Pure Adaptive Search In

Monte Carlo Optimization

Nitin R. Patel
Robert L. Smith
Zelda B. Zabinsky

Pure Adaptive Search In
Monte Carlo Optimization

Nitin R. Patel
Robert L. Smith
Zelda B. Zabinsky

Abstract

Pure adaptive search constructs a sequence of points uniformly distributed within a corresponding sequence of nested regions of the feasible space. At any stage, the next point in the sequence is chosen uniformly distributed over the region of feasible space containing all points that are equal or superior in value to the previous points in the sequence. We show that for convex programs the number of iterations required to achieve a given accuracy of solution increases at most linearly in the dimension of the problem. This compares to exponential growth in iterations required for pure random search.

PURE ADAPTIVE SEARCH IN MONTE CARLO OPTIMIZATION

Monte Carlo optimization is concerned with using randomly generated points to converge to optimal solutions of mathematical programs. The simplest of Monte Carlo optimization algorithms is pure random search (Dixon and Szego [1978], Rinnooy Kan and Timmer [1984]). This algorithm generates a sequence of independent, uniformly distributed points in the feasible region and returns the best point as an approximation to the optimal solution. While the approach in itself is inefficient, it is frequently used to generate starting points for deterministic search algorithms (e.g., Boender, Rinnooy Kan, Stougie and Timmer [1982]). A great merit of pure random search is that its simplicity lends itself to theoretical analysis. A number of analytic results on convergence and performance have been obtained for it (e.g., Archetti, Betro and Steffe [1975], Clough [1965], Rubinstein [1981], De Haan [1981], Patel and Smith [1983]).

In this paper we provide a theoretical analysis of what may be considered to be the next simplest Monte Carlo algorithm after pure random search. We call this algorithm pure adaptive search. The algorithm proceeds by generating a sequence of points uniformly distributed in a sequence of nested regions of the feasible space. At any stage, the next point in the sequence is uniformly distributed over the region of feasible space containing all points that are equal or superior in value to the previous points in the sequence.

The pure adaptive search algorithm gives a surprising improvement over pure random search. For convex programs, we show that the computational complexity of pure adaptive search increases at most <u>linearly</u> in the dimension of the problem. We also show that the asymptotic distribution of improvement in objective function value using pure adaptive search is bounded from below by a lognormal distribution. Although at this time there is no known efficient implementation of pure adaptive search, any algorithm that generates solutions that can be shown to be stochastically superior to those of pure adaptive search will also share this property of linear time complexity.

1

1. Pure Adaptive Search

Consider the standard convex program,

$$\min f(x)$$

$$\text{subject to } x \in S \tag{P}$$

where x is an n-dimensional vector, S is a convex subset of R^n, and f is a real-valued convex function defined over S. We will assume that S is closed and bounded, f is bounded, and there exists a unique solution denoted x_*. Let

$$y_* = f(x_*) = \min_{x \in S} f(x) \quad \text{and}$$

$$y^* = \max_{x \in S} f(x).$$

The algorithm for solving P begins by generating a point X_1 uniformly distributed within the feasible region of P. The associated objective function value is labeled $Y_1 = f(X_1)$. The next point is generated from a uniform distribution over the convex region formed by the intersection of the feasible region with the level set of points with objective function values equal to or less than Y_1. The procedure proceeds iteratively in this fashion until a preset stopping criterion is satisfied.

More formally,

Pure Adaptive Search: Step 0. Set $k = 0$, $S_0 = S$, and $Y_0 \geq y^*$

Step 1. Generate X_{k+1} uniformly distributed in

$$S_{k+1} = \{x: \ x \in S_k \text{ and } f(x) \leq Y_k\}$$

Step 2. Set $Y_{k+1} = f(X_{k+1})$. If stopping criterion is met, stop. Otherwise set $k = k+1$ and return to Step 1.

By construction, Y_k is a decreasing sequence of points that almost surely converges to the global minimum y_*. Although obvious here, a general approach to formally demonstrating such convergence is given in Solis and Wets [1981].

2

However, the more interesting question is the performance of the algorithm as it converges to y_*. This issue will be addressed in the next section.

