

A Hybrid Method For Optimizing A Complex System Reliability

Wafae El Alem^{1,2,a}, Abdelkhalak El Hami², Rachid Ellaia¹, Mohammed Souissi¹

¹Laboratory of Study and Research for Applied Mathematics, Mohammed V university— Engineering Mohammedia School, Rabat, BP. 765, Ibn Sina avenue, Agdal, Morocco.

²Laboratory of Mechanics of Rouen, National Institute for Applied Sciences - Rouen, BP 08, avenue de l'université 76801, St Etienne du Rouvray Cedex, France.

Received 10 December 2009, Accepted 05 March 2010

Abstract- In this paper we present a new hybrid method for maximizing a complex system reliability. The hybridization of two methods, namely the Penalty Simultaneous Perturbation Stochastic Approximation (PSPSA) and the heuristic method Sharma-Venkateswaran allowed as to design a new method, called Penalty Simultaneous Perturbation Stochastic Approximation Restarted with Sharma-Venkateswaran (PSPSARSV) method, which is expected to have some ability to look for a global minimum. The concept of this hybrid method, in which both stochastic and heuristic methods are combined, is based on the view that, as stochastic search algorithms are good for finding global optima but slow at converging, while heuristic search algorithms are good at fine-tuning but often converge on local optima, the hybrid algorithm can compensate for their individual shortcomings and outperform either one individually. The proposed hybrid method is applicable for any system configuration where subsystems are composed of identical parallel elements. PSPSARSV is simple, fast and capable of handling problems subject to any number of constraints which are not necessarily linear. An example of bridge complex system illustrates the method.

Key words: Reliability; Optimization; Redundancy; Heuristic; Stochastic.

1 Introduction

Reliability optimization plays an important role in the planning and design of modern technological systems. As it is well-known reliability of an overall system can be increased by adding redundant components or increasing the reliability levels of subsystems, but any kind of redundancy imposes many difficulties such as increase in system cost, weight or power.

Decision makers are often faced with the problem of maximizing the reliability of a system subject to certain resource constraints which in general leads to an integer nonlinear programming problem. Many authors [10], [1] have developed different algorithms for solving this problem and several heuristic, dynamic programming, linear programming, non-linear programming with integer variables, geometric programming methods were well summarized in [13].

All these methods look at the same problem in different ways and are quite different in how they carry out their search toward an optimal solution, but we believe

that methods collaborating in some way have all chances to give a better performance than the methods on their own.

In this paper we present a new hybrid algorithm implemented with Matlab which can solve the problem of constrained redundancy optimization in complex systems.

2 Notation

n : number of subsystems
 x : vector of design variables or redundant components which should be integers (x_1, x_2, \dots, x_n)
 x_i : number of redundant components in subsystem i
 K_j : total available resource for constraint j
 R_i : reliability of a single component of subsystem i
 $R_i(x_i)$: reliability of subsystem i with x_i components
 R_s : system reliability
 $g_{ij}(x_i)$: resource j consumed in subsystem i with x_i components.

^aCorresponding author: welalem@yahoo.fr

3 Problem statement

Assumptions

- (1) There are n subsystems in the system which may be in series or complex.
- (2) All the components used in each subsystem are statistically independent.
- (3) All components in parallel in the same subsystem are identically distributed.
- (4) All components in each subsystem are simultaneously working and for a subsystem to fail all the components must fail.

The redundancy allocation problem is to choose x so as to maximize system reliability R_s ,

$$\begin{cases} \text{Maximize } R_s = R_s(R_1(x_1), R_2(x_2), \dots, R_n(x_n)) \\ \text{Subject to : } \sum_{i=1}^n g_{ij}(x_i) \leq K_j, & \text{for all } j \\ \text{where each } x_i \text{ has to be a positive integer.} \end{cases}$$

In order to solve this problem, we present three methods to run on a complex bridge network system. the three methods are:

- PPSA
- Sharma and Venkateswaran
- PPSARSV

3.1 PPSA Method

This method centers on solving the problem with SPSA by focusing on a penalty function method. As it's well-known SPSA is an unconstrained stochastic method and since many optimization problems, including the redundancy allocation problem, lead to a constrained nonlinear integer problems we have used a penalty function, added to the loss function to penalize violations of the constraints either of type integer or inequality. That we have called PPSA (Penalty Simultaneous Perturbation Stochastic Approximation).

