Risk-informed Simulation Optimization for engineering asset management

Jérôme Lonchampt[a], William Lair[a]
[a] EDF R&D, Chatou, France

Abstract: This paper present a general method coupling genetic algorithms and Monte-Carlo simulation to address simulation optimization issues in the field of engineering asset management. After a description of the method, parameters tuning issues are analyzed through a test-case.

Keywords: Simulation-Optimization, Genetic Algorithm, Monte-Carlo simulation, Asset Management

1. INTRODUCTION

Optimizing the maintenance schedule for a component or a system, a classic problem in Engineering Asset Management (EAM), faces two major challenges. The first one is to build a realistic model that can be used to assess the efficiency of a given maintenance strategy. The second one is to handle the important combinatory of the optimization problem since, on a year based maintenance, the solution space size is growing exponentially with the operating time remaining.
In this paper we present a general framework for risk-informed constrained maintenance scheduling optimization coupling a Genetic Algorithm (GA) and Monte-Carlo simulation algorithm. The performance and the parameters tuning of this Genetic Algorithm for Simulation Optimization (GASO) will be discussed based on a test case (finding the replacement dates minimizing the global owning cost of a single component with a Value at Risk constraint) with a special focus on the fitness function for which two alternatives have been studied.

2. SIMULATION OPTIMIZATION REVIEW

Taken separately, efficiency assessment issue and asset management strategy optimization issue have been addressed successfully and are widely described in the literature. Reliability and, more recently, industrial asset management models have been developed for decades to assess the efficiency of a maintenance strategy, on the technical point of view (reliability, availability or safety indicators) or on the financial one (discounted cash-flows, Net Present Value…). Many mathematical models and associated tools are used (Markov graphs, Piecewise Deterministic Markov Process, Petri Net…). These models are solved thanks to numerical calculation techniques or Monte-Carlo Simulation if the underlying model is more complex.

On the other hand, finding the optimal schedule for maintaining a component is a hard optimization problem, as it is often impossible to write down the goal function as an easily optimizable function (linear, convex…), whether because the asset to model or the indicators to be optimized are too complex. Exact methods are then not usable (cutting plane methods for linear problems) or not efficient enough (branch and bound). An efficient alternative is to use approximation methods such as metaheuristics, among which one of the most popular is the genetic algorithm.

Solving optimization problem for goal function assessed through simulation is a research area known as “Simulation Optimization” and it has been increasingly studied in the past fifteen years. A recent survey ([1]) identifies metaheuristics as the best methods for global integer optimization. The difficulty is then to associate Monte-Carlo simulation and Genetic Algorithm, as a matter of fact Monte-Carlo simulations need a large number of replications to narrow the results confidence interval and Genetic Algorithms need to evaluate a large number of solutions to converge toward a global optimum, a simplistic coupling could lead to calculation durations too long to be tractable. Examples of coupling such algorithms have been described in cases where simulation durations were not an issue ([2]). In other cases ([3]), the confidence interval is taken into account to penalize the goal function but the convergence of solutions is not improved throughout the process. A sequential method, named OCBA, improving the convergence of “good solutions” throughout optimization, has been described in [4] and coupled to evolutionary algorithms for manufacturing or design problems ([5]).

In the field of maintenance and asset management, there have been very few uses of these methods but it would respond to a growing concern of decision makers who want to take multi-objectives and multi-constraints decisions for complex assets. It is then impossible to avoid using simulation to assess the objectives or to check constraints (especially risk-informed ones). One of the few examples found ([6]) does not seem to have been widely used since, perhaps because the goal function (expected availability of a redundant system) could be approximated with a Markov process numerically calculable.

3. METHOD

3.1. Asset Management Simulation Model

A generic asset management model has been developed within EDF R&D to evaluate the profitability of an investments strategy for large assets of power plants. This model relies on a parallel evaluation of two strategies, the reference one and a new one that decision makers want to assess, replicated with a Monte-Carlo simulation tool. The only source of uncertainty taken into account in this model are the failure dates of the components, which is an aleatoric uncertainty as opposed to epistemic uncertainties such as maintenance costs or spare parts supply delays which aren’t modeled in EDF tool but analyzed through sensitivity analysis. The generic method and tool have been discussed in [7].

