International Journal of Electronics and Communication Engineering (IJECE)
ISSN(P): 2278-9901; ISSN(E): 2278-991X
Vol. 4, Issue 5, Aug - Sep 2015, 37-44

International Academy of Science, Engineering and Technology
Connecting Researchers; Nurturing Innovations

RELIABILITY OPTIMIZATION USING HARDWARE REDUNDANCY: *A PARALLEL BB-BC BASED APPROACH*

AJAY SINGH & SHAKTI KUMAR

Baddi University of Emerging Sciences and Technolgy, Baddi, Himachal Pradesh, India

ABSTRACT

Reliability optimization using hardware redundancy or reliability redundancy allocation problem (RRAP) is a complex, NP hard problem. This paper proposes a parallel big bang big crunch based apporach to reliability optimization using hardware redundancy or RRAP. The apporach was implemented in MATLAB and is vallidated using two examples. One of the examples is a 14 dimension well known problem from the literature. The results are compared from the seven other algoritms namely C Programing Language Simplex optimization software (CPLEX), Genetic Algorithm (GA), Ant Colony Optimization (ACO), Immune Algorithm (IA), Tabu Search (TS), Hybrid Parallel Genetic Algorithm (HPGA) and Big Bang Big Crunch (BB-BC) approach. It is observed that the proposed method has a very fast converging rate as compared to BB-BC and has better accuracy as compared to all the approaches. The results clearly indicate that the proposed approach out performs all other seven approaches.

KEYWORDS: Reliability Optimization Using Hardware Redundancy

I. INTRODUCTION

System Reliability can be defined as the probability that system under consideration performs its intended function adequately when operated under stated environmental and operating conditions [1]. Reliability assumes a very high significance in the case of systems such as single shot systems, systems intended for particular mission and application such as space systems etc. In order to minimize cost for a given mission, the system designers are forced to design systems for the optimal reliability under given set of constraints. This paper proposes a system reliability optimization approach using hardware redundancy. Many approaches for reliability optimization using hardware redundancy, based on dynamic programming approach [2], integer programming [3], heuristic algorithm based optimization approaches [4][5] can be found in the literature. However, with the increase in system complexity i.e. with the increase in problem dimensions, it becomes increasingly difficult to compute optimal redundancy using classical techniques. Under these circumstances soft computing or nature inspired computing based approaches [6], [7] turn out to be excellent alternatives. Some of these approaches like GAs [8], [9], [10], [11] and Tabu search [12] have been successfully applied to this problem. Ajay and Shakti proposed a big bang big crunch (BB-BC) based approach to solve RRAP problem [13].

This paper proposes a multi-population Parallel BB-BC optimization algorithm [14] based approach to system reliability optimization using hardware redundancy. The paper is divided into five sections. Section-II presents problem definition. Section-III presents Parallel BB-BC algorithm. Section IV discusses the application of PBB-BC to reliability optimization using hardware redundancy. The section IV also presents simulation, results and discussion. Section V concludes the paper.

II. RELIABILITY OPTIMIZATION USING HARDWARE REDUNDANCY PROBLEM

We consider a system shown in figure 1 below, let there be “n” number of stages in a system connected in series where “i^{th} stage” is a parallel configuration of X_i components each with reliability P_i. It is assumed that all elements are working simultaneously and for the system to fail at least one stage must fail i.e. all the parallel elements of a particular stage must fail. For the whole system to be operating all the stages must be operating; for a stage to be operational one of the elements out of all the components of the stage must be operational. Under such conditions, the series-parallel system of figure 1 reduces to that of figure 2.

