Reliability Optimization: A Particle Swarm Approach

Sangeeta Pant, Anuj Kumar and Mangey Ram

Abstract In recent years, substantial efforts related to the applications of Particle Swarm Optimization (PSO) to various areas in engineering problems have been carried out. This chapter briefly gives the details of PSO development and its applications to reliability optimization.

Keywords Reliability · Optimization · Particle swarm optimization

1 Recent Works and Advances of PSO

Presently, we have many variants of Particle Swarm Optimization (PSO) and are expected to grow further rapidly. Figure 1 describes the basic variants and modifications in PSO over the years. Various modifications to the original PSO has been proposed so far [43]. Also, novel ideas from other disciplines such as evolutionary algorithms have been imported to the framework of PSO. PSO algorithms can be divided into the global version (*gbest* model) and the local version (*lbest* model) types, with the ability of the *lbest* model to prevent a solution being trapped in local minima. The *gbest* model, on the other hand, has more chance to get trapped into a local optimum. However, the global version is superior to the local version in terms of the speed of convergence to the optimum solution and the computation time.

To reduce the possibility of particles flying out of the problem space, Eberhart et al. [42] put forward a clamping scheme that limited the speed of each particle to a range $[-V_{\max}, V_{\max}]$. To assist with the balance between exploration and exploitation a modified PSO, incorporating an inertia weight, w was introduced [128]. The initial experiments suggested that a value between 0.8 and 1.2 provided

S. Pant · A. Kumar (✉)
Department of Mathematics, University of Petroleum and Energy Studies, Dehradun 248007, India
e-mail: anuj4march@gmail.com

M. Ram
Department of Mathematics, Graphic Era University, Dehradun 248002, India

M. Ram and J.P. Davim (eds.), *Advances in Reliability and System Engineering*, Management and Industrial Engineering, DOI 10.1007/978-3-319-48875-2_7

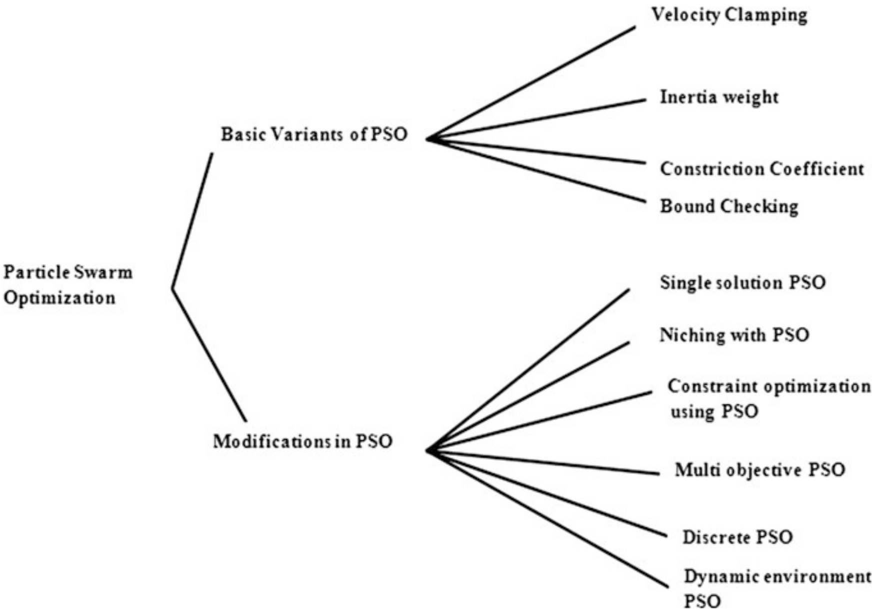

Fig. 1 Basic variants of PSO

good results, although in later work Eberhart and Shi [40] indicated that the value is typically set to 0.9 (reducing the stepwise movement of each particle, allowing greater initial exploration) reducing linearly to 0.4 (speeding convergence to the global optimum) during an optimization run. A summary of various existing inertia weight strategies is given in Table 1 [94]. Constriction is an alternative method for controlling the behaviour of particles in the swarm. Rather than applying inertia to the velocity memory, Clerc and Kennedy (developed 1999, published [20]) applied a constriction factor to the new velocity. Eberhart and Shi [40] showed that with judicious parameter settings, the two approaches were algebraically equivalent and improved performance could be achieved across a wide range of problems.

PSO has also been successfully applied to solve the constrained optimization problems. A variety of approaches [4, 15, 56, 103, 134, 137] have been developed to work with constrained optimization methods.

Although the basic PSO was developed to find single solutions to optimization problems, but later it is observed that PSO has an inherent ability to find multiple solutions. Niching is an important technique for multimodal optimization. PSO can be used as an effective niching method for maintaining stable subpopulations (or niches) [44, 76, 95, 135, 144].

