01 May 2006

Comparison of PSO and GA for K-Node Set Reliability Optimization of a Distributed System

G. A. Bakare

I. N. Chiroma

Ganesh K. Venayagamoorthy
Missouri University of Science and Technology

Recommended Citation

G. A. Bakare et al., "Comparison of PSO and GA for K-Node Set Reliability Optimization of a Distributed System," *Proceedings of the IEEE Swarm Intelligence Symposium, 2006*, Institute of Electrical and Electronics Engineers (IEEE), May 2006.

Comparison of PSO and GA for K-Node Set Reliability Optimization of a Distributed System

G. A. Bakare
Electrical & Electronics Eng'g Programme
Abubakar Tafawa Balewa University,
Bauchi, Nigeria
bakare_03@yahoo.com

I. N. Chiroma
University Computer Centre
Abubakar Tafawa Balewa
University, Bauchi, Nigeria
inchiroma@yahoo.com

G. K. Venayagamoorthy
Real-Time Power and Intelligent Systems Laboratory,
University of Missouri-Rolla,
MO 65409, USA.
gkumar@ieee.org

***Abstract*- Particle Swarm Optimization (PSO), as a novel evolutionary computing technique, has succeeded in many continuous problems, but quite a little research on discrete problem especially combinatorial optimization problem has been reported. In this paper, a discrete PSO algorithm is proposed to solve a typical combinatorial optimization problem: K-Node Set Reliability (KNR) optimization of a distributed computing system (DCS) which is a well-known NP-hard problem is presented. It computes the reliability of a subset of network nodes of a DCS such that the reliability is maximized and specified capacity constraint is satisfied. The feasibility of the proposed algorithm is demonstrated on 8 nodes 11 links DCS topology. The test results are compared with those obtained by the genetic algorithm (GA) method in terms of solution quality and convergence characteristics. Experimental study shows that the proposed PSO algorithm can achieve good results.**

***Index Terms*: Distributed computing system, particle swarm optimization, genetic algorithm and K-node set reliability optimization.**

I. INTRODUCTION

A Distributed Computing System (DCS) is a collection of processor-memory pairs connected by a communication link and logically integrated by means of a distributed communication network. The development of computer network and low cost computing elements has led to increasing interest in DCS. Among the numerous merits of using a DCS include more effective resource sharing, better fault tolerance, and higher reliability. In designing such systems, reliability must be considered, which heavily relies on the topological layout of the communication links [1], [2].

Almost all of the optimization problems relevant to distributed computing system design are NP-complete. That is, for most problems, there is no known algorithm that could guarantee finding the global optimum in a polynomial amount of time.

Several DCS reliability measures have been developed and one of these distributed system reliability measures, K-Node reliability (KNR), is adopted in [3]. Reliability optimization is the design of distributed computing systems, where KNR is viewed as the probability that all K (a subset of the processing elements) nodes in the DCS can be run successfully. They presented a heuristic algorithm for maximizing reliability by the node select problem to obtain an optimal design DCS. This work did not consider an exhaustive method because it is too time consuming. Instead, they applied the exact algorithm, which examine K–node set reliability optimization with a capacity constraint to find an optimal solution of K-node reliability. The heuristic algorithm works well for large DCS problems as it largely avoids unnecessary generation of spanning trees. They regarded the DCS as a weighted graph, in which the weight of each node represents its capacity and generate K-node disjoint terms using a K-tree disjoint reduction method to obtain the KNR. This reduces the disjoint terms and hence reduces the computation time.

In [4], K-node set reliability optimization with capacity constraint for a DCS was examined using exact method. This method can obtain an optimal solution but cannot effectively reduce the problem space. Moreover, exact method spends more execution time with a large DCS. In fact, most distributed system problems are large and an increase in the number of nodes causes the exact method execution time to grow exponentially. Occasionally, therefore an application requiring an efficient algorithm with an approximate solution is highly attractive.