Nomenclature

P_i	=	Reliability of i^{th} component/subsystem
Q_i	=	Unreliability of i^{th} component/ subsystem
R	=	System reliability
Q	=	System unreliability
X_i	=	Number of total components connected in parallel at stage-i
n	=	Number of stages
K_j	=	Available resource for constraint-j
M	=	Total number of different types of constraints
$C_{ij}(X_i)$	=	Resource-j consumed in stage-i with X_i components connected

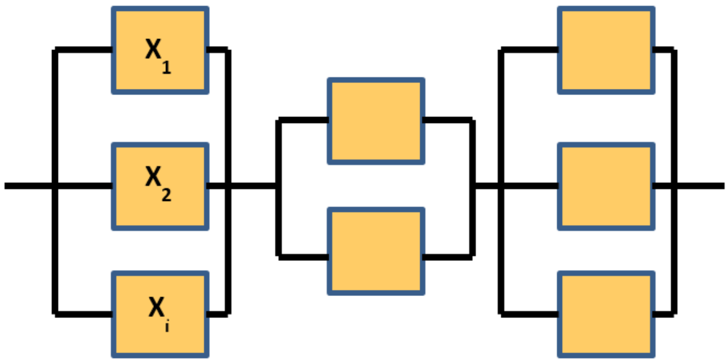

Figure 1

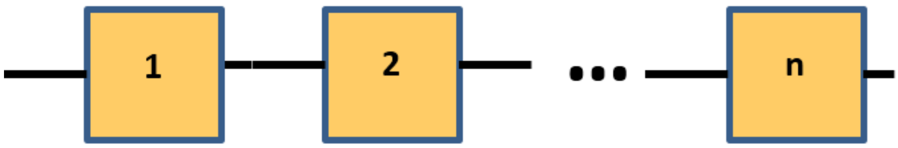

Figure 2

For a the system represented by figure 1 and 2 with n stages in series with X_i redundant components at stage-i, the system reliability can be represented by expression 1

$$R = \prod_{i=1}^{n} \left(1-(1-P_i)^{X_i}\right) = \prod_{i=1}^{n}\left(1-(Q_i)^{X_i}\right) \quad (1)$$

For the system shown in figure 1 and 2 the problem is to maximize system reliability R subject to

$$\sum_{i=1}^{m} C_{ij}(X_i) \leq K_j \; ; \; j = 1,2,\ldots,m$$

Most of the classical optimization techniques, (except the heuristic approaches) are very time consuming. Heuristic methods are approximate methods for the optimum allocation of redundancies. These methods, do not have any guarantee for best solution, although they may provide an optimal solution in most of the cases. In this paper we propose a new approach to reliability optimization using “Big Bang Big Crunch (BB-BC)” optimization approach.

III. PARALLEL BB-BC ALGORITHM

BB-BC theory of the evolution of the universe [15] [16] is the most popular scientifically established theory. According to the Big Bang theory, the universe began to evolve about twelve to fifteen billion years ago in a violent explosion. Initially all the energy was condensed to a single point, this huge amount of energy being unstable exploded out to form the galaxies. The universe is expanding since then but at a very slow speed. This phase is called the BIG BANG phase. There are two forces acting on the galaxies; outward force due to big bang and inward pull due to gravitational pull of each other.

In the BIG CRUNCH phase, the inward pull and outward push equal. Thereafter, as the outward force gets spent the inward pull starts dominating as the outward force and entire matter starts moving towards each other to meet at a single point as a single point of energy [16]. In the big bang phase outward force is maximum and in the big crunch phase outward force becomes minimal.

In conclusion, in the Big Bang phase, energy dissipation produces disorder and randomness is the main feature of this phase; whereas, in the Big Crunch phase, randomly distributed particles are drawn into an order. BB-BC is a single universe/population based search and optimization algorithm. Parallel BB-BC algorithm was first proposed by Shakti et. al. [14]. It is a multi-population based algorithm. Being parallel in nature its convergence rate is quite high. The approach is found to be more accurate than the BB-BC algorithm.

The parallel BB-BC optimization algorithm [14] can be presented as given below:

Parallel Big Bang Big Crunch Algorithm

Begin

/* Big Bang Phase */

Generate N populations each of size NC candidates randomly;

/* End of Big Bang Phase */

While not TC /* TC is a termination criterion */

/* Big Crunch Phase */

for i = 1: N

Compute the fitness value (centre of mass using Equation 3) of all the candidate solutions of ith population;

$$x_c = \frac{\sum_{i=1}^{NC} \frac{1}{f^i} x_i}{\sum_{i=1}^{NC} \frac{1}{f^i}} \quad \ldots\ldots\ldots \qquad \ldots\ldots\ldots(3)$$

Best fit individual can be chosen as the centre of mass instead of using Equation 3;

