05$ found by bundle-based stochastic optimization. The value of the objective function at this point is 142.8.

Table D.1
Network structure for stochastic PERT problems T2 and U2

Arc	Origin node	Terminal node	Arc	Origin node	Terminal node	Arc	Origin node	Terminal node	Arc	Origin node	Terminal node
1	1	2	29	19	23	57	39	26	84	58	59
2	1	30	30	18	22	58	26	27	85	26	28
3	1	29	31	22	57	59	38	40	86	56	60
4	2	3	32	12	13	60	40	41	87	58	60
5	3	4	33	13	24	61	43	51	88	59	61
6	3	5	34	23	27	62	36	42	89	60	61
7	4	6	35	24	27	63	36	44	90	52	53
8	4	10	36	25	24	64	36	45	91	53	41
9	10	5	37	25	26	65	45	47	92	53	54
10	10	14	38	12	25	66	45	46	93	54	55
11	3	11	39	12	39	67	46	48	94	55	56
12	2	12	40	8	39	68	46	49	95	56	62
13	2	8	41	29	39	69	44	49	96	55	63
14	3	9	42	29	37	70	42	50	97	41	28
15	9	13	43	30	33	71	38	52	98	54	65
16	11	13	44	31	37	72	52	67	99	65	66
17	6	13	45	30	31	73	51	67	100	64	66
18	6	19	46	31	32	74	50	49	101	41	69
19	6	7	47	32	34	75	49	64	102	69	68
20	5	7	48	34	38	76	67	64	103	52	68
21	5	16	49	32	35	77	48	66	104	66	70
22	16	17	50	35	36	78	20	57	105	68	70
23	7	15	51	33	34	79	28	21	106	63	70
24	15	18	52	36	34	80	27	56	107	62	70
25	19	17	53	35	43	81	21	58	108	61	70
26	17	21	54	43	52	82	21	59	109	68	70
27	18	20	55	37	38	83	57	58	110	57	62
28	14	18	56	39	40						

Table D.2
Triangular distribution parameters, objective cost coefficients and approximate optimal solution for problem T2 (Arcs 1–56)

Arc	Cost	Parameters			Soln.	Arc	Cost	Parameters			Soln.
i	k_i	a_i	c_i	b_i	x_i	i	k_i	a_i	c_i	b_i	x_i
1	2.071	2.003	2.262	2.597	1.239	29	0.241	2.190	2.391	2.509	3.292
2	0.295	2.072	2.218	2.414	0.600	30	0.504	2.199	2.270	2.460	1.177
3	0.218	2.090	2.317	2.541	1.852	31	2.579	2.056	2.243	2.596	2.721
4	0.947	2.047	2.351	2.598	0.926	32	1.182	2.071	2.250	2.491	7.978
5	2.577	2.024	2.208	2.445	1.699	33	0.276	2.127	2.281	2.566	1.093
6	1.582	2.196	2.382	2.513	4.980	34	0.496	2.137	2.233	2.415	4.630
7	2.613	2.041	2.202	2.433	2.724	35	2.895	2.154	2.362	2.499	3.511
8	2.472	2.038	2.386	2.415	2.156	36	1.350	2.153	2.227	2.425	1.162
9	0.722	2.142	2.228	2.538	1.491	37	0.633	2.042	2.203	2.561	3.234
10	2.361	2.049	2.375	2.416	3.847	38	1.458	2.061	2.339	2.410	4.742
11	0.965	2.053	2.319	2.414	3.843	39	0.702	2.189	2.305	2.587	3.463
12	1.037	2.192	2.276	2.486	2.679	40	1.214	2.190	2.350	2.572	3.343
13	0.994	2.154	2.212	2.502	2.884	41	0.735	2.187	2.228	2.486	5.480
14	0.844	2.187	2.274	2.590	3.972	42	1.106	2.151	2.272	2.511	5.659
15	1.979	2.174	2.353	2.424	5.892	43	2.492	2.182	2.264	2.530	4.341
16	2.611	2.065	2.219	2.464	6.301	44	2.547	2.079	2.374	2.405	6.506
17	2.457	2.091	2.289	2.492	5.749	45	0.039	2.173	2.223	2.490	0.600
18	0.416	2.020	2.352	2.543	2.366	46	0.804	2.001	2.231	2.405	1.090
19	2.043	2.013	2.334	2.459	2.977	47	0.931	2.044	2.294	2.564	5.141
20	1.173	2.053	2.335	2.542	1.962	48	0.341	2.051	2.298	2.465	0.912
21	2.404	2.120	2.358	2.528	4.658	49	2.250	2.023	2.226	2.481	1.879
22	0.778	2.096	2.344	2.544	2.735	50	2.135	2.043	2.299	2.498	1.238
23	1.850	2.193	2.388	2.570	1.893	51	0.881	2.134	2.379	2.422	2.600
24	2.459	2.053	2.216	2.512	2.233	52	1.235	2.127	2.232	2.553	2.380
25	1.874	2.097	2.315	2.497	5.995	53	1.308	2.151	2.302	2.497	2.111
26	1.239	2.016	2.231	2.412	2.932	54	2.476	2.007	2.341	2.472	3.060
27	0.409	2.136	2.226	2.457	1.104	55	0.053	2.043	2.291	2.529	0.763
28	2.313	2.029	2.379	2.549	3.653	56	2.098	2.150	2.263	2.585	3.466

Appendix E. Convexity properties of PERT problems

This appendix shows that the PERT problems analyzed in Section 4.1 satisfy Assumption 1 (strong stochastic convexity) and explains how to compute subgradients of the expected-value objective functions.

