Optimization of complex simulation models with stochastic gradient methods

Alexei A. Gaivoronski
Department of Industrial Economics and Technology Management
Norwegian University of Science and Technology
Trondheim, Norway
Alexei.Gaivoronski@ntnu.no

***Abstract*—We describe the structure of stochastic optimization solver SQG (Stochastic QuasiGradient), which implements stochastic gradient methods for optimization of complex stochastic simulation models. The solver finds the equilibrium solution when the simulation model describes the system with several actors. The solver is parallelizable and it performs several simulation threads in parallel. It is capable of solving stochastic optimization problems, finding stochastic Nash equilibria, stochastic bilevel problems where each level may require the solution of stochastic optimization problem or finding Nash equilibrium. We provide several complex examples with applications to water resources management, energy markets, pricing of services on social networks.**

***Index Terms*—stochactic optimization, stochastic equilibrium, optimization of simulation models, stochastic gradient methods**

I. Introduction

This paper describes the solver SQG, which implements stochastic (quasi)gradient methods for solution of stochastic optimization problems. Its particular strength lies in its capability to solve the problems with substantial nonlinearities and optimize functions, defined by complex stochastic simulation models. It can also solve the stochastic equilibrium problems of different types when the studied system includes constellations of independent actors, choosing their own decisions. Finding Nash equilibrium and solving bilevel stochastic optimization problems and stochastic Stackelberg games are among the capabilities of this solver. Also in the case of equilibrium problems the payoff functions of individual actors can be obtained by simulation models.

The simplest problem addressed by SQG is

$$\min_{x \in X} \mathbb{E} f\left(x, \omega\right) \tag{1}$$

which finds the values of decision variables x of a single actor on the feasible set $X \subseteq \mathbb{R}^n$. Expectation in (1) is taken with respect to the vector of random parameters ω, which models the uncertainty and is defined on appropriate probability space. The SQG solves this problem by generating the sequence of points x^s starting from some initial point x^0 and applying the recursive rule

$$x^{s+1} = \Pi_X \left(x^s - \rho_s \xi^s\right) \tag{2}$$

where $\Pi_X \left(\cdot\right)$ is the projection operator on set X, ξ^s is a statistical estimate of the gradient of function $F(x) = \mathbb{E} f\left(x, \omega\right)$ at point x^s, meaning that it satisfies the property

$$\mathbb{E}\left(\xi^s \mid \mathbb{B}_s\right) = F_x(x^s) + b_s \tag{3}$$

where the conditional expectation in (3) is taken with respect to the $\sigma-$field $\mathbb{B}_s$ describing the history of the process and b_s is some diminishing term. The step size ρ_s satisfies the property

$$\rho_s \geq 0, \; \sum_{s=0}^{\infty} \rho_s = \infty, \tag{4}$$

which is weaker than what is normally required in stochastic approximation [1] because for optimization purposes we need a weaker notion of convergence than what is normally expected in statistics. Under additional technical assumptions the sequence x^s converges to the solution of (1) [2], [3], [4].

The basic problem (1) is used in SQG as a building block for construction of considerably more complex problems, including the equilibrium and bilevel problems mentioned above. Besides, SQG has the capability to process the problems where the expectation operators are present not only in the objective (1), but also in constraints. Such problems occur, for example, in portfolio optimization with risk constraints [5]. This and other capabilities, like integration with simulation models, required substantial additional conceptual and algorithmic development, described in the rest of the paper.

In Section II we define the three level problem hierarchy implemented in SQG together with the basic algorithmic tools utilized there. This follows by discussion of challenges of concurrent optimization and simulation, showing the approach taken by SQG on this issue in Section III. Section IV describes examples of applied problems solved by SQG, concentrating on concurrent simulation and optimization.

We do not present in this paper the mathematical results on the convergence of algorithms, which underlie the operation of SQG. For such results we refer a reader to [2], [3], [1], [6].