The principal computational effort of the algorithm lies in generating a point uniformly distributed in the improving convex set, as described in Step 1. This is a challenging problem that has as yet yielded no satisfactory solution. Conventional approaches include the rejection and transformation techniques (see Rubinstein [1981], Schmeiser [1981]). The procedure combining the two techniques is to first enclose the set in a simpler region, say a sphere. One then generates a series of points uniformly distributed in the enclosing region by efficient transformational procedures until a point falls in the enclosed region. The accepted point will then be uniformly distributed within the enclosed region. Although the problem of efficiently generating many points uniformly distributed within a single bounded region has recently met with some success (see Smith [1984]), the problem of efficiently generating one uniformly distributed point in each of many bounded regions is still unsolved. Still if an algorithm can generate random points (uniform or not) with associated objective function values stochastically less than those of uniformly distributed points, then the performance of the new algorithm would be bounded by the performance of the pure adaptive search algorithm. It is here in non-linear convex optimization where the proposed pure adaptive search algorithm and its analysis have potential value.

.2.   A Stochastically Worst Case Problem

The first step in the analysis of pure random search is to identify a stochastically worst case convex program. We first define a conical convex program and then show in Lemma 2.1 and Theorem 2.2 that it represents a stochastically worst case for the algorithm.

Consider a standard convex program P. We formally define a conical convex program Q corresponding to the convex program P as follows:

$$\min \ g(x)$$
$$\text{subject to } x \in S \qquad\qquad (Q)$$

where S is the same feasible region as in P, and $g(x) = \inf \ [y: (x, y) \in$ convex hull of (x_*, y_*) and $(S, y^*)]$.

Note that the conical program Q has the same solution x_* and the same range in objective function values $[y_*, y^*]$ as P. Also note that the objective function $g(x)$ has a conical shape, and hence the name.

In order to compare the two programs P and Q, we define Y_k^P to be the random variable representing the kth objective function value generated by the algorithm on P, and similarly, let Y_k^Q be the random variable representing the kth objective function value generated by the algorithm on Q.

Lemma 2.1: Let P be a standard convex program, and let Q be the corresponding conical program. Then P has one-step dominance over Q, that is,

$$Pr(Y_{k+1}^Q \leq z | Y_k^Q = w) \leq Pr(Y_{k+1}^P \leq z | Y_k^P = w)$$

for all $k = 0, 1, 2, \ldots$ and $y_* \leq z, w \leq y^*$.

Proof: First consider the case when $y_* \leq w \leq z \leq y^*$. Let P_z and Q_z be the level sets at objective function value z for problems P and Q respectively where $P_y = \{x: x \in S$ and $f(x) \leq y\}$. In this case $P_w \subset P_z$ and $Q_w \subset Q_z$ due to convexity, and therefore $Pr(Y_{k+1}^Q \leq z | Y_k^Q = w) = 1 = Pr(Y_{k+1}^P \leq z | Y_k^P = w)$.

Now consider the more interesting case when $y_* \leq z < w \leq y^*$. Then, clearly,

$$Pr(Y_{k+1}^Q \leq z | Y_k^Q = w) \leq Pr(Y_{k+1}^P \leq z | Y_k^P = w) \text{ if and only if}$$

$$\frac{\nu(P_z)}{\nu(P_w)} \geq \frac{\nu(Q_z)}{\nu(Q_w)} \quad \text{for all } y_* \leq z < w \leq y^*,$$

where $\nu(\cdot)$ denotes the content of the set.

In order to prove this equivalent statement, we need to introduce a similarity transformation and some additional notation. Let $\lambda_{w,z}: R^n \to R^n$ be an affine function defined by

$$\lambda_{w,z}(x) = x_* + \left(\frac{z - y_*}{w - y_*}\right)(x - x_*) \text{ for } y_* \le z < w \le y^*.$$

Let $\tilde{P}_{w,z} = \lambda_{w,z}(P_w) = \left\{ \tilde{x}: \ \tilde{x} = x_* + \left(\frac{z - y_*}{w - y_*}\right)(x - x_*), \text{ for } x \ \epsilon \ P_w\right\}$ be the

level set P_w shrunken by a factor of $\left(\dfrac{z - y_*}{w - y_*}\right)$ and rerooted at x_* so that all $\tilde{P}_{w,z}$

is contained in P_z. (See Lemma A.1 in the Appendix for a proof that $\tilde{P}_{w,z} \subset P_z$).

Similarly, let $\tilde{Q}_{w,z} = \lambda_{w,z}(Q_w)$.