For the purpose of this paper, the asset management model has been simplified to a single repairable component with a non constant failure rate. Repairs are assumed instantaneous, unavailability of the component after failure being taken into account in the sole total costs. Preventive replacement, assumed to be an As Good As New (AGAN) maintenance task, may be performed according to a date-based maintenance program. Figure 1 shows the graph for such a model, for which:

- λ is the failure rate of the component depending on its age
- δ is the dirac function
- π is the deterministic law modeling preventive replacements
- t is the time
- a is the age of the component
- CF is the cumulated cash-flows
- C_{CM} is the corrective maintenance cost (including spare part costs, forced outage costs…)
- C_{PM} is the preventive maintenance cost
- α is the discount rate

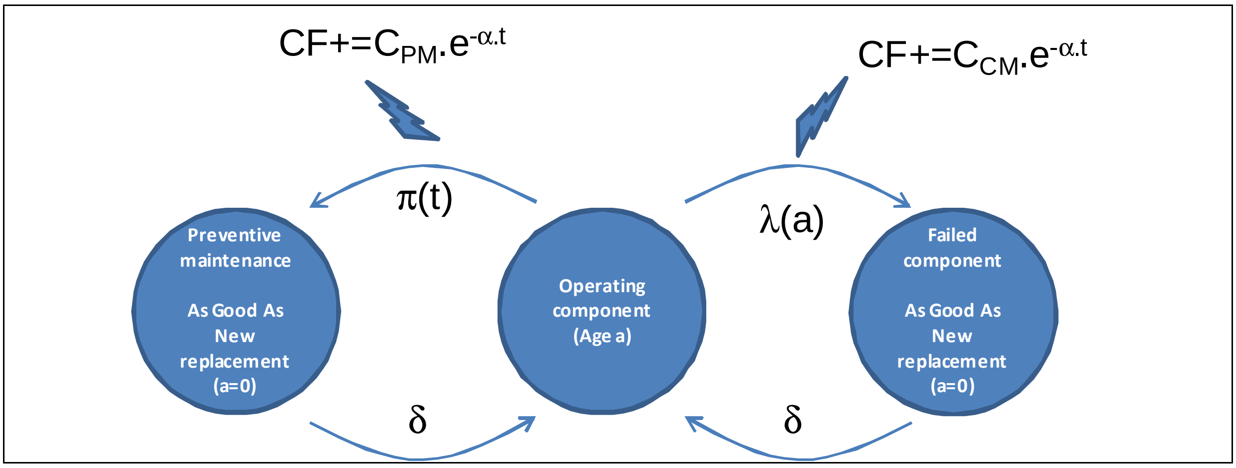

Figure 1 - Pseudo-Markov graph with transitions impacts on cash-flows for a single repairable component

The Monte-Carlo simulation used to estimate the probabilistic distribution of the cumulated cash flows is described in Table 1.

```
while r<NReplications
        Total_Cash_Flow(r)=0;
        t_failure=reliability_law(rand);
        t_preventive_replacement=[t_pr_1;t_pr_2;…];
        while min(t_failure ;t_preventive_replacement)<T_final
                t= min(t_failure ;t_preventive_replacement) ;
                if t=t_failure
                        Total_Cash_Flow(r)+=corrective_cost*exp(-alpha.t);
                        t_failure=t+ reliability_law(rand);
                else
                        Total_Cash_Flow(r)+=preventive_cost*exp(-alpha.t);
                        t_failure=t+reliability_law(rand);
                        delete(t,t_preventive_replacement);
                end;
        end;
        r+=1;
end;
```

Table 1 - Pseudo-code for the single repairable component Monte-Carlo simulation

3.2. Genetic Algorithm

The Genetic Algorithm (GA) was introduced by Holland in [9] and popularized by Goldberg [10]; it is a evolutionist meta-heuristic widely used for combinatorial optimization. It is based on an analogy to Darwin's evolution theory on natural selection stating that, within a population, organisms with a high fitness are more likely to reproduce and to create offsprings with higher fitness.

Genetic algorithms include different operators mixing global search (selection of best solutions, cross-over to build new ones) and local search (mutation) in order to find a good approximation to an optimization problem. Figure 2 presents a generic scheme of such an algorithm.