II. Problem hierarchy in SQG

SQG is developed for solution of stochastic optimization and equilibrium problems involving decision models of $I \geq 1$ actors. We refer to an instance of such problems as *stochastic* *decision problem* (StDP). Such instance is defined by a collection of functions Φ, which is the basic structure in SQG:

$$\Phi = \left\{ f_{ij}\left(x^i, x^{-i}, y^i, \omega\right), i = 1 : I,\ j = 0 : J_i \right\} \tag{5}$$

Here $x^i \in \mathbb{R}^{n_i}$ is the vector of *decision variables* of actor i, while x^{-i} is the vector of decision variables of all other actors, $x^{-i} = \left\{ x^l, l = 1 : I, l \neq i \right\}$. We denote by x the vector of all decision variables: $x = \left(x^i, x^{-i}\right)$ for any i. The vectors x^i take values from feasibility sets X_i and collection Ψ of these sets constitute another basic structure in SQG:

$$\Psi = \{X_i, i = 1 : I\} \tag{6}$$

The vector ω is the vector of all uncertain parameters in the decision models of all actors. We consider ω to be a random vector defined on appropriate probability space $(\Omega, \mathbb{B}, \mathbb{P})$ with Ω being the event set equipped with σ-field $\mathbb{B}$, on which the probability measure $\mathbb{P}$ is defined.

The vector y^i is the vector of state variables of the decision model of actor i. It is assumed that the state y^i evolves in discrete time $t = 1 : T$, where the time horizon T can be finite or infinite. Then the state y^i takes the value y^{it} at time period t. The subsequent values of state variables are connected with the state equation

$$y^{i,t+1} = \Theta^i\left(x^i, x^{-i}, y^{it}, \omega\right) \tag{7}$$

The functions Θ^i can be quite complex and be defined by a simulation model. We define by $\Theta = \Theta^i, i = 1 : I$ the collection of all state equations for all actors. The SQG provides the meta language and conventions for defining the structure (Φ, Ψ, Θ) described above. Some of the elements of this structure can be empty. In particular, the state variables y^i can be absent, then the state equations (7) and the notion of time disappear. In this case the decision problems become static. Besides, the set of actors can contain a single actor, $I = 1$. In this case there will be no stochastic equilibrium problem and StDP will become a stochastic optimization problem. We shall simplify our notations appropriately in such cases. For example, the functions from (5) will be denoted $f_j(x, \omega), j = 1 : J$ in the case of the single actor without the state variables.

Starting from some initial point $x^{i0}, i = 1 : I$ the SQG generates iteratively the sequence of points x^{is} possibly with additional auxiliary sequences, which converge in a certain probabilistic sense to the solution of collection of problems defined on the structure (Φ, Ψ, Θ) while the number of iterations s tends to infinity. Naturally, the SQG solver does not perform the infinite number of iterations, instead it stops the iteration process when a certain stopping criterion is satisfied. The solved problems and corresponding iterative processes are organized in the hierarchy shown on Figure 1.

A. Lower level: estimation (E level)

Obtain statistical estimates of the values and gradients of functions $F_{ij}\left(x^i, x^{-i}\right)$, which are expected values of functions from (5). These estimates serve as an input to the upper levels of the problem hierarchy as specified in what follows. More precisely, SQG computes these estimates for one of the following functions

$$F_{ij}\left(x^i, x^{-i}\right) = \mathbb{E} f_{ij}\left(x^i, x^{-i}, \omega\right) \tag{8}$$

$$F_{ij}\left(x^i, x^{-i}\right) = \mathbb{E}\frac{1}{T}\sum_{t=1}^{T} f_{ij}\left(x^i, x^{-i}, y^{it}, \omega\right) \tag{9}$$

$$F_{ij}\left(x^i, x^{-i}\right) = \mathbb{E}\lim_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T} f_{ij}\left(x^i, x^{-i}, y^{it}, \omega\right) \tag{10}$$

where $\mathbb{E}$ is the expectation operator with respect to random variables ω. The case (8) occurs in the absence of the state variables y^i. The case (9) refers to the dynamic problem with the finite time horizon, while the case (10) deals with the estimation of the steady state function values on the infinite time horizon.

1a. Estimation of function values. These estimates are obtained by the moving average iterative process

$$F_{ij}^{s+1} = \left(1 - \alpha_i^s F_{ij}^s\right) + \alpha_i^s \varphi_{ij}^s \tag{11}$$

where φ_{ij}^s is an observation of the quantities under the expectation sign in (8)-(10) satisfying the property

$$\mathbb{E}\left(\varphi_{ij}^s \mid \mathbb{B}_s\right) = F_{ij}\left(x^{is}, x^{-is}\right) + a_{ij}^s \tag{12}$$

where a_{ij}^s tends to zero with $s \to \infty$ and $\mathbb{B}_s$ is the $\sigma-$field describing the history of the iterative process, which generates the sequences $\left(x^{is}, F_{ij}^s\right)$ prior to iteration s. Some examples of observations φ_{ij}^s :

