OPERATIONS RESEARCH
Vol. 00, No. 0, September 2016, pp. 000–000
ISSN 0030-364X | EISSN 1526-5463 | 16 | 0000 | 0001

INFORMS
DOI 10.1287/xxxx.0000.0000

Noisy Simulation Optimisation with Simulated Annealing

Alvaro Cabrejas Egea
Centre for Complexity Science, University of Warwick, Coventry, CV4 7AL, a.cabrejas-egea@warwick.ac.uk

In this paper we investigate and compare the performance under the presence of noise of two different Stochastic Annealing algorithms: Optimal Stochastic Annealing and Thermostatistical Stochastic Annealing. Optimal Stochastic Annealing is a new algorithm and the analysis contained here are its first test against commercial software. To this end a test problem has been selected and different kinds of noise used, studying convergence quality, speed, the effect of the different exploration strategies and different random numbers strategies. We show the effect of these different components and try to give some general guidelines to select Stochastic Annealing algorithms depending on the requirement of the problem.

Key words: Stochastic Annealing, Optimal Stochastic Annealing, Thermostatistical Stochastic Annealing, Optimisation, Convergence

1. Introduction

Operational Research has benefited massively of the advances in computing power achieved over the last decades. Simulation has become an invaluable tool for analysing and understanding complex systems. The natural next step is to use simulation not only to analyse a complex system, but to optimise it, by automatically searching through a vast array of possible alternative configurations of an otherwise unmanageable parametric space.

The simulation itself can be based on a variety of approaches, but the most commonly used optimisation procedures - linear, non-linear and (mixed) integer programming - require a mathematical formulation which is generally impossible to obtain for practical applications. On the other hand, there are sets of relatively recent tools that adapt much better to processes with stochastic nature, from evolutionary and genetical algorithms to neural networks and simulated annealing Fu (2002). These have proven to be effective on difficult, mixed-decision-variable optimisation problems.

Optimisation via Simulation (OvS) of a stochastic process consists of an optimisation where the evaluation of a solution has to be obtained by running a simulation. It often uses black

box optimisation heuristics for the search which seeks to assign a performance measure to the parametric point being assessed. A major challenge here is that since simulations are usually stochastic, i.e. when two different solutions are being evaluated, it is not necessarily clear which one is best, implying a degree of uncertainty that can be due to noise or simulation constraints.

In general, these problems can be represented as

$$\min f(x),\ x \in \Theta, \tag{1}$$

where x is a vector of input variables belonging to the parameter space Θ, the objective function $f(x)$ represents the expected value of a random variable $E[X(x,\omega)]$, with ω representing a sample path (simulation replication) and X being the sample performance measure. In these cases the probability distribution of $X(x,\omega)$ is an unknown function of the vector of input variables x, but said distribution can be sampled through simulation (Hong et al. (2015)).

The practical applications are enormous since having solvers for these problems solidly implemented by simulation software developers could help many companies in designing and optimising complex systems. The benefits in manufacturing, computer and communication networks and business processes follow immediately, allowing to directly take on problems such as call-center staffing, traffic light optimisation, manufacturing facilities optimisation, inverse modelling of complex systems, vaccination campaigns, strategies design, portfolio optimisation, etc.

2. Theoretical Background

Nearly every commercial discrete event simulation software package has a built-in OvS solver. As a result, the modelling environments are all different, and the OvS solver must be customised to work with the specific simulation software as noted by Nelson (2010).
This paper is focused on optimisation via Stochastic Annealing, since it is one of the method being used by Lanner Ltd. in their commercial Packages.

2.1. Stochastic Annealing

In this scope, Simulated Annealing Kirkpatrick et al. (1983) (SA) is a probabilistic technique for approximate the global optimum of a given function. Specifically, it is a metaheuristic to approximate global optimisation in a large search space. It is often used when the search space is discrete.
For problems where finding the precise global optimum is less important than finding an acceptable local optimum in a fixed amount of time, SA is the best option when compared to alternatives such

as brute-force search or gradient descent. Brute-force search would provide the global optimum, but might not be able to run in finite (or acceptable) time; gradient descent operates in a quick way but has no way to avoid getting stuck in local minima. SA is a compromise between the two, much faster than brute force and statistically converging to at least a high-fitness sub-optimal solution, while slower than gradient descent but able to move out of local optima.

Any Stochastic Annealing algorithm requires for the initial timestep $t_0 = 0$:

• Energy or performance function $f(x)$, defined for all Θ, such that it can compute a difference in energy.

$$\Delta = f(x_{t+1}) - f(x_t) \tag{2}$$

• Initial solution x_0 with its corresponding measure $f(x_0)$.

• Initial temperature T_0.

• Annealing schedule capable of generating a suitable T_{t+1}, generally tuned by a parameter α. In this paper, a geometric annealing schedule has been used:

$$T_{t+1} = T_t * \alpha; \quad T_t = T_0 * \alpha^{t-1} \tag{3}$$

• Acceptance criterion $P_{tr}(x_t \to x_{t+1})$, capable of generating a probability transition as function of Δ and T_t. Typically the Metropolis Criterion Metropolis et al. (1953) is used:

$$P(\Delta, T) = P_{tr}(x_t \to x_{t+1}) = \min\left[1, \exp\left(\frac{-\Delta}{T}\right)\right] = \min\left[1, \exp\left(\frac{f(x_t) - f(x_{t+1}}{T}\right)\right] \tag{4}$$

The basic Stochastic Annealing with Metropolis Criterion functions as follows:

1. Propose a new neighboring solution to the current one and measure performance $f(x_{t+1})$.
2. IF the new solution has better performance ($f(x_{t+1} > f(x_t) \implies \Delta > 0$), accept it and make it the current solution.
3. ELSE IF the new solution has a worse performance ($f(x_{t+1} < f(x_t) \implies \Delta < 0$), accept it with probability $P(\Delta, T)$.
4. ELSE, reject the proposed solution.
5. Reduce Temperature, generating a T_{t+1} according to the chosen Annealing Schedule.
6. Iterate.