Now, consider $\dfrac{\nu(Q_z)}{\nu(Q_w)}$. Since $Q_z = \tilde{Q}_{w,z}$ (see Lemma A.2 in the Appendix), we have

$\dfrac{\nu(Q_z)}{\nu(Q_w)} = \dfrac{\nu(\tilde{Q}_{w, z})}{\nu(Q_w)}$. Since $\lambda_{w, z}$ is an affine transformation,

$$\frac{\nu(\tilde{Q}_{w, z})}{\nu(Q_w)} = \left| \det \lambda_{w, z}\right|.$$

Hence $\dfrac{\nu(Q_z)}{\nu(Q_w)} = \left| \det \lambda_{w, z}\right|$ for all $y_* \le z < w \le y^*$. Similarly, consider $\dfrac{\nu(P_z)}{\nu(P_w)}$.

Since $\tilde{P}_{w,z} \subset P_z$, we have $\dfrac{\nu(P_z)}{\nu(P_w)} \ge \dfrac{\nu(\tilde{P}_{w,z})}{\nu(P_w)}$, and also $\dfrac{\nu(\tilde{P}_{w, z})}{\nu(P_w)} = \left| \det \lambda_{w, z}\right|$.

Hence $\dfrac{\nu(P_z)}{\nu(P_w)} \ge \left| \det \lambda_{w, z}\right|$ for all $y_* \le z < w \le y^*$. Consequently, $\dfrac{\nu(P_z)}{\nu(P_w)} \ge \dfrac{\nu(Q_z)}{\nu(Q_w)}$

for all $y_* \le z < w \le y^*$. ∎

Lemma 2.1 states that, for any specified objective function values, the conditional probability that a point chosen uniformly from P_w lies in the improved region P_z is greater than the conditional probability that a point chosen uniformly from Q_w lies in the improved region Q_z. This establishes that, on any specific iteration, pure adaptive search performs stochastically better in one step on P than on Q.

Theorem 2.2 states that the entire sequence of points generated by the algorithm on P are stochastically better (less) than the sequence of points generated by the algorithm on Q.

Theorem 2.2: $\{Y_k^Q\} \geq st\{Y_k^P\}$ for all k = 1, 2,... .

Proof: The proof is by induction on k. We show that $Pr(Y_k^Q > z) \geq Pr(Y_k^P > z)$ for all k = 1, 2,... and $z \in [y_*, y^*]$. Note that

$$Pr(Y_1^P \leq z) = Pr(Y_1^P \leq z | Y_0^P = y^*) \text{ and}$$

$$Pr(Y_1^Q \leq z) = Pr(Y_1^Q \leq z | Y_0^Q = y^*)$$

by the definition of the algorithm. But then $Pr(Y_1^Q \leq z) \leq Pr(Y_1^P \leq z)$ for all $y_* \leq z \leq y^*$ follows immediately from Lemma 2.1. Hence $Y_1^Q \geq st Y_1^P$.

We assume the induction hypothesis that $Y_k^Q \geq st Y_k^P$ for some $k \geq 1$, and show that $Y_{k+1}^Q \geq st Y_{k+1}^P$. By Lemma 2.1, we have $Pr(Y_{k+1}^Q \leq z | Y_k^Q = w) \leq Pr(Y_{k+1}^P \leq z | Y_k^P = w)$ for all z, $w \in [y_*, y^*]$.

For ease of notation, for any fixed k and $z \in [y_*, y^*]$, let q(w) = $Pr(Y_{k+1}^Q > z | Y_k^Q = w)$ and p(w) = $Pr(Y_{k+1}^P > z | Y_k^P = w)$. Since q(w) is a non-decreasing function, we have $E(q(Y_k^Q)) \geq E(q(Y_k^P))$ (see Ross [1983], p. 252). Also since q(w) $\geq$ p(w), we have $E(q(Y_k^P)) \geq E(p(Y_k^P))$. Therefore $E(q(Y_k^Q)) \geq E(p(Y_k^P))$. But, $E[q(Y_k^Q)] = E[Pr(Y_{k+1}^Q > z | Y_k^Q)] = Pr(Y_{k+1}^Q > z)$, and similarly, $E[p(Y_k^P)] = Pr(Y_{k+1}^P > z)$. Therefore $Pr(Y_{k+1}^Q > z) \geq Pr(Y_{k+1}^P > z)$ for all $y_* \leq z \leq y^*$. ■

Theorem 2.2 states that the objective function values generated on Q are stochastically larger than those generated on P. We can conclude that a conical convex program is a stochastically worse case than its corresponding convex program. We now turn to the analysis of performance of the pure adaptive search algorithm on the conical program.

