International Institute for
Applied Systems Analysis
IIASA www.iiasa.ac.at

Algorithmic Procedures for Stochastic Optimization

Wets, R.J.-B.

IIASA Working Paper

WP-84-049

July 1984

ALGORITHMIC PROCEDURES FOR STOCHASTIC OPTIMIZATION

Roger J.-B. Wets

July 1984
WP-84-49

INTERNATIONAL INSTITUTE FOR APPLIED SYSTEMS ANALYSIS
A-2361 Laxenburg, Austria

ALGORITHMIC PROCEDURES FOR STOCHASTIC OPTIMIZATION

Roger J.B. Wets
IIASA, A-2361 Laxenburg and
Chr. Michelsen Institute, N-5036 FANTOFT

For purposes of preliminary discussion, it is convenient to identify stochastic optimization problems with:

$$\text{find } x \in R^n \text{ that minimizes } z = E\{f(x,\tilde{\xi})\}$$

where $\tilde{\xi}$ is a random N-vector with distribution function, P, $f: R^n \times R^N \to R \cup \{+\infty\}$ is a lower semicontinuous function, possibly convex, where dom $f(.,\xi) = \{x \mid f(x,\xi) \text{ is finite}\}$, corresponds to the set of acceptable choices for x when ξ is the observed value of the random vector $\tilde{\xi}$, and

$$E\{f(x,\tilde{\xi})\} = \int f(x,\xi)\, dP(\xi).$$

To simplify matters, we may even take f finite, it will not affect much the discussion of the numerical obstacles, that must be overcome to solve stochastic optimization problems. With

$$F(x) := \int f(x,\xi)\, dP(\xi),$$

we see that our original problem is equivalent to the deterministic optimization problem:

$$\text{find } x \in R^n \text{ that minimizes } z = F(x).$$

Thus, in principle, any algorithmic procedure developed for nonlinear optimization problems could be used. In fact, we have:

$$F \text{ is convex if } f(.,\tilde{\xi}) \text{ is convex a.s.,}$$

and, assuming that dom $f(.,\xi)$ is independent of ξ, then

$$\partial F(x) = \int \partial f(x,\xi)\, dP(\xi).$$

Moreover, in general

$$\nabla F(x) = \int \nabla f(x,\xi)\, P(d\xi) \text{ if } f(.,\underset{\sim}{\xi}) \text{ is a.s. differentiable}$$

although usually F is nondifferentiable and then with an appropriate definition of the subgradient-set, we have

$$\partial F(x) \subset \int \partial f(x,\xi)\, dP(\xi)$$

Related formulas can also be found for second order derivatives, so that, in theory at least, all what we need to do is to include an integration subroutine in a standard nonlinear (nondifferentiable) optimization package to have state-of-the-art software for stochastic optimization problems. And indeed this would work very well if $\underset{\sim}{\xi}$ is a random varialbe i.e. N=1 -- excellent integration subroutines are available in this case -- or even possibly when N=2 or 3 and the analytic description of f is not too complicated. However, most applications that are of interest have N much larger that 2 or 3, in some cases all the coefficients of a given problem have stochastic components that cannot be ignored in which case N could be quite large. Except for certain specific distribution functions, such as for gamma or normal distribution functions and then with $N \leq 4$, the only known multidimensional integration subroutines available rely on Monte-Carlo techniques, involving generating pseudo- or quasi-random numbers. And for these methods to be effective we need ready access to functions values or to (sub)gradients, etc, and, as we shall see later, in stochastic optimization this is the exception rather than the rule.

Thus, in one way or another, we must design solution procedures that do not rely on multidimensional integration subroutines. Excluding certain specific classes of problems, such as stochastic programs with simple recourse and some stochastic programs with probabilistic constraints, where the properties of the problem at hand make it possible to carry out at low cost the required integration, the suggested solution strategies can be divided in two major categories: (i) "descent" methods that rely on directions determined by statistical estimates of the subgradients of F, and (ii) approximation methods that replace to original distribution function P by a discrete distribution P^{ν} involving a sufficiently small number of probability mass points, so that $\int f(x,\xi)\, dP^{\nu}(\xi)$ -- now corresponding to a finite sum -- is numerically feasible. The remainder of this presentation is devoted to a brief description of the major features these solutions procedures and of their actual or potential implementations.

