Noname manuscript No.
(will be inserted by the editor)

Parallel stochastic optimization based on descent algorithms

Olivier Bilenne

Abstract This study addresses the stochastic optimization of a function unknown in closed form which can only be estimated based on measurements or simulations. We consider parallel implementations of a class of stochastic optimization methods that consist of the iterative application of a descent algorithm to a sequence of approximation functions converging in some sense to the function of interest. After discussing classical parallel modes of implementations (Jacobi, Gauss-Seidel, random, Gauss-Southwell), we devise effort-saving implementation modes where the pace of application of the considered descent algorithm along individual coordinates is coordinated with the evolution of the estimated accuracy of the convergent function sequence. It is shown that this approach can be regarded as a Gauss-Southwell implementation of the initial method in an augmented space. As an example of application we study the distributed optimization of stochastic networks using a scaled gradient projection algorithm with approximate line search, for which asymptotic properties are derived.

1 Introduction

We are concerned with the parallel minimization of a real-valued function $g : \mathbb{R}^m \mapsto (-\infty, \infty]$ unknown in closed form and which can only be estimated by means of inexact measurements or Monte-Carlo simulations. The objective of the study is to derive parallel implementations of a stochastic optimization method suggested in [22,19] and based on the approximation of the unknown function g—called the *true function*—by a sequence of function models which converges toward g in some sense, combined with the iterative application to

Olivier Bilenne
Control Systems Group, Technical University of Berlin, Germany
Tel.: +49 (0)30 314-78692
Fax: +49 (0)30 314-21137
E-mail: bilenne@control.tu-berlin.de

the model sequence of an effective descent algorithm taken from the nonlinear optimization theory. Given any arbitrary descent algorithm $\mathcal{M}$, we question how the algorithm can be parallelized in the stochastic optimization context. Section 2 surveys the traditional modes of implementation of the parallel and distributed nonlinear optimization framework, such as Jacobi, Gauss-Seidel, or random implementations [5]. Implementability issues more specific to the considered stochastic optimization setting are discussed in Section 3. The purpose of our developments is illustrated in Section 4 with numerical results based on cyclic gradient projections for a stochastic network optimization problem. Asymptotic considerations are included in the Appendix. In the rest of this introduction we specify the requirements for parallel optimization (Section 1.1) and recall the basics of the stochastic optimization methods based on descent algorithms (1.2).

Notation — In this paper vectors are column vectors and denoted by $x = (x_1, ..., x_n)$, where $x_1, ..., x_n$ are the coordinates of x. Subscripts are reserved for vector coordinates. The transpose of x is denoted by x'. For any real space $\mathbb{R}^p$, $\mathrm{Lsc}(\mathbb{R}^p)$ denotes the class of functions $\mathbb{R}^p \mapsto (-\infty, \infty]$ proper and lower semicontinuous.

1.1 Requirements for parallel analysis

In this study, the notion of parallel analysis for the stochastic minimization of a function $g \in \mathrm{Lsc}(\mathbb{R}^m)$ is understood as the ability to derive descent directions along individual coordinates or blocks or coordinates. We assume that a set of coordinate directions $N = \{1, ..., n\}$ is implicitly defined by g, and denote by $m_1, ..., m_n$ the respective dimensions of the coordinates, where $\sum_{i=1}^{n} m_i = m$.

We symbolize the optimization of any $f \in \mathrm{Lsc}(\mathbb{R}^m)$ at a point $y \in \mathrm{dom}(f)$ along a particular coordinate direction $i \in N$ by the function $f_{i:y} \in \mathrm{Lsc}(\mathbb{R}^{m_i})$ obtained from $f(y)$ by fixing the other coordinates, i.e.

$$f_{i:y}(z) = f(y_1, ..., y_{i-1}, z, y_{i+1}, ..., y_n), \qquad \forall z \in \mathbb{R}^{m_i}. \tag{2}$$

For the minimization of f over any set $X \subseteq \mathrm{dom}(f)$, we use iterative descent algorithms, the executions of which are terminated when the produced sequence of points gets close enough to a specific subset of X, denoted by $S^f(X)$, where descent can no longer be guaranteed. The set $S^f(X)$ ordinarily consists of fixed points, stationary points in the case of gradient methods, or minima if f is convex.

Since, typically, the convergence of function sequences is effective on bounded vector sets, we constrain the minimization of g within a (fixed) closed set

[0] An example of norm topology suitable for the case when g is continuously differentiable on its domain, i.e. $F(X) \subset \mathbb{C}^1(X)$ on any $X \subseteq \mathrm{dom}(g)$, is to consider the norm [19,17]

$$f \in \bar{F}(X) \mapsto \|f\|_{1,X} = \sup_{x \in X} |f(x)| + \sup_{x \in X} \|\nabla f(x)\|. \tag{1}$$

$\bar{Y} \subseteq \mathrm{dom}(g)$ enjoying a Cartesian product structure as required by parallel analysis, i.e. $\bar{Y} = \prod_{i=1}^{n} \bar{Y}_i$, where $\bar{Y} \subseteq \mathrm{dom}(g)$ and $\bar{Y}_i \subseteq \mathbb{R}^{m_i}$ for $i \in N$. Lastly, we let $F(X) \subset \mathrm{Lsc}(\mathbb{R}^p)$ define, for any $X \subseteq \mathbb{R}^p$, a functional class of interest equipped[1] with a norm $\|\cdot\|_X$. With all the above in mind, we characterize the function g as follows.

Assumption 1 (Parallel analysis over $\bar{Y}$) *Let $g \in \bar{F}(\bar{Y})$, where $\bar{F}(\bar{Y})$ is a subclass of $F(\bar{Y})$ such that, for any $f \in \bar{F}(\bar{Y})$ and $y \in \bar{Y}$, one has $f_{i:y} \in F(\bar{Y}_i)$ $\forall i \in N$ and*

$$y \in S^f(\bar{Y}) \Leftrightarrow y_i \in S^{f_{i:y}}(\bar{Y}_i) \ \forall i \in N. \tag{3}$$

A consequence of (3) is that the minimization over $\bar{Y}$ of any function of $\bar{F}(\bar{Y})$ can be done in parallel along each coordinate.

Assumption 1 covers the classes of functions usually assumed to allow for parallel analysis, including, for instance, the accepted form [20]

$$g(y) = f(y) + h(y), \tag{4}$$

where $h \in \mathrm{Lsc}(\mathbb{R}^m)$ is such that $h(y)$ is additively separable with respect to n coordinates $y_1, ..., y_n$, i.e. $h(y) = \sum_{i=1}^{n} h_i(y_i)$, with $h_i \in \mathrm{Lsc}(\mathbb{R}^{m_i})$ convex for $i = 1, ..., n$, and $f : \mathbb{R}^m \mapsto \mathbb{R}$ is continuously differentiable over $\mathrm{dom}(h)$ [4]. Indeed, suppose that g is given by (4) and let $d = (d_1, ..., d_n)$ be a descent direction for g at a point $y \in \mathrm{dom}(g)$, i.e.

$$a\nabla f(y)'d + h(y + ad) - h(y) + o(a) < 0, \tag{5}$$

where $\nabla f = (\nabla_1 f, ..., \nabla_n f)$. By additive separability of h, (5) rewrites as

$$\sum\nolimits_{i=1}^{n} [a\nabla_i f(y)'d_i + h_i(y_i + ad_i) - h_i(y_i)] + o(a) < 0. \tag{6}$$

Hence a global descent direction exists at y iff one can find $i \in N$ such that a descent direction exists (in the subspace $\mathbb{R}^{m_i}$) for $g(y_1, ..., y_{i-1}, \cdot, y_{i+1}, ..., y_n)$ at y_i—a property summarized by (3). The model (4) is met for instance in bound-constrained optimization, where h is of the type $h(y) = 0$ if $l \leq y \leq u$ $(l, u \in \mathbb{R}^m)$ and $h(y) = +\infty$ otherwise. The dual function of the separable constrained optimization problem studied in Section 4 falls into this category.

1.2 Stochastic optimization based on descent algorithms

We consider an approach to minimizing g on the set $\bar{Y}$ where the stochastic optimization algorithms take the recursive form

$$y^{k+1} \in \mathcal{M}(g^k, y^k), \qquad k = 0, 1, 2, ... \ , \tag{7}$$

where $\{g^k\}$ is a sequence of functions in $\bar{F}(\bar{Y})$ which is expected to converge to g in the norm topology of $\bar{F}(\bar{Y})$, and the point-to-set mapping $\mathcal{M} : \bar{F}(\bar{Y}) \times \bar{Y} \mapsto 2^{\bar{Y}}$ is closed on $\bar{F}(\bar{Y}) \times \bar{Y}$ and a descent algorithm for the minimization of g on $\bar{Y}$.

Let us recall the notions of closed mappings and descent algorithms [19] for the generic class of functions $F(\cdot)$ introduced in Assumption 1.

Definition 1 (Closed mapping) Let X be a closed vector set. A point-to-set mapping $\mathcal{M} : F(X) \times X \to 2^X$ is said to be closed at $(f, y) \in F(X) \times X$ if for any sequence $\{(f^k, y^k)\}$ in $F(X) \times X$ such that $(f^k, y^k) \to (f, y)$ and any vector sequence (z^k) such that $z^k \to z$ and $z^k \in \mathcal{M}(f^k, y^k)$ for all k, we have $z \in \mathcal{M}(f, y)$. For a given $f \in F(X)$, $\mathcal{M}$ is said to be closed at f if it closed at (f, y) for every $y \in X$. The mapping $\mathcal{M}$ is said to be *closed* on $F(X) \times X$ if it is closed at each point of $F(X) \times X$.

Definition 2 (Descent algorithm) Consider a closed vector set X and a mapping $\mathcal{M} : F(X) \times X \to 2^X$. Given a function $f \in F(X)$ and a set $S \subset X$, we say that a continuous, real-valued function $\Delta^f : X \to \mathbb{R}$ is a *descent function* for $\mathcal{M}$ with respect to f and S if:

(i) If $y \in X \setminus S$ and $z \in \mathcal{M}(f, y)$, then $\Delta^f(z) < \Delta^f(y)$.
(ii) If $y \in S$ and $z \in \mathcal{M}(f, y)$, then $\Delta^f(z) \leq \Delta^f(y)$.

The mapping $\mathcal{M}$ is called a *descent algorithm* is it possesses a descent function.

If $\bar{Y}$ is compact, $S^g(\bar{Y})$ is nonempty, and the mapping $\mathcal{M}$ is closed at g and a descent algorithm with respect to g and S^g, then sequences generated by (7) prove to converge to $S^g(\bar{Y})$ [19, Theorem 2.1]. An example of a descent algorithm for continuously differentiable functions is given in Appendix A.