$$\varphi_{ij}^s = f_{ij}\left(x^{is}, x^{-is}, \omega^s\right) \tag{13}$$

$$\varphi_{ij}^s = \frac{1}{T}\sum_{t=1}^{T} f_{ij}\left(x^{is}, x^{-is}, y^{is}, \omega^{st}\right) \tag{14}$$

$$\varphi_{ij}^s = f_{ij}\left(x^{is}, x^{-is}, y^{is}, \omega^s\right) \tag{15}$$

where ω^s and ω^{st}, $s = 1, ..., t = 1, ...$ are independent observations of random parameters ω. In expressions above (13) corresponds to (8), (14) corresponds to (9) and (15) corresponds to (10). In the case (15),(10) the time step t corresponds to a single iteration step s. That is, in order to obtain the new update of the estimate F_{ij}^s through the process (11) it is enough to simulate the dynamics of the system for just one time step obtaining the observation ω^s of random parameters, computing y^{is} as in (7):

$$y^{is} = \Theta^i\left(x^{is}, x^{-is}, y^{i,s-1}, \omega^s\right)$$

and obtaining φ_{ij}^s from (15). The SQG obtains the observations φ_{ij}^s by calling the external function `fun` written by the SQG user following the interface rules, which make it callable from SQG. This function takes as the input the values $\left(x^{is}, x^{-is}\right)$ and additionally the value of $y^{i,s-1}$ in case (15). Its output comprises the value φ_{ij}^s and the new state y^{is} in the case (15). During the call this function generates the new observation ω^s of the random parameters, if it is required by the estimation process. The value of the additional input parameter signals this necessity to fun. Alternatively, in the case of several concurrent estimation processes with different inputs of decision parameters, this function provides the observations of φ_{ij}^s with the same values of random parameters. See the section on estimation of the gradient for the explanation of such necessity.

The estimates F_{ij}^s are fed to the upper levels of the problem hierarchy, where they are used in the iterative solution process of the upper level problems. In particular, they are used in the stopping criterion of the whole iterative process, for the output of the optimal values of the objective functions, for estimation of gradients and for processing of constraints. They should satisfy the following fundamental convergence property

$$\left|F_{ij}^s - F_{ij}\left(x^{is}, x^{-is}\right)\right| \to 0 \tag{16}$$

in a certain probabilistic sense as $s \to \infty$. In other words, we do not need here the precise estimates of the function values from (8)-(10) for any fixed value of decision variables $\left(x^i, x^{-i}\right)$. Instead, the estimates should trace with gradually increasing precision the changing value of these functions while $\left(x^{is}, x^{-is}\right)$ changes in the course of iterations. This is a very mild requirement, allowing to make only a single observation of random parameters and a single estimation step per one update of decision parameters $\left(x^{is}, x^{-is}\right)$. However, it requires a specific coordination between the estimation process (11) and the iterative updating process of decision variables $\left(x^i, x^{-i}\right)$. Namely, the steps α_i^s from (11) should be larger than the updating steps for the decision variables and their ratio should asymptotically approach infinity.

Besides tracing the values of functions (8)-(10) at changing $\left(x^{is}, x^{-is}\right)$, other concurrent estimation processes of type (11) may compute the estimates F_{ij}^{ls}, which trace the values of these functions for other sequences of points $\left(x^{ils}, x^{-ils}\right)$, which are connected with the original sequence. In particular, the output of such estimation processes may be used on the upper levels of the problem hierarchy for the estimation of the gradients of functions (8)-(10) using finite differences. In this case one can take

$$x^{ils} = x^{is} + \delta_i^s e_l^i,\ l = 1 : n_i \tag{17}$$

where e_l^i is a unit vector of space $\mathbb{R}^{n_i}$.