A. Stochastic quasigradient methods.

Let us consider the case:

find $x \in C$ that minimizes $E\{g(x,\underset{\sim}{\xi})\}$

where

C is a closed convex set of R^n,

$g(.,\xi)$ is convex for all ξ

and the random vector $\underset{\sim}{\xi}$ is as defined above. In terms of our original formulation we would set

$$f(x,\xi) = \begin{cases} g(x,\xi) \text{ if } x \in C \\ +\infty \text{ otherwise.} \end{cases}$$

The algorithm generates a sequence $\{x^1, x^2, \ldots\}$ of points of C through the recursive formula

$$x^{\nu+1} = \text{prj}_C(x^\nu - \rho_\nu h^\nu)$$

where prj_C denotes projection on the set C, $\{\rho_\nu, \nu=1, \ldots\}$ is a sequence of scalars and

h^ν is a stochastic quasigradient of G at x^ν

with

$$G(x) := \int g(x,\xi)\, dP(\xi).$$

By stochastic quasigradient, one means a realization of a random n-vector $\underset{\sim}{h}^\nu$ satisfying

$$E\{\underset{\sim}{h}^\nu \mid x^1, \ldots, x^\nu\} \in \partial G(x^\nu).$$

Typically

$$h^{\nu} \in \partial g\ (x^{\nu}, \xi^{\nu})$$

with the $\{\xi^{\nu},\ \nu=1,\ \ldots\}$ independent random samples of $\underset{\sim}{\xi}^{\nu}$, or more

generally

$$h^{\nu} = (1/L)\ \sum_{l=1}^{L} h^{l\nu}$$

where for all l, $h^{l\nu} \in \partial g(x^{\nu},\ \xi^{\nu l})$ and the $\{\xi^{\nu l};\ l=1,\ \ldots,\ L,\ \nu=1,\ \ldots\}$ are independent random samples.

The sequence of feasible solutions $\{x^{\nu},\ \nu=1,\ ..\}$ converges with probability 1 to an optimal solution -- assuming naturally that it exists -- provided that the scalars ρ_{ν} are chosen so as to satisfy

$$\rho_{\nu} \geq 0,\ \sum_{\nu}\rho_{\nu} > \infty \text{ and } \sum_{\nu}\rho_{\nu}^{2} < \infty\ ;$$

$\rho_{\nu} = 1/\nu$ is such a sequence. The proof can be derived from a modified super-martingale convergence argument.

In the implementation of this method we must contend with three possible stumbling blocks:

the projection on C,
the choice of the step size,
the stopping criterion.

The <u>projection</u> of a point on a closed set C is easy only if C is "simple" by which we mean that C is a bounded interval, a sphere, ... The most general case that we know how to handle quite efficiently, has the set C as the intersection of a bounded interval and 1 nonlinear (or linear) constraint of the type

$$\sum_{j=1}^{n} a_j(x_j) + a_0 \leq 0$$

where a_j is convex and differentiable with $a'_j > 0$ on the bounded interval. If C is a polyhedron it may be possible to develop a technique based on the observation that for ν sufficiently large, the x^{ν} are liable to be quite close to each other and thus project in all likelihood on the same face of C. If C is a general convex set then each projection involves minimizing a

quadratic function on a convex set. For all practical purposes this complicated projection operation would make every step of the algorithm very expensive as soon as we approach the boundary of C. The objections we might have about using this method on these grounds, could be overcome by relying on penalization approximates -- see the literature on nonlinear programming -- such as for example

$$f^{\nu}(x,\xi) = g(x,\xi) + \nu \, \mathrm{dist}(x,C)$$

and h^{ν} is then a stochastic quasigradient of

$$F^{\nu}(x) = \int f^{\nu}(x,\xi) \, dP(\xi).$$

However, experimental results have shown that due to the steepness of the subgradients, penalization has a tendency to destabilize the method whenever the optimal solution lies on the boundary of C.

The choice of the step-size ρ_{ν} is in principle prescribed by the convergence requirements. However, since in practice only the short run properties of the sequence $\{x^{\nu}, \nu=1, \ldots\}$ are of interest, there is at present a gap between theory and practice where the choice of the step-size is usually guided by some adaptive rule that tries to estimate the progress made during the last M iterations, $M \geq 1$. Some preliminary results that begin to fill this gap having recently been obtained.