The underlying problem is defined by an acyclic directed network with nonnegative (random) arc lengths, in which we try to find the longest path between two distinguished nodes, *Finish* and *Start*. This is the dual (activity-on-arc) form of the PERT problem; the arc lengths represent times required to perform various activities, and the length of the longest path is, by duality, also the shortest time in which one can complete all activities while observing the required precedence relations. For additional detail see, e.g. [2, Section 8.3], and for terms and results from convex analysis see [34].

Suppose we denote by $\mathcal{N}$ the set of flows in the network that are feasible for our problem (that is, that result in sending one unit of flow from *Finish* to *Start* while conserving flow at all other nodes), and the arc lengths by $g := (g_1, \ldots, g_n)$. Then the

Table D.3
Triangular distribution parameters, objective cost coefficients and approximate optimal solution for problem T2 (Arcs 57–110)

Arc	Cost	Parameters			Soln.	Arc	Cost	Parameters			Soln.
i	k_i	a_i	c_i	b_i	x_i	i	k_i	a_i	c_i	b_i	x_i
57	0.712	2.042	2.278	2.418	4.320	84	2.175	2.056	2.248	2.559	2.701
58	2.958	2.144	2.263	2.567	1.266	85	1.528	2.172	2.273	2.532	3.757
59	1.308	2.078	2.329	2.492	2.529	86	1.299	2.081	2.243	2.572	2.229
60	2.096	2.145	2.249	2.477	2.387	87	2.133	2.060	2.255	2.529	1.625
61	0.896	2.071	2.361	2.521	0.693	88	0.205	2.159	2.211	2.596	0.600
62	1.135	2.196	2.321	2.484	4.612	89	2.950	2.038	2.268	2.505	1.783
63	2.558	2.022	2.347	2.463	6.846	90	0.988	2.200	2.236	2.518	1.376
64	0.207	2.028	2.356	2.566	1.767	91	1.481	2.085	2.203	2.552	2.879
65	1.455	2.168	2.400	2.440	9.303	92	2.588	2.048	2.250	2.441	2.779
66	0.296	2.143	2.387	2.466	2.170	93	1.571	2.109	2.278	2.500	2.215
67	2.126	2.135	2.211	2.567	5.984	94	2.805	2.130	2.315	2.535	3.131
68	0.596	2.142	2.272	2.526	6.909	95	1.916	2.009	2.219	2.451	2.724
69	2.177	2.193	2.338	2.488	6.202	96	2.696	2.124	2.291	2.578	6.667
70	1.081	2.104	2.383	2.465	4.506	97	2.166	2.167	2.303	2.441	2.268
71	0.038	2.157	2.331	2.499	0.600	98	0.309	2.185	2.352	2.566	4.303
72	1.662	2.027	2.291	2.541	6.953	99	0.198	2.161	2.306	2.552	3.571
73	1.780	2.137	2.250	2.539	9.084	100	0.335	2.053	2.294	2.410	1.423
74	0.845	2.177	2.338	2.466	3.948	101	2.088	2.044	2.342	2.573	1.042
75	0.312	2.134	2.378	2.523	1.656	102	2.884	2.195	2.296	2.474	4.891
76	0.652	2.108	2.378	2.468	3.709	103	0.256	2.094	2.287	2.560	5.215
77	2.353	2.133	2.286	2.440	6.199	104	1.267	2.135	2.292	2.421	2.307
78	2.840	2.112	2.351	2.493	2.828	105	2.844	2.005	2.332	2.590	11.633
79	2.271	2.093	2.208	2.420	2.136	106	0.102	2.001	2.247	2.403	1.395
80	0.802	2.048	2.293	2.567	1.798	107	2.586	2.180	2.226	2.587	2.543
81	1.327	2.015	2.324	2.405	1.557	108	2.112	2.195	2.347	2.563	1.198
82	2.705	2.116	2.313	2.475	4.200	109	2.258	2.121	2.354	2.435	4.374
83	2.034	2.145	2.389	2.571	1.864	110	2.086	2.122	2.289	2.534	3.798

length of the longest path is $I^*_{\mathcal{N}}(g)$ (the support function of $\mathcal{N}$, evaluated at g), and this is a closed convex function of g. In our problems g will be an n-dimensional random variable, and we wish to compute the expectation of $I^*_{\mathcal{N}}(g)$.

We now suppose that each g_i may depend continuously on a vector of parameters $z_i \in \mathbb{R}^{n_i}$, and we let

$$z = (z_1, \ldots, z_n) \in \mathbb{R}^N, \qquad g(z) = (g_1(z_1), \ldots, g_n(z_n)),$$

where $N = \sum_{i=1}^{n} n_i$. Therefore we can write the length of the longest path as a composite function of z:

$$\phi(z) := I^*_{\mathcal{N}}(g(z)).$$

We wish to investigate various properties of ϕ (such as convexity) and to develop a formula for its subdifferential in terms of other quantities already introduced.