For an extended empirical analysis of SA on several combinatorial problems see e.g., Johnson et al. (1989) and Johnson et al. (1991). However, since the simulation model is stochastic, and multiple repetitions lead to different results, optimisation becomes difficult. If the measurement and thus the performance have some noise, it is not clear whether an observed quality difference is real or only rue to the randomness in the evaluation. Averaging over multiple replications reduces the effect

of noise, but quickly becomes computationally intractable. For these reasons, there is a need for efficient optimisation algorithms that only require a minimal number of simulation replications - enough to progress in the optimisation but not more than necessary in order to limit the required computational time. Initial attempts to design such algorithms have been going on for some years, some of them in this university as seen in the papers by Ball et al. (2003) and Branke et al. (2008).

3. Optimal Stochastic Annealing

OSA is a new variant of SA specifically for use with noisy performance evaluations (as in the case of stochastic simulations), currently being developed by Ball et al. (2016). In the scope of noisy measurements OSA becomes a very useful optimisation tool to obtain the closest results to the actual optimum while keeping the computational complexity in line.

Within a simulation run, if the evaluations are disturbed by noise and assuming it follows a normal distribution, then the difference in the observed energies δ is a random variable

$$\delta = N(\Delta, \sigma^2) \tag{5}$$

where Δ is the true value of the energy difference and σ^2 is the variance of this energy difference. This further introduction of stochasticity in the process, heavily impacts the SA's skill to take correct acceptance decisions. Denoting the probability distribution function of the noise as $p(z)$, the new acceptance probability can be calculated as

$$P_A(\Delta) = \int_{-\infty}^{+\infty} P_{tr}(x)\, p(x-\Delta) dx \tag{6}$$

To reduce the impact of the noise, sequential samples of the same energy level can be drawn in order to achieve a confident estimation, but this is very expensive computationally. This leads to the need of finding an optimum between pure estimation and infinite sampling, such that the resulting algorithm can restore its decision ability.

OSA introduces a procedure that, at each timestep and after taking a sample y_i and codifying it in cumulative variables

$$c_0 = 0,\ c_i = c_{i-1} + y_i,\ i > 0 \tag{7}$$

takes a three-way decision based on the history of observations $C_n = c_1, \cdots, c_n$ with probabilities

- $A(C_n)$ to accept
- $R(C_n)$ to reject
- $S(C_n)$ to continue to the next observation

such that $A+R+S=1$.

Obedience to Detailed Balance is demanded from the system at each timestep, assuring thermodynamic equilibrium. According to statistical mechanics, a system is in equilibrium if the probability of it being in a state x, $P(x)$ follows the Boltzmann-Gibbs distribution as shown in Gibbs (2014)

$$P(x) \propto e^{-\beta E(x)} \tag{8}$$

with $E(x)$ denoting the energetic level of a configuration and $\beta = \frac{1}{k_B T}$ depending on the Temperature T and the Boltzmann constant k_B.

A sufficient condition (Graham and Haken (1971)) for equilibrium on any system is fulfilling detailed balance

$$P_{tr}(x_1 \to x_2)P(x_1) = P_{tr}(x_2 \to x_1)P(x_2) \forall x_1, x_2. \tag{9}$$

Imposing this restriction leads to an optimal acceptance rule with acceptance probability

$$A(C_n) = \min(1, e^{(-2(c_n+\beta\sigma^2/2)(c_{n-1}+\beta\sigma^2/2)/\sigma^2)}). \tag{10}$$

Rejection happens when $c_n > c^* \in [-\beta\sigma^2/2, \inf)$, setting c^* as parameter of the system.

This acceptance rule is remarkably universal in the sense that it achieves the absolute highest possible chance of move acceptance at each step, and this is independent of the choice of move rejection decisions. The rejection parameter can also be tuned to an optimum level under the assumption that the Δ of the following move is the same as for the current move, demonstrating to work well in practice. Fulfillment of detailed balance is the key point here. It guarantees OSA's convergence to the global minimum of the annealing is slow enough. Even more, since it is obeyed at every time step it also guarantees to maximise the acceptance probabilities per iteration, greatly improving performance, since t has been shown that a higher probability to accept a movement leads to a faster convergence of the algorithm to the thermodynamical equilibrium (Hastings and Apr (1970)).

The basic Optimal Stochastic Annealing functions as follows:

1. Propose a new neighboring solution to the current one and measure performance $f(x_{t+1})$ and performance difference Δ and codify it in the cumulative variables C_n, generating an acceptance probability $A(C_n, T)$.

2. IF accepted, iterate.

3. ELSE IF the current $c_n < c^*$, draw more samples y_i for both points and update C_n, generating a new $A(C_n, T)$.

   (a) IF accepted, iterate.

   (b) ELSE IF the current $c_n < c^*$, draw more samples y_i for both points and update C_n, generating more $A(C_n, T)$ and iterating inside this inner loop.

(c) ELSE IF the current $c_n < c^*$, reject the solution, exit the loop.

4. ELSE IF the current $c_n < c^*$, reject the solution.

5. Reduce Temperature, generating a T_{t+1} according to the chosen Annealing Schedule (this step depends on the schedule, i.e. it can be based on number of samples or accepted solutions .

6. Iterate

4. Thermostatistical Stochastic Annealing

This algortihm is developed and used by Lanner Ltd. in WITNESS Horizon (WH), a commercial software for discrete-event simulation, being one of the main optimisation tools.

At the time of this research, the details of the algorithm have not been released by the owner company. The main testing method has been to directly implement the test problem and execute within WITNESS itself, being able to monitor which are the best and current solutions at any point of a run.

5. Methodology

In this section the experimental setup and different approaches taken for the comparison between Optimal Stochastic Annealing and Thermostatistical Stochastic Annealing is explained.