Finding a good stopping criterion is still very much an open question. As already mentioned earlier, in stochastic optimization problem evaluating $F = E\{f(.,\xi)\}$ may be quite expensive -- and this is why we rely on the method of stochastic quasigradients in the first place -- so it is out of question to use value comparisons between F at x^{ν} and $x^{\nu+1}$. The quantity

$$F_a(x^{\nu}) = (1/M+1) \sum_{l=\nu-M}^{M} f(x^l,\xi^l)$$

has been suggested as an estimate for $F(x^{\nu})$. The algorithm is to terminate when no improvement is observed in the value of F_a. The fact that we never really know if we have or have not reached an optimal, or nearly optimal, or sufficiently optimal solution is Achille's heel of this class of methods. Finding stopping criteria based on probabilistic error bounds, and the related question of step-size, is an area ripe for research and experimentaion.

B. Approximate solutions by discretization.

If P^ν is a distribution function that approximates the given distribution P we may hope that an optimal solution x^ν of the approximating stochastic optimization problem:

$$\text{find } x \in R^n \text{ that minimizes } F^\nu(x) = \int f(x,\xi)\, P^\nu(d\xi),$$

will be an approximate solution of the original problem. And, in fact this is the case provided f is not too exotic and P^ν is not chosen so selectively that it generates the unusual. Every distribution function P can be approximated as closely as desired by a piecewise constant distribution function P^ν which corresponds to assigning probabilities

$$p_1, p_2, \cdots, p_L$$

to a finite collection of vectors

$$\xi^1, \xi^2, \ldots, \xi^L.$$

Moreover, it can usually be shown that the approximation error, measured by the quantity

$$|F(x^\nu) - \inf F|,$$

is a function of the goodness of fit of P^ν to P, even proportional to it in the polyhedral case, i.e. when $f(.,\xi)$ is a convex piecewise linear function. If we are satisfied with an approximate solution -- and often we shall not have any alternative -- and we choose a discrete distribution P^ν close enough to P, we could solve the problem

$$\text{find } x \in R^n \text{ that minimizes } F^\nu(x) := \sum_{k=1}^{L} p_k\, f(x,\xi^k).$$

No longer is there any need for a multidimentional integration scheme, gradients and values for F^ν only involve computing a finite sum. However, we should not be lulled into believing that in this way we have licked the multidimensional integration problem. Unless $N \leq 3$, the number L of points that we need to approximate P sufficiently closely so as to guarantee an acceptable error bound

for the solution may be truly astronomical. For example if N=10 and we have 10 independent random variables taking on each 10 possible values or we have approximated each marginal distribution function by a discrete distribution with 10 density points, then L=10 billion ! Thus even if the orginal problem itself involves a discrete distribtuion we may shy away from solving such types of problems.

The alternative is to choose a very rough approximate of P involving only a small number of density points, and this even if P itself is a discrete distribution, and hope that the resulting solution x^ν is nonetheless a good approximate. This actually works (!), at least with our limited computational experience. There is actually some basic justification for this: The optimal solutions of stochastic optimization problems exhibit surprising stability properties with respect to perturbations of the distribution function of the random variables.

However, we can no longer rely on the proximity of P and P^ν to obtain error bounds, this must be obtained through other means. What could be done is to choose a pair of discrete distributions P^l and P^u in such a way that, if we solve

$$\text{find } x \in R^n \text{ that minimizes } F^l(x) := \sum_{k=1}^{L} p_k^l \, f(x,\xi^{lk})$$

we obtain a lower approximate for the original problem, and if we solve

$$\text{find } x \in R^n \text{ that minimizes } F^u(x) := \sum_{k=1}^{L'} p_k^u \, f(x,\xi^{uk})$$

we obtain an upper approximate, i.e. we have

$$\inf F^l \leq \inf F \leq \inf F^u$$

This of course gives an error estimate that can be used as a termination criterion. If we feel that the error bound provided by these approximates is not tight enough we can refine either P^l or P^u or both, to obtain a better bracketing of the optimal value. In fact we could design a solution procedure that systematically refines the approximating distribution while carrying out the steps of the algorithm.