1b. Estimation of gradient values. The SQG computes the estimates ξ_{ij}^s of the gradient of function $F_{ij}\left(x^i, x^{-i}\right)$ at point $\left(x^{is}, x^{-is}\right)$ from (8)-(10) with respect to variables x^i. These estimates are sent to the higher levels of the problem hierarchy and should satisfy the following condition

$$\mathbb{E}\left(\xi_{ij}^s \mid \mathbb{B}_s\right) = \nabla_{x^i} F_{ij}\left(x^{is}, x^{-is}\right) + b_{ij}^s \tag{18}$$

where b_{ij}^s tends to zero with $s \to \infty$. There are two possibilities

- The user includes the computation of ξ_{ij}^s in the body of the external function fun. Then this function returns the value of ξ_{ij}^s with each call of fun performed by SQG. This option is preferable when it is relatively easy to compute one of the following entities

$$\xi_{ij}^s = \nabla_{x^i} f_{ij}\left(x^{is}, x^{-is}, \omega^s\right) \tag{19}$$

$$\xi_{ij}^s = \frac{1}{T}\sum_{t=1}^{T} \nabla_{x^i} f_{ij}\left(x^{is}, x^{-is}, y^{is}, \omega^{st}\right) \tag{20}$$

$$\xi_{ij}^s = \nabla_{x^i} f_{ij}\left(x^{is}, x^{-is}, y^{is}, \omega^s\right), \tag{21}$$

which similarly to (13)-(15) corresponds to the cases (8)-(10).

- The complexity of functions (8)-(10) typical of simulation models makes it difficult to compute the values (19)-(21) directly. Then the estimates ξ_{ij}^s are computed by the estimation level of SQG using only the function observations (12) and function estimates (11). In order to do this $n_i + 1$ tracing moving average processes (11) are performed in parallel for the same sequence of observations of random parameters ω^s or ω^{st} and $n_i + 1$ sequences of decision parameters x^{is}, x^{ils} from (17). These processes produce the estimates F_{ij}^s, F_{ij}^{ls} and the SQG computes

$$\xi_{ij}^s = \frac{1}{\delta_i^s}\sum_{l=1}^{n_i}\left(F_{ij}^{ls} - F_{ij}^s\right) \tag{22}$$

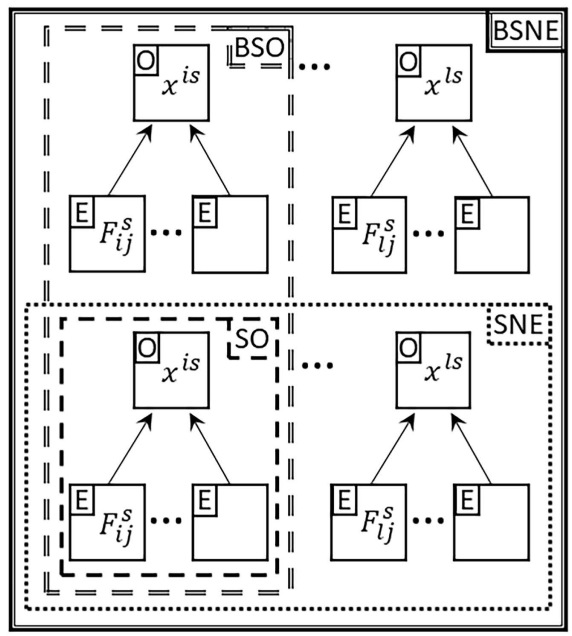

Figure 1. Problem and process hierarchy in SQG

B. *Intermediate level: optimization* (O level)

This level takes the values of the estimates F_{ij}^s, ξ_{ij}^s supplied by the estimation level and generates the sequences $x^{is},\ i = 1 : I,\ s = 1, ...$ which solves the following problem

$$\text{"}\min_{x^i}\text{" ("}\max\text{")}\ F_{i0}\left(x^i, x^{-i}\right) \tag{23}$$

subject to

$$F_{ij}\left(x^i, x^{-i}\right) \leq 0,\ j = 1 : J \tag{24}$$

$$x^i \in X_i \tag{25}$$

That is, the generated sequence x^{is} attempts to minimize or maximize the objective $F_{i0}\left(x^i, x^{-i}\right)$ of actor i with respect to decision variables x^i of this actor. Such sequences are generated for each actor. These sequences may or may not actually converge to the solution of respective optimization problems due to dependence of the objectives not only on the decision variables of respective actors, but also on decision variables of all actors. Still, depending on the coordination between the step sizes used for their generation, the sequences may converge to the solution of optimization problems in a certain sense. In other cases they will converge to the solutions of equilibrium problems, defined on the upper level of the problem hierarchy. For this reason we put min or max from (23) in quotation marks. In the case of a single actor the problem (23)-(25) reduces to a proper optimization problem. We drop the index i then and the problem becomes the following:

$$\min_{x\in X}\,(\max)\;\; F_0\left(x\right) \tag{26}$$

subject to

$$F_j\left(x\right)\leq 0,\;\; j=1:J \tag{27}$$

The problem (23)-(25) looks similar to deterministic nonlinear optimization (or equilibrium) problems. This similarity is deceitful, however, because functions $F_{ij}\left(x^i,x^{-i}\right)$, usually can not be computed with precision required for application of deterministic nonlinear programming techniques. This is due to the presence of the expectation operator in the definition of these functions (8)-(10), which can not be computed precisely for the problems of realistic dimensions and complexity. Therefore the problem (23)-(25) belongs to the family of stochastic optimization (or equilibrium) problems [7]. Another reasons are long computing times necessary for precise simulations, when these functions are obtained through simulation models, and the transient dynamic behavior in the case of the steady state optimization (10) discussed in more detail in Section III.

The SQG generates the sequences x^{is} following the stochastic gradient iteration similar to (2):

$$x^{i,s+1}=\Pi_{X_i}\left(x^{is}-\rho_{is}\left(\xi_{i0}^s+\sum_{j=1}^{m_i}C_{ij}\eta_{ij}^s\xi_{ij}^s\right)\right) \tag{28}$$

$$\eta_{ij}^s=\max\left(0,F_{ij}^s\right)\vee\mathbf{1}_{\left\{F_{ij}^s>0\right\}} \tag{29}$$

Observe that in the case of a single actor, $i=1$ and the absence of expectation type constraints, $m_i=0$, the sequence (28),(29) coincides with the basic SQG process (2). In the case of several actors the step size ρ_{is} is individual for actor i. This is important for solution of bilevel stochastic optimization problems and Stackelberg games. The second term in the internal parenthesis in (28) is dedicated to processing of expectation constraints. It makes (28) to be a basic SQG process of type (2) for minimization of function $F_{i0}\left(x^i,x^{-i}\right)$ with added penalty term, which penalizes the violation of constraints (24). The positive constant C_{ij} is the penalty coefficient, vector ξ_{ij}^s is the stochastic gradient of constraint satisfying (18) and η_{ij}^s from (29) indicates the violation of constraint. Since exact values of $F_{ij}\left(x^i,x^{-i}\right)$ are unknown, η_{ij}^s substitutes them with the estimates F_{ij}^s provided by the estimation level of the problem hierarchy. The two alternatives for η_{ij}^s shown in (29) correspond to two types of penalties. The first alternative is the classical quadratic penalty, while the second one is the nonsmooth linear penalty. In the case of maximization the signs in (28) are changed to the opposite ones.

C. *Upper level: equilibrium problems* (Q level)

This level constructs specific problem types and their solutions from solution sequences x^{is} and function estimates F_{ij}^s provided by the two lower levels, see Figure 1. We consider the following four problem types.

3a. Stochastic optimization (SO). This is the simples type of SQG problems with the empty third level of SQG hierarchy. In this case only single actor is present, $i=1$. The problem formulation is shown in (26)-(27) where $F_j\left(x\right)$ can belong to one of the types (8)-(10). Observe that also the problems of this type can be quite complex because they may include expectation constraints (27) and the observations of functions $F_j\left(x\right)$ can be obtained by complex simulations.

3b. Bilevel stochastic optimization (BSO). These problems are known otherwise as leader-follower games or Stackelberg games with two actors [8]. We describe them here in the interpretation of the leader-follower game. In this case there are two actors, $i=2$. The actor 1 called the *leader* announces his decision $\bar{x}^1$ to actor 2 called the *follower*. Knowing the decision of the leader the follower chooses his decision $\bar{x}^2$ solving stochastic optimization problem

$$\min_{x^2\in X_2}\;\; F_{20}\left(x^2,\bar{x}^1\right) \tag{30}$$

$$F_{2j}\left(x^2,\bar{x}^1\right)\leq 0,\;\; j=1:J \tag{31}$$

for fixed $x^1=\bar{x}^1$. In this way the decision $\bar{x}^2$ will depend on $\bar{x}^1:\bar{x}^2=\bar{x}^2\left(\bar{x}^1\right)$. Knowing this, the leader chooses his decision solving the problem

$$\min_{x^1\in X_1}\;\; F_{10}\left(x^1,\bar{x}^2\left(x^1\right)\right) \tag{32}$$

$$F_{1j}\left(x^1,\bar{x}^2\left(x^1\right)\right)\leq 0,\;\; j=1:J \tag{33}$$

Bilevel optimization problems are quite challenging numerically even in the deterministic linear case. The SQG solves them by running two concurrent SQG processes (28)-(29) with asymptotically vanishing ratio between leader and follower step sizes: $\rho_{1s}/\rho_{2s}\to 0$.

