Comparison of Simulated Annealing and Particle Swarm Optimization on Reliability-Redundancy Problem

Martin Bagaram

Comparison of Simulated Annealing and Particle Swarm Optimization on Reliability-Redundancy Problem

Martin B. Bagaram
University of Washington

IND E 516: Assignment 3

March 15, 2018

Abstract

Reliability-redundancy is a recurrent problem in engineering where designed systems are meant to be very reliable. However, the cost of manufacturing very high reliability components increases exponentially, therefore redundancy of less reliable components is a palliative solution. Nonetheless, the question remains how many components of low reliability (and of what extent of reliability) should be coupled to produce a system of high reliability. In this paper, I compare the performance of particle swarm optimization (PSO) and simulated annealing (SA) on a system of electricity distribution in a rural hospital. The results proved that PSO outperformed SA. In addition, considering the problem as reliability maximization and cost minimization bi-objective give a useful insight on how the cost increase exponentially at a certain given reliability of the system.

1. Introduction

Reliability-redundancy allocation problems is a recurrent problem in engineering. The layout of the problem in this paper is sometimes known as "**Complex Bridge System**." The objective of the design is to produce a very reliable system at a minimum cost. This can be achieved by either using more reliable material an/or using redundant material in parallel. The increase of the redundancy comes with the increase of the cost, the volume, and the weight of the system. It is therefore necessary to specify the maximum redundancy acceptable in the system. The challenge is to find the redundancy that is acceptable and will maximize the total reliability of the system even with components that have limited reliability. The advantage of the redundancy is that it gives guaranty that the whole system will continue operating even if one component fails.

In this exercise, I consider the problem as an application of electricity distribution to a rural hospital from a power plant located in city. However, the power line follows a circuit on which there are couple of villages that need power as well. The power company therefore has to build power transformers in vicinity of each village. The main objective is to maximize the reliability that the power reaches the hospital without failure (or limited failure). The design of the network is imposed by villages needing electricity. The figure 1 displays the layout of the distribution design. However, the power company can increase the reliability of the system by coupling many power transformations if necessary or buying very reliable ones or both. The objective is to find the reliability of each component and how many of each component should be coupled in parallel. I test simulated annealing and particle swarm optimization implementations to meet the aforementioned objective. I also explore the bi-objective approach in order to find the efficient Pareto frontier for the cost minimization and reliability maximization.

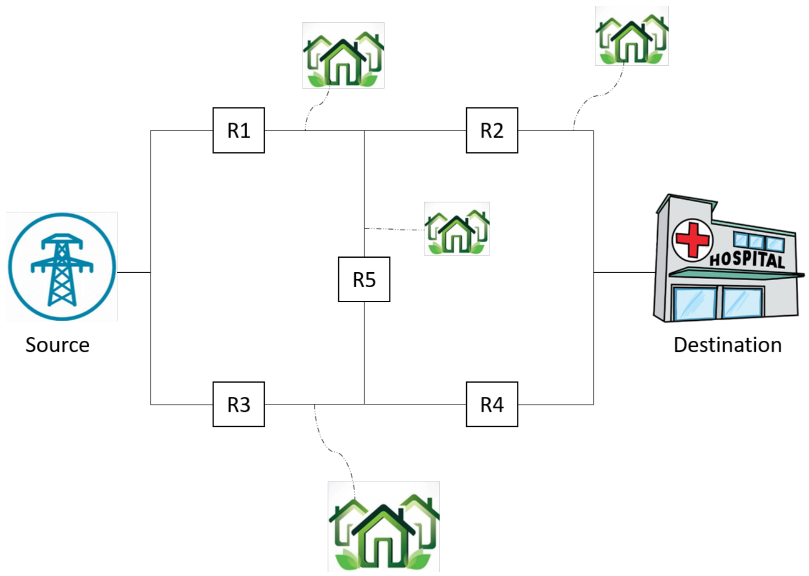

Figure 1: *Power distribution layout*

2. Mono-objective formulation

The problem used in this paper has been presented in several papers including [1] and solved using different algorithms ranging from particles swarm optimization to genetic algorithms.