The design of discrete distributions P^l and P^u with the desired properties, either relies on convexity or concavity properties of f(x,.), i.e. with respect

to the random parameter, or we try to identify a class of distributions Σ that contains P and such that for all x, or at least for some region in the neighborhood of an optimal solution,

$$P^l \in \operatorname{argmin}_{Q \in \Sigma} \int f(x,\xi)\, dQ(\xi),$$

and

$$P^u \in \operatorname{argmax}_{Q \in \Sigma} \int f(x,\xi)\, dQ(\xi).$$

If the distributions in Σ are restricted to a fixed compact support and we choose to define Σ as the class of distribution functions that have the same moments up to order r as P, then P^l and P^u as defined above are discrete distributions having about as many points of support as the number of moments that we want to match. The bounds obtained through convexity or concavity of $f(x,.)$ rely on Jensen's inequality and the fact that $\sup_\xi f(x,\xi)$ is attained at an extreme point of the (convex hull of) the support of P. Assuming convexity, it yields

$$f(x, E\underset{\sim}{\xi}) \leq E\{f(x,\underset{\sim}{\xi})\}$$

Thus,

$$P^l \text{ which assigns probability 1 to } E\underset{\sim}{\xi}$$

yields a lower bound since then $F^l(x) = f(x, E\underset{\sim}{\xi})$. On the other hand if

$$\xi^u \in \operatorname{argmax}\{f(x,\xi) \mid \xi \in \text{support of } P\}$$

then with

$$P^u \text{ which assigns probability 1 to } \xi^u,$$

we have an upper bound since $F^u(x) = f(x,\xi^u)$.

All these bounds can be substantially refined by partitioning the support of the distribution function P and taking conditional moments or conditional extreme points instead of moments or extreme points as here above.

All we have done so far is lay the ground work to justify limiting our attention to

$$\text{find } x \in R^n \text{ that minimizes } w = \sum_{k=1}^{L} p_k f(x,\xi^k)$$

in the development of solution techniques for stochastic optimization problems, with L relatively small, maybe a few hundreds or thousands. We could now rely on standard linear or nonlinear optimization techniques for solving this class of problems, and this would work well enough (and in some cases we actually can proceed in this manner), except that in most applications the function f is quite difficult to evaluate, the same being true about subgradients as well as other related quantities. To see this we need to examine a little bit more closely the type of functions f that we have to deal with in stochastic optimization.

As a first example, let us consider a simple version of <u>stochastic programs with probabilistic constraints</u>:

$$\text{find } x \in R_+^n \text{ such that } Ax = b,\ P(Tx) \geq \alpha$$

$$\text{and } z = cx \text{ is minimized.}$$

The constraint $P(Tx) \geq \alpha$ -- recall P is here the distribution function -- means that with probability α we want the values of $\tilde{\xi}$ to be less than Tx. The function

$$f(x,\xi) = \begin{cases} cx \text{ if } x \geq 0,\ Ax = b,\ \alpha - P(Tx) \leq 0, \\ + \text{ otherwise} \end{cases}$$

does not really depend on ξ but to check if $x \in \text{dom} f(.,\xi)$, or equivalently if x is a feasible solution of the stochastic program, we must evaluate the integral

$$\int_0^{Tx} dP(\xi),$$

which can be replaced by a finite sum to obtain upper and lower bounds. We could then refine the approximation in the neighborhood of the suspected optimal value of Tx to obtain tighter bounds. The most efficient and reliable algorithm for solving such problems appears to be a primal-dual procedure that works as follows: with

$$\rho(v) := \inf[vx \mid P(x) \geq \alpha]$$

we can show that

find $u \in R^{m1}$, $v \in R_+^{m2}$ such that $c = uA + vT$

and $w = ub + \rho(v)$ is maximized,

is dual to our stochastic program, at least when

$\{x|P(x)\geq \alpha\}$ is convex,

the function ρ is then concave. Suppose (u^k,v^k) is a feasible solution of this dual program, let