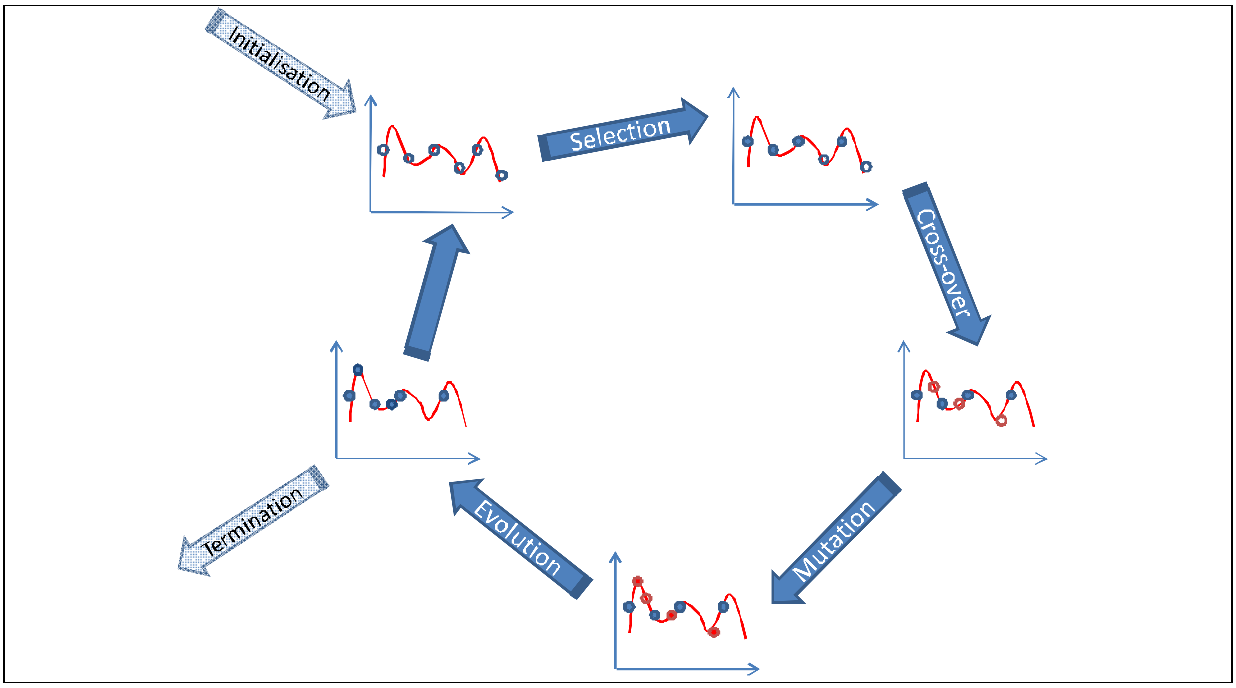

Figure 2 – Generic Genetic Algorithm

This method is one of the most popular meta-heuristic used in EAM optimization problems [8]. It has been implemented in IPOP software to solve investments planning problems with budget constraint for EDF (for a complete description of the specific algorithm and a discussion on parameters tuning, one should read [11]). The goal function used in IPOP is a mean indicator whose evaluation is fast enough to be efficiently computed with regular Genetic Algorithms.

3.3. Genetic Algorithm for Simulation Optimization (GASO)

When the goal function is expensive to compute, which is often the case for simulation evaluation, it is very difficult to use GA as described previously. As a matter of fact, for complex models, industrial assets simulation of one given strategy may take up to several minutes or hours to compute converged estimators. On the other hand, GA may need the evaluations of several thousands of different strategies, leading to calculations that would last days or months. Even if it does not seem impractical, and that calculations time could be shortened using supercomputers, it is not a workable method as this kind of calculations needs to be assessed daily by system engineers or business planners, often with very short delays.

The main idea of the method described in this paper is to improve the convergence of the simulations throughout the optimization process. At the initialization step, all solutions, chosen randomly, are evaluated with a small number of replications N then offspring are created using usual cross-over and mutation operators except that, when it comes to evaluating the offsprings, the simulation is ran in two different ways:

1. If the new solution already exists: M replications (with $M<N$) are added to the existing solution and the number of offsprings is not incremented
2. If the new solution is not present in the population: it is initialized with N replications and considered as an offspring.

When the number of offsprings reaches the limit required to update the population, a $(\lambda+\mu)$ evolution strategy is applied keeping the best solutions of both the original population and the offsprings, and N replications are added to all solutions.

To the regular termination criteria of a Genetic Algorithm (maximum number of generations reached or no improvement of the best solution after X generations) are added criteria on the simulation convergence, such as a maximal width of confidence intervals, non overlapping intervals for the best solutions or reaching a maximal total number of replications (resource-limited computing). **Figure 3** gives a simple view of such a Genetic Algorithm for Simulation Optimization (GASO).