$$\begin{aligned} \text{Maximize} \qquad R_s = {} & R_1R_2 + R_3R_4 + R_1R_4R_5 + R_2R_3R_5 \\ & -R_1R_2R_3R_4 - R_1R_2R_3R_5 - R_1R_2R_4R_5 \\ & -R_1R_3R_4R_5 - R_2R_3R_4R_5 + 2R_1R_2R_3R_4R_5 \end{aligned} \tag{1}$$

subject to:

$$g_1(r,n) = \sum_{i=1}^{m} w_i v_i^2 n_i^2 \leq V \tag{2}$$

$$g_2(r,n) = \sum_{i=1}^{m} \alpha_i \left(-\frac{T}{\log r_i}\right)^{\beta_i} [n_i + \exp(0.25 n_i)] \leq C \tag{3}$$

$$g_3(r,n) = \sum_{i=1}^{m} w_i n_i \exp(0.25 n_i) \leq W \tag{4}$$

$$R_i = 1 - (1 - r_i)^{n_i} \tag{5}$$

$$0 \leq r_i \leq 1 \qquad r_i \in \mathbb{R} \qquad \text{and} \qquad 1 \leq n_i \leq 5 \qquad n_i \in \mathbb{Z} \tag{6}$$

The equations 2, 3 and 4 are constraints about the system volume, the cost and the weight,

respectively. The equation 2 can also be seen as the combination of redundancy/volume and weight constraint. It imposes the total volume of materials composing the system. The values of parameters used in the formulation are given in Table 1. They have been gathered from [2] and [1].

Table 1: *Summary of parameters used in the study*

Parameter	Definition	Value
V	The upper limit on the sum of the subsystems' product of volume and weight	110
W	The upper limit of the weight of the system	200
C	The upper limit on the cost of the system	175
T	Time during which the material should not fail	1000
r_i	Reliability of the component i	-
n_i	The number of components in the i^{th} subsystem (redundancy)	$1 \leq i \leq m$
w_i	The weight of each component in the subsystem i	[7, 8, 8, 6, 9]
v_i	Volume of each component in the subsystem i	[1, 2, 3, 4, 2] [1]
β_i and α_i	Physical characteristics of the system components	
m	Number of subsystems in the system	5
R_i	Reliability of the subsystem i	-

3. Resolution algorithms

3.1 Particle Swarm Optimization (PSO)

The PSO optimization is a stochastic global optimization method. It is inspired from the behavior of some schooling animals such as birds and fish. The algorithm can be summarized as follows: It starts by initializing a given number of particles randomly over a searching space. The particles moves with a velocity and find the global best position after a number of iterations. At each iteration, each particle adjust its velocity based on its best position (p_{best}) as well as the best position of its neighbors (g_{best}) and then compute the new position that the particle moves to. If the new position is better than the previous p_{best} then updates the p_{best}. Similarly if the new position is better than the best global position g_{best} then update the g_{best}.
In computation, the number of particles was set to 100 and the number of iteration was set to 100 as well. The stopping criteria were either the number of iterations or if there is no improvement higher than 10^{-8} from an iteration to another. The package used is

[1] The values correspond to $w_i v_i^2$

provided by *pyswarm*[2] implemented in Python. The package has the ability to handle constraints.

3.2 Simulated Annealing

Simulated annealing is a global optimization belonging to the field of the stochastic optimization and metaheuristics [3]. The algorithm works by generating an initial state with a starting temperature. For each iteration, a candidate point is generated and accepted if its fitness leads to improvement of the solution. The candidate point could be accepted with a probability (that depends on on the temperature) even if its fitness does not improve the solution. At each iteration, temperature is cooled down following a cooling schedule. The candidates points selection simulates Boltzmann distribution.