In general, a constrained nonlinear integer programming problem can be written as follows.

$$\begin{cases} \text{Minimize } f(x), & x \in \mathbb{R}^n \\ \text{Subject to : } c_j(x) \leq 0, & j = 1, \dots, p \\ & x_i \leq b_i, & i = 1, \dots, n \\ & x_i, \text{ integer,} & i = 1, \dots, n \end{cases} \quad (1)$$

3.1.1 Penalty Function

Assume that $f(x)$ and $c_j(x)$ are twice continuously differentiable, and define the following sets

$$S_1 = \{x \in \mathbb{R}^n \mid c_j(x) \leq 0, j = 1, 2, \dots, p,$$

$$x_i \leq b_i, i = 1, \dots, n\}$$

$$S_2 = \{x \in \mathbb{R}^n \mid x_i \text{ is integer, } i = 1, 2, \dots, n\}$$

Definition 3.1 x' is an integer point if its components x'_i ($i = 1, \dots, n$) are all integers. For an integer point x' , the set $N(x') = \{x : \|x - x'\|_\infty \leq \frac{1}{5}\}$ is called an $\frac{1}{5}$ -cubic neighborhood of the integer point x' .

We construct a penalty function

$$\begin{aligned} \phi_{\mu,k}(x) = f(x) + \mu \sum_{j=1}^p \max(0, c_j(x))^2 \\ - k \sum_{i=1}^n \cos 2\pi x_i \end{aligned} \quad (2)$$

Then we have

Theorem 3.1 Suppose that μ and k are large enough and satisfies some conditions (see for instance [8], [6]), if the global minimizer x^* of $\phi_{\mu,k}(x)$ is in a $\frac{1}{5}$ -cubic neighborhood of an integer point $\hat{x} \in S_2$, then \hat{x} is a solution of (1).

After penalizing the problem (1) the use of the SPSA method become perfectly possible, since the obtained problem (2) is an unconstrained one.

3.1.2 SPSA Method

a) Introduction

Stochastic Approximation(SA) has long been applied for problems of minimizing loss functions or root finding when only noisy measurements of the objective function are available.

In the following we will focus on SPSA for reasons of its relative efficiency. SPSA is based on a highly efficient and easily implemented simultaneous perturbation approximation to the gradient: this gradient approximation uses only two loss-function measurements, independent of the number of parameters being optimized.

b) Background

Consider the problem of minimizing a (scalar) differentiable loss function $f(x)$, where $x \in \mathbb{R}^n$, $n \geq 1$, and where the optimization problem can be translated into finding the minimizing x^* such that the gradient $g(x^*) = 0$. It is assumed that measurements of $f(x)$ are available at various values of x , these measurements may or may not include added noise. No direct measurements of $g(x)$ are assumed available. The recursive procedure we consider is in the general SA form:

$$\hat{x}_{k+1} = \hat{x}_k - a_k \hat{g}_k(\hat{x}_k) \quad (3)$$

where $\hat{g}_k(\hat{x}_k)$ is the estimate of the gradient ∇f at the iterate \hat{x}_k . The essential part of Eq. (3) is the gradient approximation $\hat{g}_k(\hat{x}_k)$. We discuss below the form that have attracted the most of attention that is of Simultaneous Perturbation.

- Simultaneous Perturbation (SP) :

Let $y(\cdot)$ denote a measurement of $f(\cdot)$ at a design level represented by the dot [i.e., $y(\cdot) = f(\cdot) + \text{noise}$], and c_k , be some (usually small) positive number. All elements of \hat{x}_k are randomly perturbed together to obtain two measurements $y(\cdot)$, but each component of $\hat{g}_k(\hat{x}_k)$ is formed from a ratio involving the individual components in the perturbation vector and the difference in the two corresponding measurements.

$$\hat{g}_{ki}(\hat{x}_k) = \frac{y(\hat{x}_k + c_k \Delta_k) - y(\hat{x}_k - c_k \Delta_k)}{2c_k \Delta_{ki}} \quad (4)$$

where the distribution of the user-specified random perturbations for SP,

$\Delta_k = (\Delta_{k1}, \Delta_{k2}, \dots, \Delta_{ki}, \dots, \Delta_{kn})^T$, satisfies conditions that will be mentioned later.

c) Convergence of the SPSA Algorithm

As with any optimization algorithm, it is of interest to know whether the iterate x_k will converge to x^* as k gets large. In fact, one of the strongest aspects of SA is the rich convergence theory that has been developed over many years. Researchers and analysts in many fields have noted that if they can show that a particular stochastic optimization algorithm is a form of SA algorithm, then it

may be possible to establish formal convergence. Note that since we are in a stochastic context, convergence is in a probabilistic sense. In particular, the most common form of convergence established for SA is in the almost sure (a.s.) sense.