3c. Stochastic Nash equilibrium (SNE). In this case we have $I>1$ actors with payoffs $F_{i0}\left(x^i,x^{-i}\right)$ having one of the types (8)-(10) and possible constraints (24). The SQG runs I concurrent SQG processes (28)-(29), which under additional technical assumptions converge to the Nash equilibrium [9] if it exists.

3d. Bilevel stochastic Nash equilibrium (BSNE). This is the combination of problems 3b and 3c. There are leader and follower levels as in (30)-(33), but there are $I_1\geq 1$ leaders and $I_2\geq 1$ followers, $I=I_1+I_2$. Suppose that the actors $i=1:I_1$ are the leaders and the actors $i=I_1+1:I$ are the followers. Then SQG runs I concurrent SQG processes

(28)-(29) with asymptotically vanishing ratios between leader step sizes and follower step sizes: $\rho_{is}/\rho_{ls} \to 0$ if $1 \le i \le I_1$ and $I_1 + 1 \le j \le I$.

We emphasize here again that in all the problem types mentioned above the payoff and constraint functions can originate from complex simulations.

III. Concurrent optimization and simulation with SQG

Here we discuss in more detail the issues regarding integration of simulation and optimization using SQG. As we have mentioned above, the functions $f_{ij}\left(x^i, x^{-i}, \omega\right)$ or $f_{ij}\left(x^i, x^{-i}, y^{it}, \omega\right)$ from (8)-(10) can result from running of complex simulation model implemented by user inside function `fun` required by SQG. Such simulation can be quite complex, include in its body *yes* or *no* decisions of actors based on reaching of certain triggering quantities certain thresholds, like acceptance or rejection of an investment project based on the predictions of profit. Or, decision to release a certain amount of water from a reservoir based on the level of water in it. It can also include solutions of simpler equilibrium or optimization problems.

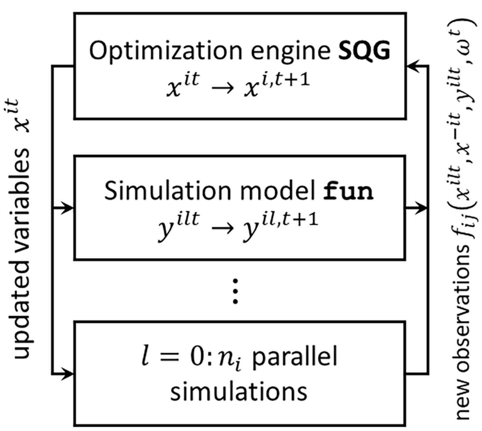

Figure 2. Concurrent optimization and simulation on the infinite horizon

To be more specific, let us consider optimization of the steady state function $F_{ij}\left(x^i, x^{-i}\right)$ on the infinite horizon (10), operating in discrete time $t = 1, ...,$ where we assume that the state transformation from y^{it} to $y^{i,t+1}$ is performed by simulation model. We describe here how the optimal or equilibrium values of decision variables x^i can be obtained using $I + \sum_i n_i$ parallel simulation runs. In this case one simulation time step t coincides with one SQG iteration s from (28), so we use t also for indexing of SQG iterations. The concurrent simulation and optimization process is presented on Figure 2. It is performed as follows.

1. Initialization. At the start the initial values x^{i0} are selected as the starting points for SQG iterations (28). Simulation processes Λ^{li}, $i = 1 : I$, $l = 0 : n_i$ are initialized. The points x^{i0} are sent to simulation processes Λ^{0i} and points x^{il0}, $l = 1 : n_i$ obtained as in (17) for $s = 0$ are sent to processes Λ^{li}, $l = 1 : n_i$. The initial values of simulation state variables y^{il0} for processes Λ^{li} are selected: $y^{il0} = y^{i0}$ for $l = 0 : n_i$.

2. Generic step. By the beginning of simulation step t the current approximation to the optimal values of decision variables are x^{it} and the current states of the simulation processes are y^{ilt}. The following actions are performed at step t.