Sample-average approximations of expectation functions. In many problems, the function g can be expressed as the expectation of another function $\hat{g} : \mathbb{R}^m \times \Omega \to (-\infty, \infty]$ which is known and varies randomly with a parameter ω defined on a probability space $(\Omega, \mathcal{F}, P)$, i.e.

$$g(y) = \mathrm{E}[\hat{g}(y, \omega)], \qquad \forall y \in \mathbb{R}^m, \tag{8}$$

where $\mathrm{E}[\cdot] \equiv \int_\Omega \cdot \, P(d\omega)$ denotes the expectation with respect to the random parameter ω , and P may be unknown. It is usually assumed for such problems that sequences of random realizations of ω can be observed and used to estimate g, or that random samples of growing sizes can be generated for ω by Monte-Carlo simulations. In the context of two-stage stochastic programming, for instance, the quantity $\hat{g}(y, \omega)$ is given by the optimal value of the second-stage problem [17].

Suppose that (8) holds, and that a sequence $\{\omega^{k,l}\}_{l=0}^{q(k)-1}$ of independent realizations of ω is available at each step k, with $q(k) \to \infty$ as $k \to \infty$. A natural choice for the sequence (g^k) is given by the sample average estimator

$$g^k(y) = \frac{1}{q(k)} \sum\nolimits_{l=0}^{q(k)-1} \hat{g}(y, \omega^{k,l}), \qquad k = 0, 1, 2, \ldots , \tag{9}$$

which is known to converge almost surely and uniformly towards g on any compact set C where $\hat{g}(\cdot, \omega)$ is continuous for P-almost every $\omega \in \Omega$ and $\mathrm{E}[\sup_{y \in C} |\hat{g}(y, \omega)|] < \infty$ [15]. It follows from the central limit theorem that (9) is asymptotically normal at every $y \in \mathrm{dom}(g)$, i.e.

$$q(k)^{-\frac{1}{2}} [g^k(y) - g(y))] \xrightarrow{d} \nu(y), \tag{10}$$

where $\xrightarrow{d}$ denotes convergence in distribution and $\nu(y)$ is a centered normal random variable with variance $\sigma^2(y) = \mathrm{Var}\,[\hat{g}(y,\omega)]$. In particular, when (7) is used with (9) and the optimal value algorithm $\mathcal{M}(f,y) \equiv S^f(\bar{Y})$, it reduces to the sample average approximation estimator (SAA), for which asymptotic properties are known [17].

Notice that the computational complexity of (9) grows with k and may become prohibitive for the stochastic optimization algorithm (7), especially when $\hat{g}$ is unavailable in closed form. Viable implementations of (9) can nonetheless be designed by controlling the generated samples in the case of simulation-based optimization (see Section 3), or by using variance reduction techniques [16, 18,17]. The complexity of (7) may also increase quickly with the dimension of $\mathbb{R}^m$ and the cardinality of Ω, which is sometimes expected to expand exponentially with the size of the problem [9]. This dimensionality issue can be addressed, under certain conditions, by parallel computing.

2 Parallel stochastic optimization

Let the operator $\mathcal{M}$ symbolize, for any closed vector set X, a point-to-set mapping $F(X) \times X \mapsto 2^X$ which is both closed on $F(X) \times X$ and a descent algorithm with respect to any $f \in F(X)$ and its corresponding set $S^f(X)$, and let Δ^f denote the associated descent function. We are interested in optimization algorithms based on the application of $\mathcal{M}$ along coordinates or blocks of coordinates and consider, in the rest of this section, various parallel modes of implementation for $\mathcal{M}$. In Sections 2.1 and 2.2 it is assumed that the optimization process is operated in parallel by n computers (sometimes called nodes), each of them assigned to a particular coordinate $i \in N$, and which collaborate in minimizing g along their respective directions.

2.1 Synchronous implementations

In the synchronous mode of implementation, the mapping $\mathcal{M}$ is applied simultaneously—as in the Jacobi method—by all the computers, i.e.

$$y_i^{k+1} \in \mathcal{M}(g_{i:y^k}, y_i^k), \qquad \forall i \in N,\ k = 0, 1, 2, \ldots\ . \tag{11}$$

In the general case (11) is not a descent algorithm because simultaneous descent along $g^k_{i:y^k}$ for every $i \in N$ in accordance with (11) does not imply descent along g^k. This is a well-known issue of the Jacobi methods which is typically addressed by scaling the coordinate descents with step-sizes so as to guarantee descent at the global level. We refer to e.g. [5] for related results.

2.2 Cyclic implementations

Consider, for $i \in N$, the mapping $\mathcal{M}_i : \bar{F}(\bar{Y}) \times \bar{Y} \mapsto 2^{\bar{Y}}$ defined by

$$\mathcal{M}_i(f,y) = \{(y_1, \ldots, y_{i-1}, z, y_{i+1}, \ldots, y_n) \,|\, z \in \mathcal{M}(f_{i:y}, y_i)\} \tag{12}$$

for all $(f, y) \in \bar{F}(\bar{Y}) \times \bar{Y}$. By assumption on $\bar{Y}$, it is straightforward to show that $\mathcal{M}_1, ..., \mathcal{M}_n$ are closed on $\bar{F}(\bar{Y}) \times \bar{Y}$ if $\mathcal{M}$ is closed. By applying the mappings $\mathcal{M}_1, ..., \mathcal{M}_n$ sequentially as in the *Gauss-Seidel* method, we can devise a *cyclic* implementation of $\mathcal{M}$,

$$y^{k+1} \in \mathcal{C}(g^k, y^k), \qquad k = 0, 1, 2, ... \,, \tag{13}$$

where we define

$$\mathcal{C} = \mathcal{M}_n \circ \mathcal{M}_{n-1} \circ \cdots \circ \mathcal{M}_1 \tag{14}$$

and $\circ$ denotes the composition operator[2]. The mapping $\mathcal{C}$ inherits the closedness and descent properties of $\mathcal{M}$ under the following condition.

Condition 1 (Sequential analysis of $(\mathcal{M}, \bar{F})$) *If $f \in \bar{F}(\bar{Y})$, $y \in \bar{Y} \backslash S^f(\bar{Y})$, and if for some $i, j \in N$ we have $y_i \in S^{f_{i:y}}(\bar{Y}_i)$ and $y_j \notin S^{f_{j:y}}(\bar{Y}_j)$, then $y_j \notin S^{f_{j:(y_1,...,y_{i-1},z,y_{i+1},...,y_n)}}(\bar{Y}_j)$ for every $z \in \mathcal{M}(g_{i:y}, y_i)$.*

Note that Condition 1 is usually satisfied in implementable settings, in particular when $\mathcal{M}$ is a descent algorithm and $S^f(\bar{Y})$ is the set of stationary points of f (e.g. gradient projection methods for (4), or extensions [21]), in which case we have $\mathcal{M}(f_{i:y}, y_i) = \{y_i\}$ if $y_i \in S^{f_{i:y}}(\bar{Y}_i)$.

Result 1 *When Condition 1 holds, the mapping $\mathcal{C}$ is closed on $\bar{F}(\bar{Y}) \times \bar{Y}$ and a descent algorithm with respect to g and $S^g(\bar{Y})$ with descent function Δ^g.*

Proof We first show by induction that $\mathcal{C}$ is closed. We already know that $\mathcal{M}_1$ is closed. Now, for $i = 2, ...n$, assume that $\mathcal{N}_i = \mathcal{M}_{i-1} \circ \cdots \circ \mathcal{M}_1$ is closed, and let $\{(f^k, y^k)\}$ be a sequence in $\bar{F}(\bar{Y}) \times \bar{Y}$ with $(f^k, y^k) \to (f, y)$ and $\{z^k\}$ a sequence in $\bar{Y}$ such that $z^k \in (\mathcal{M}_i \circ \mathcal{N}_i)(f^k, y^k)$ for all k and $z^k \to z$. Consider the sequence $\{\hat{y}^k\}$ such that $\hat{y}^k = (z_1^k, ..., z_{i-1}^k, y_i^k, ..., y_n^k)$ for all k. By assumption on $\mathcal{N}_i$ and $\mathcal{M}_i$, we have $\hat{y}^k \in \mathcal{N}_i(f, y^k)$ and $z^k \in \mathcal{M}_i(f, \hat{y}^k)$ for all k, and $\hat{y}^k \to \hat{y} = (z_1, ..., z_{i-1}, y_i, ..., y_n) \in \bar{Y}$. Since $\mathcal{N}_i$ and $\mathcal{M}_i$ are closed, we successively find $\hat{y} \in \mathcal{N}_i(f, y)$, then $z \in \mathcal{M}_i(f, \hat{y}) \subset (\mathcal{M}_i \circ \mathcal{N}_i)(f, y)$. Hence $\mathcal{M}_i \circ \mathcal{N}_i$ is closed and it follows by induction that $\mathcal{C}$ is closed.

Now we show that Δ^g meets the conditions of Definition 2 for $\mathcal{C}$ with respect to g. For every $i \in N$ we have, by definition of $\mathcal{M}_i$,

$$\Delta^g(z) = \Delta^{g_{i:y}}(z_i) \leq \Delta^{g_{i:y}}(y_i) = \Delta^g(y), \qquad \forall y \in \bar{Y},\ z \in \mathcal{M}_i(g, y). \tag{15}$$

By induction on i, we find $\Delta^g(z) \leq \Delta^g(y)$ for any $y \in \bar{Y}$, $z \in \mathcal{C}(g, y)$. It remains to show that, for any vector $y \in \bar{Y} \setminus S^g(\bar{Y})$, $\mathcal{C}(g, y)$ produces a strict descent along Δ^g. It follows from (3) that one can find at least one coordinate direction $i \in N$ such that $y_i \notin S^{g_{i:y}}(\bar{Y}_i)$. Among such directions, denote that of smallest index by j. Let $z \in \mathcal{C}(g, y)$. There exists a sequence $\hat{y}^0, ..., \hat{y}^n$ such that $\hat{y}^0 = y$, $\hat{y}^n = z$, and $\hat{y}^t \in \mathcal{M}_t(g, \hat{y}^{t-1})$ for $t = 1, ..., n$. By Condition 1, we have $\hat{y}_j^{j-1} \notin S^{g_{j:\hat{y}^{j-1}}}(\bar{Y}_j)$. Since $\hat{y}_j^j \in \mathcal{M}(g_{i:\hat{y}^{j-1}}, \hat{y}_j^{j-1})$, we find $\Delta^{g_{i:\hat{y}^{j-1}}}(\hat{y}_j^j) < \Delta^{g_{i:\hat{y}^{j-1}}}(\hat{y}_j^{j-1})$, i.e. $\Delta^g(\hat{y}^j) < \Delta^g(\hat{y}^{j-1})$. Using (15), we also have $\Delta^g(\hat{y}^t) \leq \Delta^g(\hat{y}^{t-1})$ for $t \neq j$. All in all, we find $\Delta^g(z) < \Delta^g(y)$, which completes the proof. □

[2] Given two mappings $\mathcal{M}, \mathcal{N} : F(X) \times X \mapsto 2^X$, the composition of $\mathcal{M}$ and $\mathcal{N}$ is defined by $\mathcal{N} \circ \mathcal{M} : (f, x) \in F(X) \times X \mapsto (\mathcal{N} \circ \mathcal{M})(f, x) = \{z \in \mathcal{N}(f, y) \,|\, y \in \mathcal{M}(f, x)\} \in 2^X$.