Note for future reference that $I^*_{\mathcal{N}}$ is nondecreasing in the vector g: that is, if $g' \geqslant g$ then $I^*_{\mathcal{N}}(g') \geqslant I^*_{\mathcal{N}}(g)$, because any feasible flow in $\mathcal{N}$ will be nonnegative. Therefore,

Table D.4
Uniform distribution spreads, objective cost coefficients and approximate optimal solution for problem U2

Arc i	Cost k_i	Spread s_i	Soln. μ_i	Arc i	Cost k_i	Spread s_i	Soln. μ_i	Arc i	Cost k_i	Spread s_i	Soln. μ_i
1	6.902	0.017	5.000	38	2.499	0.294	7.646	75	4.859	0.303	7.871
2	4.452	0.986	3.133	39	3.910	0.263	10.691	76	2.519	0.051	12.569
3	0.983	0.361	5.827	40	0.154	0.712	3.000	77	2.340	0.946	14.175
4	7.695	0.070	4.320	41	8.012	0.601	19.122	78	4.571	0.934	7.913
5	0.728	0.451	2.000	42	1.816	0.642	10.170	79	4.046	0.950	5.816
6	7.334	0.703	14.552	43	2.594	0.481	8.120	80	6.551	0.859	5.720
7	3.158	0.234	5.108	44	3.419	0.718	9.239	81	2.450	0.935	4.613
8	1.317	0.989	5.000	45	6.168	0.964	3.860	82	9.156	0.432	12.130
9	8.591	0.121	7.705	46	0.563	0.852	2.000	83	3.687	0.756	4.892
10	2.347	0.226	10.501	47	8.196	0.265	12.676	84	6.921	0.557	7.378
11	5.272	0.981	10.936	48	2.767	0.558	3.269	85	8.306	0.909	18.945
12	6.618	0.565	9.374	49	6.246	0.483	4.656	86	3.619	0.650	16.524
13	8.711	0.207	16.994	50	2.148	0.485	3.586	87	8.490	0.395	7.733
14	4.618	0.167	9.022	51	4.129	0.080	10.282	88	0.566	0.025	2.405
15	8.240	0.189	12.068	52	3.631	0.062	4.538	89	0.131	0.864	2.000
16	4.348	0.077	9.935	53	1.362	0.682	6.316	90	9.330	0.450	5.092
17	2.408	0.708	13.678	54	2.421	0.285	8.447	91	2.681	0.003	11.089
18	4.815	0.688	7.972	55	7.710	0.144	8.723	92	9.975	0.024	5.416
19	7.869	0.244	15.038	56	9.433	0.747	11.337	93	3.105	0.219	3.149
20	7.085	0.082	7.500	57	0.803	0.949	9.503	94	7.818	0.820	6.000
21	3.215	0.264	7.513	58	5.659	0.544	7.346	95	1.136	0.255	26.000
22	4.709	0.070	9.221	59	1.679	0.994	10.873	96	7.336	0.327	20.409
23	3.455	0.961	4.687	60	6.182	0.301	8.639	97	7.501	0.115	9.019
24	5.301	0.428	5.792	61	8.598	0.278	9.944	98	6.061	0.405	15.278
25	3.313	0.771	15.993	62	4.636	0.979	10.215	99	7.117	0.216	16.342
26	7.575	0.509	19.249	63	3.941	0.355	15.263	100	8.048	0.489	9.202
27	2.815	0.933	9.753	64	0.416	0.456	4.853	101	2.936	0.672	14.368
28	5.148	0.949	14.860	65	0.921	0.635	10.230	102	2.436	0.108	13.391
29	6.596	0.868	9.342	66	7.570	0.830	17.593	103	4.116	0.636	17.279
30	4.540	0.122	6.424	67	1.655	0.684	12.543	104	6.430	0.767	7.131
31	8.702	0.325	11.518	68	4.312	0.076	13.999	105	4.359	0.756	17.626
32	0.959	0.320	10.373	69	9.650	0.769	21.164	106	3.171	0.484	15.196
33	8.189	0.454	8.765	70	8.905	0.496	13.886	107	8.254	0.033	4.002
34	5.775	0.458	8.746	71	4.501	0.765	3.607	108	3.914	0.362	3.437
35	1.385	0.098	3.529	72	8.881	0.126	20.327	109	0.177	0.216	5.010
36	3.911	0.715	16.745	73	2.111	0.212	13.668	110	3.130	0.646	13.956
37	6.809	0.067	13.033	74	6.884	0.806	12.306				

if $g \geqslant 0$ then also $I^*_{\mathcal{N}}(g) \geqslant 0$. Also, we have $\operatorname{dom} I^*_{\mathcal{N}} = \mathbb{R}^n$ since the network is acyclic and the number of arcs is finite. Therefore $\phi = I^*_{\mathcal{N}} \circ g$ is finite for each z such that $g(z) \geqslant 0$.

The first result is a simple extension of [34, Theorem 5.1].