5.1. Experimental Setup

WH is mainly intended for mid-size to big discrete-event simulations in areas such as processing or manufacturing industry. This usually involves operators executing tasks assigned to process or processed elements. These elements normally are stored in buffers or in queue waiting to be processed.

This implies that some of the approaches embedded in the simulation toolbox would not be the optimal for a lighter or more general optimisation process. WH uses by construction a fixed sample with fixed size approach to optimisation. The number of potential solutions to be evaluated has to be defined as a hard limit. Which solutions are sampled is determined by one of ten thousand pseudo-random number streams that the program allows to use. This approach is taken so the Common Random Numbers method, as described in Kleinman et al. (1999), can be used to reduce the variance of the simulations, by using the same random numbers to simulate all configurations, a specific random number used for a specific purpose in one configuration is used for exactly the same purpose in all other configurations. This decrease in randomness also makes the exact same results obtainable by using the same original seed.

OSA normally uses a variable sample of variable size which had to be modified to a fixed size limiting the total number of samples that the program is allowed to take. This limit was imposed

to make comparison with WH easier.

In any SA comparison, one of the main choices is the annealing schedule. WH defines it in terms of initial temperature, annealing factor α, and number of temperature drops. For these experiments the following custom schedule was selected:

- $T_0 = 20$
- $\alpha = 0.95$
- 150 Temperature drops over 3000 input vectors

In traditional Stochastic Annealing an initial temperature of 20 would be low. The rationale for this choice is that in order for the comparison to be meaningful, OSA needs to be operate in a regime such that there is a meaningful amount of resampling. This, given the noise levels that have been used, only occurs for low temperatures. At the same time, too much temperature reduction prevents WH from improving on the solution. A mid point has been chosen, such that WH still maintains its typical behavior with options to improve until the end.

WH also offers the option to allow the program to select its own schedule. This option was used in some experiments to compare a manually fine-tuned approach with the automatic Thermostatistical setting.

The main limitation to the experimentation was WH's running times. So the first two experiments were centered about minimising running times so enough data could be generated for the results to have statistical significance.

5.2. Test Problem

Both algorithms were compared using two different versions of the Ronsenbrock Function in 2D. The landscape results of the analytical formula:

$$f(x,y) = (a - x^2) + b(y - x^2)^2 \tag{11}$$

It has a global minimum at (a, a^2) with value $f(x,y) = 0$. Usually $a = 1$ and $b = 100$. For these experiments and due to the precision limitation of WH, the function was defined in $[-3, 3)$ for both x and y with a resolution (minimum step) of 0.001 and a total interval length of $L = 6$.

For those experiments studying convergence quality and speed, Gaussian noise was added to the evaluation of the function at the new proposed solution such that

$$f(x,y)_{measured} = f(x,y) + Normal(0,1) \tag{12}$$

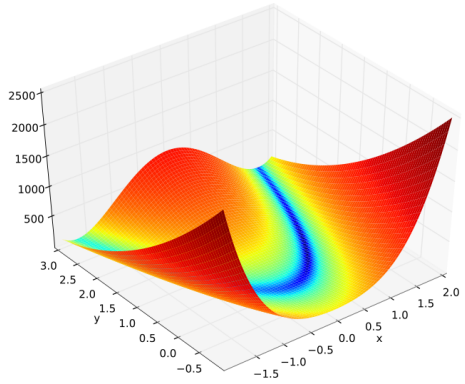

Figure 1 Landscape of undisturbed Rosenbrock Function resulting from Equation 11.

This however, showed that due to the random number strategies that WH uses, in which the random number generator is given the same seed at every noise evaluation, leads to WH optimising over an deterministic offset landscape, in which at every iteration the value of the noise will be the same as at the initial time. The only option to reduce this offset effect is to allow the program to make more parallel replications of the same analysis, since every replication will be seeded with a different random number, and the average of these, if enough are allowed will be the actual value of the function. These additional replications use all the same inputs (x, y), the only difference being the random numbers used, so their effect is not the same than doing extra repetitions of the same experiment. This situation greatly complicates assessing the effect of noise since both algorithms process it in different ways, given that OSA uses independent noise at every timestep.

To assess the effect of the random number and the stepping strategies, a different kind of noise was defined in which the measured quality would be the original function disturbed by a Random Fourier Series:

$$f(x,y)_{measured} = f(x,y) + Fourier(x,y) \tag{13}$$

Given a noise level ϵ, the disturbance is given by:

$$\begin{aligned} Fourier(x,y) = \Bigg[r_1 * \sin\left(\frac{2\pi}{L}(x+3) + 2\pi r_2\right) + r_3 * \sin\left(\frac{4\pi}{L}(x+3) + 2\pi r_4\right) + \\ + r_5 * \sin\left(\frac{2\pi}{L}(y+3) + 2\pi r_6\right) + r_7 * \sin\left(\frac{4\pi}{L}(y+3) + 2\pi r_8\right) \Bigg] * \epsilon, \quad r_i \in \text{Uniform}(0,1) \end{aligned} \tag{14}$$

This noise results on average into a Normal Distribution with zero mean and $\sigma = 0.86$.

6. Results

6.1. Replication Scaling

One approach is to allow, on average, the same number of samples to WH than OSA takes. Based on the previous point and according to OSA results, between 2 and 3 replications per point should

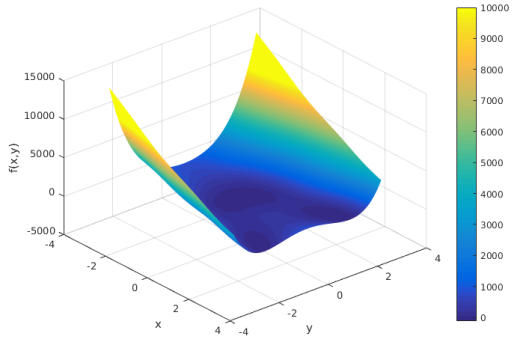

Figure 2 Landscape of Rosenbrock Function disturbed by a Fourier Random Series as shown in Equation 13 with $\epsilon = 20$.

be allowed to WH. The reader must take into account that this means that every point will be sampled 2 or 3 times, and the used value will be the average of the different offset qualities of the observations.
The results for this experiment can be found in Figures 3 and 4.