Remark 1 (Random implementations) The order in which $\mathcal{M}_1, ..., \mathcal{M}_n$ are applied in (14) is arbitrary, and $\mathcal{C}$ remains a descent algorithm for the composition of any arrangement (with possible repetitions) of $\mathcal{M}_1, ..., \mathcal{M}_n$ provided that each $\mathcal{M}_i$ appears at least once. By considering the union of the mappings generated by all these possibilities, we obtain the mapping of a descent algorithm where the order of the directional descents $\mathcal{M}_i$ may change randomly at each step k, hence a random parallel implementation of $\mathcal{M}$. Convergence to $S^g(\bar{Y})$ is however not guaranteed for the parallel implementations where every $\mathcal{M}_i$ would not be used at each step.

2.3 Implementations based on block-coordinate selection

In this section we consider algorithms where $\mathcal{M}$ is applied at each step to a block of one or several coordinates. The successive coordinate blocks are chosen according to a specific coordinate selection policy ensuring convergence (e.g. of the *Gauss-Southwell* type [21]). Since in the sequel the coordinates directions are treated by blocks, we consider the family $2^N = \{I \,|\, I \subset N\}$ of all the subsets of N and the family $\mathcal{N} = 2^N \setminus \{\emptyset\}$ of the nonempty blocks of coordinate directions, denoted herein by capital letters. The subscript notation previously used for coordinates can be extended without ambiguity to block coordinates. For any $I \in \mathcal{N}$, we define $\bar{Y}_I = \prod_{i \in I} \bar{Y}_i$ and, for $y \in \bar{Y}$, denote by $y_I \in \bar{Y}_I$ the composite vector of the coordinates (ordered by coordinate index) of y along I. Extending (2) and (12) to coordinates blocks, we define, for any $f \in \bar{F}(\bar{Y})$ and $I \in \mathcal{N}$,

$$f_{I:y}(z) = f(x) \text{ where } x_I = z,\ x_{N\setminus I} = y_{N\setminus I}, \qquad \forall y \in \bar{Y}, z \in \bar{Y}_I, \tag{16}$$

and the mapping $\mathcal{M}_I : \bar{F}(\bar{Y}) \times \bar{Y} \mapsto 2^{\bar{Y}}$ given by

$$\mathcal{M}_I(f, y) = \{z \,|\, z_{N\setminus I} = y_{N\setminus I}, z_I \in \mathcal{M}(f_{I:y}, y_I)\}, \qquad \forall (f, y) \in \bar{F}(\bar{Y}) \times \bar{Y}. \tag{17}$$

We first characterize the coordinate selection procedure.

Definition 3 (Coordinate selection) Given a descent algorithm $\mathcal{M}$ with descent function Δ^f, we call *coordinate selection policy* over $\bar{F}(\bar{Y}) \times \bar{Y}$ any mapping $K : \bar{F}(\bar{Y}) \times \bar{Y} \mapsto \mathcal{N}$ closed on $\bar{F}(\bar{Y}) \times \bar{Y}$ and such that, for all $(f, y) \in \bar{F}(\bar{Y}) \times \bar{Y}$, we have $\Delta^f(z) < \Delta^f(y)$ if $y \notin S^f(\bar{Y})$, $I \in K(f, y)$ and $z \in \mathcal{M}_I(f, y)$.

Remark 2 The conditions of Definition 3 are satisfied, independently of the descent algorithm $\mathcal{M}$, by any closed mapping $K : \bar{F}(\bar{Y}) \times \bar{Y} \mapsto \mathcal{N}$ such that $K(f, y) \subset \{I \in \mathcal{N} \,|\, y_I \notin S^{f_{I:y}}(\bar{Y}_I)\}$ holds for all $(f, y) \in \bar{F}(\bar{Y}) \times \bar{Y}$.

Now, let $K : \bar{F}(\bar{Y}) \times \bar{Y} \mapsto \mathcal{N}$ be a coordinate selection policy over $\bar{F}(\bar{Y}) \times \bar{Y}$ in the sense of Definition 3. We consider the algorithm

$$y^{k+1} \in \mathcal{K}(g^k, y^k), \qquad k = 0, 1, 2, ... , \tag{18}$$

where $\mathcal{K}$ is a mapping $\bar{F}(\bar{Y}) \times \bar{Y} \mapsto \bar{Y}$ defined by

$$\mathcal{K}(f, y) = \cup_{I \in K(f,y)} \mathcal{M}_I(f, y). \tag{19}$$

Result 2 *The mapping $\mathcal{K}$ is closed on $\bar{F}(\bar{Y}) \times \bar{Y}$ and a descent algorithm with respect to g and $S^g(\bar{Y})$ with descent function Δ^g.*

Proof It is immediate from (19) and Definition 3 that $\mathcal{K}$ is a descent algorithm. We show that $\mathcal{K}$ is closed. Let $\{(f^k, y^k)\}$ be a sequence in $\bar{F}(\bar{Y}) \times \bar{Y}$ with $(f^k, y^k) \to (f, y)$ and $\{z^k\}$ a sequence in $\bar{Y}$ such that $z^k \in \mathcal{K}(f^k, y^k)$ for all k and $z^k \to z$. Assume that coordinate selection yields a certain coordinate block sequence $\{I^k\}$ during the generation of $\{z^k\}$. Suppose that $I \in \mathcal{N}$ appears an infinity of times in the block sequence. One can find a subsequence $\{I^{\kappa(k)}\}$ with $I^{\kappa(k)} = I$ for all k. Since K is closed, $I \in K(f, y)$. Consider now the subsequence $\{z^{\kappa(k)}\}$. We have $z^{\kappa(k)} \in \mathcal{M}_I(f^{\kappa(k)}, y^{\kappa(k)})$ for all k. Since $\mathcal{M}_I$ is closed, we find $z \in \mathcal{M}_I(f, y) \subset \mathcal{K}(f, y)$. Hence $\mathcal{K}$ is closed. □

Example 1 One possible coordinate selection policy, denoted by L, assigns to every (f, y) the block coordinate index I for which the descent from y along Δ^f is potentially the most effective for some test points generated by $\mathcal{M}_J(f, y)$ $(J \in \mathcal{N})$. We define $L : \bar{F}(\bar{Y}) \times \bar{Y} \mapsto \mathcal{N}$ by

$$L(f, y) = \{I \in \mathcal{N} \mid \max_{J \in \mathcal{N}} \min_{z \in \mathcal{M}_I(f,y), \hat{z} \in \mathcal{M}_J(f,y)} [\Delta^f(z) - \Delta^f(\hat{z})] \leq 0\}. \tag{20}$$

Result 3 *If the graph of $\Delta^{\cdot}(\cdot)$ (seen as a function on $\bar{F}(\bar{Y}) \times \bar{Y}$) is closed, then the mapping L is a coordinate selection policy over $\bar{F}(\bar{Y}) \times \bar{Y}$.*

Proof We first show that L is closed. Let $\{(f^k, y^k)\}$ be a sequence in $\bar{F}(\bar{Y}) \times \bar{Y}$ with $(f^k, y^k) \to (f, y)$ and $\{I^k\}$ a block-sequence in $\mathcal{N}$ such that $I^k \in L(f^k, y^k)$ for all k and $I^k \to I \in \mathcal{N}$. For any $J \in \mathcal{N} \backslash \{I\}$, consider in accordance with (20) two sequences $\{z^k\}$ and $\{\hat{z}^k\}$ such that $z^k \in \mathcal{M}_I(f^k, y^k)$, $\hat{z}^k \in \mathcal{M}_J(f^k, y^k)$ and $\Delta^{f^k}(z^k) \leq \Delta^{f^k}(\hat{z}^k)$ for all k. We can find convergent subsequences $\{z^{\kappa(k)}\}$ and $\{\hat{z}^{\kappa(k)}\}$ with $z^{\kappa(k)} \to z$, $\hat{z}^{\kappa(k)} \to \hat{z}$, and $\Delta^{f^{\kappa(k)}}(z^{\kappa(k)}) \leq \Delta^{f^{\kappa(k)}}(\hat{z}^{\kappa(k)})$ for all k. By assumption on $\Delta^{\cdot}(\cdot)$, we infer $\Delta^f(z) \leq \Delta^f(\hat{z})$, where $z \in \mathcal{M}_I(f, y)$ and $\hat{z} \in \mathcal{M}_J(f, y)$ since $\mathcal{M}_I$ and $\mathcal{M}_J$ are closed. Repeating this rationale for all $J \in \mathcal{N}$, we find $z \in L(f, y)$, and L is closed. □

3 Parallel implementations with computer standby

One relevant topic of investigation in relation to algorithm (7) lies in the coordination of the applications of the descent algorithms with the generation of the model sequence for the true function (see e.g. [7]). In fact, there exists a contrast between the prevalent descent algorithms, sometimes approaching superlinear convergence rates, or their coordinate descent implementations, which in most cases converge linearly (see Appendix B), and the generation of $\{g^k\}$, which often is a much slower process—recall (10) for the sample

average model—inclined to hamper the execution of (7). The competition experienced by the two processes may be addressed by temporarily suspending the successive applications of the descent algorithm when the precision of the current function model g^k is too poor to expect sensible improvements in minimizing the true function g [18]. In section 3, we parallelize this idea by assuming that the sequence $\{g^k\}$ is given and that each individual computer of a parallel setting, which in the current framework is programmed to apply the descent mapping at its coordinate level, can decide on its own to refrain from doing so when no significant improvement is to be expected.

3.1 Considerations on the sample average model

Suppose that (7) is used with any parallel implementation $\mathcal{M}$ of a given descent algorithm. Consider an operator δ such that, for any given set X and $(f, y) \in \bar{F}(X) \times X$, $\delta(f, y)$ is a quantity related to the optimality of the point y with regard to the minimization of f on X, so that $\delta(f, y) = 0$ iff $y \in S^f(X)$. Further assume that δ has the $\delta = (\delta_1, ..., \delta_n)$ on $\bar{F}(\bar{Y}) \times \bar{Y}$, where, for $i \in N$, δ_i takes its values in some vector space $\mathbb{R}^{p_i}$, we have $\delta_i(f, y) \equiv \delta(f_{i:y}, y_i)$ at every $(f, y) \in \bar{F}(\bar{Y}) \times \bar{Y}$, and thus $\delta_i(f, y) = 0$ iff $y_i \in S^{f_{i:y}}(\bar{Y}_i)$.