3. Analytic Performance Bounds for Pure Adaptive Search

The second part of the analysis is to measure performance of the pure adaptive search algorithm on the conical convex program and in this manner obtain bounds on performance for all convex programming.

In this section we define a measure of complexity that is appropriate for a stochastic algorithm, and then provide two computational complexity results. Analyzing the performance of the algorithm for this conical convex program then

provides a worst case distribution for the number of iterations necessary to achieve a desired accuracy of solution. The first result gives the probability distribution for the quantity of improvement per iteration, and shows that the asymptotic minimum (scaled) objective function value obtained from pure adaptive search is stochastically bounded from above by a lognormal distribution. The second result is that the necessary number of iterations grows at most linearly in the dimensionality of the problem.

In order to compare performance of the algorithm across convex programs, we define a standardized measure of improvement. Given any feasible objective function value y of P where $y_* \leq y \leq y^*$, define the standardized objective function value of y, as

$$z = \frac{(y - y_*)}{(y^* - y_*)}.$$

Let Z_k^P be the standardized objective function value corresponding to the kth point X_k^P generated by the algorithm on P, $Z_k^P = \frac{f(X_k^P) - y_*}{y^* - y_*}$. A standardized value of $z = \frac{1}{m}$ where m is a positive integer will be referred to as an m-fold improvement. We say that the algorithm achieves m-fold improvement by the kth iteration if $Z_k^P \leq \frac{1}{m}$, or $\frac{y^* - y_*}{Y_k^P - y_*} \geq m$. Notice that while the objective function y ranges between $[y_*, y^*]$, the standardized value of improvement of ranges between $[0, 1]$.

We now define a measure of performance that will be evaluated for the pure adaptive search algorithm. Let $K_{\alpha,m}^P$ be the number of iterations of the pure adaptive search algorithm applied to P necessary to ensure an m-fold improvement with probability at least $1 - \alpha$. Formally, $K_{\alpha,m}^P = \min\{k: \Pr(Z_k^P \leq \frac{1}{m}) \geq 1 - \alpha\}$. Let $K_{\alpha,m}^Q$ be the same quantity applied to the conical convex program Q.

The following result states that the conical convex program provides a worst case distribution for standardized improvement and thus provides an upper bound on the number of iterations necessary to achieve a desired accuracy of solution. It follows as an immediate corollary to Theorem 2.2.

7

<u>Corollary 3.1</u>: $\{Z_k^P\} \le st\{Z_k^Q\}$ for all $k = 1, 2,\ldots,$ and $K_{\alpha,m}^P \le K_{\alpha,m}^Q$ for all

$m > 1$, $0 < \alpha < 1$.

We now turn to an analysis of algorithm performance on conical convex programs

to establish worst case performance criteria.

Our first computational complexity result characterizes the probability

distribution of improvement on the conical program. The result, stated below in

Theorem 3.2, is that the asymptotic distribution of Z_k^Q is lognormal.

<u>Theorem 3.2</u>: Let Z_k^Q be the random variable corresponding to the standardized

objective function value on the kth iteration for a conical convex program Q. Let

$Z_0^Q = 1$. Then $\qquad\qquad \mu_k = E(Z_k^Q) = \left(\dfrac{n}{n+1}\right)^k$ and

$\sigma_k^2 = Var\ (Z_k^Q) = \left(\dfrac{n}{n+2}\right)^k \left(1 - \left(\dfrac{n(n+2)}{(n+1)(n+1)}\right)^k\right)$ for all k, with $Z_k^Q \sim$ lognormal (μ_k, σ_k^2) for large k.

<u>Proof</u>: First we drop the superscript Q with the understanding that this proof is

for the conical program Q. Then $Z_k = Y_k\ Y_{k-1}\ \cdots\ Y_1$ where $Y_k = \dfrac{Z_k}{Z_{k-1}}$.

The Y_k are independent identically distributed random variables with mean $\dfrac{n}{n+1}$

and variance $\dfrac{n}{(n+2)(n+1)^2}$ (See Appendix, Lemma A.3). Thus $E(Z_k) = \left(\dfrac{n}{n+1}\right)^k$,

$E(Z_k^2) = \left(\dfrac{n}{n+2}\right)^k$, and hence $Var(Z_k) = \left(\dfrac{n}{n+2}\right)^k \left(1 - \left(\dfrac{n(n+2)}{(n+1)^2}\right)^k\right)$. It remains to

show that Z_k approaches a lognormal distribution. Since Z_k is the product of

independent identically distributed random variables, we can write $\ln Z_k = \sum\limits_{i=1}^{k} \ln Y_i$,

and using the central limit theorem we get that $\ln Z_k$ approaches a normal

distribution for large k, or $Z_k \sim$ lognormal (μ_k, σ_k^2) for large k. ∎

Our second result gives an upper bound on $K_{\alpha,m}^Q$ and hence $K_{\alpha,m}^P$.