4. Bi-objective formulation

One alternative formulation of the problem is to consider it as a bi-objective optimization problem where the first objective remains the same as in the equation 1 and then consider the equation 3 as minimization objective. In this case we want to maximize the reliability while minimizing the cost. The formulation becomes:

$$\begin{aligned} \text{Maximize} \qquad R_s = {} & R_1R_2 + R_3R_4 + R_1R_4R_5 + R_2R_3R_5 \\ & -R_1R_2R_3R_4 - R_1R_2R_3R_5 - R_1R_2R_4R_5 \\ & -R_1R_3R_4R_5 - R_2R_3R_4R_5 + 2R_1R_2R_3R_4R_5 \\ \text{Minimize} \qquad g_2(r,n) = {} & \sum_{i=1}^{m} \alpha_i \left(-\frac{T}{\log r_i} \right)^{\beta_i} [n_i + \exp(0.25 n_i)] \end{aligned} \tag{7}$$

subject to:

Equations 2, 4, 5 and 6.

The bi-objective was resolved using the ϵ-method. The algorithm can be summarized in three steps.

Algorithm 1 ϵ-method for soving bi-objective problem

1: Resolve for R_s alone (get $\bar{R_1}$). Determine the corresponding cost ($\bar{c_1}$). The point ($\bar{R_1}, \bar{c_1}$) is an endpoint of efficient frontier

2: Resolve for g_2 alone (get $\bar{c_2}$). Determine the corresponding reliability ($\bar{R_2}$). The point ($\bar{R_2}, \bar{c_2}$) is the other endpoint of efficient frontier

3: Keep R_s and add g_2 to the set of constraints, and vary its right hand side (by amount ϵ)

[2] https://pythonhosted.org/pyswarm/

5. Numerical Results

5.1 Mono-objective results

The two algorithms were run 50 times. The "best known results" are taken from [2] who did a comparative study and concluded their method yielded the best results compared to all other algorithms available at the time. They have used in their study "PSO using Gaussian distribution and chaotic distribution".
The most salient results from the SA, PSO and the best known results suggest all that the redundancy for the subsystems 1, 2 and 5 should be 3, 3, and 1, respectively. In addition, PSO and SA concord that 3 transformers can be set in parallel in subsystem 3 (Table 2). The reliability of all components are required to be higher than 0.5 with the highest reliability required in the subsystem 3 with a value of about 0.91. Overall, the best reliability of the system obtained from the two algorithm are equal for 3 digital decimal. The PSO yielded better results. It can be inferred as well that the cost is the limiting resource for reliability maximization (Table 2) and SA provided less satisfactory results with a higher standard deviation (Figure 2).

Table 2: *Results of Particle Swam Optimization and Simulated Annealing compared to the best known results*

Parameter	Best known results[2]	PSO	SA
n	(3, 3, 2, 4, 1)	(3, 3, 3, 3, 1)	(3, 3, 2, 3, 1)
r_1	0.826678	0.826176260	0.745312312
r_2	0.857172	0.863356826	0.816112405
r_3	0.914629	0.864910125	0.915260424
r_4	0.648918	0.714651387	0.766155776
r_5	0.715291	0.717516082	0.668408642
R_s	0.99988957	0.99989175	0.99971838
Slack 1 (volume)	5	18.0	33.0
Slack 2 (cost)	0.000339	0.00230	2.793
Slack 3 (weight)	1.5604	4.26477	28.693
Mean	0.99988594	0.9998333	0.99857200
Std. Dev.	6.9e-07	5.31e-05	8.63e-04

5.2 Bi-objective results

The bi-objective results using the PSO gives an insight on the trade-off between the reliability and the cost of the materials. From a cost of about 60 and a reliability averaging 099, a marginal increase of the reliability lead to an exponential increase of the cost (Figure 3). For instance, if we set the cost to **60**, then reliability of the system $R_s = 0.995131$, $r = [0.6952, 0.7647, 0.7871, 0.4337, 0.5192]$, $n = [3, 3, 2, 3, 2]$, slack 1 = 46.0 (Volume), and slack 3 = 0.00024 (Weight). In this case, the bounding constraint is the weight. Theoretically it means we could design a system with a reliability of 0.995 and a volume of 64 (reducing the original volume by 46).