Lemma 1. *Let f be a convex function from $(-\infty, +\infty]^n$ to $(-\infty, +\infty]$ that is nondecreasing with respect to the partial order induced on $\mathbb{R}^n$ by $\mathbb{R}^n_+$, and such that if any component of x_i is $+\infty$ then $f(x) = +\infty$. Let $g_1, \ldots, g_n$ be proper convex functions on $\mathbb{R}$, and for $z \in \mathbb{R}^N$ define $g(z) = (g_1(z_1), \ldots, g_n(z_n))$ and let $\phi := f \circ g$. Then ϕ*

is convex. Further, if we suppose that the g_i are continuous and real-valued, that f is lower semicontinuous, and that there is a point $\hat{z} \in \mathbb{R}^n$ with $g(\hat{z}) \in \operatorname{dom} f$, then ϕ is closed and proper.

Proof. Let z^1 and z^2 be points of $\mathbb{R}^n$ and $\lambda \in (0,1)$. Then

$$\begin{aligned}\phi[(1-\lambda)z^1 + \lambda z^2] &= f[g((1-\lambda)z^1 + \lambda z^2)] \\ &\leqslant f[(1-\lambda)g(z^1) + \lambda g(z^2)] \\ &\leqslant (1-\lambda)\phi(z^1) + \lambda\phi(z^2),\end{aligned}$$

where the first inequality holds because for each i,

$$g_i[(1-\lambda)z_i^1 + \lambda z_i^2] \leqslant (1-\lambda)g_i(z_i^1) + \lambda g_i(z_i^2)$$

by the convexity of g_i, and because of the nondecreasing property of f. Noting that ϕ cannot take $-\infty$ because f cannot, we see that ϕ is convex by [34, Theorem 4.1].

With the additional assumptions we see that $\phi(\hat{z}) < +\infty$, so that ϕ is not everywhere $+\infty$. We already noted that it cannot take $-\infty$, so it is proper. It will therefore be closed if it is lower semicontinuous, and this property follows immediately from continuity of g and lower semicontinuity of f. □

A very simple calculation using the nondecreasing property shows that for any f satisfying the conditions of Lemma 1, we have $\operatorname{im} \partial f \subset \mathbb{R}^n_+$.

To apply Lemma 1 to the PERT problems of Section 4, note that for the case of uniform distributions, a random number $u \in [0,1]$ will generate an arc length $l_i = \mu_i + s_i(u - .5)$; therefore the function $g_i(\mu_i)$ is just μ_i plus a constant. This is certainly continuous and convex; it will be positive if we place appropriate lower bounds on the μ_i. For the case of triangular distributions, the parameter is x_i and the resulting density function has support in $[a_i x_i, b_i x_i]$, with mode $c_i x_i$. A random number $u \in [0,1]$ generates an arc length

$$l_i = \begin{cases} x_i[a_i + \{u(c_i - a_i)(b_i - a_i)\}^{1/2}] & \text{if } u \in [0, (c_i - a_i)/(b_i - a_i)] \\ x_i[b_i - \{(1-u)(b_i - c_i)(b_i - a_i)\}^{1/2}] & \text{if } u \in [(c_i - a_i)/(b_i - a_i), 1]. \end{cases}$$

In this case $g_i(x_i)$ is linear in x_i, and this function is also continuous and convex; it is positive whenever $x_i > 0$. We have already noted that $l^*_{\mathcal{N}}$ is closed, proper, convex, and componentwise nondecreasing.

Lemma 1 now shows that for any fixed $u \in [0,1]$ the composite function $\phi = l^*_{\mathcal{N}} \circ g$ will be closed proper convex as a function of the decision variables μ or x. We already noted that there is an open set G, of the form $\{\mu \mid \mu_i > \beta_i\}$ or $\{x \mid x_i > 0\}$ on which $g(z) \geqslant 0$, where here z may be either μ_i or x_i. Therefore ϕ is continuous and convex on G. The cost function we consider is the sum of $\phi(z)$ and a sum of terms of the form k_i/z_i. Averages of such functions, which we use in the minimization procedure, will clearly also be continuous and convex. Therefore Assumption 1 holds for the PERT problems that we consider.

We now proceed to determine the form of the subdifferential of a function ϕ of the kind defined in Lemma 1. We shall obtain an inclusion under very weak assumptions, and an equality when those assumptions are slightly strengthened.

Proposition 2. *Let f, g_i, and ϕ be as in Lemma 1. Then for each $z \in \mathbb{R}^N$ one has*

$$\partial\phi(z) \supset Q(z),$$

where $Q(z)$ is the set of all points of the form $(x_1^* z_1^*, \ldots, x_n^* z_n^*)$, where $x^* \in \partial f[g(z)]$ and for each i, $z_i^* \in \partial g_i(z_i)$.