In many cases, sophisticated heuristics have to be developed to achieve satisfying results. Previous approaches have either been enumerative-based, which are applicable only for small DCS sizes [5], or heuristic-based, which can only be applied to larger networks, but do not guarantee optimality. Therefore other heuristic approaches are required. There exist a number of researches, which deal with such approaches as discussed [6]. Genetic algorithm (GA) is one of such heuristics that have been found to be a very good computational tool for problem of this nature. They can be applied to search large, multimodal, complex problem spaces [7], [8]. In [9], a genetic algorithm to determine the link capacities of a network, initially generated by the software called DESIGNET was proposed. In [10], GA approach is employed in the backbone network design under the constraint: minimal total link cost and 1-FT (fault-tolerant to 1 link-failure).

Kennedy and Eberhart proposed a new evolutionary computation technique called particle swarm optimization (PSO) in 1995. This method was developed through the simulation of a simplified social system and has been found to be robust in solving continuous non linear optimization problems [11]-[14]. Swarm algorithms differ from evolutionary algorithms most importantly in both metaphorical explanations and how they work. The individuals (particles) persist over time, influencing one another's search of the problem space, unlike genetic algorithms where the weakest chromosomes are immediately discarded. The particles in PSO (similar to chromosomes in GA) are known to have fast convergence to local/global optimum position (s) over a small number of iterations.

In this paper, particle swarm optimization, specifically discrete PSO for K-node set reliability optimization of a distributed computing system is presented. For comparison purposes, the same problem using GA [15] is solved in this paper. The feasibility of the proposed algorithm is demonstrated on a typical DCS topology obtained from [10]. The test results are compared with those obtained by the GA method in terms of solution quality and convergence characteristics. Experimental study shows that the PSO algorithm can achieve good results.

II. PROBLEM FORMULATION

The KNR problem can be characterized as follows: Given the topology of an undirected DCS, the reliability of each communication link, the capacity of each node and a possible set of data files, and assume that each node is perfectly reliable and each link is either in the working state or failed state. The problem can be mathematically stated thus:

$$\text{Maximize } R\ (G_K) \qquad (1)$$

$$\text{Subject to: } \quad \sum c\ (v_i) \geq C_{\ constraint} \qquad (2)$$

Where G_K is the graph G with set K of nodes specified, K ≥ 2, $R(G_K)$ is the reliability of the K-node set, v_i is an i^{th} node which represents the i^{th} processing element, $c(v_i)$ is the capacity of the i^{th} node and $C_{constraint}$ is the total capacity constraint in the DCS.

III. OVERVIEW OF PARTICLE SWARM OPTIMIZATION

Particle swarm optimization (PSO) is an evolutionary computation technique that was originally developed in 1995 by Kennedy and Eberhart [11]. It has been developed through simulation of simplified social models and has been found to be robust for solving non-linear, non-differentiability multiple optimal and multi-objective problems. The features of the method are as follows:

- The method is based on the researches about swarms such as fish schooling and bird flocking.
- It is based on a simple concept and has high quality solution with stable convergence.
- It was originally developed for non-linear optimization with continuous variables; however it is easily expanded to treat problems with discrete variables. Therefore it is applicable to K-node reliability optimization of a distributing computing system.

PSO is an evolutionary technique that does not implement survival of the fittest. Unlike other evolutionary algorithms where an evolutionary operator is manipulated, each individual in the swarm flies in the search space with a velocity which is dynamically adjusted according to its own flying experience and its companions flying experience.

