Variance Reduction in Population-Based Optimization: Application to Unit Commitment

Jérémie Decock, Jialin Liu, Olivier Teytaud
TAO, Inria, Univ. Paris-Sud, UMR CNRS 8623
Bat 660 Claude Shannon Univ. Paris-Sud, 91190 Gif-sur-Yvette, France
{*firstname.lastname*}*@inria.fr*

ABSTRACT

We consider noisy optimization and some traditional variance reduction techniques aimed at improving the convergence rate, namely (i) common random numbers (CRN), which is relevant for population-based noisy optimization and (ii) stratified sampling, which is relevant for most noisy optimization problems. We present artificial models of noise for which common random numbers are very efficient, and artificial models of noise for which common random numbers are detrimental. We then experiment on a desperately expensive unit commitment problem. As expected, stratified sampling is never detrimental. Nonetheless, in practice, common random numbers nonetheless provided, by far, most of the improvement.

Keywords

Noisy Optimization; Variance Reduction; Stratified Sampling; Common Random Numbers

1. ALGORITHMS

1.1 Different forms of pairing

For each request x_n to the objective function oracle, the algorithm also provides a set $Seed_n$ of random seeds; $Seed_n = \{seed_{n,1}, \ldots, seed_{n,m_n}\}$. $\mathbb{E}f(x_n, w)$ is then approximated as $\frac{1}{m_n}\sum_{i=1}^{m_n} f(x_n, seed_{n,i})$.

One can see in the literature different kinds of pairing. The simplest one is as follows: all sets of random seeds are equal for all search points evaluated during the run, i.e. $Seed_n$ is the same for all n. The drawback of this approach is that it relies on a sample average approximation: the good news is that the objective function becomes deterministic; but the approximation of the optimum is only good up to the relevance of the chosen sample and we can not guarantee convergence to the real optimum. Variants consider m_n increasing and nested sets $Seed_n$, such as $\forall(n \in \mathbb{N}^+, i \leq m_n)$, $m_{n+1} \geq m_n$ and $seed_{n,i} = seed_{n+1,i}$. A more sophisticated version is that all random seeds are equal inside an offspring, but they are changed between offspring (see discussion above). We will test this, as an intermediate step between CRN and no pairing at all. In Section 1.2, we explain on an illustrative example why in some cases, pairing can be detrimental. It might therefore make sense to have partial pairing. In order to have the best of both worlds, we propose in Section 1.3 an algorithm for switching smoothly from full pairing to no pairing at all.

1.2 Why common random numbers can be detrimental

The phenomenon by which common random numbers can improve convergence rates is well understood; correlating the noise between several points tends to transform the noise into a constant additive term, which has therefore less impact - a perfectly constant additive term has (for most algorithms) no impact on the run. Setting $\alpha = 1$ in Eq. 1 (below), modelizing an objective function, provides an example in which pairing totally cancels the noise.

$$f(x, w) = ||x||^2 + \alpha w' + 20(1 - \alpha)w'' \cdot x \tag{1}$$

We here explain why CRN can be detrimental on a simple illustrative example. Let us assume (toy example) that

- We evaluate an investment policy on a wind farm.
- A key parameter is the orientation of the wind turbines.
- A crucial part of the noise is the orientation of wind.
- We evaluate 30 different individuals per generation, which are 30 different policies - each individual (policy) has a dominant orientation.
- Each policy is evaluated on 50 different simulated wind events.

With CRN: If the wind orientation (which is randomized) was on average more East than it would be on expectation, then, in case of pairing (i.e. CRN), this "East orientation bias" is the same for all evaluated policies. As a consequence, the selected individuals are more East-oriented. The next iterate is therefore biased toward East-oriented.

Without CRN: Even if the wind orientation is too much East for the simulated wind events for individual 1, such a bias is unlikely to occur for all individuals. Therefore, some individuals will be selected with a East orientation bias, but others with a West orientation bias or other biases. As a conclusion, the next iterate will incur an average of many uncorrelated random biases, which is therefore less biased.