The SPSA algorithm works by iterating from an initial guess of the optimal x , where the iteration process depends on the above-mentioned simultaneous perturbation stochastic approximation to the gradient $g(x)$. The form of the SPSA gradient approximation was presented above.

J. Spall (1992) presents sufficient conditions for convergence of the SPSA iterate ($\hat{x}_k \rightarrow x^*$ a.s.). In particular, we must impose conditions on both gain sequences (a_k and c_k), the user-specified distribution of Δ_k and the statistical relationship of Δ_k to the measurements $y(\cdot)$. The main conditions are that (a_k and c_k) both go to 0 at rates neither too fast nor too slow, that $f(x)$ is sufficiently smooth (several times differentiable) near x^* , and that the Δ_{ki} are independent and symmetrically distributed about 0 with finite inverse moments $E(|\Delta_{ki}^{-1}|)$ for all k, i . One particular distribution for Δ_{ki} that satisfies these latter conditions is the symmetric Bernoulli ± 1 distribution; two common distributions that do not satisfy the conditions (in particular, the critical finite-inverse-moment condition) are the uniform and the normal.

Although the convergence result for SPSA is of some independent interest, the most interesting theoretical results in Ref. [11], and those that best justify the use of SPSA, are the asymptotic efficiency conclusions that follow from an asymptotic normality result. It can be shown that

$$k^{\beta/2}(\hat{x}_k - x^*) \rightarrow N(\mu, \Sigma) \quad \text{as } k \rightarrow \infty \quad (5)$$

where $\beta > 0$ depends on the choice of gain sequences (a_k and c_k), μ depends on both the Hessian and the third derivatives of $f(x)$ at x^* , and Σ depends on the Hessian matrix at x^* .

d) Algorithm

Step 1: Initialization and coefficient selection.

Set counter index $k = 1$. Pick initial guess and non-negative coefficients a, c, A and γ in the SPSA gain sequences $a_k = a/(A + k)^\alpha$ and $c_k = c/k^\gamma$. The choice

of the gain sequences (a_k and c_k) is critical to the performance of SPSA (as with all stochastic optimization algorithms and the choice of their respective algorithm coefficients).

Step 2: Generation of the simultaneous perturbation vector.

Generate a n -dimensional random perturbation vector Δ_k , where each of the n components of Δ_k are independently generated. A simple (and theoretically valid) choice for each component of Δ_k is to use a Bernoulli ± 1 distribution.

Step 3: Loss function evaluations.

Obtain two measurements of the loss function $y(\cdot)$ based on the simultaneous perturbation around the current \hat{x}_k : $y(\hat{x}_k + c_k\Delta_k)$ and $y(\hat{x}_k - c_k\Delta_k)$ with the c_k and Δ_k from Steps 1 and 2.

Step 4: Gradient approximation.

Generate the simultaneous perturbation approximation to the unknown gradient $g(\hat{x}_k)$:

$$\hat{g}_k(\hat{x}_k) = \frac{y(\hat{x}_k + c_k\Delta_k) - y(\hat{x}_k - c_k\Delta_k)}{2c_k} \begin{bmatrix} \Delta_{k1}^{-1} \\ \vdots \\ \Delta_{kn}^{-1} \end{bmatrix}$$

where Δ_{ki} is the i^{th} component of the Δ_k vector (which may be ± 1 random variables as discussed in Step 2);

Step 5: Updating x estimate.

Use the standard SA form :

$$\hat{x}_{k+1} = \hat{x}_k - a_k \hat{g}_k(\hat{x}_k)$$

to update \hat{x}_k to a new value \hat{x}_{k+1} .

Step 6: Iteration or termination.

Return to Step 2 with $k + 1$ replacing k . Terminate the algorithm if there is little change in several successive iterates or the maximum allowable number of iterations has been reached.

3.2 Sharma and Venkateswaran Method

Step 1 : Assign $x_j = 1$ for $j = 1, 2, \dots, n$. There must be at least one component at each stage and also the

system should not violate any constraints.