The system initially has a population of random solutions. Each potential solution, called a particle, is given a random velocity and is flown through the problem space. The particles have memory and each particle keeps track of its previous best position, called the *pbest* and its corresponding fitness. There exist a number of *pbest* for the respective particles in the swarm and the particle with greatest fitness is called the global best (*gbest*) of the swarm. The basic concept of the PSO method lies in accelerating each particle towards its *pbest* and *gbest* locations, with random weight acceleration at each time step. The modified velocity of each particle can be computed using the current velocity and the distance from *pbest* and *gbest* as given by:

$$\begin{aligned} V_{id}^{k+1} &= W \times V_{id}^{k} + c_1 \times rand_1 \times \left(pbest_{id} - x_{id}^{k} \right) + \\ & c_2 \times rand_2 \times \left(gbest_{id} - x_{id}^{k} \right) \end{aligned} \qquad (3)$$

In the discrete version of the PSO, the trajectories are change in the probability that a coordinate will take on a binary value (0 or 1). Therefore, the positions are computed using:

$$\begin{aligned} & if \quad (rand < S(V_{id}^{k+1})) \quad then\ x_{id}^{k+1} = 1 \\ & \qquad\qquad\qquad\qquad else \quad x_{id}^{k+1} = 0 \end{aligned} \qquad (4)$$

Where:

S(V) is a sigmoid limiting transformation function given by:

$$S(V) = 1 \big/ \left(1 + e^{-V} \right)$$

rand, $rand_1$ and $rand_2$: random numbers between 0 and 1
V^{k}_{id} : current velocity of individual *i* at iteration *k*
$V^{min}_{d} \leq V^{k}_{id} \leq V^{max}_{d}$
V^{k+1}_{id} : modified velocity of individual *i*
x^{k}_{id} : current position of individual *i* at iteration *k*
$pbest_i$: pbest of individual *i*
$gbest_i$: gbest of the group
c_1 and c_2: the weighting of the stochastic acceleration that pulls each particle towards *pbest* and *gbest*.
W: inertia weight factor that controls the exploitation and exploration of the search space by dynamically adjusting the velocity. It is computed using:

$$W = W_{max} - \frac{W_{max} - W_{min}}{max\,gen} \times iter \qquad (5)$$

maxgen : maximum generation
iter : current iteration number.

IV. REALIZATION OF PROPOSED METHOD

Particle swarm K-node reliability optimization is developed as explained below.

A. Representation of Individual String

Before using the binary PSO algorithm to solve the KNR combinatorial optimization problem, the representation of a particle must be defined. A particle is also called an individual. The problem involves a capacity constraint. The network topology is fixed. The size of every node is also fixed. Binary coding scheme is therefore employed. The length of a chromosome is equal to the number of network nodes of the form:

$$x_n \, x_{n-1} \ldots. \; x_{i-1} \ldots.. \; x_2 \, x_1 \qquad (6)$$

Each bit indicates whether the node is selected for the K-node set, where $x_i = 1$ if v_i is selected; otherwise $x_i = 0$.

B. Initialization

The DCS parameter such as the number of nodes (*n*), the number of links (*e*), each link's reliability (*a*), each node's capacity, *c(vi)*, capacity constraint (constant), etc are entered into the computer. The PSO parameters such as population size (*ps*), minimum and maximum inertia weights, W_{min} and W_{max}, the limit of velocity change V_{min} and V_{max}, maximum generation (genmax), acceleration constants c_1 and c_2, etc are also entered.

For each node v_i, if the degree of v_i is $d(v_i)$ and if the links $e_{i,k1}, e_{i,k2}, e_{i,k3}, \ldots, e_{i,kd(vi)}$ are adjacent to v_i, then its weight can be computed using [17]:

$$w(v_i) = a_{i,k_1} + b_{i,k_1} * (a_{i,k_2} + (b_{i,k_2} * (a_{i,k_3} + (..(b_{i,k_{d(v_i)}}))..) \qquad (7)$$

Where $i, k_1, k_2, \ldots, k_{d(vi)} \in \{1, , n\}$. The heaviest node is then determined.

Initial population of particles $X_i = [x_n, .., x_1] \; \varepsilon \; X$ are randomly generated with the heaviest node included. The node capacities are computed. If the total capacity of a particle satisfies the capacity constraint, then it is appended to the population, otherwise it is discarded.