Sort the population from best to worst based on fitness (cost) value;

Select local best candidates $l_{best}(i)$ for ith population;

end

From amongst "N" l_{best} candidates select the globally best g_{best} candidate;

for i =1: N

With a given probability replace a gene of $l_{best}(i)$ with the corresponding

gene of global best g_{best} candidate

end

/* End of Big Crunch Phase */

/* Big Bang Phase */

Calculate new candidates around the centre of mass by adding or subtracting a normal random number whose value decreases as the iterations elapse using Equation 4;

$$x^{new} = x^c + l(rand)/k \quad \ldots \qquad \ldots(4)$$

/* End of Big Bang Phase */

end while

end

IV. APPLICATION OF THE PROPOSED APPROACH

In order to validate our proposed approach we implemented it in MATLAB and simulated the results. We considered the following examples.

A. Example 1 [6]

Consider a four stage system for optimum redundancy allocation with two linear constraints K_1, and K_2. The data are:

$n = 4$, $K_1 \leq 56$, $K_2 \leq 120$

$P_1 = 0.80$, $C_{11} = 1.2$, $C_{12} = 5$

$P_2 = 0.70$, $C_{21} = 2.3$, $C_{22} = 4$

$P_3 = 0.75$, $C_{31} = 3.4$, $C_{32} = 8$

$P_4 = 0.85$, $C_{41} = 4.5$, $C_{42} = 7$

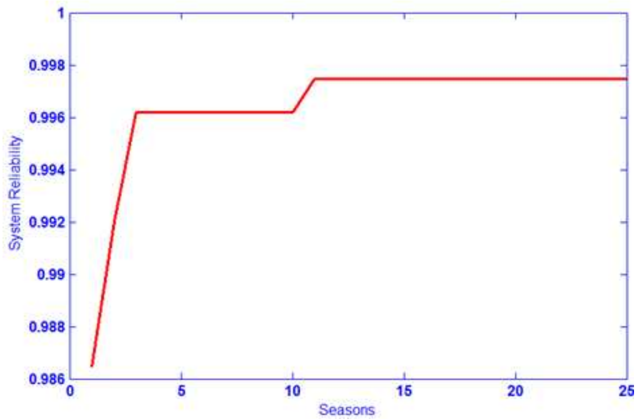

Figure 3

We took 25 trials runs. Since every time the simulation produced same result as given in [8]. We observed that on every trial the simulation produced the optimal result of 0.99747 as was produced by other approaches as given in [8]. Hence, the observation process was terminated with the conclusion that since, the problem was very small the best result has been obtained. The simulation run for one of the trials is embedded here as figure 3

B. Example 2

In order to validate our approach we selected one of the frequently used problem for the integer nonlinear reliability-redundancy allocation problem (RRAP), as given in Fyffe et. al. [9], and Nakagawa & Miyazaki [2]. The problem is a 14 stage series–parallel system where each stage could have three or four components of choice. The objective is to maximize the reliability of the series–parallel system subject to the cost and weight constraint of 130 and 191 respectively. The upper limit on the number of components is 6 in each subsystem. The component data for testing problem is listed in Table 1. The problem is an NP hard problem.

The approach was implemented in MATLAB and 50 trials were taken on an Alienware Laptop powered by Core-i7 @2.5 GHz processor with 8GB RAM. The performance results are placed as table 2

Table 1

Subsystem No.	Component choice											
	Choice 1			Choice 2			Choice 3			Choice 4		
	R	C	W	R	C	W	R	C	W	R	C	W
1	0.90	1	3	0.93	1	4	0.91	2	2	0.95	2	5
2	0.95	2	8	0.94	1	10	0.93	1	9	-	-	-
3	0.85	2	7	0.90	3	5	0.87	1	6	0.92	4	4
4	0.83	3	5	0.87	4	6	0.85	5	4	-	-	-
5	0.94	2	4	0.93	2	3	0.95	3	5	-	-	-
6	0.99	3	5	0.98	3	4	0.97	2	5	0.96	2	4
7	0.91	4	7	0.92	4	8	0.94	5	9	-	-	-
8	0.81	3	4	0.90	5	7	0.91	6	6	-	-	-
9	0.97	2	8	0.99	3	9	0.96	4	7	0.91	3	8
10	0.83	4	6	0.85	4	5	0.90	5	6	-	-	-
11	0.94	3	5	0.95	4	6	0.96	5	6	-	-	-
12	0.79	2	4	0.82	3	5	0.85	4	6	0.90	5	7
13	0.98	2	5	0.99	3	5	0.97	2	6	-	-	-
14	0.90	4	6	0.92	4	7	0.95	5	6	0.99	6	9