Proof. Let $z^0 \in \mathbb{R}^N$. If $Q(z_0)$ is empty there is nothing to prove. Otherwise, select a point of $Q(z^0)$ and suppose that x^* and $z_1^*, \ldots, z_n^*$ generate it according to the definition of $Q(z^0)$. As $x^* \in \partial f[g(z^0)]$ one has for each $z \in \mathbb{R}^n$,

$$\begin{aligned}\phi(z) &= f[g_1(z_1), \ldots, g_n(z_n)] \\ &\geqslant f[g_1(z_1^0), \ldots, g_n(z_n^0)] + \sum_{i=1}^{n} x_i^*[g_i(z_i) - g_i(z_i^0)].\end{aligned}$$

But for each i, $x_i^* \geqslant 0$ as previously observed, and

$$g_i(z_i) - g_i(z_i^0) \geqslant \langle z_i^*, z_i - z_i^0 \rangle.$$

Hence

$$\phi(z) \geqslant \phi(z^0) + \langle (x_1^* z_1^*, \ldots, x_n^* z_n^*), z - z^0 \rangle,$$

which proves the assertion. □

Note that no $g_i(z_i^0)$ can be $+\infty$ in the above proof, since otherwise $f[g(z^0)]$ would be $+\infty$ and its subdifferential (which would be empty) could not contain x^*.

One might worry that the set $Q(z)$ considered in Proposition 2 could be nonconvex, or could be much smaller than $\partial\phi(z)$. We shall show that this is not so; in fact $Q(z)$ is always convex, and under some additional assumptions it is identical to the subdifferential. The following lemma shows that a set of the form of $Q(z)$ is convex.

Lemma 3. *Let X be a convex set in $\mathbb{R}_+^n$ and for $i = 1, \ldots, n$ let S_i be a convex set in $\mathbb{R}^{n_i}$. Then the set*

$$C = \{ (x_1 s_1, \ldots, x_n s_n) \mid x \in X, \quad s_i \in S_i \quad (i = 1, \ldots, n) \}$$

is convex in $\mathbb{R}^N$, where $N = \sum_{i=1}^n n_i$.

Proof. Let x and x' be elements of X and s_i and s_i' be elements of S_i for $i = 1, \ldots, n$. Let $\lambda \in (0, 1)$, and consider the point

$$z = (1 - \lambda)(x_1 s_1, \ldots, x_n s_n) + \lambda(x_1' s_1', \ldots, x_n' s_n').$$

Let $y = (1-\lambda)x + \lambda x'$; evidently $y \in X$. Now fix an index i between 1 and n. If $y_i = 0$ then both x_i and x_i' are zero, so $z_i = 0 = y_i s_i$. Otherwise, we have

$$z_i = y_i[(1-\lambda)x_i y_i^{-1} s_i + \lambda x_i' y_i^{-1} s_i'],$$

and the quantity in square brackets belongs to S_i by convexity. Therefore z is of the form $(y_1 w_1, \ldots, y_n w_n)$, where for each i $w_i \in S_i$, and so $z \in C$. □

The next proposition shows that under some additional regularity assumptions, $Q(z)$ is in fact the subdifferential of ϕ at z (note that the inclusion in one direction was established in Proposition 2; here we prove the opposite inclusion).

Proposition 4. *Let f, g_i, and ϕ be as in Lemma 2. Suppose $z = (z_1, \ldots, z_n)$ is a point of $\mathbb{R}^N$ such that for each i, $z_i \in \operatorname{int}\operatorname{dom} g_i$, and that $g(z) \in \operatorname{int}\operatorname{dom} f$. Then $\partial\phi(z) \subset Q(z)$, where $Q(z)$ is the set defined in Proposition 3.*

Proof. We first prove that the directional derivative $\phi'(z; \cdot)$ minorizes $I^*_{Q(z)}$. Choose some $w = (w_1, \ldots, w_n) \in \mathbb{R}^N$, and temporarily fix i. The condition $z_i \in \operatorname{int}\operatorname{dom} g_i$ implies that g_i is locally Lipschitzian at z_i, that $\partial g_i(z_i)$ is compact, and that $g_i'(z_i; \cdot) = I^*_{\partial g_i(z_i)}$. Therefore there is some $w_i^* \in \partial g_i(z_i)$ such that $g_i'(z_i; w_i) = \langle w_i^*, w_i\rangle$. So, for some function $r_i(t) = o(t)$, one has

$$g_i(z_i + tw_i) = g_i(z_i) + t\langle w_i^*, w_i\rangle + r_i(t),$$

and for small t the left side is finite. Accordingly, we can write

$$g(z + tw) = g(z) + t(\langle w_1^*, w_1\rangle, \ldots, \langle w_n^*, w_n\rangle) + r(t),$$

where $r(t) = (r_1(t), \ldots, r_n(t))$.

Now f is locally Lipschitzian at $g(z)$ because $g(z) \in \operatorname{int}\operatorname{dom} f$, so for some function $s(t) = o(t)$ and all small t we have

$$f[g(z + tw)] = f[g(z) + t(\langle w_1^*, w_1\rangle, \ldots, \langle w_n^*, w_n\rangle)] + s(t).$$

An argument similar to that just made for g shows that for some point $x^* \in \partial f[g(z)]$ and some function $r_f(t)$ one has

$$\begin{aligned} &f[g(z) + t(\langle w_1^*, w_1\rangle, \ldots, \langle w_n^*, w_n\rangle)] \\ &\quad = f[g(z)] + t\langle x^*, (\langle w_1^*, w_1\rangle, \ldots, \langle w_n^*, w_n\rangle)\rangle + r_f(t), \end{aligned}$$

and $r_f(t) = o(t)$. By combining these expressions we then obtain

$$\phi(z + tw) = \phi(z) + t\sum_{i=1}^{n}\langle x_i^* w_i^*, w_i\rangle + o(t).$$

This immediately implies that

$$\phi'(z; w) = \langle (x_1^* w_1^*, \ldots, x_n^* w_n^*), w\rangle,$$

and the expression in parentheses on the right side is an element of $Q(z)$. It follows that $\phi'(z;\cdot) \leqslant I^*_{Q(z)}$.