Recall the sample average model sequence given in (9) and let $i \in N$. It is convenient to assume that, at any $y \in \bar{Y}$, $\sqrt{q(k)}[\delta_i(g^k, y) - \delta_i(g, y)]$ is asymptotically normal as $k \to \infty$ with a certain covariance $\Sigma_i(y)$—see Section 4 for an example of such a mapping δ in the context of continuously differentiable functions. In $\delta_i(g^k, y)$ we find a consistent estimate of $\delta_i(g, y)$ at $y \in \bar{Y}$. If, in addition, we can derive a sequence $\{\hat{\Sigma}_i^k(y)\}$ of approximate covariance matrices converging to $\Sigma_i(y)$, then our estimator $\delta_i(g^k, y)$ may be seen, due to sample averaging, as an approximately normal variable with mean $\delta_i(g, y)$ and covariance $\hat{\Sigma}_i^k(y)/q(k)$. It follows that the hypothesis $\delta_i(g, y) = 0$ (or equivalently $y_i \in S^{g_{i:y}}(\bar{Y}_i)$) can be tested at every step k and point $y \in \bar{Y}$ by inspection of the statistic $\delta_i(g^k, y)'[\hat{\Sigma}_i^k(y)/q(k)]^\dagger \delta_i(g^k, y)$, which is asymptotically chi-squared with $\mathrm{rk}(\Sigma_i(y))$ degrees of freedom, where $\dagger$ denotes the Moore-Penrose pseudoinverse and $\mathrm{rk}(\cdot)$ the matrix rank. We suggest the heuristic

$$\begin{aligned} 0 \in &\{\delta_i(g^k, y) + [\hat{\Sigma}_i^k(y)]^{\frac{1}{2}} x \,|\, x \in \mathbb{R}^{p_i}\} \\ &\cap \{x \in \mathbb{R}^{p_i} \,|\, [x - \delta_i(g^k, y)]'[\hat{\Sigma}_i^k(y)/q(k)]^\dagger [x - \delta_i(g^k, y)] \leq \beta_{\mathrm{rk}(\hat{\Sigma}_i^k(y))}(\pi)\}, \end{aligned} \tag{21}$$

where the parameter $\pi \in [0; 1]$ is an arbitrary p-value, and $\beta_d(\pi)$ is the maximum squared Mahalanobis distance between 0 and $\delta_i(g^k, y)$ observed with probability π under the $\delta_i(g, y) = 0$ hypothesis, approximately given by the cumulative chi-squared distribution with d degrees of freedom [8]. If $\hat{\Sigma}_i^k(y)$ has full rank, then (21) reduces to $\delta_i(g^k, y)'[\hat{\Sigma}_i^k(y)]^{-1}\delta_i(g^k, y) \leq \beta_{p_i}(\pi)/q(k)$.

When (21) holds, it may be considered that y_i is close enough to the set $S^{g_{i:y}}(\bar{Y}_i)$ and that further improvements along coordinate i can only be obtained by reducing its covariance, i.e. by increasing k and thus $q(k)$. The

optimization algorithms considered in this section have the property to ignore coordinate i at step k whenever (21) is true by setting $y_i^{k+1} = y_i^k$, thus placing computer i into 'standby' mode as long as (21) is satisfied.

In summary, we consider effort-saving implementations of (7), in which descent is only performed at the coordinates where significant progress along the true function can be expected, and designed based on a quantity $q(k)^{-1}\hat{\Sigma}^k$, where $q(k)^{-1}$ is a scalar sequence decreasing to 0 and $\hat{\Sigma}^k = (\hat{\Sigma}_1^k, ..., \hat{\Sigma}_n^k)$ is a bounded vector sequence of functions on $\bar{Y}$.

3.2 Standby policies

Less specifically, consider a continuous function $d : \mathbb{R}_{\geq 0} \mapsto \mathbb{R}_{\geq 0}$ such that $d(0) = 0$ and $0 < d(x) < x$ if $x > 0$, and a decreasing sequence $\{w^k\}$ in $W \equiv [0; +\infty)$ such that $w^0 > 0$ and $w^{k+1} = d(w^k)$ for all k. Assume that one can compute (in complement to $\{g^k\}$) a function sequence $\{v^k\}$ in a functional set $V(\bar{Y})$ equipped with a norm $\|\cdot\|_{\bar{Y}}^V$ and such that $v^k = (v_1^k, ..., v_n^k)$ for all k and $\sup_{k \geq t} \|v^k(y)\|_{\bar{Y}}^V < \infty$ for some $t \geq 0$. It follows that $\{w^k v^k\}$ vanishes uniformly on $\bar{Y}$. The process of selection of the active and inactive computers is represented by a mapping $Z((g^k, v^k), (y^k, w^k))$, which differs from the coordinate selection policies of Section 2.3 in the presence of arguments (v^k and w^k) outside $\bar{F}(\bar{Y})$ and $\bar{Y}$. This difficulty is circumvented by integrating the additional variables into an augmented space, which considers w^k as an $(n+1)^{\text{th}}$ coordinate and is specified as follows.

Definition 4 (Descent in $\tilde{Y}$) Consider the set $\tilde{Y} = \bar{Y} \times W$, and the functional set $\tilde{F}(\tilde{Y})$ defined by $\tilde{F}(\tilde{Y}) = \{(f, v) : (y, w) \in \tilde{Y} \mapsto (f(y), v(y)) \,|\, (f, v) \in \bar{F}(\bar{Y}) \times V(\bar{Y})\}$ and equipped with the norm $\|(f, v)\| = \|f\|_{\bar{Y}}$, where $\|\cdot\|_{\bar{Y}}$ denotes the accepted norm in the topology of $\bar{F}(\bar{Y})$. We introduce the function $\tilde{g} \in \tilde{F}(\tilde{Y})$ such that $\tilde{g}(y, w) = g(y)$ for $(y, w) \in \tilde{Y}$, the function $\tilde{\Delta}^{\tilde{g}}$ defined on $\tilde{Y}$ by $\tilde{\Delta}^{\tilde{g}}(y, w) = \Delta^{\tilde{g}}(y) + w$, and the mapping $\tilde{\mathcal{M}} : \tilde{F}(\tilde{Y}) \times \tilde{Y} \mapsto \tilde{Y}$ such that $\tilde{\mathcal{M}}((f, v), (y, w)) = \mathcal{M}(f, y) \times \{d(w)\}$ for all $(f, v) \in \tilde{F}(\tilde{Y})$ and $(y, w) \in \tilde{Y}$.

Result 4 *The mapping $\tilde{\mathcal{M}}$ is closed on $\tilde{F}(\tilde{Y}) \times \tilde{Y}$ and a descent algorithm with respect to $\tilde{g}$ and $S^{\tilde{g}}(\tilde{Y}) \equiv S^g(\bar{Y}) \times W$ with descent function $\tilde{\Delta}^{\tilde{g}}$. The minimization of $\tilde{g}$ over $\tilde{Y}$ is equivalent the minimization of g over $\bar{Y}$.*

Proof First notice that the last statement is immediate from Definition 4.

Consider sequences $\{(f^k, v^k)\}$ in $\tilde{F}(\tilde{Y})$ and $\{(y^k, w^k)\}$ in $\tilde{Y}$ which respectively converge to $(f, v) \in \tilde{F}(\tilde{Y})$ and $(y, w) \in \tilde{Y}$, and a sequence $\{(z^k, u^k)\}$ in $\tilde{Y}$ such that $(z^k, u^k) \in \tilde{\mathcal{M}}((f^k, v^k), (y^k, w^k))$ for all k, and $(z^k, u^k) \to (z, u)$. By definition of $\tilde{\mathcal{M}}$ and continuity of d we find $u = d(w)$ and, since $\mathcal{M}$ is closed, $z \in \mathcal{M}(f, y)$. Hence $(z, u) \in \tilde{\mathcal{M}}((f, v), (y, w))$, and $\tilde{\mathcal{M}}$ is closed.

It remains to show that $\tilde{\mathcal{M}}$ is a descent algorithm. For any $((g, v), (y, w)) \in \tilde{F}(\tilde{Y}) \times \tilde{Y}$ and $(z, u) \in \tilde{Y}$ such that $(z, u) \in \tilde{\mathcal{M}}((g, v), (y, w))$, we have $z \in \mathcal{M}(g, y)$ and $u = d(w)$. Hence $\Delta^g(z) \leq \Delta^g(y)$ and $u \leq w$. Thus, $\tilde{\Delta}^{\tilde{g}}(z, u) \leq \tilde{\Delta}^{\tilde{g}}(y, w)$. If, in addition, $(y, w) \notin S^{\tilde{g}}(\tilde{Y})$ or, equivalently, $y \notin S^g(\bar{Y})$, then

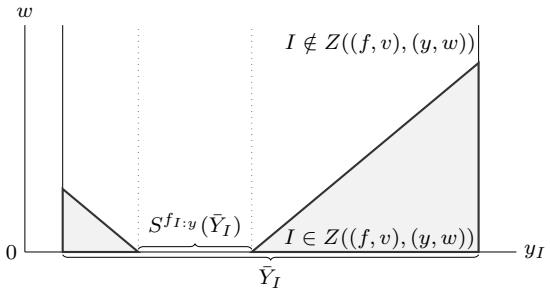

Fig. 1 Graph $\{(y, w, Z((f,v),(y,w))) \,|\, y \in \mathbb{R}^m, w \in W\}$ of a standby policy Z along one block coordinate y_I, for some $f \in \bar{F}(\bar{Y})$, $v \in V(\bar{Y})$, and $I \in \mathcal{N}$. The shaded area contains the points (y_I, w) where I may be chosen by $\mathcal{Z}((f,v),(y,w))$. Notice that for any $y_I \notin S^{f_{I:y}}(\bar{Y}_I)$, this area is eventually reached when $w \downarrow 0$.

$\Delta^g(z) < \Delta^g(y)$. We then find $\tilde{\Delta}^{\tilde{g}}(z,u) < \tilde{\Delta}^{\tilde{g}}(y,w)$, and $\tilde{\mathcal{M}}$ is a descent algorithm. □

Next, we characterize the standby policies. An example of standby policy is illustrated in Fig. 1.

Definition 5 (Standby policy) We call *standby policy* over $\tilde{F}(\tilde{Y}) \times \tilde{Y}$ any mapping $Z : \tilde{F}(\tilde{Y}) \times \tilde{Y} \mapsto 2^N$ closed on $\tilde{F}(\tilde{Y}) \times \tilde{Y}$ and such that

(i) for every $(f,v) \in \tilde{F}(\tilde{Y})$, $y \in Y \setminus S^f(\bar{Y})$ and $w \in W$, we have $I \notin Z((f,v),(y,w))$ if $I \in \mathcal{N}$ and $y_I \in S^{f_{I:y}}(\bar{Y}_I)$;
(ii) for every $(f,v) \in \tilde{F}(\tilde{Y})$ and $y \in Y \setminus S^f(\bar{Y})$, we have $\emptyset \notin Z((f,v),(y,0))$;
(iii) for any $y \in \bar{Y}$, one can find $\epsilon > 0$ such that $I \notin Z((f,v),(y,w))$ for every $I \in \mathcal{N}$ and $(v,w) \in V(\bar{Y}) \times W$ satisfying $w\|v_I(y)\|_{\bar{Y}_I}^V > \frac{1}{\epsilon}$.