GECCO '15 July 11-15, 2015, Madrid, Spain

ACM ISBN 978-1-4503-3488-4/15/07.
DOI: http://dx.doi.org/10.1145/2739482.2764638

1.3 Proposed intermediate algorithm

We have seen that pairing can be efficient or detrimental depending on the problem. We will here propose an intermediate algorithm (Algorithm 1), somewhere in between the paired case ($g(r) = r$) and the totally unpaired case ($g(r) >> r$).

Algorithm 1 One iteration of a population-based noisy optimization algorithm with pairing.

Require: A population-based noisy optimization algorithm (in particular, rule for generating offspring)
Require: n: current iteration number
Require: $r \in \mathbb{N}^+$: a resampling rule
Require: λ: a population size
Require: $g : \mathbb{N}^+ \to \mathbb{N}^+$: a non-decreasing mapping such that $g(r) \geq r$

1: Generate λ individuals $i_1, \ldots, i_\lambda$ to be evaluated at this iteration
2: Compute the resampling number r by the resampling rule
3: Generate $P_{r,g(r)} = (w_{r,1}, \ldots, w_{r,g(r)})$ a set of $g(r)$ random seeds (we will see below different rules)
4: Each of these λ individuals is evaluated r times with r distinct random seeds randomly drawn in the family $P_{r,g(r)}$.

The $P_{r,g(r)}$ can be

- Nested, i.e. $\forall (i,r),\ g(r) \geq i \Rightarrow w_{r,i} = w_{r+1,i}$. The $(w_{r,i})_{i \leq g(r)}$ for a fixed r are then independent.
- Independent, i.e. all the $w_{r,i}$ are randomly independently identically drawn.

SAA is equivalent to the nested case with $n \mapsto r(n)$ constant, i.e. we always use the same set of random seeds. [1] corresponds to the nested case. Classical CRN consists in $g(r) = r$ and independent sampling.

We will design, in Section 2, an artificial testbed which smoothly (parametrically depending on α in Eq. 1) switches

- from an ideal case for pairing (testbed in which pairing cancels the noise, as $\alpha = 1$ in Eq. 1);
- to worst case for pairing (counterexample as illustrated above, Section 1.2).

and which (depending on $g(\cdot)$) switches from fully paired to fully independent. We will compare stratified sampling and paired sampling on this artificial testbed. Later, we will consider a realistic application (Section 3).

2. ARTIFICIAL EXPERIMENTS

All experiments are reported to the extended version of the present paper.

3. REAL WORLD EXPERIMENTS

All experiments are reported to the extended version of the present paper.

4. CONCLUSIONS

We tested, in an artificial test case and a Direct Policy Search problem in power management, paired optimization (a.k.a common random numbers) and partial variants of it. We also tested stratified sampling. Both algorithms are easy to implement, "almost" black-box and applicable for most applications. Paired optimization is unstable; it can be efficient in simple cases, but detrimental with more difficult models of noise, as shown by results on $\alpha = 1$ (positive effect) and $\alpha = 0$ (negative effect) in the artificial case (Eq. 1). We provided illustrative examples of such a detrimental effect (Section 1.2). Stratification had sometimes a positive effect on the artificial test case and was never detrimental. Nonetheless, on the realistic problem, pairing provided a great improvement, much more than stratification. Pairing and stratification are not totally black box; however, implementing stratification and pairing is usually easy and fast and we could do it easily on our realistic problem. We tested an intermediate algorithm with a parameter for switching smoothly from fully paired noisy optimization to totally unpaired noisy optimization. However, this parametrized algorithm (intermediate values of β) was not clearly better than the fully unpaired algorithm ($\beta = \infty$). It was not more robust in the case $\alpha = 0$, unless β is so large that there is essentially no pairing at all. As a conclusion, we firmly recommend common random numbers for population-based noisy optimization. Realistic counter-examples to CRN's efficiency would be welcome - we had such detrimental effects only in artificially built counter-example. There are probably cases (e.g. problems with rare critical cases) in which stratification also helps a lot, though this was not established in our application (which does not have natural strata).

5. REFERENCES

[1] V. de Matos, A. Philpott, and E. Finardi. Improving the performance of stochastic dual dynamic programming. *Applications – OR and Management Sciences (Scheduling)*, 2012.