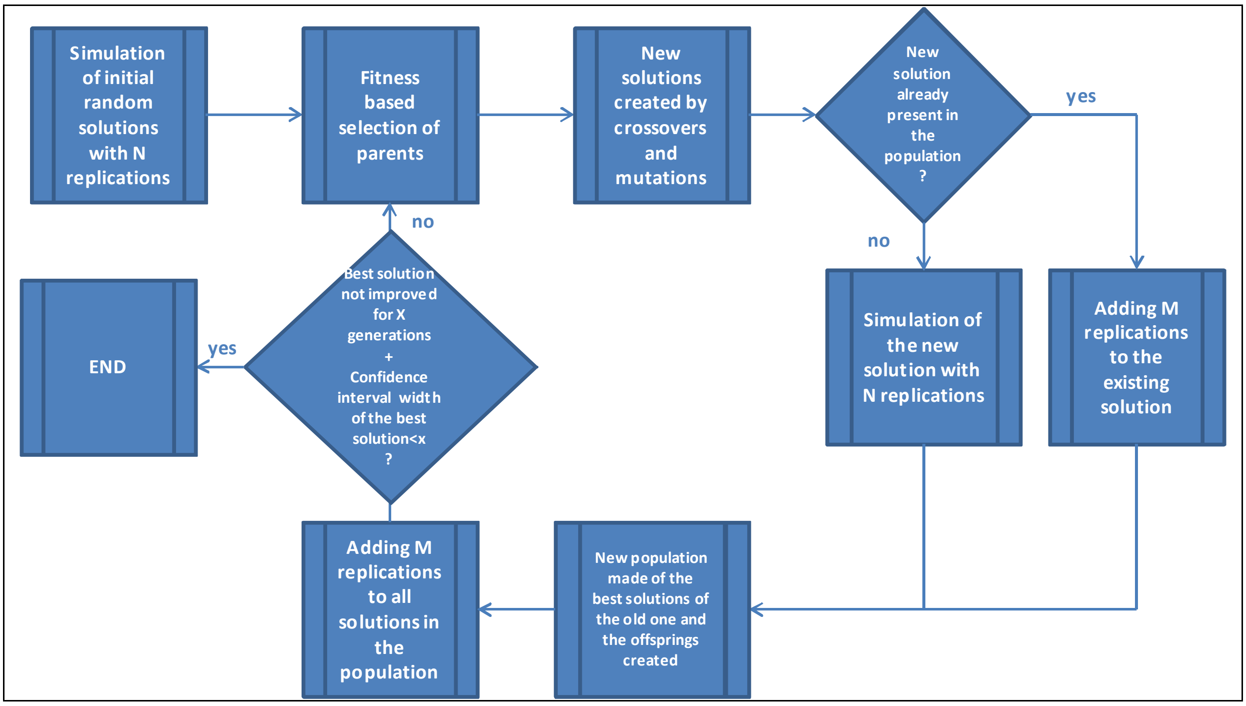

Figure 3 – Genetic Algorithm for Simulation Optimization (GASO)

Three specificities of this method, compared with regular GA, should be addressed.

- Parameters tuning: it is a well-known limitation of Genetic Algorithms that their efficiency is highly dependent on the different parameters like the size of the population or the mutation rate. No simple rule actually exists to evaluate these parameters according to the problem characteristics, so the right tuning of parameters often depends on the skills of the analyst. Adding two more parameters corresponding to the numbers of replications M and N makes the tuning more difficult.
- Goal function: in this method the fitness of a solution will change throughout the optimization process, as it is an estimator depending on the growing sample of replications. A solution then needs to be associated with more than one single piece of data per objective, the number of values needed will depend on the type of statistic. For a statistic like a probabilistic moment two values are sufficient as the couple (empirical value, number of replications) contains all the information needed to update the estimator when replications are added. On the opposite, using a quantile for objective is more complicated and it seems necessary to store the results of all replications, leading to memory issues.
- Confidence interval: as the value of a solution is estimated, solutions may be compared taking into account the accuracy of the estimators. Depending on the type of statistic the confidence interval may be more or less easily calculated for a given solution and updated when new replications are added. If the goal function is the mean value for instance, the solution in the population will need to be associated with the empirical standard-deviation so that an approximation of the confidence interval may be computed according to the central limit theorem. Once a confidence interval is available and easily updatable for the solutions it may be used at two different steps of the GA:
    1. <u>Selection of solutions candidates to cross-over</u>: whatever the selection method is, as long as it is based on a ranking of the population (that is to say all usual selection methods except the random one), a mathematical order should be defined. The question then comes whether solutions must be ranked according to the estimator of the goal function, regardless of the number of replications, or according to the

confidence interval bounds. These two different possibilities will be discussed on a test-case in §4.