Step 2 : Find the stage which is the most unreliable. Add a redundant component to that stage.

Step 3 : Check the constraints :

- (a) If any constraint is violated, go to Step 4.
- (b) If no constraint has been violated, go to Step 2.
- (c) If any constraint is exactly satisfied, stop.

The current x_j 's are then the optimum configuration of the system.

Step 4 : Remove the redundant component added in Step 2. The resulting number is the optimum allocation for that stage. Remove this stage from further consideration.

Step 5 : If all stages have been removed from consideration, the current x_j 's are then the optimum configuration of the system. Otherwise, go to Step 2.

4 The proposed method

In this paper we present a new hybrid method viz, PSP-SARSV, its algorithm can be shown as follows :

Step 1 : Assign $x_j = 1$ for $j = 1, 2, \dots, n$. x should be a feasible integer solution.

Step 2 : Use PSPSA to solve the problem

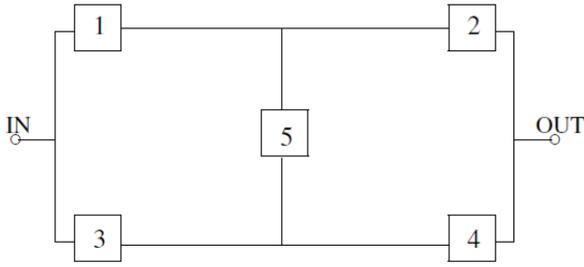
Step 3 : Keep the solution found by PSPSA and start Sharma-Venkateswaran's approach with this one.

Step 4 : If all stages have been removed from consideration, or if any constraint is exactly satisfied the current x_j 's are then the optimum configuration of the system, stop.

5 NUMERICAL EXAMPLE

5.1 Positioning of the Problem

It is about given a model for the bridge system represents below, Since the components can not be repaired the method of the reliability diagram is revealed the most adequate so much by its representation or by methods of calculation that it presents.



(a) Design domain

Fig. 1: A complex bridge network system

Suppose that the reliability of every component is known, a mathematical model of optimization with an objective function and constraints adapted to the present problem is given as follows.

5.1.1 Assessment of the Objective Function

To evaluate the reliability of the system in Figure 1, we have used the decomposition method.

o Decomposition Method

The decomposition method is an application of the law of total probability. It involves choosing a "key" component and then calculating the reliability of the system twice: once as if the key component failed and once as if the key component succeeded. These two probabilities are then combined to obtain the reliability of the system, since at any given time the key component will be failed or operating. Using probability theory, the equation is:

$$R_s = P(S \cap A) + P(S \cap \bar{A})$$

$$= P(S \setminus A)P(A) + P(S \setminus \bar{A})P(\bar{A})$$

Now consider our five bridge system. First, select a "key" component for the system.

Selecting Unit 5, the probability of success of the system is:

$$R_{sys} = P(S \setminus R_5)R_5 + P(S \setminus \bar{R}_5)(1 - R_5)$$

If Unit 5 is good, then:

$$P(S \setminus R_5) = (1 - (1 - R_1)(1 - R_3))(1 - (1 - R_2)(1 - R_4))$$

If Unit 5 fails, then:

$$P(S \setminus \bar{R}_5) = 1 - (1 - R_1 R_2)(1 - R_3 R_4)$$

Hence, the reliability of our system is :

$$R_{sys} = R_5(1 - (1 - R_1)(1 - R_3))(1 - (1 - R_2)(1 - R_4))$$

$$+ (1 - R_5)(1 - (1 - R_1 R_2)(1 - R_3 R_4))$$

By adding similar redundant components we obtain the final expression of our objective function.

$$R_{sys} = (1 - (1 - R_5)^{x_5})(1 - (1 - R_1)^{x_1}(1 - R_3)^{x_3})$$

$$(1 - (1 - R_2)^{x_2}(1 - R_4)^{x_4}) + (1 - R_5)^{x_5}$$

$$[1 - (1 - (1 - (1 - R_1)^{x_1})(1 - (1 - R_2)^{x_2}))$$

$$(1 - (1 - (1 - R_3)^{x_3})(1 - (1 - R_4)^{x_4}))].$$

5.1.2 Constraints

o Inequality Constraints

In general reliability's constraints are about cost, weight and power. In our problem we only have one constraint of type inequality and it's about cost [1], it has the form as shown below

$$2x_1 + 3x_2 + 2x_3 + 3x_4 + x_5 \leq 20$$

o Integer constraints

All the components of the vector x should be integer.