Now, consider the mapping $\mathcal{Z} : \tilde{F}(\tilde{Y}) \times \tilde{Y} \mapsto \tilde{Y}$ defined by

$$\mathcal{Z}(\tilde{f},\tilde{y}) = \cup_{J \in Z(\tilde{f},\tilde{y})} \tilde{\mathcal{M}}_J(\tilde{f},\tilde{y}), \qquad \forall (\tilde{f},\tilde{y}) \in \tilde{F}(\tilde{Y}) \times \tilde{Y}, \tag{22}$$

where $Z : \tilde{F}(\tilde{Y}) \times \tilde{Y} \mapsto 2^N$ is a standby policy in the sense of Definition 5, and $\tilde{\mathcal{M}}_J$ is defined based on (17), for all $(f,v) \in \tilde{F}(\tilde{Y})$ and $(y,w) \in \tilde{Y}$, by

$$\tilde{\mathcal{M}}_J((f,v),(y,w)) = \begin{vmatrix} \mathcal{M}_J(f,y) \times \{d(w)\} & \text{if } J \in \mathcal{N} \\ \{(y, d(w))\} & \text{if } J = \emptyset \end{vmatrix} . \tag{23}$$

We derive the algorithm

$$y^{k+1} \in \mathcal{Z}((g^k, v^k),(y^k, w^k)), \qquad k = 0, 1, 2, \ldots , \tag{24}$$

in which $w^0 > 0$. It is easily seen that (24) is an application of (18) in the augmented space with the coordinate selection policy $\tilde{K}$ defined by

$$\tilde{K}(\tilde{f},\tilde{y}) = \{I \cup \{n+1\} \,|\, I \in Z(\tilde{f},\tilde{y})\}, \qquad (\tilde{f},\tilde{y}) \in \tilde{F}(\tilde{Y}) \times \tilde{Y}. \tag{25}$$

Result 5 *For any standby policy Z over $\tilde{F}(\tilde{Y})\times\tilde{Y}$, the mapping given by (25) is a coordinate selection policy over $\tilde{F}(\tilde{Y})\times\tilde{Y}$ with descent algorithm $\tilde{\mathcal{M}}$ and descent function $\tilde{\Delta}^{\tilde{g}}$.*

Proof By definition of Z we already know that $\tilde{K}$ is a closed mapping with values in $2^{N\cup\{n+1\}}\setminus\{\emptyset\}$. It remains to show that the last condition of Definition 3 is satisfied. Let $((g,v),(y,w))\in\tilde{F}(\tilde{Y})\times\tilde{Y}$. Suppose that $(y,w)\notin S^{\tilde{g}}(\tilde{Y})$, i.e. $y\notin S^{g}(\bar{Y})$, and choose any coordinate block $I\cup\{n+1\}\in\tilde{K}((g,v),(y,w))$ and point $(z,u)\in\tilde{\mathcal{M}}_I((g,v),(y,w))$.

First assume that $w=0$. It follows from (ii) in Definition 5 that $I\neq\emptyset$, and from (i) that $y_I\notin S^{f_{I:y}}(\bar{Y}_I)$. By (23) and since $\mathcal{M}$ is a descent algorithm, we have $\Delta^g(z)<\Delta^g(y)$. Using $u\leq w$, we find $\tilde{\Delta}^{\tilde{g}}(z,u)<\tilde{\Delta}^{\tilde{g}}(y,w)$.

If now $w>0$, we find $u<w$ and $\Delta^g(z)\leq\Delta^g(y)$. Hence $\tilde{\Delta}^{\tilde{g}}(z,u)<\tilde{\Delta}^{\tilde{g}}(y,w)$ as well, and $\tilde{K}$ satisfies the conditions of Definition 5. □

The next result is a corollary of Results 2 and 5.

Result 6 *The mapping $\mathcal{Z}$ is closed on $\tilde{F}(\tilde{Y})\times\tilde{Y}$ and a descent algorithm with respect to $\tilde{g}$ and $S^{\tilde{g}}(\tilde{Y})$ with descent function $\tilde{\Delta}^{\tilde{g}}$.*

Remark 3 Notice from the proof of Result 5 that only (i) and (ii) in Definition 5 are needed by Z to work as a valid coordinate-block selection policy. The role of Condition (iii) is to improve the computational efficiency by ensuring that the application of the mapping $\mathcal{M}$ is avoided—totally or partially for a subset of coordinates—when the model g^k is inaccurate.

Remark 4 The dimension of the problem is only augmented in Definition 4 for analysis needs, with no effect on the implemented algorithm.

Remark 5 The algorithm (24) will minimize g over $\bar{Y}$ for any descent algorithm closed on $\bar{F}(\bar{Y})\times\bar{Y}$. It follows from Results 1 and 2 that algorithms such as $\mathcal{C}$ or $\mathcal{K}$ may also be used with standby policies.

4 Application: stochastic network optimization

4.1 Problem description

The function g takes the form (8) for instance, when it is the dual function of a convex stochastic optimization problem, formulated below in standard form.

Problem 1 (Convex stochastic optimization) Let $\omega\in\Omega$ be a random parameter defined on a probability space $(\Omega,\mathcal{F},P)$ and $f:\mathbb{R}^p\times\Omega\to(-\infty,\infty]$ a cost function such that $f(\cdot,\omega)$ is convex for all $\omega\in\Omega$. Consider the problem

$$\begin{array}{ll}\underset{x}{\text{minimize}} & \mathrm{E}[f(x(\omega),\omega)]\\ \text{subject to} & \mathrm{E}[d(x(\omega),\omega)]\leq 0\\ & \mathrm{E}[h(x(\omega),\omega)]=0\end{array}\tag{26}$$

where the function $x:\Omega\times\mathbb{R}^p$ is the unknown, $d:\mathbb{R}^p\times\Omega\to(-\infty,\infty]^v$, $h:\mathbb{R}^p\times\Omega\to(-\infty,\infty]^u$, and $d(\cdot,\omega)$ is convex and $h(\cdot,\omega)$ affine for all $\omega\in\Omega$.

Separability — When placed in a network environment composed of a collection $N = \{1, ..., n\}$ of n computing nodes, Problem 1 frequently enjoys the property to be separable, suggesting a distributed analysis of the problem. The problem variables and parameters are then stored and managed locally: every $x \in \mathbb{R}^p$ is seen as a vector $x = (x_1, ..., x_n)$, where $x_i \in \mathbb{R}^{p_i}$ is local to node i and $\sum_{i=1}^n p_i = p$. One condition for separability is that the cost function be additively separable with respect to $x_1, ..., x_n$, i.e., $f(x, \omega) = \sum_{i=1}^n f_i(x_i, \omega)$ for all $x \in \mathbb{R}^p$ and $\omega \in \Omega$, where each $f_i(x_i, \omega)$ is a function $\mathbb{R}^{p_i} \times \Omega \to (-\infty, \infty]$ convex in x_i. Furthermore, the constraints of the problem must be locally assignable to the nodes, i.e. $d = (d_1, ..., d_n)$ and $h = (h_1, ..., h_n)$, so that each d_i or h_i is only concerned with a node subset $N_i \subset N$, called the neighborhood of i and typically including i and a few other nodes (its neighbors) located in the communication range of i. By gathering the inequality and equality constraints, we can introduce $c_i = (h_i, d_i)$ with dimension $m_i = u_i + v_i$, where c_i rewrites as $c_i(x, \omega) = \sum_{j \in N_i} \varsigma_{ij}(x_j, \omega)$ for some functions $\varsigma_{ij} : \mathbb{R}^{p_j} \times \Omega \to (-\infty, \infty]^{m_i}$. When these separability conditions are met, Problem 1 falls into a class of problems sometimes referred to as *stochastic network utility maximization* (NUM) [13,12,6]. Distributed methods for solving the stochastic NUM problem include Lyapunov optimization frameworks [11, 12], and the dual methods [13], which are addressed in this section.

The dual function of the separable problem is given by (8), i.e. $g(y) = \mathrm{E}[\hat{g}(y, \omega)]$, where the dual variable $y = (y_1, ..., y_n) \in \mathbb{R}^m$ is such that each coordinate $y_i \in \mathbb{R}^{m_i}$, and $\hat{g}$ is given by $\hat{g}(y, \omega) = \sum_{i=1}^n \hat{g}_i(y, \omega)$ for all $y \in \mathbb{R}^m$ and $\omega \in \Omega$, where we define

$$\hat{g}_i(y, \omega) = -\inf_{x \in \mathbb{R}^{p_i}} [f_i(x, \omega) + \sum\nolimits_{j \in N_i} y_j' \varsigma_{ji}(x, \omega)], \quad \forall y \in \mathbb{R}^m, \omega \in \Omega, i \in N. \tag{27}$$

In particular, when $f(\cdot, \omega)$ is strictly convex with nonempty compact domain and $d(\cdot, \omega)$ continuous for every $\omega \in \Omega$, then $\hat{g}(\cdot, \omega)$ is continuously differentiable for all ω over the set $Y = \prod_{i=1}^m Y_i$, where $Y_i \equiv \mathbb{R}_{\geq 0}^{v_i} \times \mathbb{R}^{u_i}$ and, for every $i \in N$, $\arg\inf_{x \in \mathbb{R}^{p_i}} \{f_i(x, \omega) + \sum_{j \in N_i} y_j' \varsigma_{ji}(x, \omega)\}$ reduces to a singleton that we denote by $\{x_i^*(y, \omega)\}$ [2, Lemma 6.3.2]. It follows from Danskin's theorem [17] (see also [2, Theorem 6.3.3]) that the function g is (under mild conditions) continuously differentiable on Y with gradient given by $\nabla g = (\nabla_1 g, ..., \nabla_n g)$, where $\nabla_i g(y) = -\mathrm{E}[(d_i(x^*(y, \omega), \omega), h_i(x^*(y, \omega), \omega))]$, where $x^*(y, \omega) = (x_1^*(y, \omega), ..., x_n^*(y, \omega))$.

Under a constraint qualification, Problem 1 has the same optimal value as the dual problem of minimizing g on Y. Then, one says that strong duality holds, and a solution $\bar{x}$ of Problem 1 can be recovered indirectly from any solution $\bar{y}$ of the dual problem by solving $\bar{x}(\omega) \in x^*(\bar{y}, \omega)$ $\forall \omega \in \Omega$.

4.2 Stochastic optimization

In the following example g is minimized over a convex compact set $\bar{Y} \subset Y$ with the Cartesian product structure $\bar{Y} = \prod_{i=1}^n \bar{Y}_i$. We consider the func-

tional set $F^l(\bar{Y})$ and a cyclic implementation of the gradient projection mapping $\mathcal{G}$, both introduced in Appendix A. A model sequence $\{g^k\}$ converging with probability one towards g in the norm topology (1) is generated in a functional set $\bar{F}^l(\bar{Y})$ satisfying Assumption 1 with respect to $\bar{Y}$ and $F^l(\bar{Y})$. Based on (14) and (31) we derive the cyclic algorithm $\mathcal{C}^{\mathcal{G}} = \mathcal{C}_n^{\mathcal{G}} \circ \cdots \circ \mathcal{C}_1^{\mathcal{G}}$, where $\mathcal{C}_i^{\mathcal{G}}$ is defined as in (12) by $\mathcal{C}_i^{\mathcal{G}}(f,y) = \{(y_1,...,y_{i-1},z,y_{i+1},...,y_n) \,|\, z = \mathcal{G}(f_{i:y}, y_i)\}$ for $(f,y) \in \bar{F}^l(\bar{Y}) \times \bar{Y}$ and $i \in N$.