<u>Theorem 3.3</u>: $K_{\alpha,m}^P \le 2(n+1)\ \ln(m(1+1/\sqrt{\alpha}))$ for all $m > 1$ and $0 < \alpha < 1$.

<u>Proof</u>: From Chebyshev's inequality

$$Pr\left(E(Z_k) - (1/\sqrt{\alpha})\sigma(Z_k) \le Z_k \le E(Z_k) + (1/\sqrt{\alpha})\sigma(Z_k)\right) \ge 1 - \alpha$$

where Z_k is the standardized objective function value on the kth iteration for the

conical program Q. For m-fold improvement, we should have $Pr(Z_k \le \frac{1}{m}) \ge 1 - \alpha$.

From Theorem 3.2, $E(Z_k) = (n/n+1)^k$ and $\sigma(Z_k) = \left(\dfrac{n}{n+2}\right)^{k/2}\left[1 - \left(\dfrac{n^2+2n}{n^2+2n+1}\right)^k\right]^{1/2}$.

Combining all of these, we find that at most k iterations are required for an m-fold improvement where k is the smallest integer that satisfies

$$h(k) \equiv \left(\frac{n}{n+1}\right)^k + \left(\frac{1}{\sqrt{\alpha}}\right)\left(\frac{n}{n+2}\right)^{k/2}\left[1 - \left(\frac{n^2+2n}{n^2+2n+1}\right)^k\right]^{1/2} \leq \frac{1}{m}.$$

Let $g(k) = \left(1 + \dfrac{1}{\sqrt{\alpha}}\right)\left(\dfrac{n}{n+1}\right)^{k/2}$. Note that (i) $g(k)$ is monotone decreasing in k for $k > 0$; (ii) $g(k) > h(k)$ since

$$g(k) \geq \left(\frac{n}{n+1}\right)^k + \frac{1}{\sqrt{\alpha}}\left(\frac{n}{n+1}\right)^{k/2} \geq \left(\frac{n}{n+1}\right)^k + \frac{1}{\sqrt{\alpha}}\left(\frac{n}{n+2}\right)^{k/2} > h(k).$$

From (i) and (ii) it is clear that if k' is an integer satisfying $g(k') \leq \dfrac{1}{m}$, then all integers k greater than or equal to k' satisfy $h(k) \leq \dfrac{1}{m}$ and thus k' provides an upper bound on $K^P_{\alpha,m}$. Since $\dfrac{2\ln\left(m(1 + 1/\sqrt{\alpha})\right)}{\ln(1 + \frac{1}{n})}$ is a root of $g(x) = \dfrac{1}{m}$, we may take k' equal to the smallest integer greater than or equal to this root. Therefore, since $\ln(1 + \dfrac{1}{n}) \geq \dfrac{1}{n+1}$ for $n \geq 1$, $K^P_{\alpha,m} \leq \dfrac{2\ln\left(m(1 + 1/\sqrt{\alpha})\right)}{\ln(1 + 1/n)} \leq 2(n+1)\ln\left(m(1 + \dfrac{1}{\sqrt{\alpha}})\right)$. ■

The importance of a bound for $K^P_{\alpha,m}$ lies not only in providing a stopping criterion for terminating the algorithm, but it also has implications on the computational effectiveness of the algorithm.

Theorem 3.3 states that the number of iterations needed for a worst case situation will at most grow linearly in the number of dimensions. In fact, since this is not a tight upper bound, the performance is better than linear. Table 1 gives upper bounds on performance versus dimension and for a million fold improvement with 99% certainty.

The linear bound on complexity for pure adaptive search is perhaps a surprising improvement over the exponential complexity of pure random search. It suggests that steady improvement is an important feature for an algorithm (Adler [1983]), and that even a "blind" search using a uniform distribution yields good performance. If another algorithm could be shown to be stochastically better than the pure adaptive search algorithm, then the same analysis and results would apply.

In fact, other random search optimization algorithms have experimentally witnessed linear results on convex programs such as in Solis and Wets [1981], Schumer and Steiglitz [1968], and Schrack and Borowski [1972]. These findings suggest possible extension of the pure adaptive search algorithms' linear behavior to more practical algorithms.