2. Termination criterion: for regular GA the termination criterion is usually the fact that the best solution has not been improved for several generations. This kind of criterion proved to be not sufficient enough in the case of GASO as a solution with a very large confidence interval may dominate the population of solutions for many generations. This is the reason why the termination criterion for GASO is a double one with the best solution not improved for X generations **and** having a confidence interval width lower then Y%. Another possibility would have been that the top Z solutions have non-overlapping confidence intervals, but, as it will be discussed in §4.2., this criterion is often impractical.

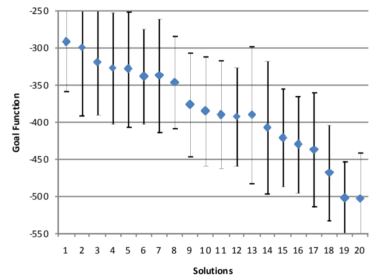

Figure 4 – Mean fitness and confidence intervals of the initial random population

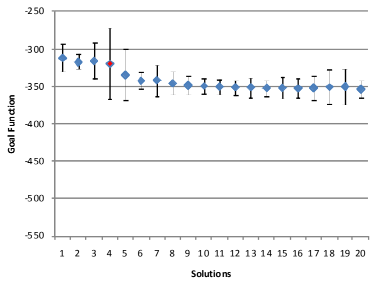

Figure 5 – Mean fitness and confidence intervals of the population after 500 generations

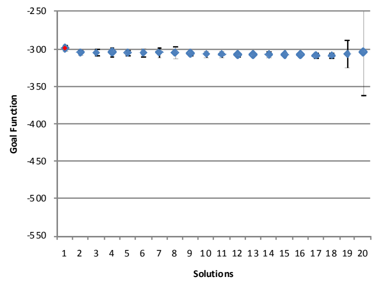

Figure 6 – Mean fitness and confidence intervals of the population when termination criterion is fulfilled

Figure 4 to **Figure 6** give an example of the evolution of a GA for a maximization problem. After 500 generations (**Figure 5**) the best mean fitness is actually worse than the best mean fitness of the initial population (**Figure 4**), but the average confidence interval width is much smaller. The best solution, at this point, does not have the narrowest confidence interval, showing that it is a "young" solution present in the population for few generations. When the algorithm terminates the top solutions all have narrow confidence intervals. This example also highlights the fact that the best solution may be present in the population before being considered optimal, as the red mark, representing the optimal solution, is identified as a good one but ranked number four after 500 generations (**Figure 5**).

4. TEST-CASE

4.1. Test-case description

The test case consists in finding the optimal preventive replacement planning for a repairable component. Both preventive and corrective actions are perfect maintenance. The component reliability is modeled by a Weibull distribution. All costs are discounted using a discount rate. The parameters of the test case are given in Table 2.

Parameter	Value
Scale parameter λ	0.05
Shape parameter β	2.3
Time horizon	40 years
Corrective cost	1000
Preventive cost	100
Discount Rate	7.5%/year

Table 2 - Asset parameters

As for the optimization problem, it is a cost minimization one with a risk constraint:

$$\begin{cases} \min Cost(x) \\ subject\,to\, \mathrm{Prob}(Cost(x) \geq 800) \leq 0.06 \end{cases} \quad (1)$$

The replacements schedules being annual ones, the size of the search space is $2^{40} \approx 1{,}1.10^{12}$.

The test-case aim is to study the impact of both the numbers of replications at each step of the GASO and the selection criterion based on the mean values of the different objectives or the upper bound of the confidence intervals.

4.2. GASO

As explained in §3.3, GASO is a regular GA with an iterative enhancement of the simulations convergence. For the regular part of the GA, the methods and parameters used for this test case are given in Table 3.

Method/Parameter	Value
Population size	20
Offspring per generation	1
Selection method	Tournament with three candidates
Selection elitism rate	0.9
Crossover method	Uniform
Crossover rate	0.7
Mutation method	Neighbors swap
Mutation rate	0.1
Evolution strategy	λ+μ

Table 3 – GA features

As for the GASO specific features, the aim of the test-case is to study the impact of both the numbers of replications and the selection criterion. The impact of the termination criterion was not studied in this test case. A limit on the total number of replications is applied to control the computing cost of the optimization.

Method/Parameter	Value
Initial number of replications (N)	100/1000/10000
Enhancement number of replications (M)	10/100/1000
Selection order	• Mean values • Upper bound of the 95% confidence interval
Termination criterion	**Dispersion (width of the confidence interval over the mean estimator) of the best solution lower 1%** **AND** **Best solution ranked first for at least 20 generations** OR **Total number of replications higher than 50.10^6**

Table 4 – GA features

The design of experiment is to run 10 trials for each of the 18 sets of parameters (initial number of replications, enhancement number of replications and selection order measures).