C. Evaluation Function

The objective function of a chromosome is computed using [17]:

$$f = \left(ratioprdl_k + ratio\,deg\right) / \sqrt{|k| + 2} \qquad (8)$$

Where

$$ratioprdl_k = \left(\sum_{v_i \varepsilon G_k} \sum_{v_i \varepsilon G_k} a_{i,j} \right) \Big/ k(k-1) \qquad (9)$$

$$ratio\,deg_k = \left(\sum_{v_i \varepsilon G_k} d(v_i) \right) \Big/ (n-1)k \qquad (10)$$

and $2 \le |k| \le n$.

The fitness values that determine which particles are to be carried onto the next generation are computed from the objective function equation.

D. Implementation of the PSO for KNR

The main steps involved in the searching procedures of the proposed PSO methods are described as follows:

Step 1: Read the DCS data and the PSO parameters.

Step 2: Generate randomly initial population of particles with random positions and velocities and the heaviest node included. The node capacities of the particles are computed. If the sum of the capacity of a particle satisfies the capacity constraint, then it is appended to the population, otherwise it is discarded.

Step 3: Compute the fitness value of the initial particles in the swarm using the objective function in (8). Set the initial *pbest* to current position of each particle and the initial best evaluated values among the swarm is set to gbest.

Step 4: Update the generation count.

Step 5: Update the velocities and the positions according to equations (3) and (4) respectively.

Step 6: Compute the fitness value of the new particles in the swarm using the objective function given by (8). Update the *pbest* with new positions if the particle's present fitness is better than the previous one. Also update the *gbest* with the best in the population swarm.

Step 7: Repeat steps 4 to 6 until the preset convergence criterion: maximum number of generations is fulfilled.

Step 8: Compute the reliability $R(G_K)$ of the optimal K-node set result in the particle of the population using the Monte Carlo technique [18] and output the K-node set corresponding to the $R(G_K)$.

V. SIMULATION RESULTS AND DISCUSSION

The procedure described above was implemented using the FORTRAN language and the developed software program was executed on a 450 MHz Pentium III PC. For comparison purpose, the problem using GA on the same platform [15] is studied in this paper. To illustrate the effectiveness of the proposed method, a distributed computing system topology obtained from [4] was considered. The results obtained are compared with that of GA in terms of solution quality and convergence criteria.

The topology of the distributed computing system with eight (8) nodes, and eleven (11) links, was considered as a case study where $c(v_i)$ represents the capacity of nodes v_i, and a_{ij} represents the reliability of the link e_{ij}.

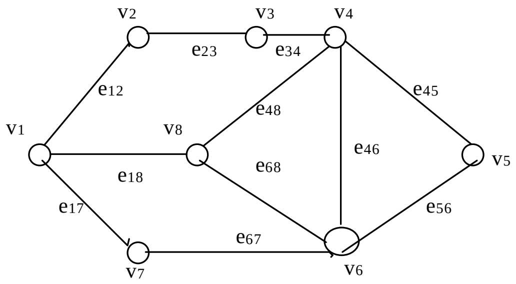

Fig 1: A DCS with 8 nodes and 11 links.

Benchmark input data for DCS are:

$c\ (v_1) = 39$, $c(v_2) = 45$, $c(v_3) = 38$, $c(v_4) = 53$, $c(v_5) = 47$,
$c\ (v_6) = 49$, $c(v_7) = 51$, $c(v_8) = 41$.
$a_{12}=0.89$, $a_{17}=0.81$, $a_{18}=0.93$, $a_{23}=0.85$, $a_{34}=0.91$, $a_{45}=0.82$, $a_{46}=0.83$, $a_{48}=0.96$, $a_{56}=0.87$, $a_{67}=0.84$, $a_{68}=0.88$.
$C_{onstraint} \geq 100$.

The proposed approach as well as GA was then applied to this problem and the optimum parameter settings for both methods are as shown in Table I.