Dimension n	Upper Bound for $K^P_{\alpha,m}$ with $\alpha = .01$, $m = 10^6$ (number of iterations)
1	65
2	98
5	195
10	357
50	1,654
100	3,276
500	16,246
1,000	32,460
5,000	162,167
10,000	324,301

Table 1. An upper bound for $K^P_{\alpha,m}$ versus

dimension for convex programs for

million fold improvement with 99% certainty.

4. Conclusion

We have shown that pure adaptive search greatly improves on pure random search. Iterations for pure adaptive search are linear in the dimension of convex programs.

However, at the present time we do not have efficient procedures for generating uniformly distributed points in general convex regions. For this reason, despite its substantial improvement over pure random search, pure adaptive search cannot compete with existing algorithms for the general case. For special regions it can be more effective. More importantly, our analysis clearly indicates that simple linear-time Monte-Carlo algorithms for convex optimization are possible if we can develop efficient procedures to generate points uniformly distributed over convex regions.

References

1. Adler, I. [1983], Private communication with R. L. Smith, University of California, Berkeley, California.

2. Archetti, F., B. Betro and S. Steffe [1975], "A Theoretical Framework for Global Optimization via Random Sampling", Quaderno dei gruppi di recerca matematica del C.N.R., University of Pisa.

3. Boender, C. G. E., A. H. G. Rinnooy Kan, L. Stougie and G. T. Timmer [1982], "A Stochastic Method for Global Optimization", Mathematical Programming, Vol. 22, pp. 125-140.

4. Clough, D. J. [1965], "An Asymptotic Extreme-Value Sampling Theory for Estimation of a Global Maximum", CORS Journal, Vol. 7, pp. 96-109.

5. De Haan, L. [1981], "Estimation of the Minimum of a Function Using Order Statistics", Journal of the American Statistical Association, Vol. 76, pp. 467-469.

6. Dixon, L. C. W. and G. P. Szego [1978], eds., Towards Global Optimization 1 and 2, North-Holland, Amsterdam.

7. Patel, N. R. and R. L. Smith [1983], "The Asymptotic Extreme Value Distribution of the Sample Minimum of a Concave Function Under Linear Constraints", Operations Research, Vol. 31, pp. 789-794.

8. Rinnooy Kan, A. H. G. and G. T. Timmer [1984], "Stochastic Methods for Global Optimization", American Journal of Mathematical and Management Sciences, Vol. 4, pp. 7-39.

9. Ross, S. M. [1983], Stochastic Processes, John Wiley and Sons, New York.

10. Rubinstein, R. Y. [1981], Simulation and the Monte Carlo Method, John Wiley and Sons, New York.

11. Schmeiser, B. W. [1981], "Random Variate Generation: A Survey", Simulation With Discrete Methods: A State of the Art View, T. I. Oren, C. M. Shab, P. F. Roth (eds.), IEEE, pp. 79-104.

12. Schrack, G. and N. Borowski [1972], "An Experimental Comparison of Three Random Searches", Numerical Methods for Non-linear Optimization, F. Lootsma, ed., pp. 137-147, Academic Press, London.

13. Schumer, M. A. and K. Steiglitz [1968], "Adaptive Step Size Random Search", IEEE Trans. Automatic Control AC-13, pp. 270-276.

14. Smith, R. L. [1984], "Efficient Monte Carlo Procedures for Generating Points Uniformly Distributed Over Bounded Regions", Operations Research, Vol. 32, pp. 1296-1308.

15. Solis, F. J. and R. J. Wets [1981], "Minimization by Random Search Techniques", Mathematics of Operations Research, Vol. 6, pp. 19-30.

Lemma A.1: For any convex program P, the shrunken set $\tilde{P}_{w,z} = \lambda_{w,z}(P_w)$ is contained in its level set P_z;

$$\tilde{P}_{w,z} \subset P_z \text{ for all } y_* \leq z < w \leq y^*.$$

Proof: Let $\tilde{x} \in \tilde{P}_{w,z}$. By the definition of $\tilde{P}_{w,z} = \lambda_{w,z}(P_w)$, there exists $x \in P_w$ such that

$$\tilde{x} = \lambda_{w,z}(x) = x_* + \left(\frac{z - y_*}{w - y_*}\right)(x - x_*).$$