5.2 Final Formulation

The reliability optimization problem in complex system can be formulated as a constrained nonlinear programming problem:

$$\begin{cases} \text{Maximize } R_{sys} \\ \text{Subject to : } 2x_1 + 3x_2 + 2x_3 + 3x_4 + x_5 \leq 20 \\ x_i \text{ is integer for } i=1, \dots, 5 \end{cases}$$

The optimum allocations are given in Table 1

5.3 Results and Discussion

The optimal solution provided by PSPSA is (2,2,2,1,1) with system reliability 98.81 and by sharma and vankateswaran is (2,1,2,2,3) with system reliability 98.84. Whereas PSPSARSV provided the solution (3,2,2,1,1) with system reliability 99.32. It's noted that the solution of the same example in the literature [1] gives

the system configuration (3,1,2,2,1) with system reliability 99.14, which is inferior to the solution obtained by our hybrid method.

Table 1: Performance of PPSARSV on five bridge system

Methods	x_1	x_2	x_3	x_4	x_5	optimum
PPSPA	2	2	2	1	1	98.81
Sharma and Venkateswaran	2	1	2	2	3	98.84
PPSARSV	3	2	2	1	1	99.32
Aggarwal [1]	3	1	2	2	1	99.14

6 Conclusion

According to the results found above, one can claim that our hybrid method gives more robust results. Effectively it does, since the acknowledged strengths of each approach could ideally compensate for the weaknesses in the other, however we should pay very attention. For successful collaboration to take place, the two methods need to complement each other in solving the same problem, often this means that they need to be sufficiently different approaches to provide different views on the problem. However, they also need to be sufficiently similar so that the shared information can be used easily.

Acknowledgements

This work was supported by CMIFM, A.I. EGIDE, Numero: MA/07/173.

References

1. K.K. Aggarwal, *Redundancy optimization in general systems*, IEEE Transactions on Reliability, **25**, 330-332 (1976).
2. O. Bendaou, A. El Hami, A. Aannaque and M. Agouzoul, *Calculation time optimization for stochastic analysis of an industrial structure*, Int. J. Simul. Multidisci. Des. Optim., **2(2)**, 135-141 (2008).
3. G. Kharmanda, A. Mohsine, A. Makloufi, A. El Hami, *Recent methodologies for reliability-based design optimization*, Int. J. Simul. Multidisci. Des. Optim., **2(1)**, 11-24 (2008).
4. N.L. Kleinman, S.D. Hill, V. A. Ilenda, *SPSA/SIMMOD Optimization of Air Traffic Delay Cost*, American Control Conference Proceedings of the 1997, **2**, 1121- 1125 (1997).
5. N.L. Kleinman, S.D. Hill, V. A. Ilenda, *Simulation Optimization of Air Traffic Delay Cost*, Simulation Conference Proceedings, **2**, 1177-1181 (1998).
6. D.G. Luenberger, *Introduction to Linear and Nonlinear Programming*, Addison Wesley, (1973).
7. J.L. Maryak and J.C. Spall, *Simultaneous Perturbation Optimization for efficient Image Restoration*, IEEE Transactions on Aerospace and Electronic Systems, **41**, 356-361 (2005).
8. G. Renpu and C. Huang, *A Continuous Approach to Nonlinear Integer Programming*, Applied Mathematics and Computation, **34**, 39-60 (1989).
9. J.E. Rojas, A. El Hami and D.A. Rade, *Reliability analysis based on gradient and heuristic optimization techniques of composite laminates using element-free Galerkin method*, Int. J. Simul. Multidisci. Des. Optim., **2(2)**, 157-169 (2008).
10. J. Sharma and K.V. Venkateswaran, *A direct method for maximizing the system reliability*, IEEE Transaction on Reliability, **20**, 256-259 (1971).
11. J.C. Spall, *Multivariate Stochastic Approximation using a Simultaneous Perturbation Gradient Approximation*, IEEE transactions on automatic control, **37**, 332-341 (1992).
12. J.C. Spall, *Adaptive Stochastic Approximation by the Simultaneous Perturbation Method*, IEEE transactions on automatic control, **45**, 1839-1853 (2000).
13. A. Tillman, C. Hwang, W. Kuo, *Optimization of Systems Reliability*, Industrial Engineering, (1980).