The maximum fitness value of 0.701 was achieved and the chromosomes are 0001010 (from left to right). The K-node set in the population chromosome is $\{v_4, v_6\}$, i.e. nodes 4 and 6. The algorithm then computes the reliability and output the K-node set, which is the highest reliability K-node set, and R $(\{v_4, v_6\}) = 0.9974$. It can therefore be seen that both methods achieved the same solution quality. The convergence characteristics of the methods are comparatively shown in Fig. 2. As can be seen from this figure, both the PSO and GA have rapid convergence characteristics. GA achieved the maximum fitness earlier than the PSO: second and third generations respectively. This shows that PSO has good convergence characteristics and is less prone to being trapped into local optimal. Finally, PSO has less parameter settings as compared with GA.

Comparing the complexity of the proposed method with that of exact method, the complexity of the exact method is $O\ (2^e \times 2^n)$ where e denotes the number of edges (links) and n represents the number of nodes [3]. With the proposed algorithm, however, in the worst case, the complexity of evaluating the weight of the each node is $O(e)$, that of selecting the heaviest node is $O(n)$, and that of computing the reliability of the K- node set is $O(m^2)$, where m represents the number of paths of the selected K-node set [16]. Therefore, the complexity of the proposed algorithm is $O\ ((n \times ps \times tng) + m^2)$, where ps is the population size, and tng is the total number of generation. Although the conventional techniques can yield an optimal solution, they cannot effectively reduce the number of reliability computations. Some applications, especially NP-complete problems often require an efficient algorithm for computing the reliability. Under this circumstance deriving the optimal reliability may not be feasible. So an efficient algorithm, even when it yields only approximate reliability is preferred.

TABLE I: OPTIMUM PARAMETER SETTING FOR PSO AND GA

PARTICLE SWARM OPTIMIZATION	GENETIC ALGORITHM
c_1 : 2	Mutation rate: 0.02
c_2 : 2	Crossover Probability:0.9
V^{max} : 2	Elitism : Yes
V^{min} : -2	Uniform crossover : Yes
Maximum generation : 50	Maximum generation: 50
Particle size : 20	Population Size : 100
W^{max} : 0.9	
W^{min} : 0.4	

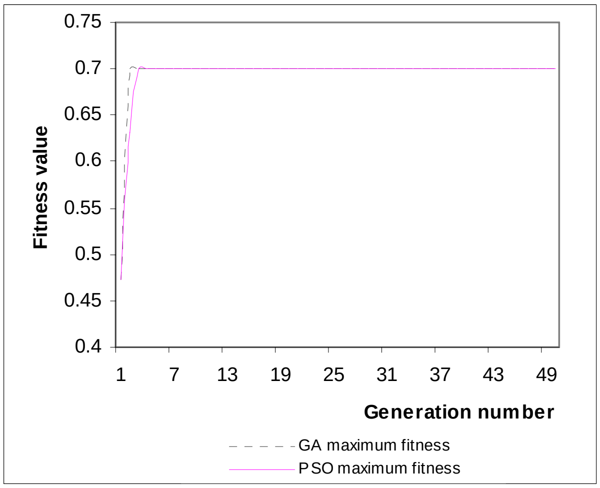

Fig. 2: Comparison of Convergence Characteristics of PSO & GA.

VI. CONCLUSION

In this paper, comparison is made between the particle swarm optimization and genetic algorithm based K-node set reliability optimization. The efficiency and accuracy of the proposed algorithm has been verified by implementing the procedure using a FORTRAN program on a PC. Verification on a typical DCS topology revealed that both methods achieved good quality solutions. PSO exhibits good convergence characteristics and is less prone to being caught at the local optimal. The PSO algorithm can efficiently obtain the maximal K-node set reliability with a capacity constraint.

REFERENCES

[1] R.H. Jan, F.J. Hwang, and S.T. Cheng, "Topological Optimization of a Communication Network Subject to a Reliability Constraint. *IEEE Trans on Reliability*, Vol,.42, N.o1, 1993, pp.63-70.