In order to improve the computational efficiency of $\mathcal{C}^{\mathcal{G}}$, the network nodes are granted the possibility to suspend their individual efforts in accordance with the developments of Section 3.2. We consider the sample average model (9) for the sequence $\{g^k\}$, and set $\delta(f,y) \equiv \mathrm{P}^{\perp}_{H^*(\bar{Y},y)}(-\nabla f(y))$ for any $(f,y) \in \bar{F}^l(\bar{Y}) \times \bar{Y}$, where $H^*(\bar{Y},y)$ is defined in Appendix A as the polar cone of the normal vectors of all the hyperplanes supporting $\bar{Y}$ at y, and $\mathrm{P}^{\perp}_{H^*(\bar{Y},y)}$ denotes the orthogonal projection on $H^*(\bar{Y},y)$. If, for some $i \in N$ and $y \in \bar{Y}$, $\mathrm{P}^{\perp}_{H^*(\bar{Y}_i,y_i)}$ is continuously differentiable at $-\nabla_i g(y)$ with Jacobian matrix $\mathrm{JP}^{\perp}_{H^*(\bar{Y}_i,y_i)}$. It follows from the delta method [17] that the asymptotic covariance of $\sqrt{q(k)}[\delta_i(g^k,y) - \delta_i(g,y)]$ reduces to

$$\Sigma_i(y) = [\mathrm{JP}^{\perp}_{H^*(\bar{Y}_i,y_i)}(-\nabla_i g(y))]\Gamma_i(y)[\mathrm{JP}^{\perp}_{H^*(\bar{Y}_i,y_i)}(-\nabla_i g(y))]', \qquad (28)$$

where $\Gamma_i(y) = \int_\Omega [\nabla_i \hat{g}(y,\omega) - \nabla_i g(y)][\nabla_i \hat{g}(y,\omega) - \nabla_i g(y)]' P(d\omega)$. A sample-average estimate of (28) is given by

$$\hat{\Sigma}_i^k(y) = [\mathrm{JP}^{\perp}_{H^*(\bar{Y}_i,y_i)}(-\nabla_i g(y))]\hat{\Gamma}_i^k(y)[\mathrm{JP}^{\perp}_{H^*(\bar{Y}_i,y_i)}(-\nabla_i g(y))]', \qquad (29)$$

where $\hat{\Gamma}_i^k(y) = \frac{1}{q(k)-1}\sum_{l=0}^{q(k)-1}[\nabla_i \hat{g}(y,\omega^{k,l}) - \nabla_i g^k(y)][\nabla_i \hat{g}(y,\omega^{k,l}) - \nabla_i g^k(y)]'$ is a consistent estimate of $\Gamma_i(y)$. Note that bounds on $\Sigma_i(y)$ and $\hat{\Sigma}_i^k(y)$ can also be derived at points y where the Jacobian $\mathrm{JP}^{\perp}_{H^*(\bar{Y}_i,y_i)}(-\nabla_i g(y))$ is not defined, based on the directional derivatives of $\mathrm{P}^{\perp}_{H^*(\bar{Y}_i,y_i)}$.

Consider now the decreasing sequence $\{w^k\}$ with $w^k = 1/q(k)$ for all k, and the sequence $\{v^k\}$ such that $v^k = (v_1^k,...,v_n^k)$ and $v_i^k = \hat{\Sigma}_i^k$ for all k and $i \in N$. We let $V(\bar{Y}) = \prod_{i=1}^n V_i(\bar{Y})$, where $V_i(\bar{Y})$ is of the type $\bar{Y}_i \mapsto \mathbb{R}^{p_i \times p_i}$. Using (22) and Definition 4, we devise the algorithm $\mathcal{Z}^{\mathcal{G}}$ given by

$$\mathcal{Z}^{\mathcal{G}}(\tilde{f},\tilde{y}) = \cup_{J \in Z(\tilde{f},\tilde{y})} \tilde{\mathcal{C}}_J^{\mathcal{G}}(\tilde{f},\tilde{y}), \qquad \forall \tilde{f} \in \bar{F}^l(\bar{Y}) \times V(\bar{Y}), \tilde{y} \in \bar{Y} \times W, \qquad (30)$$

where $\tilde{\mathcal{C}}_J^{\mathcal{G}}$ is defined as in (23) via (17), in which we set $\mathcal{M} \equiv \mathcal{C}^{\mathcal{G}}$, and Z is a standby policy[3] that can produce the heuristic specified by (21) in Section 3.1.

[3] Such a policy can be defined by extrapolation from the points of interest $((f,v^k),(z,w^k))$, for all k and every $(f,z) \in \bar{F}^l(\bar{Y}) \times \bar{Y}$. Proceeding by induction, we set $\bar{Z}_0((f,v^k),(z,w^k)) = \emptyset$ and, for $i \in N$, $\bar{Z}_i((f,v^k),(z,w^k)) = \bar{Z}_{i-1}((f,v^k),(z,w^k))$ if (21) holds with strict inequality at every $y \in \mathcal{C}_J^{\mathcal{G}}(f,z)$ with $J \in \bar{Z}_{i-1}((f,v^k),(z,w^k))$, $\bar{Z}_i((f,v^k),(z,w^k)) = \{J \cup \{i\} \,|\, J \in \bar{Z}_{i-1}((f,v^k),(z,w^k))\}$ if one can find $y \in \mathcal{C}_J^{\mathcal{G}}(f,z)$ with $J \in \bar{Z}_{i-1}((f,v^k),(z,w^k))$ such that (21) does not hold, and $\bar{Z}_i((f,v^k),(z,w^k)) = \bar{Z}_{i-1}((f,v^k),(z,w^k)) \cup \{I \cup \{i\} \,|\, I \in \bar{Z}_{i-1}((f,v^k),(z,w^k))\}$ otherwise. It is easily seen that $Z \equiv \bar{Z}_n$ satisfies the conditions of Definition 5.

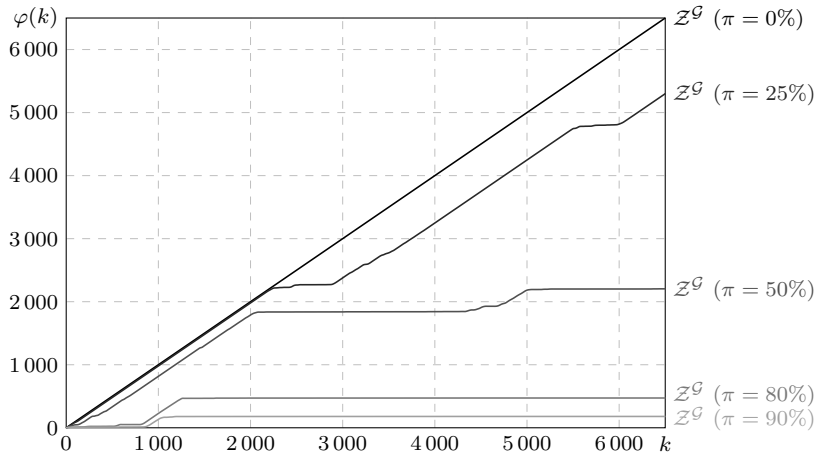

Fig. 2 Cumulative number of applications of $\mathcal{G}$ per node at step k: $\varphi(k)$

4.3 Numerical results

Problem 1 is implemented as detailed in Appendix C, and we report the performance of a run of the algorithm (30) for the minimization of the dual function g of a random instance of the problem. The generated network is limited to 10 nodes and 20 edges in order to allow for comparison with the stochastic approximation method with averaging (SA) [14]. The cyclic algorithm $\mathcal{Z}^{\mathcal{G}}$ is implemented in a large compact set $\bar{Y} \subset Y$ with Newton scaling for $\mathcal{G}$ along each coordinate, i.e. $T(f, x) = [\nabla^2 f(x)]^{-1}$, and with the standby policy specified by (21) for growing values for the parameter π.

Figure 4.3 displays a quantity $\varphi(k)$ denoting, in function of the number of iterations k, the cumulative number of applications per node of the mapping $\mathcal{G}$, which is only effective when (21) is rejected. The abrupt shifts in the slope of $\varphi(k)$ betray phases where most nodes are active and sequences $\{y^k\}$ follow trajectories in Y typical of the coordinate descent methods, alternating with other phases where the nodes remain in standby and the prevalence of which increases with π.

Table 1 reports, for several values of π and for various precisions ϵ, the number of steps $\tau(\epsilon)$ after which the duality gap $|\mathrm{E}[f(x^*(y^k, \omega), \omega)] + g(y^k)|$ remains less than ϵ in the collected data. Observe that increasing π does not considerably slow down the convergence of $\mathcal{Z}^{\mathcal{G}}$—essentially dictated by the convergence speed of the sequence $\{g^k\}$—in spite of the important reduction in the frequency of application of $\mathcal{G}$. Hence substantial savings in operations can be effected by an appropriate choice for π, as illustrated by the second table, which reports the cumulative number of applications of $\mathcal{G}$ per node needed for the duality gap to be less than ϵ, denoted by $\bar{\tau}(\epsilon) \equiv \varphi(\tau(\epsilon))$. A trade-off symbolized by the parameter π can be observed between speed of convergence

Table 1 Number of steps $\tau(\epsilon) = \min_{\bar{k}}\{\bar{k} : |\mathrm{E}[f(x^*(y^k,\omega),\omega)] + g(y^k)| < \epsilon,\ \forall k \geq \bar{k}\}$ after which the duality gap is less than ϵ, and corresponding cumulative number of applications of $\mathcal{G}$ per node $\bar{\tau}(\epsilon) \equiv \varphi(\tau(\epsilon))$

	$\tau(10^0)$	$\tau(10^{-1})$	$\tau(10^{-2})$	$\tau(10^{-3})$	$\tau(10^{-4})$	$\tau(10^{-5})$
$\mathcal{Z}^{\mathcal{G}}$ ($\pi = 25\%$)	5	6	14	85	1 424	3 879
$\mathcal{Z}^{\mathcal{G}}$ ($\pi = 50\%$)	5	6	14	150	1 424	5 240
$\mathcal{Z}^{\mathcal{G}}$ ($\pi = 80\%$)	5	5	16	177	1 244	8 572
$\mathcal{Z}^{\mathcal{G}}$ ($\pi = 90\%$)	5	5	17	82	1 102	8 800
SA	15	32	385	10 380	$> 999\,999$	

	$\bar{\tau}(10^0)$	$\bar{\tau}(10^{-1})$	$\bar{\tau}(10^{-2})$	$\bar{\tau}(10^{-3})$	$\bar{\tau}(10^{-4})$	$\bar{\tau}(10^{-5})$
$\mathcal{Z}^{\mathcal{G}}$ ($\pi = 25\%$)	3.0	4.0	10.2	76.8	1 404.3	3 127.4
$\mathcal{Z}^{\mathcal{G}}$ ($\pi = 50\%$)	2.9	3.9	7.6	62.0	1 240.0	2 199.0
$\mathcal{Z}^{\mathcal{G}}$ ($\pi = 80\%$)	2.7	2.7	5.5	13.2	456.9	472.8
$\mathcal{Z}^{\mathcal{G}}$ ($\pi = 90\%$)	2.7	2.7	5.2	7.7	167.1	180.4

and computational cost. The comparative struggle of the SA method to solve the problem shows the potential of the approach addressed in this paper.