4.3. Proof of optimality and efficiency measure

It is a well-known limitation of GA that the convergence towards a global optimal solution is often difficult to prove. For GASO the difficulty is even higher as the fitness is based on an estimator and not the real value; if a GASO is run twice, two close solutions may actually be ranked differently if the convergence of the simulation is not sufficient. The number of replications needed to achieve, with a high confidence, an optimization problem with a flat optimal neighborhood may happen to be too important to be feasible. A practical method is to run the algorithm several times and to study the frequency of the different best solutions and the dispersion of their estimators. If no solution happens to appear as the best one, additional replications should be added to narrow the width of the confidence intervals.

Such an analysis have been performed for the test-case. **Figure 7** shows the different best solutions of all instances of the design of experiment, with the mathematical order used for the selection mechanism being the mean value. It represents 90 trials of the algorithm. Not all solutions found are shown, but only the ones with a confidence interval width lower than 1% of the mean estimator (some of the trials did not converge, the algorithm ending because it reached the maximum number of replications awarded to the calculation) and appearing more than once. This leads to eight different solutions. If their confidence intervals show some intersections, making impossible to demonstrate the dominance of one of the solution, the fact that the first one appears 12 times as the optimal one, the second best appearing only 3 times is a good indication that it is a good candidate to optimality. This indication is confirmed by the fact that the best (i.e. minimal) estimator value is found for this solution.

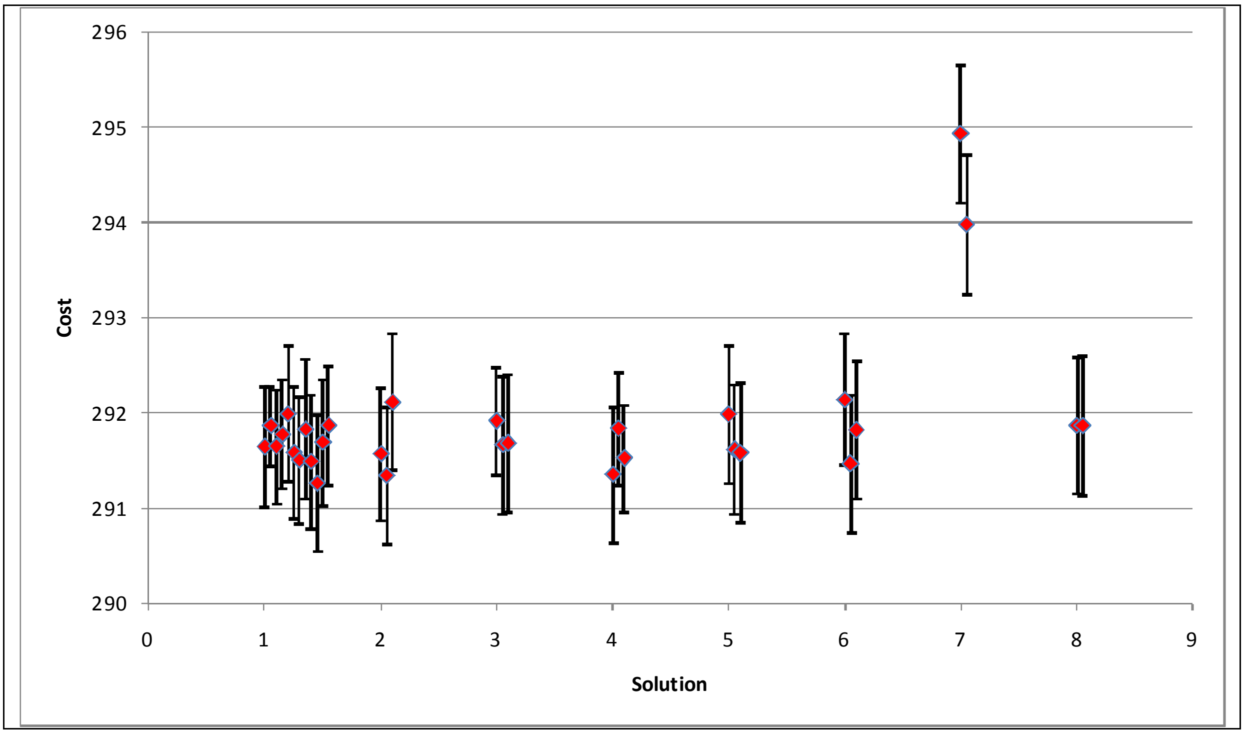

Figure 7 – Best solution dispersion for the test-case with their 95% confidence intervals

This solution corresponds to replacements scheduled at years (5, 12, 19, 26, 33). It is then used as the reference one (and called optimal solution) to evaluate the efficiency of each set of parameters of the design of experiment. This efficiency will be measured by counting the number of trials for which the algorithm finds this solution, and the number of trials for which this solution is present in the top ten solutions of the population.