It follows from what we have just proved that $I^*_{\partial\phi(z)}$, which is the closure of $\phi'(z;\cdot)$, also minorizes $I^*_{Q(z)}$, and hence that $\partial\phi(z) \subset \mathrm{cl}\, Q(z)$. But since $\partial f[g(z)]$ and all of the sets $\partial g_i(z_i)$ are compact, $Q(z)$ is in fact closed, so $\partial\phi(z) \subset Q(z)$. □

Proposition 4 is included for completeness; the most important result for computation is Proposition 2. In the (maximization) linear programming problem to be solved for the critical path, the objective will be of the form $\sum_{i=1}^n g_i(z_i)x_i$, and the constraints will be network constraints (equations expressing the node-arc incidence conditions, with nonnegative variables x_i). The optimal objective value will be $f[g(z)]$ (that is, $f = I^*_{\mathcal{N}}$), and any optimal solution $(x_1,\ldots,x_n)$ will be a subgradient of f at $g(z)$. The subdifferentials $\partial g_i(z_i)$ may well actually be derivatives, depending upon the form in which the parameters z_i enter the probability distributions.

Appendix F. Constrained optimization vs. projection

This appendix addresses the issue of why we prefer to operate with a method that incorporates constraints into the subproblems solved at each iteration, rather than dealing with them after the fact by projection, as is often done in methods of stochastic approximation. An anonymous referee has suggested that our distaste for projection in stochastic approximation is unjustified, since each method (our proposal, and stochastic approximation with projection) has to solve a quadratic programming problem at each iteration. Our view, however, is based not on the fact that quadratic programming problems have to be solved, but on the very different outcomes that those problems can produce in the two methods. The point is that projection onto the feasible set can substantially retard the progress of stochastic approximation, and that this retardation can combine with the natural slowness of small gradient-step methods in a disastrous way. The following example in $\mathbb{R}^2$ illustrates this phenomenon.

Suppose we wish to minimize the function $f(x) := \langle v^*, x\rangle$ on the nonnegative orthant $\mathbb{R}^2_+$, where for fixed $\epsilon \in (0,1)$ we define $v^* = ([1-\epsilon^2]^{1/2}, \epsilon)$. Note that $\|v^*\| = 1$, and that the unique minimizer is the origin. We shall start from $x_1 := (0,1)$, using a gradient step followed by projection. To imitate the stepsize technique generally used in stochastic approximation, we take a step size of n^{-1} at the nth step. Accordingly, if we denote the Euclidean projection of $z \in \mathbb{R}^2$ on $\mathbb{R}^2_+$ by z_+, and if $T_n := \sum_{k=1}^n k^{-1}$, then we have

$$x_{n+1} = [x_n - n^{-1} f'(x)/\|f'(x)\|]_+ = [x_n - n^{-1}v^*]_+ = (0, 1-\epsilon T_n), \tag{F.1}$$

provided that $\epsilon T_n < 1$. As soon as $\epsilon T_n \geqslant 1$ the problem is solved. Thus, for this problem the iteration (F.1) is actually finitely convergent. We shall estimate the number of steps required, namely $N(\epsilon) := \inf\{n \mid \epsilon T_n \geqslant 1\}$.

The elementary inequality

$$\sum_{k=2}^{n} k^{-1} \leqslant \int_{1}^{n} x^{-1}\,\mathrm{d}x \leqslant \tfrac{1}{2} + \sum_{k=2}^{n} k^{-1} - (2n)^{-1},$$

obtained by bounding the graph of x^{-1} on $[1, n]$ by, respectively, piecewise constant and piecewise linear functions, shows that

$$T_n - \tfrac{1}{4} \leqslant \tfrac{3}{4} + \ln n \leqslant T_n + \tfrac{1}{4} - (2n)^{-1}.$$

It follows that to have $\epsilon T_n \geqslant 1$ one must have $\epsilon(1+\ln n) \geqslant 1$: that is, $n \geqslant \exp(\epsilon^{-1} - 1)$. If for concreteness we take $\epsilon = 0.01$, then we find that $N(\epsilon) \geqslant e^{99}$, a number that is greater than 9.8×10^{42}. Therefore our finitely convergent algorithm (F.1) requires nearly 10^{43} steps, each consisting of a gradient step followed by a projection. It should be clear that such a computation is impracticable even with the fastest equipment available.