In these figures it can be seen how OSA converges faster to a solution which is better by two orders of magnitude. This occurs for the current solution and for the best found solution (Figures 3 and 4).
It can also be observed that the behavior of OSA is quite homogeneous through the run, while WH has three different regions. An inital "jumpy" exploration phase in which great jumps are made and the resulting evaluations greatly oscillate, then a smoother one where the energy differences between solutions are closer to those of OSA, and lastly, a localised search region in which no great jumps are taken. This behavior has been observed for different temperature settings, so it is thought to be a design feature rather than a consequence of the analysis' parameters.

6.2. Standard deviation Scaling

An alternative approach is to compare standard OSA with WH allowing OSA to keep using a variable sample size. There are two ways to achieve this keeping the comparison fair. The first approach involves computing the true number of samples that OSA takes per full run on average, and then reducing the standard deviation in HW's gaussian noise.
Since every time OSA samples a new solution it also resamples the current one, and denoting by n the number of times that OSA calculates energies, by N the total number of allowed different

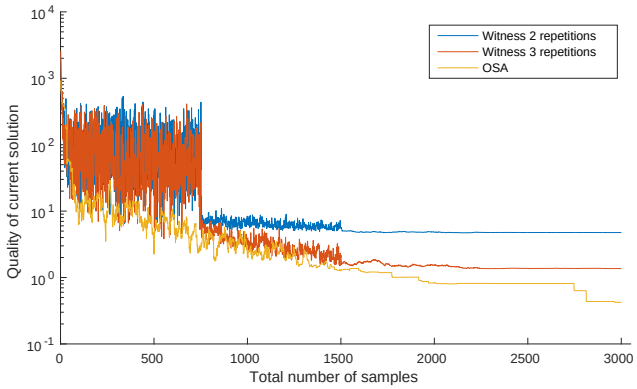

Figure 3 Convergence results for the current solution in Ronsenbrock function using replication scaling in WH, averaged over 10 executions. The methodology used to obtain these results is explained in Section 6.1. It can be seen how OSA converges steadily in a faster, more stable and better way. It can also be seen that WH uses a specific strategy for generating new solutions, as it shows three very distinct phases in the difference in the quality between subsequent solutions.

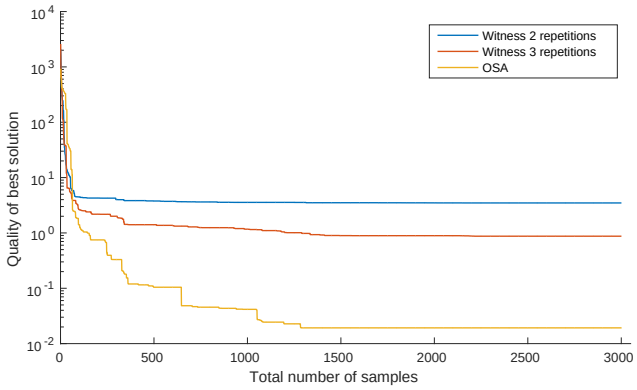

Figure 4 Convergence results for the best solution in Ronsenbrock function using replication scaling in WH, averaged over 10 executions. The methodology used to obtain these results is explained in Section 6.1. It can be seen how OSA converges steadily in a faster, better way to a best solution two orders of magnitude better than WH.

solutions to be tested and by R the ratio between the samples taken in OSA and HW, the narrowed down the standard deviation of the sample mean, σ for WH will be:

$$\sigma = \sqrt{\frac{1}{R}} = \sqrt{\frac{N}{2n}} \tag{15}$$

This achieves the same experimental conditions for this and the previous experiment requiring less replications and computational time. Averaging over 2000 OSA executions the obtained value is R=2.615, resulting on an scaled standard deviation of $\sigma = 0.6184$.

Additional experimentation was made using WH's ability to set it's own schedule, comparing against two versions of OSA with slow and quick cooling respectively.

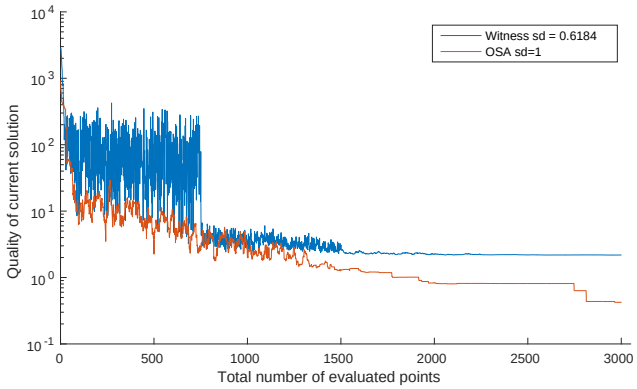

Figure 5 Convergence results for the current solution in Ronsenbrock function using standard deviation scaling in WH, averaged over 10 executions. The methodology used to obtain these results is explained in Section 6.2. It can be seen that the final quality in WH for the adjusted standard deviation of the sample mean experiment converges to a value situated between those using two and three replications in the Figure 3, due to the obtained value of R=2.615, this result validates the approach.

The results for this experiment can be found in Figure 5 through Figure 8.