Table 2: Performance of the Proposed Approach

Worst	Mean	Best
0.98767	0.98767	0.98767

The data of one of the simulation run is placed as Table 3. Here "elite" indicates the best system evolved with number of parallel components at each stage. Elite reliability is the reliability of optimal system which in this trial run was 0.98767. At each stage which all components were placed in parallel are given in the columns titled as type of components.

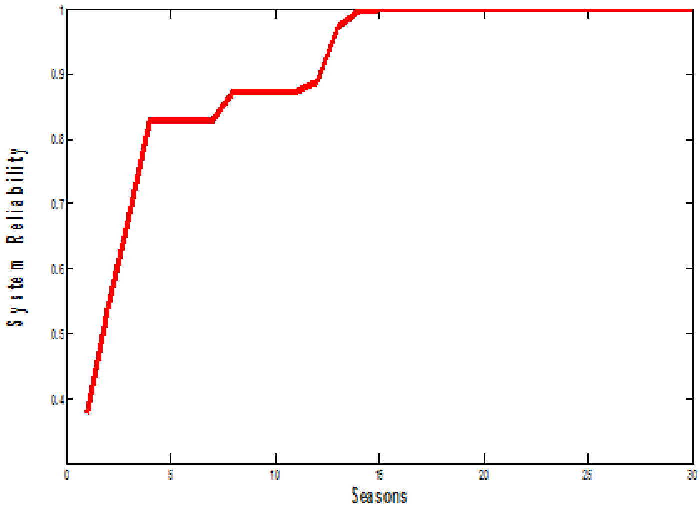

Figure 4

Figure 4 shows the seasons (iteration) Vs. System Reliability plot of this trial run. When the system complexity (dimensionality) grows, the system reliability computation becomes very time consuming and at one point become infeasible, under such circumstances the proposed computerized algorithm comes very handy for the system designers.

Table 3

Stage No.	Number of Parallel Components	Type of the Components			
1	4	3	4	2	3
2	2	2	3	0	0
3	2	3	2	0	0
4	2	2	3	0	0
5	2	2	2	0	0
6	3	2	3	2	0
7	2	3	2	0	0
8	2	3	3	0	0
9	1	2	0	0	0
10	2	2	2	0	0
11	3	3	3	3	0
12	3	3	2	4	0
13	1	3	4	2	3
14	2	2	3	0	0

Table 4: Comparison of performance of various algorithms

CPLEX	GA	TS	ACO	IA	HPGA	BB-BC	PBB-BC
0.986811	0.9867	0.986811	0.9868	0.986811	0.986811	0.9873	0.98767

Table 4 compares the performance of proposed method with the other methods [17][13].

It can be observed that performance of proposed PBB-BC based algorithm is better than the other 7 algorithms evaluated in [17][13]. The main advantage of PBB-BC as compared to BB-BC is the fast convergence time. PBB-BC takes an average of 9 seconds to optimize the reliability whereas BB-BC takes about 600 seconds to compute the optimal reliability.

V. CONCLUSIONS

As the system complexity grows reliability optimization using redundancy becomes very tedious. The classical techniques can be applied for low to medium complexity system where problem dimensionality is low. However, with the increased problem dimensionality the classical methods turn out to be either very time consuming or infeasible. Under these circumstances one is forced to migrate from exact reasoning classical approaches to approximate reasoning (soft computing) based reliability optimization approaches. This paper proposed a PBB-BC based system reliability optimization approach using hardware redundancy. We implemented our new approach using MATLAB, and compared it with the other seven approaches found in the literature. We found that the proposed approach outperformed other 7 algorithms (CPLEX, GA, TS, ACO, IA, HPGA and BB-BC) on account of accuracy, and demonstrated much faster convergence as compared to single population BB-BC approach.