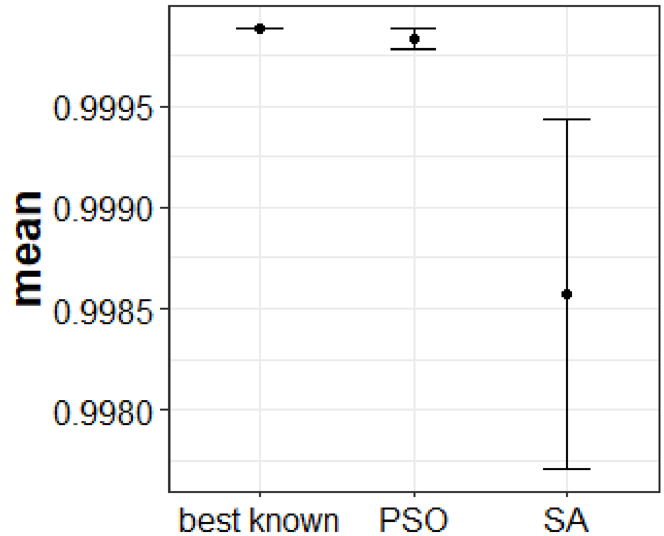

Figure 2: *Comparison of the performance of PSO, SA and the best known solution from the literature. Bars represent the standard deviation associated to each implementation*

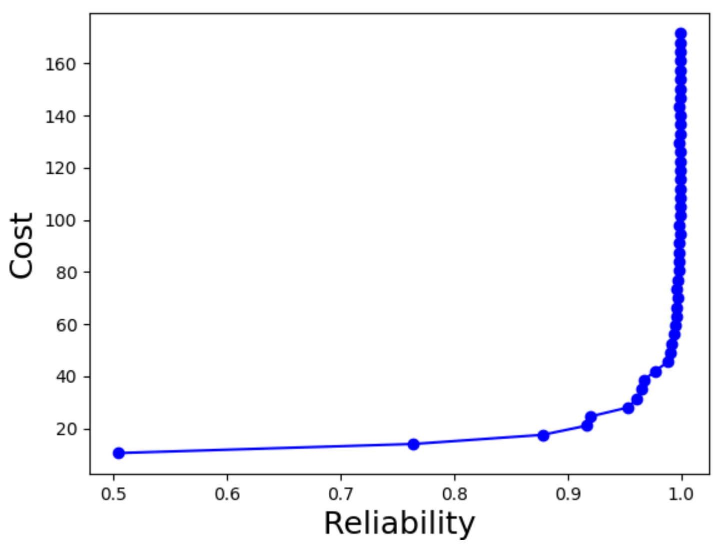

Figure 3: *Bi-objective optimal Pareto frontier*

6. Conclusion

This exercise showed the performance of two algorithm (PSO and SA). The two algorithms have some stochasticity. PSO used a population of points whereas, SA used a single particle. PSO sightly outperformed SA, however, this discrepancy may be attributed to the implementation of SA with a searching space too high or the stopping criteria.

Furthermore, bi-objective approach allowed to understand the trade-off between the cost and the reliability of the system.
However, the reliability as given in this exercise remains valid in laboratory conditions but may fail in real situation where the material will be subject to different weather and climate conditions. The best approach may be to use an algorithm giving an interval of reliability (probability branch and bound seems a good candidate algorithm). A second approach would be to use the fuzzy theory as proposed by [1] where each reliability is replaced by three reliabilities representing respectively, the expected reliability, the pessimistic (lower bound) and the optimistic one (upper bound).

References

[1] H. Garg, M. Rani, S. P. Sharma, and Y. Vishwakarma, "Bi-objective optimization of the reliability-redundancy allocation problem for series-parallel system," *Journal of Manufacturing Systems*, vol. 33, no. 3, pp. 335–347, 2014. [Online]. Available: http://dx.doi.org/10.1016/j.jmsy.2014.02.008

[2] L. d. S. Coelho, "An efficient particle swarm approach for mixed-integer programming in reliability-redundancy optimization applications," *Reliability Engineering & System Safety*, vol. 94, no. 4, pp. 830–837, apr 2009. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S0951832008002251

[3] J. Brownlee, *Clever Algorithms*, 2011. [Online]. Available: http://www.cleveralgorithms.com