5 Conclusion

In this study we addressed the stochastic optimization of functions qualifying for parallel analysis. With focus set on a particular family of stochastic optimization methods characterized by the association of (i) a descent algorithm specified by a closed mapping and (ii) a sequence of approximate functions—typically: the sample average estimator—converging in some sense to the function of interest, we took a systematic approach to the parallelization of these methods, extending a convergence result due to [19] to their parallel and distributed modes of implementation.

Besides the benefits usually credited to parallel computing, the parallelization of stochastic optimization algorithms comes out as a means to scalability, as for instance in the applications where the true function has the form of an expectation with respect to random parameters. Often, the set of possible outcomes for these parameters is expected to grow quickly in volume with the dimension of the problem. Parallelisation brings an answer to this 'curse of dimensionality' by decomposing such parameters into local components with lessened complexities invariant with the problem size.

A specificity of the considered stochastic optimization methods is that their computational cost can be reduced by coordinating the iterative pace of the descent algorithm with the accuracy of the convergent function sequence. This can be understood, in a parallelized context, as the possibility to suspend the iterative descent process along any individual coordinate whenever the current function estimate is too inaccurate to expect actual progression along that direction. The resulting algorithm, identified in this text as a Gauss-

Southwell-like implementation of the initial algorithm, was shown to lead to considerable savings in operations even for small problems.

A Scaled gradient projections with line search

Let X be a closed subset of a vector space $\mathbb{R}^p$ and and l a positive scalar constant. We consider the functional class $F^l(X) \subset \mathrm{Lsc}(\mathbb{R}^p)$ of the functions f continuously differentiable on X and such that ∇f is Lipschitz continuous on X with Lipschitz constant l. Given two positive scalars $\underline{\lambda}$ and $\bar{\lambda}$ with $0 < \underline{\lambda} \leq \bar{\lambda} < \infty$, we let $\mathcal{T}(p)$ define the set of the symmetric, positive definite scaling matrices in $\mathbb{R}^{p\times p}$ bounded by $\underline{\lambda}$ and $\bar{\lambda}$, i.e. $\mathcal{T}(p) = \{T \in \mathbb{R}^{p\times p} : \underline{\lambda} I \preceq T \preceq \bar{\lambda} I\}$, where I denotes the identity matrix. A descent algorithm based on scaled projected gradient descents may be formulated as follows.

Definition 6 (Scaled gradient projection algorithm) Let $T : F^l(X) \times X \to \mathcal{T}(p)$ be a scaling mapping, and $\beta, \sigma \in (0,1)$ fixed scalar parameters. Consider the point-to-point mapping $\mathcal{G} : F^l(X) \times X \mapsto X$ defined at every $(f,x) \in F^l(X) \times X$ by $\mathcal{G}(f,x) = \bar{x}(\hat{a})$, where

$$\bar{x}(a) = \arg\min_{y\in X} \nabla f(x)'(y-x) + \frac{1}{2}(y-x)'[aT(f,x)]^{-1}(y-x), \ \forall a > 0, \tag{31}$$

and $\hat{a}$ is chosen as the largest element of $\{\beta^m\}_{m=0}^{\infty}$ satisfying

$$f(x) - f(\bar{x}(\hat{a})) \geq \sigma(\bar{x}(\hat{a}) - x)'[\hat{a}T(f,x)]^{-1}(\bar{x}(\hat{a}) - x). \tag{32}$$

In (31)-(32) the step-size is selected using an approximate line search ruleof the type Armijo [1]. From [3] we know that the step-sizes computed by (32) are restricted to a set $[\underline{a}, 1]$, where $\underline{a} > 0$ is a function of the Lipschitz constant l.

Notice that Algorithm $\mathcal{G}$ is covered by the present framework as a particular implementation of the the algorithm $\mathcal{H} : F^l(X) \times X \to 2^X$ defined by [6]

$$\mathcal{H}(f,x) = \{\bar{x}_{f,x,T}(a) \,|\, a \in [\underline{a}, 1], T \in \mathcal{T}(p), f(x) - f(y) \geq \tfrac{\sigma}{a}\|y-x\|^2_{T^{-1}}\}, \tag{33}$$

where $\bar{x}_{f,x,T}(a) = \arg\min_{y\in X} \nabla f(x)'(y-x) + \frac{1}{2}(y-x)'[aT]^{-1}(y-x)$. It is easy to see that $\mathcal{H}$ is a descent algorithm with respect to any $f \in F^l(X)$ and the set of stationary points

$$S^f(X) \equiv \{y \in X \,|\, \exists \epsilon > 0 \ \nabla f(y)'(z-y) \geq 0 \ \forall z \in X^\epsilon(y)\}, \tag{34}$$

where $X^\epsilon(y) = \{z \in X \,|\, \|z-y\| < \epsilon\}$ denotes a neighborhood of y within X, and with descent function $\Delta^f \equiv f$. The closedness of $\mathcal{H}$ on $F^l(X) \times X$ follows from continuity arguments and the properties of the scaled projection operator (Proposition 3.7 in [5, section 3.3]).

Optimality condition on convex sets — When X is a convex set, (34) reduces to $S^f(X) = \{y \in X \,|\, \nabla f(y)'(z-y) \geq 0 \ \forall z \in X\}$. For any $f \in F^l(X)$ and $y \in X$, we find $y \in S^f(X)$ iff $-\nabla f(y) \in H(X,y)$, where $H(X,y) = \{v \in \mathbb{R}^p \,|\, v'(z-y) \leq 0 \ \forall z \in X\}$ defines the cone of the normal vectors of all the hyperplanes supporting X at y. If $H^*(X,y) = \{v \in \mathbb{R}^p \,|\, v'w \leq 0 \ \forall w \in H(X,y)\}$ denotes the polar cone of $H(X,y)$ and $\mathrm{P}^{\perp}_{H^*(X,y)}(\cdot)$ the orthogonal projection on $H^*(X,y)$, it follows from the projection theorem [5] that

$$y \in S^f(X) \Leftrightarrow \mathrm{P}^{\perp}_{H^*(X,y)}(-\nabla f(y)) = 0. \tag{35}$$

If X is has the Cartesian product form $X = \prod_{i=1}^n$, then $H^*(X,y) = \prod_{i=1}^n H^*(X_i, y_i)$ and (35) rewrites as $y_i \in S^{f_{i;y}}(X_i) \Leftrightarrow \mathrm{P}^{\perp}_{H^*(X_i,y_i)}(-\nabla_i f(y)) = 0$ for $i \in N$. Hence the projected gradient can be used to devise standby policies as introduced in Section 3.

B Local convergence of gradient projection methods

In some cases it is possible to sudy the asymptotic properties of the algorithm (7). In this section we consider, for the sake of illustration, any parallel implementation of the gradient projection mapping $\mathcal{G}$ in a convex, compact, polyhedral set $\bar{Y} \subseteq \mathrm{dom}(g)$ and used with the functional set $F^l(\bar{Y})$ (see Appendix A). The sample average model sequence $\{g^k\}$ is generated in $F^l(\bar{Y})$ according to (9) so that it converges with probability one towards g in the norm topology $\|\cdot\|_{1,\bar{Y}}$. We further assume that g has a unique minimizer z on $\bar{Y}$ ($S^g(\bar{Y}) = \{z\}$) and, similarly, that every g^k has a unique minimizer on $\bar{Y}$ denoted by z^k and equal to the SAA estimator.

Under a strict complementarity condition at z, i.e. $-\nabla g(y) \in \mathrm{int}(H(\bar{Y}, z))$ using the notations of Appendix A, [6, Proposition 4.6] extends to the stochastic optimization framework and local convergence to z takes place in a reduced space $\mathbb{R}^{\tilde{m}}$ ($0 \leq \tilde{m} \leq m$). Then, one can find an $m \times \tilde{m}$ matrix E with orthonormal columns such that any sequence $\{y^k\}$ generated by the considered stochastic optimization algorithm and converging almost surely towards z satisfies, with probability one, $y^k = z + E\tilde{y}^k$ for large k and for some vectors $\tilde{y}^k \in \mathbb{R}^{\tilde{m}}$. Similarly, for k large enough, there exist vectors $\tilde{z}^k \in \mathbb{R}^{\tilde{m}}$ such that $z^k = z + E\tilde{z}^k$. Under stronger assumptions—typically g twice continuously differentiable in a neighborhood of z and $\nabla^2 g(z)$ positive definite—, an asymptotic convergence rate can be derived for the operator $\mathcal{M}(g^k, \cdot)$ in the form of an $\tilde{m} \times \tilde{m}$ matrix $\tilde{R}(g^k, z^k) \preceq I$ given as a function of $\nabla^2 g^k(z^k)$ by the Taylor theorem through an equation of the type:

$$\tilde{y}^{k+1} - \tilde{z}^k = \tilde{R}(g^k, z^k)(\tilde{y}^k - \tilde{z}^k) + \rho(g^k, \tilde{y}^k)(\tilde{y}^k - \tilde{z}^k)(\tilde{y}^k - \tilde{z}^k)'. \tag{36}$$

We refer to [6] for the derivation of $\tilde{R}$ for various implementations of the gradient projection method (e.g. Jacobi, Gauss-Seidel, or more sophisticated settings such as in [10]). The remainder $\rho(g^k, \tilde{y}^k)$ in (36) is a function of second derivatives of g^k and, with probability one, it is uniformly bounded if $\nabla^2 g^k(z^k)$ exists for large k and $\{\nabla^2 g^k\}$ converges uniformly towards $\nabla^2 g$ on a neighborhood of z. Since $E'E = I$, (36) then rewrites as

$$y^{k+1} - z = \tilde{A}^k(y^k - z) + \tilde{B}^k(z^{k+1} - z) + o(\|y^k - z\|), \tag{37}$$

where $\tilde{A}^k = E\tilde{R}(g^k, z^k)E'$, $\tilde{B}^k = E(I - \tilde{R}(g^k, z^k))E'$, and I is the $\tilde{m} \times \tilde{m}$ identity matrix.