Therefore, since f is a convex function,

$$f(\tilde{x}) \leq \left(1 - \left(\frac{z - y_*}{w - y_*}\right)\right) f(x_*) + \left(\frac{z - y_*}{w - y_*}\right) f(x).$$

We have that $f(x_*) = y_*$ and $f(x) \leq w$ since $x \in P_w$, hence

$$f(\tilde{x}) \leq \left(\frac{w - z}{w - y_*}\right) y_* + \left(\frac{z - y_*}{w - y_*}\right) w = \frac{z(w - y_*)}{w - y_*} = z.$$

Therefore $f(\tilde{x}) \leq z$, and $\tilde{x} \in S$ by convexity, giving the result, $\tilde{x} \in P_z$ for all $y_* \leq z < w \leq y^*$. ∎

Lemma A.2: For any conical convex program Q, the shrunken set $\tilde{Q}_{w,z} = \lambda_{w,z}(Q_w)$, is identical to its level set Q_z;

$$\tilde{Q}_{w,z} = Q_z \text{ for all } y_* \leq z < w \leq y^*.$$

Proof: i) First, $\tilde{Q}_{w,z} \subset Q_z$ for all $y_* \leq z < w \leq y^*$ by Lemma A.1 because Q is a convex program. ii) Now, we show that $Q_z \subset \tilde{Q}_{w,z}$. If $z = y_*$, then $Q_z = \{x_*\} = \tilde{Q}_{w,z}$ by definition. Consider $y_* < z < w \leq y^*$, and let $\tilde{x} \in Q_z$ with $\tilde{y} = g(\tilde{x})$. Define $x' = \tilde{x}\left(\frac{w - y_*}{z - y_*}\right) - x_*\left(\frac{w - z}{z - y_*}\right)$ for any $y_* < z < w \leq y^*$. Note that

$$\lambda_{w,z}(x') = x_* + \left(\frac{z - y_*}{w - y_*}\right)(x' - x_*), \text{ and substituting in } x',$$

$$= x_* + \left(\frac{z - y_*}{w - y_*}\right)\left[\tilde{x}\left(\frac{w - y_*}{z - y_*}\right) - x_*\left(\frac{w - z}{z - y_*}\right) - x_*\right]$$

$$= \tilde{x} + x_*\left[\left(\frac{w - y_*}{w - y_*}\right) - \left(\frac{w - z}{w - y_*}\right) - \left(\frac{z - y_*}{w - y_*}\right)\right]$$

$$= \tilde{x}.$$

Since $\tilde{x} = \lambda_{w,z}(x')$, if $x' \in Q_w$ we would have $\tilde{x} \in \tilde{Q}_{w,z}$. It remains to show that $x' \in Q_w$.

Since $(\bar{x}, \bar{y}) \epsilon$ convex hull of (x_*, y_*) and $\{(x, y^*): x \epsilon S\}$, we know that

there exists $\bar{x} \epsilon S$ such that $(\bar{x}, \bar{y})$ is on the line segment connecting (x_*, y_*) and

$(\bar{x}, y^*)$; $\bar{x} = (1 - r)x_* + r\bar{x}$ where $r = \dfrac{\bar{y} - y_*}{y^* - y_*}$ $(o < r < 1)$. By our definition of

x', we have $x' = (1 - s)x_* + s\bar{x}$ where $s = \dfrac{w - y_*}{z - y_*}$ and $s > 1$. Substituting in for $\bar{x}$;

$$x' = (1 - s)x_* + s(1 - r)x_* + sr\,\bar{x}$$

$$= (1 - sr)x_* + sr\,\bar{x}.$$

Now $sr = \left(\dfrac{w - y_*}{z - y_*}\right)\left(\dfrac{\bar{y} - y_*}{y^* - y_*}\right)$, and $y_* \leq \bar{y} \leq z$, hence

$sr \leq \left(\dfrac{w - y_*}{z - y_*}\right)\left(\dfrac{z - y_*}{y^* - y_*}\right) = \dfrac{w - y_*}{y^* - y_*} \leq 1$. Therefore $x' \epsilon S$ because it is on a line

segment connecting x_* and $\bar{x}$, and S is convex. Now, $g(x') \leq (1 - sr)\,g(x_*) +$