2a. SQG sends decision variables to simulation processes. The points x^{it} are sent to simulation processes Λ^{0i} and points x^{ilt}, obtained as in (17) for $s = t$ are sent to processes Λ^{li}, $l = 1 : n_i$.

2b. Observation of random parameters. The new observation ω^t of random parameters Λ^{li} common to all simulation processes is made.

2c. Simulation step. Simulation processes Λ^{li} obtain the observations $f_{ij}\left(x^i, x^{-i}, y^{it}, \omega\right)$ and update the respective state variables y^{ilt} to $y^{il,t+1}$ with equation (7), where $x^i = x^{ilt}$, $x^{-i} = x^{-it}$, $y^{it} = y^{ilt}$, $\omega = \omega^t$ for process Λ^{li}.

2d. Simulation processes send function observations to SQG. The observations $f_{ij}\left(x^{ilt}, x^{-it}, y^{it}, \omega^t\right)$ are sent to SQG.

2e. SQG updates the decision variables. SQG uses observations of $f_{ij}\left(x^{ilt}, x^{-it}, y^{it}, \omega^t\right)$ obtained from simulation processes to update the function estimates F_{ij}^t, F_{ij}^{lt} as in (11), compute stochastic gradients ξ_{ij}^t as in (22) and obtain the update of decision variables $x^{i,t+1}$ as in (28).

2f. Checking of stopping criterion. Stopping criterion is checked and if satisfied the value of $x^{i,t+1}$ or its average over all or part of preceding iterations is taken as the solution of the problem, see [10], Otherwise the computations proceed with step $t + 1$.

In this way the optimal or equilibrium values of decision parameters will be obtained during a single concurrent run of $I + \sum_i n_i$ simulations. For a finite simulation horizon as in (9) one has two choices. If T is small enough then the updates of decision parameters as in (28) can be made after the whole simulation run is performed and the values of the averaged observations as in the right hand side of (9) are sent to the SQG. If T is large then the updates can be performed after each simulated time step as in the case of the infinite time horizon described above.

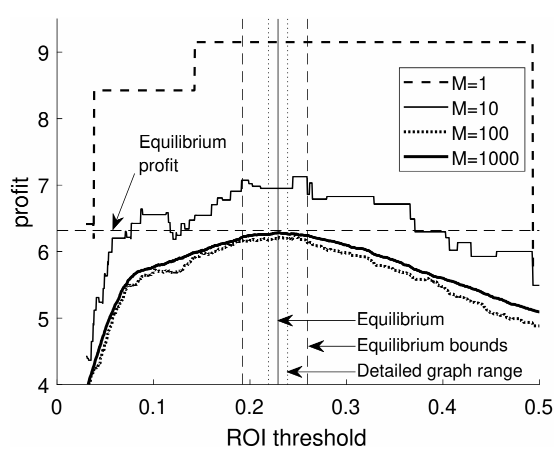

Figure 3. Sample average approximation of payoff function

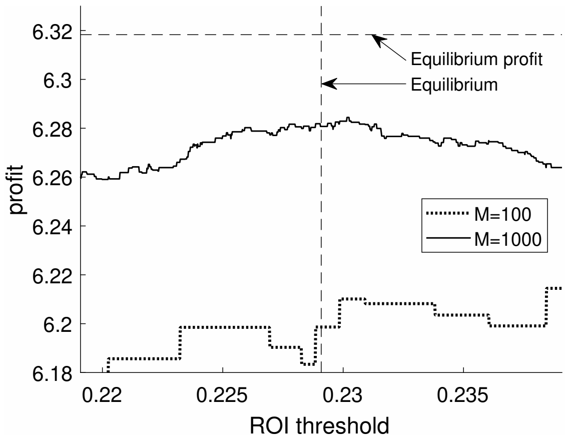

Figure 4. Sample average approximation of payoff function, magnified around equilibrium

In the case of the finite time horizon the alternative methods like nonlinear programming algorithms (NLP) or genetic algorithms encounter additional pitfalls related to the *sample average approximation* [11]. This approach generates a large sample M of observations ω^m and substitutes the original problem with expectation functions $F_{ij}\left(x^i, x^{-i}\right)$ from (9) by sample average approximation $F_{ij}^M\left(x^i, x^{-i}\right)$ of these functions:

$$F_{ij}^M\left(x^i, x^{-i}\right) = \frac{1}{M}\sum_{m=1}^{M}\frac{1}{T}\sum_{t=1}^{T} f_{ij}\left(x^i, x^{-i}, y^{itm}, \omega^m\right). \quad (34)$$