In these figures it can be seen again that OSA converges faster and better in all cases. The performance for WH using this approach is situated between the results for two and three replications in the previous section, demonstrating that both methods are valid. Standard deviation scaling becomes the most interesting one since it greatly decreases the number of replications needed and the required computational time This approach is taken so experimentation can proceed in a faster way, and does not relate to the real problem where σ is intrinsic to the simulation model and the only way to reduce the standard deviation of the sample mean is to allow more replications.
Unexpectedly the set schedule in WH performed better than when the program is allowed to choose its own schedule. The same three phases can be observed in the behavior of WH, while OSA performs exactly as in the previous analysis.
In Figure 7 and Figure 8 it can also be seen that OSA converges quicker with a more aggressive cooling schedule, but the final quality is better with a slower one.

6.3. Performance as function of number of samples

Once convergence quality has been analysed in equal conditions, the speed of the convergence becomes relevant.
Since the previous points have proven effective, standard deviation scaling has been used in order to reduce the total number of replications and be more computationally efficient.
For this experiment, the same schedule has been used for all run lengths, comparing the quality of the final current solution at the end of the run and also the best solution found through the run.

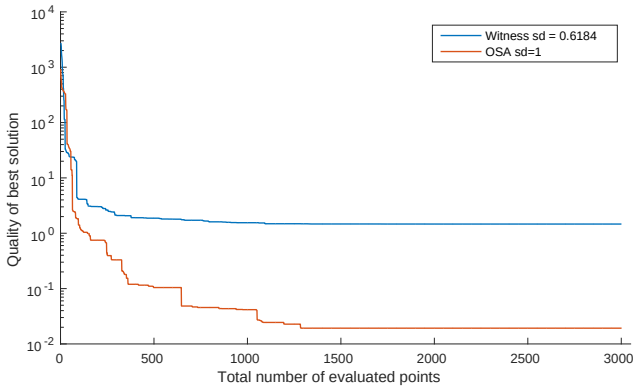

Figure 6 Convergence results for the best solution in Ronsenbrock function using standard deviation scaling in WH, averaged over 10 executions. The methodology used to obtain these results is explained in Section 6.2. It can be seen that the best quality in WH for the adjusted standard deviation of the sample mean experiment converges to a value situated between those using two and three replications in the Figure 4, due to the obtained value of R=2.615, this result validates the approach also in the case of best results in the run.

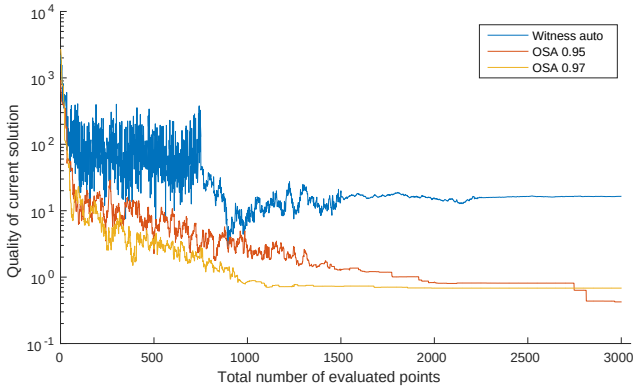

Figure 7 Convergence results for the current solution in Ronsenbrock function using standard deviation scaling in WH, allowing the program to choose its own schedule, averaged over 10 executions. The methodology used to obtain these results is explained in Section 6.2. No benefit is found from allowing WH to select its own schedule, obtaining worse results for the current solution than those found in Figures 3 and 5.

The results for this experiment can be found in Figure 9 and Figure 10.

In these figures the error bars mark one standard deviation. In some sections there is some overlapping, but in general OSA performs better for all run lengths. Both algorithms greatly benefit from the extra samples allowed at the beginning, but this effect dampens as more samples are allowed. This dampening is more severe in the case of OSA which does not seem to gain any benefit when it measures over 2000 points for the current solution and around 1200 for the best solution.

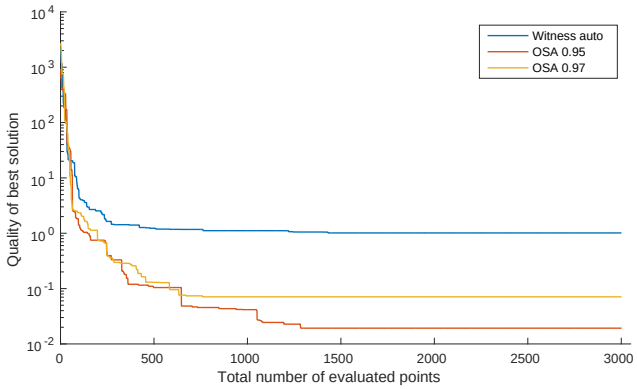

Figure 8 Convergence results for the best solution in Ronsenbrock function using standard deviation scaling in WH, allowing the program to choose its own schedule, averaged over 10 executions. The methodology used to obtain these results is explained in Section 6.2. Little benefit is found from allowing WH to select its own schedule, obtaining only slightly better results for the best solution than those found in Figures 4 and 6.

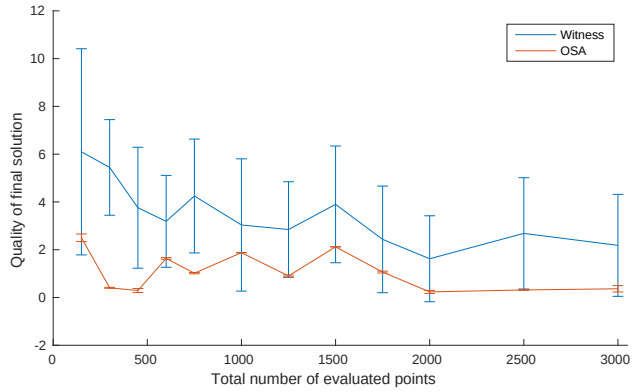

Figure 9 Convergence results for the current solution in Ronsenbrock function using standard deviation scaling in WH, taking the quality of the final current solution for different run lengths, averaged over 10 executions. The methodology used to obtain these results is explained in Section 6.3. It can be observed that the quality of the final solution of the run in OSA is better for all run lengths.