The optimal replacement schedule without risk constraint is (8, 16, 24, 32). It was also obtained using a GASO. **Table 5** presents the valuation for both solutions with 10^6 replications. The estimated cumulative distribution function is shown in Figure 8: it clearly shows that the optimal constrained solution has a higher minimal cost, corresponding to the preventive replacements without any failure (five replacements instead of four), leading to a better control of the failure risk and the respect of the constraint.

	Replacement schedule (5, 12, 19, 26, 33)	**Replacement schedule (8, 16, 24, 32)**
Mean cost	291.9±0.5	284±0.5
Prob(C>800)	0.0567±0.0001	0.0719±0.0002

Table 5 – Evaluation of optimal schedules with and without constraint

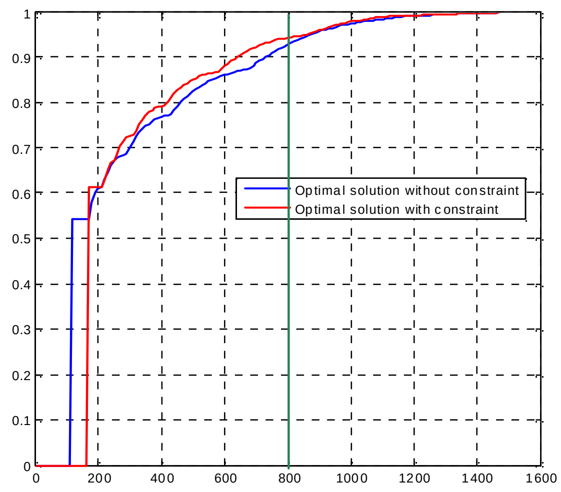

Figure 8 - Cumulative Distribution Function of the cost for both optimal strategies

4.4. Results

Table 6 and Table 7 present the number of successes of the algorithm to converge towards the optimal solution, out of ten trials. The first conclusion that can be made is that using the upper bound of the confidence interval is not efficient at all, and that it is better not to take into account the convergence of the simulations to compare two candidates in the selection step of the GASO. Using the mean value of the estimator presents a maximal frequency of success of 40% with N=100 and M=1000.

Low numbers of replications are not efficient because low convergence tends to create singular values of estimators very far from the real values. Such solutions will be at first considered as good candidates, dominating the population and expelling other good solutions, until its estimators start to converge towards its real value and is, in turn, expelled by a new solution. A detailed analysis of the evolution of such a case actually showed a cycle of the solutions, with specific solutions appearing and disappearing several times during the optimization.

As for high numbers of replications, they are not efficient because the maximal total number of replications awarded (50.10^6) is reached before the optimal termination criterion is met.

		Initial number of replications (N)		
		100	1000	10000
Enhancement number of replications (M)	10	0	3	0
	100	4	3	2
	1000	1	1	3

Table 6 – Efficiency of the algorithm to converge towards the optimal solution with the mean value as the mathematical order for selection mechanism

		Initial number of replications (N)		
		100	1000	10000
Enhancement number of replications (M)	10	0	0	0
	100	0	0	1
	1000	0	0	0

Table 7 – Efficiency of the algorithm to converge towards the optimal solution with the upper bound of the confidence interval as mathematical order for selection mechanism

Even if using the mean value, no matter the number of replications, as a selection indicator is better, it still gives mixed results. This can be explained by the fact that the dispersion of 1% considered for the termination criterion is not small enough to rank the solutions with no doubt because of overlapping confidence intervals, as illustrated in **Figure 7**. Narrowing the width of confidence intervals for the termination criterion would lead to computing durations making the calculation impractical. Another way to avoid the non-convergence of the GASO is to terminate the algorithm with the criterion defined in §4.2 and then to add replications to all solutions present at the last generation, without applying the GA mechanisms (static population), until the confidence intervals lengths drop down to 0.1%. The efficiency of the algorithm after this post-treatment is given in Table 8 and Table 9.