$$x^k \in \text{argmin } \rho(v^k)$$

and

$$x^k \in \text{argmin } [cx|\ Ax = b,\ Tx \geq \chi^k,\ x \geq 0]$$

This last problem is a linear program. Let $(\bar{u}^k,\bar{v}^k)$ be the simplex multipliers associated with the constraints at the optimum. If (u^k,v^k) matches $(\bar{u}^k, \bar{v}^k)$, we are done since then we satisfy the optimality conditions. Otherwise note that $(\bar{u}^k, \bar{v}^k)$ is an extreme point of the dual feasible region and the direction

$$(\bar{u}^k-u^k,\ \bar{v}^k-v^k)$$

is a direction of ascent for the dual problem. A new point (u^{k+1}, v^{k+1}) is selected between

$$(u^k, v^k) \text{ and } \bar{u}^k, \bar{v}^k)$$

that improves the dual objective, and the procedure is repeated until an optimal solution of the dual problem is reached: the corresponding x^k solves the original program. In fact only convergence can be claimed. The touchy part in this algorithm from a numerical viewpoint is the minimization of ρ which requires evaluating $P(x)$.

As a second example, we take f to be the essential objective function of a (linear) <u>stochastic program with recourse</u> (with random right hand sides), namely

$$f(x,\xi) = \begin{cases} cx + \inf_y [qy \mid Wy = \xi - Tx,\ y \geq 0] \\ \qquad \text{if } Ax = b,\ x \geq 0, \\ +\infty \text{ otherwise} \end{cases}$$

The stochastic program is a model for the following decision process: we choose an activity level x subject to certain deterministic constraints $Ax = b$, $x \geq 0$, and generate an output Tx before we can observe the value ξ of the random vector $\tilde{\xi}$. If there is any discrepancy between ξ and Tx, we make it up by selecting a recouse decision y at cost qy such that $Wy = \xi - Tx$, $y \geq 0$. The penalty for not matching exactly the random outcome ξ with the output Tx can be calculated by solving a linear programming problem. There are of course a myriad of variants of this model.

Unless the problem has specific structural properties, the standard solution procedure is a partial decomposition method to which one usually refers as the L-shaped algorithm. Let

$$\psi(x) = E\{Q(x,\xi)\} = \int Q(x,\xi)\, dP(\xi),$$

where

$$Q(x,\xi) = \inf_y [qy \mid Wy = \xi - Tx,\ y \geq 0].$$

The method consists of 3 steps that can be interpreted as follows. In step 1 we solve an approximation to

$$\text{find } x \in R_+^n \text{ with } Ax = b \text{ that minimizes } z = cx + \psi(x)$$

obtained by outer-linearization. The two types of additional linear constraints that appear in this linear program come from

(i) feasibility cuts -- generated in Step 2 -- that restrict x to the region where $\psi(x) < +\infty$, i.e. which render the recourse problem feasible for all possible values of $\tilde{\xi}$, and

(ii) optimality cuts -- generated in Step 3 -- that refine the linear approximation to ψ at least in the neighborhood of the optimal solution.

We give here a coarse version of this algorithm. At the outset set all counting parameters $\nu = s = t = 0$.

Step 1. Set $\nu = \nu+1$ and solve the linear program:

$$
\begin{aligned}
&\text{find } x \geq 0,\ \theta \in R \text{ such that}\\
&Ax \qquad = b\\
&D_k x \qquad \geq d_k,\ k=1, \ldots, s\\
&E_k x + \theta \geq e_k,\ k=1, \ldots, t\\
&cx + \theta \ = z \quad \text{is minimized}
\end{aligned}
$$

Let (x^ν, θ^ν) be an optimal solution. If there are no constraints involving θ, we set $\theta^\nu = -\infty$ and the variable θ is ignored in the linear program.

Step 2. For all possible realizations ξ of $\underset{\sim}{\xi}$ solve the linear programs

$$
\begin{aligned}
&\text{find } y \geq 0,\ v^+ \geq 0,\ v^- \geq 0 \text{ such that}\\
&Wy + Iv^+ - Iv^- = \xi - Tx^\nu \text{ and}\\
&\qquad ev^+ + ev^- = w^1 \text{ is minimized.}
\end{aligned}
$$

If for some ξ the corresponding value $w^1 > 0$, let σ^ν be the simplex multipliers associated to an optimal solution, and define

$$D_{s+1} = \sigma^\nu T, \text{ and } d_{s+1} = \sigma^\nu \xi.$$

Return to Step 1 adding this feasibility cut and set $s = s+1$, $\nu = \nu+1$. If for all possible vectors ξ, $w^1 = 0$ then go to Step 3.