REFERENCES

1. KK Aggarwal, “Redundancy Techniques for Reliability Optimization”, Reliability Engineering, Kluwer academic Publishers, Netherlands, 1993. pp 140-152
2. Y. Nakagawa, S. Miyazaki, Surrogate constraints algorithm for reliability optimization problems with two constraints, IEEE Trans. Reliability, R-30, pp. 175-180 (1981).
3. R. L. Bulfin, C. Y. Liu, Optimal allocation of redundant components for large systems, IEEE Trans. Reliability, R-34, pp. 241-247 (1985).
4. K.K. Govil, Simplified algorithm for optimum availability allocation and redundancy optimization subject to cost constraint, Microelectronics and Reliability, 24, 5, pp. 935-938 (1984).
5. V.D. Murty, K.B. Misra, A new heuristic approach for redundancy optimization problem, Microelectronics and Reliability, 24, 6, pp. 1013-1017 (1984).
6. Shakti Kumar & Amar Singh, “Nature Inspired Computing and its Applications”, Proceedings of the National Conference on Information Security & Software Engg., Deptt. Of Computer Science, University of Lucknow, Lucknow,UP Feb 10-11, 2014, pp 33-50.
7. Iztok Fister Jr., Xin-She Yang, Iztok Fister, Janez Brest, Duˇsan Fister, “A Brief Review of Nature-Inspired Algorithms for Optimization”, ELEKTROTEHNIˇSKI VESTNIK 80(3): 1–7, 2013.
8. Parddep Kumar, GL Pahuja, DK Chaturvedi, “Constrained Reliability Redundancy Optimization of Complex Systems using Genetic Algorithm”, MIT International Journal of Electrical and Instrumentation Engineering Vol. 1, No. 1, Jan. 2011, pp. 41-48.
9. Fyffe, D. E., Hines, W. W., and Lee, N. K., “System reliability allocation problem and a computational algorithm”, IEEE Transaction on Reliability, vol. 17, pp. 64-69, June, 1968
10. Coit, D. W., Smith, A., “Reliability optimization for series-parallel systems using a genetic algorithm”, IEEE Transactions on Reliability, vol. 45, no. 4, pp. 895-904, September, 1996.

11. Hadi Akbarzade Khorshidi1, Sanaz Nikfalazar, “Comparing Two Meta-Heuristic Approaches for Solving Complex System Reliability Optimization” Applied and Computational Mathematics 2015; 4(2-1): 1-6
12. Kulturel-Konak, S., Smith, A. E., Coit, D. W., “Efficiently Solving the Redundancy Allocation Problem Using Tabu Search," IIE Transactions, vol. 35, no. 6, June 2003, pp. 515-526.
13. Ajay Singh and Shakti Kumar,” Reliability Optimization Using Hardware Redundancy: *A BB-BC Based Approach*.” National Conference on Recent Innovations in Electronics, Electrical and Computer Engineering RIEECE-15, Baddi University of Emerging Sciences and Technology, Baddi, Distt. Solan, (HP) INDIA. May 9, 2015, pp.6-9.
14. Shakti Kumar, Sukhbir Wallia & Amar singh, “Parallel Big Bang-Big Crunch Algorithm**”** International Journal of Advanced Computing, Vol.46, Issue.3, pp 1330-1335.
15. Kumbasar, T., Yesil, E., Eksin, I. and Guzelkaya, M. 2008. Inverse fuzzy model control with online adaptation via Big Bang-Big Crunch optimization. ISCCSP 2008, Malta, 12-14 March, pp. 697-702.
16. Kripka, M. and Kripka, R. M. L. 2008. Big Crunch Optimization method. International Conference on Engineering Optimization, Brazil, 1-5 June.
17. Ki Tae Kim and Geonwook Jeon (2012). A Hybrid Parallel Genetic Algorithm for Reliability Optimization, Real-World Applications of Genetic Algorithms, Dr. Olympia Roeva (Ed.), ISBN: 978-953-51-0146-8, InTech, pp 127-146. Available from: http://www.intechopen.com/ books/real-world-applications-of-genetic-algorithms/a hybridparallel-genetic-algorithm-for-reliability-optimization.