		Initial number of replications (N)		
		100	1000	10000
Enhancement number of replications (M)	10	1	6	7
	100	9	10	9
	1000	1	3	4

Table 8 – Efficiency of the algorithm to converge towards the optimal solution after post treatment with the mean value as the mathematical order for selection mechanism

		Initial number of replications (N)		
		100	1000	10000
Enhancement number of replications (M)	10	0	1	9
	100	0	1	7
	1000	0	0	0

Table 9 – Efficiency of the algorithm to converge towards the optimal solution after post treatment with the upper bound of the confidence interval as the mathematical order for selection mechanism

These last results confirm that using the upper bound of the confidence interval as an order for the selection mechanism is not as effective as using the mean estimator. If M=100 appears to be an optimal value, it is not as clear for N with an efficiency between 90% and 100%.

One of the criteria that could be used to decide between the different values of N would be the total number of replications during the GASO as computing duration is often at stake in this kind of problem (cumulative number of replications for all solutions simulated throughout all generations of the algorithm). Table 10 shows that choosing N=100 is one third less expensive with very close success frequency.

		Initial number of replications (N)		
		100	1000	10000
Enhancement number of replications (M)	10	5756012	21035744	50005685
	100	19190330	30544850	50006930
	1000	23068760	29321300	31433300

Table 10 – Average total number of replications (using the mean value as the order measure)

This work is just a first step, applying the GASO methodology on a very simple test-case, and it seems difficult to edict generic rules based on this unique example, although the following assumptions may be stated:

1. The fitness function, used to compare and select solutions in the GA, should take into account mean estimators without consideration of their convergence.
2. Number of replications should be chosen carefully as large values of M and N will lead to a very local exploration of the search space because of the replications budget limit applied to ensure the tractability of the method. On the opposite, small values of M and N will lead to a global search based on inaccurate estimators.
3. The termination criterion does not have to be too strict on simulations convergence as it can be improved after the GA has terminated.

It would be interesting to apply the GASO to a set of various problems in order to try to identify a relation between the parameters of the algorithm and some specific data, such as the order of magnitude of a random solution, its dispersion, the size of the space-search...

5. CONCLUSION

This paper presented a methodology to couple GA with Monte-Carlo simulation function when addressing Simulation Optimization issues in the field of engineering asset management. The test-case presented here proved the GASO to be an effective answer to realistic issues a system engineer or a business manager could face. If the method may be easily implemented in any EAM tool the success of a study will depend on the user skill to configure the calculations. As a matter of fact, even if the studied example showed some interesting results, such as the existence of an optimal number of replications for each simulation, or the efficiency of a post treatment of the best solutions to narrow the confidence intervals and then to finalize the optimization, it is not sufficient to edict generic rules, even empiric ones, to tune the parameters of a GASO.
The next step would be to apply this algorithm to different and more complex cases to confirm its efficiency and try to build a rule linking the parameters to the characteristics of the problem.

References

[1] A comprehensive literature classification of simulation optimisation methods, Wak Hachicha et al., Munich Personal RePEc Archive (2010)
[2] A hybrid simulation optimization method for production planning of dedicated remanufacturing , Jianzhi Li _, Miguel Gonzalez, Yun Zhu, International Journal of Production Economic, 117 (2009) 286–301
[3] Multi-response simulation optimization using genetic algorithm within desirability function framework, Seyed Hamid Reza Pasandideh, Seyed Taghi Akhavan Niaki, Applied Mathematics and Computation 175 (2006) 366–382

[4] Optimal computing budget allocation for Monte-Carlo simulation with application to product design. Chen, C. H., Yücesan, E., Lin, J., & Donohue, K., Simulation Modeling Practice and Theory 11 (2009), 57–74.
[5] Evolutionary algorithm assisted by surrogate model in the framework of ordinal optimization and optimal computing budget allocation, Shih-Cheng Horng, Shin-Yeu Lin, Information Sciences, Volume 233, (2013), Pages 214-229
[6] Condition-based maintenance optimization by means of genetic algorithms and Monte Carlo simulation, Marzio Marseguerra, Enrico Zio, Luca Podofillini, Reliability Engineering and System Safety 77 (2002) 151–166
[7] Stock effects on exceptional maintenance tasks taking into account risks, J. LONCHAMPT et al., 2006 ASME Pressure Vessel and Piping Division Conference (2006).
[8] Genetic algorithms in optimizing surveillance and maintenance of components, Munoz A, Martorell S, Serradell V., Reliability Engineering Systems Safety 57 (1997) 107–20.
[9] Adaptation in Natural and Artificial Systems, J. H. Holland, University of Michigan Press, Ann Arbor, MI (1975)
[10] Genetic Algorithms in Search, Optimization, and Machine Learning, David Goldberg, Addison-Wesley Professional, (1989)
[11] On the use of genetic algorithm to optimize industrial assets lifecycle management under safety and budget constraints, J. Lonchampt and K. Fessart, Mathematics and Computation (2013)