Step 3. For all possible realizations ξ of $\underset{\sim}{\xi}$ solve the linear program:

$$
\begin{aligned}
&\text{find } y \geq 0 \text{ such that}\\
&Wy = \xi - Tx, \text{ and}\\
&qy = w^2 \text{ is minimized}
\end{aligned}
$$

Let $\pi(\xi)$ be the multipliers associated with the optimal solution, of course they depend on ξ. Define

$$E_{t+1} = E\{\pi(\xi)\}T$$

$$e_{t+1} = E\{\pi(\xi)\xi\}$$

and set

$$w^{2\nu} = e_{t+1} - E_{t+1}x^{\nu}.$$

If $\theta^{\nu} \geq w^{2\nu}$ we stop, x^{ν} is an optimal solution. Otherwise return to step 1 improving the outer-linearization to Φ by adding the optimality cut

$$E_{t+1}x + \theta \geq e_{t+1}$$

and set $t = t+1$, $\nu = \nu+1$.

We see that in carrying out the steps of the algorithm we are up against two major difficulties. The first one is: given $\pi(.)$ compute E_{t+1} and e_{t+1}. This has been dealt with in the earlier part of this section, and we shall assume that an acceptable discrete approximation of the distribution function P has been found that renders this calculation numerically feasible. The second difficulty is that even if the probability mass is only carried by a finite number of vectors

$$\xi^1, \ldots, \xi^L$$

and L is not too large -- of the order of 50, or 100, maybe even 1000 -- we need to solve in Step 3 L linear programs. (Step 2 allows usually for much further simplification, it often suffices to solve the linear program that appears there for a very limited number of vectors of type ξ to check if x^{ν} is feasible.)

To solve a large number of linear programs (with constant technology matrix) we rely on a discrete parametric analysis technique which goes under the name of bunching. Given x^{ν}, let B be a submatrix of W that is optimal for some $(\xi^1 - Tx^{\nu})$. Then from the optimality conditions for linear programming it follows that this basis will also be optimal for all vectors ξ^k such that

$$B^{-1}(\xi^k - Tx^{\nu}) \geq 0.$$

Since B^{-1} is already available, verifying the preceeding inequality involves substantially less work than solving a whole collection of linear programs. Moreover, because of the nature of the problem at hand it is reasonable to expect that only a small number of bases in the neighborhood of B, i.e. which can be obtained from B by a small number of pivot steps, should be sufficient to bunch all vectors $\xi^1, \ldots, \xi^L$; a bunch is the collection of vector associated to a basis by verifying if the linear inqualities above are satisfied.

Efficient bunching, the favorite one these days is a trickling down procedure that creates a tree of neighboring bases rooted at the optimal basis that corresponds to the vector $E\{\xi\}-Tx$, brings the carrying out of Step 3 of the L-shaped algorithm in the realm of possibilites. Special subroutines have been built for the case when the linear programs: for k=1, ..., L

$$\text{find } y \geq 0 \text{ with } Wy = \xi^k - Tx \text{ that minimizes } w^2 = qy$$

are transportation problems or network flow problems that are remarkably efficient, but even for general linear programs much progress has been made at the experimental level.

C. Conclusion.

I have tried to delineate the difficulties that are inherent to stochastic optimization problems and that hamper the development of efficient solution procedures. I have also suggested some strategies for the development of algorithmic procedures. Of course, I have only been able to survey a very limited corner of the on-going work. For recent results please refer to the Mathematical Programming Studies of Stochastic Programming, eds. A. Prekopa and R. Wets, that should appear in early 1985 (North-Holland). For a state-of-the-art description of computational issues in stochastic programming, please refer to the IIASA-Collaboritive Volume on "Numerical Methods in Stochastic Optimization", eds. Y. Ermoliev and R. Wets, that is now being prepared for publication. The algorithm given for stochastic programs with probabilistic constraints is due to E. Komaroni.

Acknowledgment. My views on algorithmic development for stochastic optimization problems have been shaped by too many other workers in the field to name them here, but I would like to mention Larry Nazareth (at IIASA) whose contributions and approach to software development have had a significant influence.