If instead we apply a version of the bundle algorithm to this problem, we shall obtain x_{n+1} by the iteration

$$x_{n+1} = \arg\min\{\langle v^*, x\rangle + (2t)^{-1}\|x - x_n\|^2 \mid x \in \mathbb{R}^2_+\},$$

since the linearity of f means that our piecewise linear approximating function will be f itself. There will then be no null steps, so that if x_n is of the form $(0, w_n)$ with $w_n \geqslant 0$ then we have $x_{n+1} = (0, [w_n - t\epsilon]_+)$. Hence if we start at $x_1 = (0, 1)$ with $\epsilon = 0.01$, then if $t \geqslant 100$ we solve the problem in one step, whereas if $t < 100$ then we need $\lceil 100t^{-1} \rceil$ steps. In fact, an adaptive technique such as is used in the methods of [22] or [36] will quickly adjust t upward when it finds that no null steps are taken. Therefore, not many steps will be required in any case.

This example used a very simple function to make the exposition easy. However, it should be clear that the underlying situation involved here (a gradient nearly, but not quite, normal to the boundary of the feasible set at an iterate x_n) can be expected to hold in general when one is near a constrained optimizer. Therefore the phenomenon is by no means special, but rather illustrates a fundamental disadvantage of combining small gradient steps with projection.

References

[1] H. Attouch, *Variational Convergence for Functions and Operators* (Pitman, Boston, MA, 1984).

[2] S.P. Bradley, A.C. Hax, and T.L. Magnanti, *Applied Mathematical Programming* (Addison-Wesley, Reading, MA, 1977).

[3] R. Correa and C. Lemaréchal, "Convergence of some algorithms for convex minimization," *Mathematical Programming* **62** (1993) 261–275.

[4] Y.M. Ermoliev and A.A. Gaivoronski, "Stochastic quasigradient methods for optimization of discrete event systems," *Annals of Operations Research* **39** (1992) 1–39.

[5] B.-R. Fu, "Modeling and analysis of discrete tandem production lines using continuous flow models," Ph.D. Dissertation, Department of Industrial Engineering, University of Wisconsin–Madison (Madison, WI, 1996).

[6] A.A. Gaivoronski, "Interactive program SQG-PC for solving stochastic programming problems on IBM PC/XT/AT compatibles, user guide," Working Paper WP-88-11, International Institute for Applied Systems Analysis (Laxenburg, Austria, 1988).

[7] A.A. Gaivoronski, E. Messina, and A. Sciomachen, "A statistical generalized programming algorithm for stochastic optimization problems," *Annals of Operations Research* **58** (1995) 297–321.
[8] A.A. Gaivoronski and L. Nazareth, "Combining generalized programming and sampling techniques for stochastic programs with recourse," in: G. Dantzig and P. Glynn, eds., *Resource Planning under Uncertainty for Electric Power Systems* (Stanford University, 1989).
[9] S.B. Gershwin and I.C. Schick, "Continuous model of an unreliable two-stage material flow system with a finite interstage buffer," Technical Report LIDS-R-1039, Massachusetts Institute of Technology (Cambridge, MA, 1980).
[10] S.B. Gershwin and I.C. Schick, "Modeling and analysis of three-stage transfer lines with unreliable machines and finite buffers," *Operations Research* **31** (1983) 354–380.
[11] P. Glasserman, *Gradient Estimation Via Perturbation Analysis* (Kluwer Academic Publishers, Boston, MA, 1991).
[12] P.W. Glynn, "Likelihood ratio gradient estimation: an overview," in: *Proceedings of the 1987 Winter Simulation Conference* (IEEE, Piscataway, NJ, 1987), pp. 366–374.
[13] P.W. Glynn, "Optimization of stochastic systems via simulation," in: *Proceedings of the 1989 Winter Simulation Conference* (IEEE, Piscataway, NJ, 1989), pp. 90–105.
[14] K. Healy and L.W. Schruben, "Retrospective simulation response optimization," *Proceedings of the 1991 Winter Simulation Conference* (1991) 901–907.
[15] M. Held, P. Wolfe, and H.P. Crowder, "Validation of subgradient optimization," *Mathematical Programming* **6** (1974) 62–88.
[16] J.L. Higle and S. Sen, "Statistical verification of optimality conditions for stochastic programs with recourse," *Annals of Operations Research* **30** (1991) pp. 215–240.
[17] J.L. Higle and S. Sen, "Stochastic decomposition: an algorithm for two-stage linear programs with recourse," *Mathematics of Operations Research* **16** (1991) pp. 650–669.
[18] Y.C. Ho, "Performance evaluation and perturbation analysis of discrete event dynamic systems," *IEEE Transactions on Automatic Control* **32** (1987) 563–572.
[19] Y.C. Ho and X.R. Cao, *Perturbation Analysis of Discrete Event Dynamic Systems* (Kluwer Academic Publishers, Boston, MA, 1991).
[20] J.-Q. Hu, "Convexity of sample path performance and strong consistency of infinitesimal perturbation analysis estimates," *IEEE Transactions on Automatic Control* **37** (1992) 258–262.
[21] P. Kall, "Approximation to optimization problems: an elementary review," *Mathematics of Operations Research* **11** (1986) 9–18.
[22] K.C. Kiwiel, "Proximity control in bundle methods for convex nondifferentiable minimization," *Mathematical Programming* **46** (1990) 105–122.
[23] P. L'Ecuyer, "Efficient and portable combined random number generators," *Communications of the ACM* **31** (1988) 742–774.
[24] P. L'Ecuyer and P.W. Glynn, "Stochastic optimization by simulation: convergence proofs for the GI/G/1 queue in steady state," *Management Science* **40** (1994) 1562–1578.
[25] P. L'Ecuyer, N. Giroux and P.W. Glynn, "Stochastic optimization by simulation: numerical experiments with the M/M/1 queue in steady state," *Management Science* **40** (1994) 1245–1261.
[26] D.G. Malcolm, J.H. Roseboom, C.E. Clark and W. Fazar, "Application of a technique for research and development program evaluation," *Operations Research* **7** (1959) 646–669.
[27] M.S. Meketon, "A tutorial on optimization in simulations," Unpublished tutorial presented at the 1983 Winter Simulation Conference.
[28] A. Mongalo and J. Lee, "A comparative study of methods for probabilistic project scheduling," *Computers in Industrial Engineering* **19** (1990) 505–509.
[29] G.L. Nemhauser and L.A. Wolsey, *Integer and Combinatorial Optimization* (Wiley, New York, 1988).
[30] E. Nummelin, "Regeneration in tandem queues," *Advances in Applied Probability* **13** (1981) 221–230.
[31] E.L. Plambeck, B.-R. Fu, S.M. Robinson and R. Suri, "Throughput optimization in tandem production lines via nonsmooth programming," in: J. Schoen, ed., *Proceedings of the 1993 Summer Computer Simulation Conference* (Society for Computer Simulation, San Diego, CA, 1993) pp. 70–75.
[32] H. Robbins and S. Monro, "A stochastic approximation method," *Annals of Mathematical Statistics* **22** (1951) 400–407.
[33] S.M. Robinson, "Analysis of sample-path optimization," accepted by *Mathematics of Operations Research*.
[34] R.T. Rockafellar, *Convex Analysis* (Princeton University Press, Princeton, NJ, 1970).