6.4. Effect of the strategies for proposing new solutions (stepping strategy)

In order to assess the effect of the more elaborate and unknown approach to stepping used in WH, the same Common Random Numbers strategy was implemented in Python for the classic Metropolis Algorithm, and this was compared with the results obtained from WH, applied to the deterministic Rosenbrock problem.
The results for this experiment can be found in Figure 11 and Figure 12.

In these figures it can be seen that a regular Metropolis algorithm keeps improving the value of the current and best solution until the end of the run, while WH, following the same pattern as in

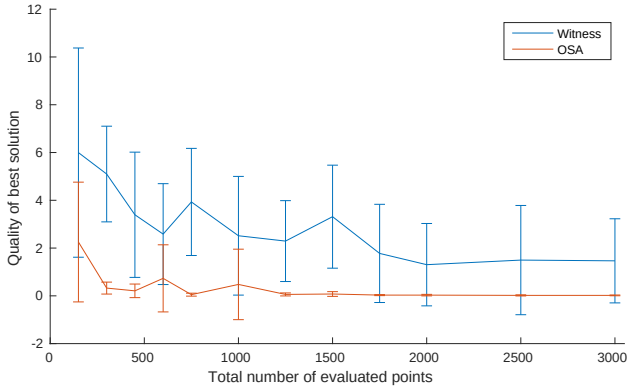

Figure 10 Convergence results for the best solution in Ronsenbrock function using standard deviation scaling in WH, taking the quality of the final current solution for different run lengths, averaged over 10 executions. The methodology used to obtain these results is explained in Section 6.3. It can be observed that the quality of the best found solution of the run in OSA is better for all run lengths, even if there exists some overlapping in the error bars.

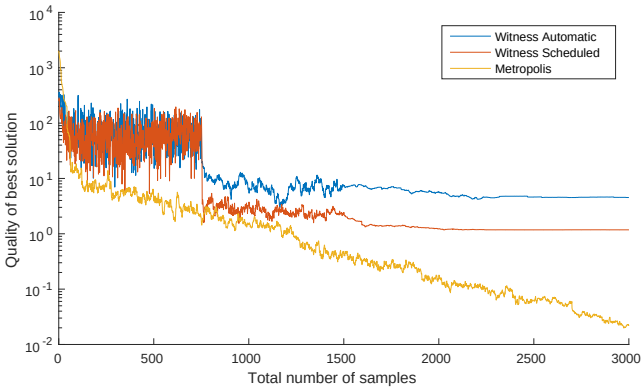

Figure 11 Convergence results for the current solution in Ronsenbrock function without scaling or noise for a fixed sample size with respect to the number of observations of the functions, not the number of input vectors. Averaged over 20 executions. The methodology used to obtain these results is explained in section 6.4. It can be observed again how even in the absence of noise, the difference in quality between subsequent solutions in WH suggests a 3 phase search strategy that prevents the algorithm from steadily approaching the minimum. Metropolis algorithm with a random generation of new solutions achieves this steady descent.

the previous experiments, settles soon after half of the analysis for a local search that prevents it from further improving the best or current solutions.

6.5. Effect of the Common Random Numbers Strategy

Since Common Random Numbers can be a positive influence for optimisation in some circumstances (For example, when optimising a queuing system, if we are comparing two different configurations cashiers in a supermarket, the random time of arrival of the $N-th$ customer should be generated

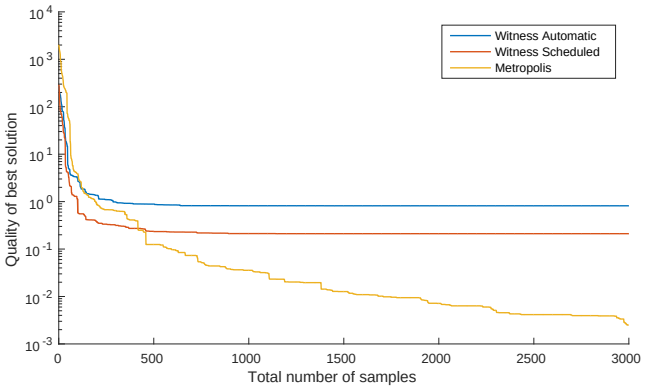

Figure 12 Convergence results for the best solution in Ronsenbrock function without scaling or noise for a fixed sample size with respect to the number of observations of the functions, not the number of input vectors. Averaged over 20 executions. The methodology used to obtain these results is explained in section 6.4. The behavior of both algorithms in the case of the best found solution is a reflection of the behavior shown in Figure 11.

using the same draw from a random number stream for both configurations). As a last experiment, Common Random Numbers with Metropolis algorithm was compared against OSA, forcing them to use exactly the same number of measurements. To this end, OSA was limited to take exactly the same number of samples (observations, measurements) as WH in a harder way than in the previous experiments, discarding a potential solution if the observation limit is reached. OSA's resampling tends to happen towards the end of the run when the ratio of noise to signal is high and it's more beneficial. This means that with this approach, OSA will effectively probe many less points than WH. The test problem chosen to optimise was the Rosenbrock function with an added Random Fourier Series as shown in equations 13 and 14. The results for this experiment can be found in Figure 13 and Figure 14.

When looking at the figures the Common Random Numbers strategy proves to be superior in this case, but the reader can notice that the results of both algorithms are quite close during most of the run. In all the cases the overlapping finishes and OSA's solutions stagnate. This is consistent with OSA taking many resamples once the temperature is low enough. The fact that the lines decouple earlier as the noise increases is further indication of this behavior.

In general this situation is expected, since now the acceptance criterion for the algorithm using Common Random Numbers is classic Metropolis, which can be obtained for some settings of OSA. The main difference between the two algorithms is how many points they are allowed to sample, resulting in a much lower number in the case of OSA due to said resampling.