If we now suppose that $1/\sqrt{q(k)}[g^k - g]$ converges in distribution and in accordance with (10) to a random element ν of $F^l(\bar{Y})$, then the hypotheses of [17, Theorem 5.8] are satisfied at z. It follows that the first order asymptotics of the SAA estimator z^k can be inferred from the second order Taylor expansion of g at z and the Delta theorem provided that $\arg\inf_{h \in C(z)}\{2h'\nabla\delta(z) + h'\nabla^2 g(z)h\}$ yields a singleton $\{\bar{h}(\delta)\}$ for every $\delta \in F^l(\bar{Y})$, where $C(z) = \{h \in H^*(\bar{Y}, z) \,|\, h'\nabla g(z) = 0\}$ is the critical cone at z. In that case we have

$$q(k)^{-\frac{1}{2}}[z^k - z] \xrightarrow{d} \bar{h}(\nu). \tag{38}$$

We see from (37) and (38) that the convergence of the sequence $\{y^k\}$ is then asymptotically analogous to that of a discrete-time random dynamical system characterized by the affine mapping sequence $\{\tilde{A}^k\}$ (converging almost surely towards an asymptotic convergence rate $\tilde{A}^\infty = E\tilde{R}(g, z)E'$) and a random noise process with variance vanishing like $O(q(k)^{-1})$.

C Implementation of a network flow allocation problem

An instance of Problem 1 is given by the network flow optimization problem studied in [6], where a network with node set $N = \{1, ..., n\}$ and edge set E is represented by a directed graph $\mathcal{G} = (N; E)$. Each edge connects an arbitrarily ordered pair of nodes of N, and each pair of nodes is connected by at most one edge. Direct transmissions of information are only allowed along edges, and the neighborhood N_i of a node $i \in N$ is assumed to coincide with its transmission range. By assigning the edges to one of their connected nodes and ordering them accordingly, the $n \times |E|$ incidence matrix A takes the block form $(A_{ij})_{n \times n}$

where A_{ij} is a line vector, and null iff $j \notin N_i$. The prospect of node failure or unavailability is modeled by the random parameter $\omega \in \Omega$, which reflects the availability of all the network nodes at a given time. For $\omega \in \Omega$, we introduce the stochastic incidence matrix $\hat{A}(\omega)$ such that $\hat{A}_{ij}(\omega) = A_{ij}$ if the nodes i and j are available under ω, and $\hat{A}_{ij}(\omega) = 0$ otherwise $(i, j \in N)$. Since the random parameter ω is the conjunction of local parameters, we can write $\Omega \subset \prod_{i=1}^{n} \Omega_i$, where each Ω_i relates to parameters local to node i.

The objective of the problem is to optimize the expectation of the additively separable cost function f with respect to a transmission flow policy $x(\omega) \in \mathbb{R}^p$ with $x = (x_1, ..., x_n)$ and $x_i : \Omega \mapsto \mathbb{R}^{p_i}$ $(i \in N)$, subject to a mean flow conservation constraint

$$\mathrm{E}[\textstyle\sum_{j \in N_i} \hat{A}_{ij}(\omega) x_j(\omega) - b_i(\omega)] = 0, \qquad \forall i \in N, \tag{39}$$

where $b_i(\omega)$ is the rate of information generated by node i under ω with $\mathrm{E}[\sum_{i=1}^{n} b_i(\omega)] = 0$, and to a convex capacity constraint

$$\mathrm{E}[\textstyle\sum_{j \in N_i} \hat{\kappa}_{ij}(x_j(\omega), \omega)] \leq 0, \qquad \forall i \in N, \tag{40}$$

which limits the mean total activity of each node $i \in N$, where $\hat{\kappa}_{ii}(x, \omega) = A_{ii}^{+} x_i^{+} - d_i$ and $\hat{\kappa}_{ij}(x, \omega) = A_{ij}^{+} x_j^{+}$ if $j \neq i$, $d = (d_1, ..., d_n)$ is a positive constant, and we introduce an operator $\cdot^{+}$ such that $(v_{ij})^{+} = (v_{ij}^{+})$ for any matrix with scalar components v_{ij}. We obtain a separable instance of the problem (26), in which $u_i = v_i = 1$, $m_i = 2$, $Y_i = \mathbb{R} \times \mathbb{R}_{\geq 0}$, $h_i(x, \omega) = \sum_{j \in N_i \cup \{i\}} \hat{A}_{ij}(\omega) x_j - b_i(\omega)$, and $d_i(x, \omega) = \sum_{j \in N_i \cup \{i\}} \hat{\kappa}_{ij}(x_j, \omega)$ for $i \in N$. We use the strictly convex cost function suggested in [23] and defined, for all $i \in N$, by $f_i(v_1, ..., v_l) = \sum_{j=1}^{l} (e^{\gamma v_j} + e^{-\gamma v_j})$, where γ is a positive constant.

In the tests of Section 4.3, it is assumed that a realization ω^k of the parameter $\omega \in \Omega$ can be measured or randomly generated prior to each step k of the stochastic optimization algorithm. A model sequence $\{g^k\}$ is built for g based on $\omega^0, ..., \omega^k$ and in accordance with the sample average model (9), in which we set $q(k) = k + 1$ for all k, $\omega^{t,k} \equiv \omega^k$ for $0 \leq t \leq k+1$, and $\hat{g} = \sum_{i=1}^{n} \hat{g}_i$ which can be computed locally at the nodes using (27). The estimation, at each node $i \in N$, of variations of g along y_i and of the derivative $\nabla_i g$ relies in practice on the estimation of the probability distribution of a multinomial variable[4] with $|\Omega_i|$ possible outcomes [8]. This distribution is specified by the $|\Omega_i|$-dimensional vector μ_i, the components of which symbolize the probabilities of the possible outcomes and sum up to 1. A consistent estimate for μ_i is given at each step k by the empirical probability vector $\hat{\mu}_i^k$ computed from $\omega^0, ..., \omega^k$. The estimation of Σ_i, required by (21), follows from (29) and we find

$$\hat{\Gamma}_i^k(y) = \frac{1}{k} (\nabla_i \hat{g}(y, \omega^0), ..., \nabla_i \hat{g}(y, \omega^k))' [\mathrm{diag}(\hat{\mu}_i^k) - \hat{\mu}_i^k \hat{\mu}_i^{k\prime}] (\nabla_i \hat{g}(y, \omega^0), ..., \nabla_i \hat{g}(y, \omega^k)), \tag{41}$$

where $\mathrm{diag}(\hat{\mu}_i^k)$ denotes the diagonal matrix with the components of $\hat{\mu}_i^k$ as diagonal entries.

References

1. Armijo, L.: Minimization of functions having lipschitz continuous first partial derivatives. Pacific Journal of Mathematics **16**(1), 1–3 (1966)
2. Bazaraa, M.F., Sherali, H.D., Shetty, C.M.: Nonlinear Programming, Theory and Algorithms. John Wiley and Sons, New York (1993)
3. Bertsekas, D.: On the Goldstein-Levitin-Polyak gradient projection method. Automatic Control, IEEE Transactions on **21**(2), 174 – 184 (1976). DOI 10.1109/TAC.1976.1101194

[4] For simplicity, it is assumed in Section 4.3 that at most one network node is unavailable at any time within a two-hop distance of each node, and that all the nodes break down with equal probability. It follows that $|\Omega_i| = \bar{n}_i + 1$, where $\bar{n}_i$ denotes the number of nodes located within a two-hop distance of i. In the general case, one would have $|\Omega_i| \leq 2^{\bar{n}_i}$.

4. Bertsekas, D.: Convex Optimization Theory. Athena Scientific (2009)
5. Bertsekas, D., Tsitsiklis, J.: Parallel and Distributed Computation: Numerical Methods. Athena Scientific (1997)
6. Bilenne, O.: Distributed Methods for Convex Optimisation – Application to Cooperative Wireless Sensor Networks. Ph.D. thesis, Technische Universität Berlin (2014). Submitted
7. Dupuis, P., Simha, R.: On sampling controlled stochastic approximation. Automatic Control, IEEE Transactions on **36**(8), 915 –924 (1991). DOI 10.1109/9.133185
8. Evans, M., Hastings, N.A.J., Peacock, B.: Statistical distributions. A. Wiley-Interscience Publication, New York (2000)
9. Hauskrecht, M., Singliar, T.: Monte-carlo optimizations for resource allocation problems in stochastic network systems. In: Nineteenth International Conference on Uncertainty in Artificial Intelligence, pp. 305–312 (2003)
10. Jadbabaie, A., Ozdaglar, A., Zargham, M.: A distributed Newton method for network optimization. In: 48th IEEE Conference on Decision and Control (CDC) combined with the 28th Chinese Control Conference, pp. 2736–2741. Shanghai, China (2009)
11. Kelly, F.: Charging and rate control for elastic traffic. European Transactions on Telecommunications (1997)
12. Neely, M.: Stochastic Network Optimization with Application to Communication and Queueing Systems. Synthesis Lectures on Communication Networks. Morgan & Claypool Publishers (2010)
13. O'Neill, D., Thian, B., Goldsmith, A., Boyd, S.: Wireless NUM: rate and reliability tradeoffs in random environments. In: IEEE Wireless Communications & Networking Conference, pp. 444–449 (2009). DOI 10.1109/WCNC.2009.4918024
14. Polyak, B., Juditsky, A.: Acceleration of stochastic approximation by averaging. SIAM J. Control Optim. **30**(4), 838–855 (1992). DOI 10.1137/0330046. URL http://dx.doi.org/10.1137/0330046
15. Rubinstein, R.Y., Shapiro, A.: Discrete event systems: Sensitivity analysis and stochastic optimization by the score function method. John Wiley & Sons Ltd., Chichester (1993)
16. Shapiro, A.: Simulation Based Optimization. In: Proceedings of the 28th Conference on Winter Simulation, WSC '96, pp. 332–336. IEEE Computer Society, Washington, DC, USA (1996). DOI 10.1145/256562.256644. URL http://dx.doi.org/10.1145/256562.256644
17. Shapiro, A., Dentcheva, D., Ruszczyski, A.: Lectures on stochastic programming : modeling and theory. MPS-SIAM series on optimization. Society for Industrial and Applied Mathematics, Philadelphia (2009)
18. Shapiro, A., Homem-de Mello, T.: A simulation-based approach to two-stage stochastic programming with recourse. Mathematical Programming **81**(3), 301–325 (1998). DOI 10.1007/BF01580086. URL http://dx.doi.org/10.1007/BF01580086
19. Shapiro, A., Wardi, Y.: Convergence analysis of stochastic algorithms. Mathematics of Operations Research **21**(3), 615–628 (1996). DOI 10.1287/moor.21.3.615. URL http://mor.journal.informs.org/content/21/3/615.abstract
20. Tseng, P., Yun, S.: A coordinate gradient descent method for nonsmooth separable minimization. Mathematical Programming **117**, 387–423 (2009)
21. Tseng, P., Yun, S.: A block-coordinate gradient descent method for linearly constrained nonsmooth separable optimization. Journal of Optimization Theory and Applications **140**(3), 513–535 (2009)
22. Wardi, Y.: Stochastic algorithms with Armijo stepsizes for minimization of functions. Journal of Optimization Theory and Applications **64**, 399–417 (1990). URL http://dx.doi.org/10.1007/BF00939456. 10.1007/BF00939456
23. Zargham, M., Ribeiro, A., Jadbabaie, A., Ozdaglar, A.: Accelerated dual descent for network optimization. CoRR **abs/1104.1157** (2011)