The optimal or equilibrium values of the decision variables are obtained then by solving the appropriate problem with functions $F_{ij}^M\left(x^i, x^{-i}\right)$ using a variety of methods from nonlinear optimization approaches to genetic algorithms. This approach encounters substantial difficulties in the very common case when the simulation model includes yes or no decisions. In such cases the approximate functions $F_{ij}^M\left(x^i, x^{-i}\right)$ will be discontinuous for any finite M even when the limiting functions $F_{ij}\left(x^i, x^{-i}\right)$ exhibit smooth and unimodal behavior. Figures 3,4 show one example with equilibrium problem on simulation model of energy market. The decision variables there are the thresholds, which trigger the acceptance of investment projects in energy sector: following the industry practice the project is accepted when the forecasted Return On Investment (ROI) exceeds the given threshold. It is necessary to find here the equilibrium values of the acceptance thresholds. The Figures 3,4 show that the functions $F_{ij}^M\left(x^i, x^{-i}\right)$ are discontinuous and piecewise constant for any finite M, which provide insurmountable difficulties for the NLP methods and even for genetic algorithms. The stochastic gradients succeed in finding the solution here because they address directly the expectation functions $F_{ij}\left(x^i, x^{-i}\right)$, which are smooth and well behaved.

IV. Examples

The SQG was applied to solution of complex optimization and equilibrium problems on simulation models: pricing of services on social networks [6], optimization of water resources networks [12], maritime transportation [13], energy markets [14].

V. Implementation

SQG runs on top of Matlab using descriptive language for defining the problem hierarchy (Section II) and the solver parameters. Matlab Parallel Computing Toolbox is used for parallelization.

VI. Conclusion

SQG is a powerful tool for solving a variety of stochastic optimization and equilibrium problems on realistic simulation models.

References

[1] G. C. Pflug, *Optimization of Stochastic Models. The Interface Between Simulation and Optimization*. Boston: Kluwer, 1996.

[2] A. A. Gaivoronski, "Nonstationary stochastic optimization problems," *Cybernetics and Systems Analysis*, vol. 14, no. 4, pp. 575–579, 1978.

[3] Y. Ermoliev, "Stochastic quasigradient methods and their application to system optimization," *Stochastics*, vol. 9, pp. 1–36, 1983.

[4] A. A. Gaivoronski, "SQG: Stochastic programming software environment," in *Applications of Stochastic Programming*, S. Wallace and W. Ziemba, Eds. SIAM, 2005.

[5] S. A. Zenios, *Practical Financial Optimization: Decision Making for Financial Engineers*. Blackwell Publishing, 2007.

[6] D. M. Becker and A. A. Gaivoronski, "Stochastic optimization on social networks with application to service pricing," *Computational Management Science*, vol. 11, no. 4, pp. 531–562, 2014.

[7] J. R. Birge and F. Louveaux, *Introduction to Stochastic Programming, Second Edition*. New York: Springer, 2011.

[8] H. V. Stackelberg, *The Theory of Market Economy*. London: Oxford University Press, 1952.

[9] J. Nash, "Equilibrium points in n-person games," *Proceedings of the National Academy of Sciences*, vol. 36, no. 1, pp. 48–49, 1950.

[10] A. Nemirovski, A. Juditsky, G. Lan, and A. Shapiro, "Robust stochastic approximation approach to stochastic programming," *SIAM Journal on Optimization*, vol. 19, no. 4, pp. 1574–1609, 2009.

[11] A. Shapiro and H. Xu, "Stochastic mathematical programs with equilibrium constraints, modelling and sample average approximation," *Optimization*, vol. 57, no. 3, pp. 395–418, 2008.

[12] A. A. Gaivoronski, G. Sechi, and P. Zuddas, "Balancing cost-risk in management optimization of water resource systems under uncertainty," *Physics and Chemistry of the Earth, Parts A/B/C*, vol. 42-44, pp. 98–107, 2012.

[13] P. Zuddas, M. Di Francesco, and A. Gaivoronski, "Container transportation problem under uncertain demand and weather conditions," in *13th International Conference on Stochastic Programming, book of abstracts*, 2013.

[14] A. Gaivoronski, Z. Su, and S. Harbo, "Modelling investment in energy markets: stochastic equilibrium problem defined on simulation model," in *ECSO 2017, book of abstracts*, 2017.