The author believes that these results are due more to an unfair comparison due to experimental

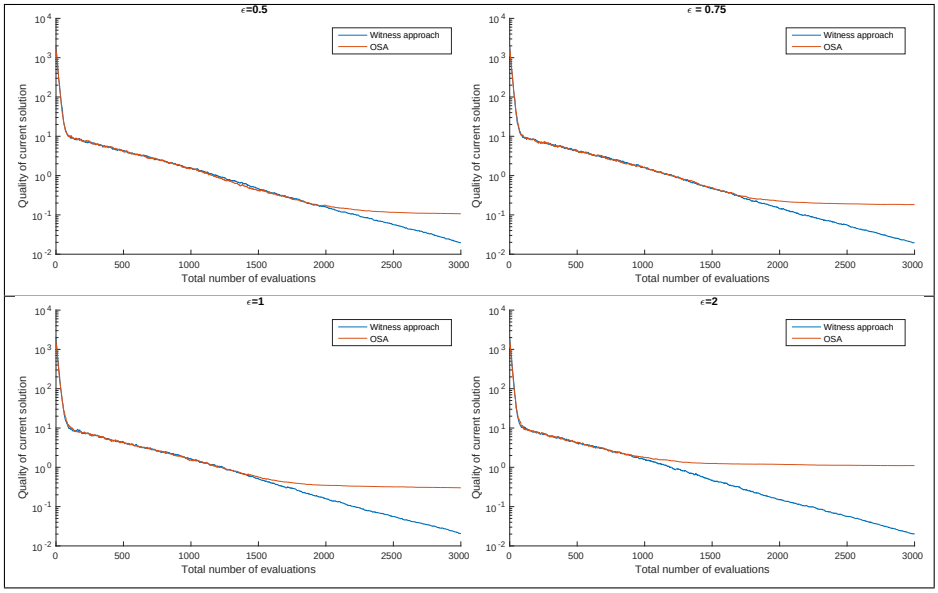

Figure 13 Convergence results for different values of ϵ of the current solution in Ronsenbrock function disturbed with a Random Fourier Series for a fixed sample size with respect to the number of observations of the functions, not the number of input vectors. Averaged over 1000 executions. The methodology used to obtain these results is explained in section 6.5. It can be observed how both algorithms advance in exactly the same fashion during the run, until close to the end, when in the presence of low temperatures and high noise-to-signal ratio, OSA starts heavily resampling. Since the comparison is made in demanding equal number of samples and not evaluated points, this plays against OSA's strengths and forces it to probe a smaller number of input vectors than Metropolis with Common Random Numbers.

setup (equality of samples vs equality of input vectors) rather than a proof that Common Random Numbers are the strategy to follow in this situation, as shown by all the previous experiments.

7. Discussion

In general and in sight of the figures, specially those corresponding to the comparison under different run lengths (Figure 9 and Figure 10), it can be said that OSA converges, in general, faster and better than the commercial implementation of WH under any run length. The author has several hypothesis for this behavior.

7.1. Witness Horizon Automatic Schedule

No benefit has been found by using the automatic scheduling system provided by the program. Since the details of this schedule are not known to this project, the author is not able to theorise about why this happens. The results from experimentation with this setting can be seen in Figure 7 and Figure 8.

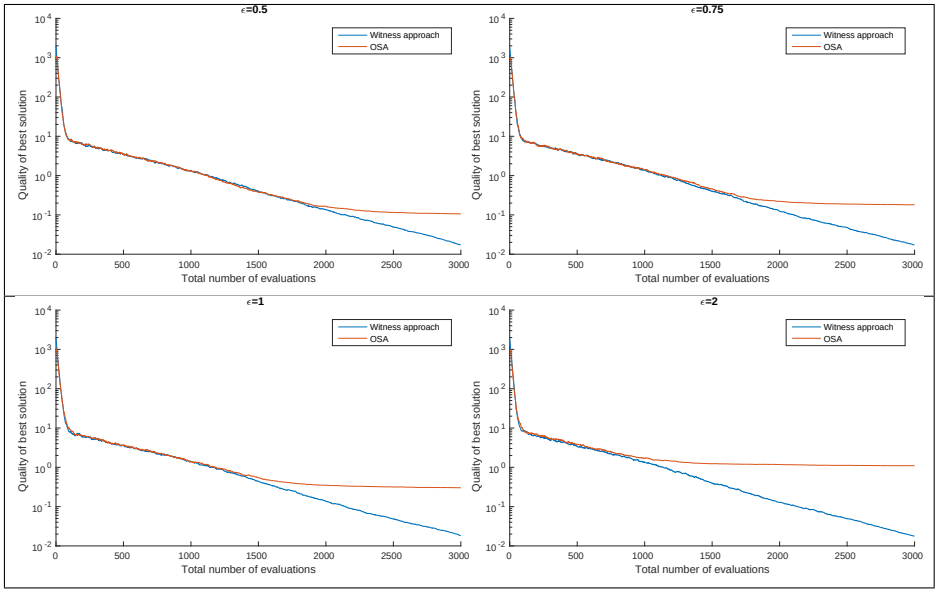

Figure 14 Convergence results for different values of ϵ of the best solution in Ronsenbrock function disturbed with a Random Fourier Series for a fixed sample size with respect to the number of observations of the functions, not the number of input vectors. Averaged over 1000 executions. The methodology used to obtain these results is explained in section 6.5. The same behavior as shown in Figure 13 can be observed for the best solution found.

7.2. Interaction of Noise with Common Random Numbers

The fact that HW uses Common Random Numbers for their simulations can have a negative effect on the simulations for low levels of replications of each scenario. Without extra points to average over, WH only optimises over an offset landscape.

The author believes that this approach was taken because WH is mostly meant to optimise big simulations, in which little changes are introduced. Using this method, it allows for better fine tuning of the different parameters of the model. However, there are different methods to achieve this goal, and SA is mostly indicated for big search spaces where not all configurations can be tested, which does not seem to be the goal here.