[2] M.S. Lin, The Reliability Analysis on Distributed Computing Systems. Ph.D Dissertation, Institute of Computer Science and Information Engineering, National Chiao Tung University, Hsinchu, Taiwan, R.O.C., 1994, pp.64-75.

[3] R.S. Chen, D.J. Chen and Y.S. Yel, "Reliability Optimization in the Design of Distributed Computing Systems", Proceedings of Int. Conference on Computing and Information Institute of Computer

Science and Engineering of National Chiao Tung University, Hsinchu, Taiwan, R.O.C., 1994 pp.422-439.

[4] R.S. Chen, D.J. Chen and Y.S. Yel, “Reliability Optimization of Distributed Computing Systems Subject to Capacity Constraint”, International Journal of Computers & Mathematics with Applications**,** 29(3), 1993, pp. 93-99.

[5] K.K. Aggarwal, Y.C. Chopra, and J.S. Bajwa,. “Topological Layout of Links for Optimizing the Overall Reliability in a Computer Communication System”, Microelectronic Reliability 22, 1982), pp.347-351

[6] S. Pierre, and G. Legault, “A Genetic Algorithm for Designing Distributed Computer Network Topologies”, IEEE Transactions on Systems, Man and Cybernetics. Part B: Cybernetics, Vol.28, No.2, 1998.

[7] J. H. Holland, Adaptation in Natural and Artificial Systems, University of Michigan Press, 1975.

[8] G.A. Bakare, Removal of Overloads and Voltage Problem in Electric Power Systems using Genetic Algorithm/Expert Systems Approaches. Ph.D Dissertation published by Shaker Verlag, Germany, 2001, pp.25-65.

[9] L. Davis, and S. Coombs,. Optimizing Network Link Sizes with Genetic Algorithms in Modelling and Simulation Methodology, Knowledge System’s Paradegins. Amsterdam, The Netherlands, Elsvier, 1987. pp.22-28.

[10] Cheng, S.T. Topological, “Optimization of a Reliable Communications Network”, *IEEE Trans. on Reliability*. Vol.47, No.3, 1998, pp. 225-233.

[11] V. G. Gudise and G. K. Venayagamoorthy, “Comparison of PSO and Backpropagation Training Algorithms for Neural Networks”, IEEE Swarm Intelligence Symposium, April 2003.

[12] M. A. Abido, “ Optimal Design of Power System Stabilizers Using PSO", IEEE Transactions on Energy and Conversion, Vol 17, No. 3, September 2002, pp. 406-413.

[13] J. Kennedy and R. Eberhart, “A Discrete Binary Version of the Particle Swarm Optimization”, Proceedings of IEEE International Conference on Neural Networks, Perth, Australia, 1997, Vol. 4, pp. 4104-4108.

[14] W. Zhang and Y. Liu, “Reactive Power Optimization Based on PSO in a Practical Power System”, IEEE PES Meeting, 2004.

[15] I. N. Chiroma, A Genetic Algorithm Based Efficient Optimization of K-Node Set Reliability for Distributed Computing System, MSc. Thesis (Computer Science), Abubakar Tafawa Balewa University, Bauchi, Nigeria, 2005.

[16] M. A. Aziz, “Path Set Enumeration of Directed Graphs by the Set Theoretic Method”, Microelectronics and Reliability, Vol. 5, No. 37, pp. 809-814.

[17] Y. S. Yeh, C. C. Chiu and R. S. Chen, “A Genetic Algorithm for K-Node Set Optimization with Capacity Constraint of a Distributed System”, Proceeding National Science Council ROC(A), Vol. 25, No. 1, 2001, pp. 27-34.

[18] M. S. Yeh, J. S. Lin and W. C. Yeh, “ A New Monte Carlo Method for Estimating Network Reliability”, Proceedings of the 16th International Conference on Computers and Industrial Engineering, 1994, pp. 723-726.