[35] R.Y. Rubinstein and A. Shapiro, *Discrete Event Systems: Sensitivity Analysis and Stochastic Optimization by the Score Function Method* (Wiley, Chichester, 1993).

[36] H. Schramm and J. Zowe, "A version of the bundle idea for minimizing a nonsmooth function: conceptual idea, convergence analysis, numerical results," Technical Report 209, Mathematisches Institut, Universität Bayreuth (Bayreuth, Germany, 1990).

[37] J.G. Shanthikumar and D.D. Yao, "Second-order stochastic properties in queueing systems," *Proceedings of the IEEE* **77** (1989) 162–170.

[38] J.G. Shanthikumar and D.D. Yao, "Strong stochastic convexity: closure properties and applications," *Journal of Applied Probability* **28** (1991) 131–145.

[39] A. Shapiro and Y. Wardi, "Nondifferentiability of the steady-state function in discrete event dynamic systems," *IEEE Transactions on Automatic Control* **39** (1994) 1707-1711.

[40] R. Suri, "Perturbation analysis: the state of the art and research issues explained via the GI/G/1 queue," *Proceedings of the IEEE* **77** (1989) 114–137.

[41] R. Suri and B.-R. Fu, "On using continuous flow lines for performance estimation of discrete production lines," in: *Proceedings of the 1991 Winter Simulation Conference* (IEEE, Piscataway, NJ, 1991), pp. 968–977.

[42] R. Suri and B.-R. Fu, "On using continuous flow lines to model discrete production lines," *Discrete Event Dynamic Systems* **4** (1994) 129–169.

[43] R. Suri and B.-R. Fu, "Using continuous flow models to enable rapid analysis and optimization of discrete production lines – a progress report," in: *Proceedings of the 19th Annual NSF Grantees Conference on Design and Manufacturing Systems Research* (Charlotte, NC, 1993), pp. 1229–1238.

[44] R. Suri and Y.T. Leung, "Single run optimization of discrete event simulations – an empirical study using the M/M/1 queue," *IIE Transactions* **34** (1989) 35–49.

[45] R. Suri and M. Zazanis, "Perturbation analysis gives strongly consistent sensitivity estimates for the M/G/1 queue," *Management Science* **34** (1988) 39–64.

[46] R.M. Van Slyke, "Monte Carlo Methods and the PERT problem," *Operations Research* **11** (1963) 839–860.

[47] R.M. Van Slyke and R.J.-B. Wets, "L-shaped linear programs with applications to optimal control and stochastic programming," *SIAM Journal on Applied Mathematics* **17** (1969) 638–663.

[48] S.W. Wallace, "Bounding the expected time-cost curve for a stochastic PERT network from below," *Operations Research Letters* **8** (1989) 89–94.

[49] K.C. Wei, Q.Q. Tsao and N.C. Otto, "Determining buffer size requirements using stochastic approximation methods," Technical Report SR-89-73, Manufacturing Systems Department, Ford Motor Company (Dearborn, MI, 1989).

[50] R.D. Wollmer, "Critical path planning under uncertainty," *Mathematical Programming Study* **25** (1985) 164–171.

[51] J. Zowe, "The BT-algorithm for minimizing a nonsmooth functional subject to linear constraints," in: F.H. Clarke et al., eds., *Nonsmooth Optimization and Related Topics* (Plenum Press, New York, 1989).