$sr\,g(\bar{x})$, since $g(x)$ is a convex function. Hence $g(x') \leq (1 - sr)y_* + sr\,y^* \leq w$ as

is easily shown. Since $x' \epsilon S$, we have $x' \epsilon Q_w$. ∎

Lemma A.3: Let $Y_k = \dfrac{Z_k}{Z_{k-1}}$ be the ratio of successive improvements on a conical

program Q. Then $\{Y_k, k = 1, 2, ...\}$ are independent identically distributed random

variables. Also, for $k = 1, 2, ...$

$$E[Y_k] = \frac{n}{n+1}$$

$$E[Y_k^2] = \frac{n}{n+2}$$

Proof: Now $\Pr(Y_k \leq y) = E_{Z_{k-1}}\left(\Pr(Z_k \leq y \; Z_{k-1} | Z_{k-1})\right)$. Since $Z_k = \dfrac{Y_k - y_*}{y^* - y_*}$, the

conditional distribution of Z_k given Z_{k-1} is

$$\Pr(Z_k \leq z | Z_{k-1} = w) = \Pr(Y_k \leq z' | Y_{k-1} = w')$$

where $z' = z(y^* - y_*) + y_*$, $w' = w(y^* - y_*) + y_*$. When $0 \leq z < w \leq 1$,

$$\Pr(Y_k \leq z' | Y_{k-1} = w') = \frac{\nu(Q_{z'})}{\nu(Q_{w'})} = |\det \lambda_{w',z'}|$$

$$= \left(\frac{z' - y_*}{w' - y_*}\right)^n = \left(\frac{z}{w}\right)^n,$$

15

since Q is a conical convex program in n dimensions. Therefore,

$$\Pr(Z_k \le z \,|\, Z_{k-1} = w) = \begin{cases} \left(\dfrac{z}{w}\right)^n & \text{for } z < w, \ 0 \le w, \ z \le 1 \\[2ex] 1 & \text{for } z \ge w, \ 0 \le w, \ z \le 1. \end{cases}$$

Let $f_{Z_{k-1}}(x)$ be the density function for Z_{k-1}. Then

$$\Pr(Y_k \le y) = \int_0^1 \Pr(Z_k \le xy \,|\, Z_{k-1} = x) \, f_{Z_{k-1}}(x) \, dx$$

$$= \int_0^1 \left(\frac{xy}{x}\right)^n f_{Z_{k-1}}(x) \, dx$$

$$= y^n \int_0^1 f_{Z_{k-1}}(x) \, dx$$

$$= y^n \text{ for } 0 \le y \le 1.$$

Thus, the random variables Y_k, for $k = 1, 2, \ldots$ are identically distributed.

To establish the independence of the Y_k, we need to show that $\Pr(Y_{k+1} \le y \,|\, Y_k, \ldots, Y_1) = \Pr(Y_{k+1} \le y))$ for all y and k. This immediately implies that $Y_1, Y_2, \ldots$ are independent random variables, i.e., for all k,

$$\Pr(Y_{k+1} \le y, \ Y_k \le y_k, \ldots, Y_1 \le y_1) = \Pr(Y_{k+1} \le y) \, \Pr(Y_k \le y_k) \cdots \Pr(Y_1 \le y_1).$$

Now, for $0 \le y \le 1$ and $0 \le y_i \le 1, i = 1, 2, \ldots, k$ and for all k,

$$\Pr(Y_{k+1} \le y \,|\, Y_k = y_k, \ldots, Y_1 = y_1)$$

$$= \Pr(Z_{k+1} \le y Z_k \,|\, Z_k = y_k \, y_{k-1} \cdots y_1, \ldots, Z_1 = y_1)$$

$$= \Pr(Z_{k+1} \le y \, y_k y_{k-1} \cdots y_1 \,|\, Z_k = y_k y_{k-1} \cdots y_1, \ldots, Z_1 = y_1)$$

$$= \Pr(Z_{k+1} \le y y_k y_{k-1} \cdots y_1 \,|\, Z_k = y_k y_{k-1} \cdots y_1) \text{ since } Z_{k+1} \text{ is conditionally}$$
independent of $Z_{k-1}, \ldots, Z_1$ given Z_k

$$= (y y_k y_{k-1} \cdots y_1 / y_k y_{k-1} \cdots y_1)^n$$
$$= y^n$$
$$= \Pr(Y_{k+1} \le y).$$

Finally the rth moment of Y_k can be easily determined:

$$E[Y_k^r] = \int_0^1 y^r n y^{n-1} dy$$

$$= n \int_0^1 y^{r+n-1} dy$$

$$= \frac{n}{r+n}.$$

In particular,

$$E[Y_k] = \frac{n}{n+1}$$

$$E[Y_k^2] = \frac{n}{n+2}. \quad \blacksquare$$