Figures 13 and 14 were obtained using the Fourier Random Series and the same stepping algorithm with Metropolis criteria. In them it can be seen that due to the high number of replications used, the negative effect of the Common Random Numbers method disappears.

7.3. Stepping Strategy

WH shows a intentional behavior in the way it chooses the next potential solution. As indicated before, three different areas can be identified, a first exploration where the quality of the solutions varies greatly, a more general search with energy differences similar to those in SA and OSA and a final highly local search that most of the time does not produce big improvements for the problems

tested.

Once this stepping strategy is removed (Figures 13 and 14), the convergence keeps improving during the entire run.

7.4. Effect of Resampling

The approaches that both algorithms take towards replications for a given input vector create affect their performance in very different ways.

HW will resample uniformly throughout the run, helping to diminish the effect of noise. This has the side effect that a lot of computational power is wasted re-evaluating solutions for which the behavior should be clear. Also the algorithm will measure a set of points of fixed size, allowing to always know which ratio of the potential search space has been examined

OSA concentrates the resampling towards the end of the run, when the noise to signal ratio becomes more meaningful. The side effect is that in unlucky runs, the algorithm can need to measure a single point many times, all of them counting towards the total amount of samples that the algorithm is allowed. This causes the decoupling in quality improvement that can be seen in Figure 13 and Figure 14. In general OSA's approach will be more useful and will generate better results for those situations in which the same number of input variables are checked. If the experimentation is limited to the number of samples, and OSA has a highly variable sample size, this will only lead to performance loss as shown in the Figures indicated above.

8. Conclusions and future research

Both algorithms are intended for very different uses, each one being adapted to maximise their utility in their field.

In the scope of heavy discrete-event simulations with intensive use of queuing, the method used by WH will reduce the variability of the models, allowing for better fine tuning.

In a more general setup, OSA is way more capable of managing uncertainty, higher levels of noise and can be applied to a wider variety of problems, not necessarily discrete.

For future research it would be most beneficial to obtain the Thermostatistical Stochastic Annealing algorithm directly from Lanner Ltd. so it can be implemented in an external platform and more extensive testing can be conducted.

This future testing could focus on better understanding the specifics of the stepping strategy used by WH and the potential benefits it can yield. A second topic is to investigate the effect of the Common Random Numbers method while optimising a simulation, not an evaluation of a function. A third topic could involve changing the test problem for a simple noisy test simulation.

Acknowledgments

The author wishes to thank all the Centre for Complexity Science staff for their dedication to provide the best academic life they can to all of their students. The author also wants to thank Juergen Branke and Robin C. Ball for the help and interesting discussions during the project. Last and most importantly to all the colleagues, family and friends that one way or another made this year possible.

References

Ball RC, Branke J, Meisel S (2016) Optimal Stochastic Annealing (preprint). *Operations Research* .

Ball RC, Fink TMa, Bowler NE (2003) Stochastic annealing. *Physical review letters* 91(3):030201, ISSN 0031-9007, URL `http://dx.doi.org/10.1103/PhysRevLett.91.030201`.

Branke J, Meisel S, Schmidt C (2008) Simulated annealing in the presence of noise. *Journal of Heuristics* 14(6):627–654, ISSN 1381-1231, URL `http://dx.doi.org/10.1007/s10732-007-9058-7`.

Fu MC (2002) Optimization for simulation: Theory vs. practice. *INFORMS Journal on Computing* 14(3):192–215, ISSN 1091-9856, URL `http://dx.doi.org/10.1287/ijoc.14.3.192.113`.

Gibbs JW (2014) *Elementary principles in statistical mechanics* (Courier Corporation).

Graham R, Haken H (1971) Generalized thermodynamic potential for Markoff systems in detailed balance and far from thermal equilibrium. *Zeitschrift fur Physik* 243(3):289–302, ISSN 14346001, URL `http://dx.doi.org/10.1007/BF01394858`.

Hastings WK, Apr N (1970) Monte Carlo Sampling Methods Using Markov Chains and Their Applications. *Biometrika* 57(1):97–109.

Hong LJ, Nelson BL, Xu J (2015) Discrete Optimization via Simulation. *Handbook of Simulation Optimization*, 9–44, number 216, ISBN 978-1-4939-1383-1, 978-1-4939-1384-8, URL `http://dx.doi.org/10.1287/opre.1050.0237`.

Johnson DS, Aragon CR, McGeoch LA, Schevon C (1989) Optimization by Simulated Annealing: An Experimental Evaluation; Part I, Graph Partitioning. *Operations Research* 37(6):865–892, URL `http://dx.doi.org/10.1287/opre.37.6.865`.

Johnson DS, Aragon CR, McGeoch LA, Schevon C (1991) Optimization by Simulated Annealing: An Experimental Evaluation; Part II, Graph Coloring and Number Partitioning. *Operations Research* 39(3):378–406, URL `http://dx.doi.org/10.1287/opre.39.3.378`.

Kirkpatrick S, Gelatt CD, Vecchi MP (1983) Optimization by Simulated Annealing. *Science* 220(4598):671–680, ISSN 00368075, URL `http://dx.doi.org/10.1126/science.220.4598.671`.

Kleinman NL, Spall JC, Naiman DQ (1999) Simulation-Based Optimization with Stochastic Approximation Using Common Random Numbers. *Management Science* 45(11):1570–1578, URL `http://dx.doi.org/10.1287/mnsc.45.11.1570`.

Metropolis N, Rosenbluth AW, Rosenbluth MN, Teller AH, Teller E (1953) Equation of state calculations by fast computing machines. *Journal Chemical Physics* 21(6):1087–1092, ISSN 00219606, URL `http://dx.doi.org/http://dx.doi.org/10.1063/1.1699114`.

Nelson B (2010) Optimization via simulation over discrete decision variables. *Tutorials in operations research INFORMS 2010* 193–207, URL `http://dx.doi.org/10.1287/educ.1100.0069`.