PSO was originally developed and applied to static problems where the objective function does not change. Later, It is realized that PSO be adapted to solve dynamic problems as well. Several simple modifications [39, 41, 54, 57, 80, 105, 157] have been applied to improve its performance in dynamic environment. One of the first

Table 1 Description of different inertia weight strategies

Label	Inertia weight strategy	Adaption mechanism	Feedback parameter	References
W1	$w = c$	Constant w	–	Shi and Eberhart [128, 132]
W2	$w = 0.5 + \frac{\text{rand}()}{2}$	Random w	–	Eberhart and Shi [41]
W3	$w(\text{iter}) = w_{\min} + \frac{\text{maxiter} - \text{iter}}{\text{maxiter}}(w_{\max} - w_{\min})$	Linear time varying	–	Eberhart and Shi [41], Shi and Eberhart [129, 130]
W4	$w(\text{iter}) = w_{\min} + \left(\frac{\text{maxiter} - \text{iter}}{\text{maxiter}}\right)^n (w_{\max} - w_{\min})$	Nonlinear time varying	–	Chatterjee and Siarry [17]
W5	$w(\text{iter}) = w_{\text{initial}} \times U^{\text{iter}}$	Nonlinear time varying	–	Jiao et al. [60]
W6	$w(\text{iter}) = w_{\min} \times Z + \frac{\text{maxiter} - \text{iter}}{\text{maxiter}}(w_{\max} - w_{\min})$	Linear time varying with random changes	–	Feng et al. [46, 47]
W7	$w(\text{iter}) = w_{\min} + \frac{1 - \left(\frac{\text{iter}}{\text{maxiter}}\right)}{1 - s\left(\frac{\text{iter}}{\text{maxiter}}\right)}(w_{\max} - w_{\min})$	Nonlinear time varying	–	Lei et al. [72]
W8	$w(\text{iter}) = \left(\frac{2}{\text{iter}}\right)^{0.3}$	Nonlinear time varying	–	Fan and Chiu [45]
W9	$w(\text{iter}) = w_{\max} + \frac{\text{maxiter} - \text{iter}}{\text{maxiter}}(w_{\min} - w_{\max})$	Linear time varying	–	Zheng et al. [158, 159]
W10	Fuzzy rules	Adaptive	Best fitness	Saber et al. [122], Shi and Eberhart [131]
W11	$w_i^t = w_{\text{initial}} - \alpha(1 - h_i^t) + \beta s$	Adaptive	Fitnees of the current and previous iterations	Yang et al. [151]
W12	$w = 1.1 + \frac{gbest}{(pbest_i)_{\text{average}}}$	Adaptive	Global best and average local best fitness	Arumugam and Rao [6]

(continued)

Table 1 (continued)

Label	Inertia weight strategy	Adaption mechanism	Feedback parameter	References
W13	$w_i = w_{\min} + (w_{\max} - w_{\min}) \frac{\text{Rank}_i}{\text{total population}}$	Adaptive	Particle rank	Panigrahi et al. [99]
W14	$w = 1 - \alpha\left(\frac{1}{1+e^{ISA_{ij}}}\right),\ ISA_{ij} = \frac{\lvert x_{ij} - P_{ij}\rvert}{\lvert p_{ij} - p_{gi}\rvert + \varepsilon}$	Adaptive	Distance to particle and global best positions	Qin et al. [110]
W15	$w = w_{\text{initial}} + \left(1 - \frac{\text{dist}_i}{\text{max\_dist}}\right),$ $\text{dist}_i = \left(\sum_{d=1}^{D} (gbest_d - x_{i,d})^2\right)^{1/2}$	Adaptive	Distance to global best position	Suresh et al. [138]

studies in the application of PSO to dynamic environment came from Carlisle and Dozier [16], where the efficiency of different velocity models has been evaluated. Coelho et al. [22] successfully applied the split adaptive PSO design proportional integral (PI) controllers for nonlinear time-varying process in discrete time domain.

As far as the main body of PSO development work is concerned it concentrated on optimization in continuous search spaces, although some research has also been conducted into the application of the algorithm to discrete problems. Kennedy and Eberhart [61] developed the first discrete version of PSO for binary problems. Later, Al-kazemi and Mohan [3] compared this with an alternative velocity update technique called Multiphase Discrete PSO (M-DiPSO). Laskari et al. [71] presented an implementation of PSO that initially operated in continuous space but truncated the real number values to integers. Afshinmanesh et al. [1] have developed a hybrid approach, combining PSO and Artificial Immune Systems (AIS), for the optimization of binary problems.

For more developments in PSO one can refer the review papers by Parsopoulos and Vrahatis [104], Banks et al. [10, 11], Dian et al. [38].

A number of approaches have been proposed to extend the PSO for multiple objective problems. One of the earliest proposals for progression of PSO strategy for solving MOOP was made by Moore and Chapman [91] in an unpublished manuscript from 1999. Since then there have been several recent attempts to use PSO for multi-objective optimization problems only. Some of these concepts have been surveyed briefly in this section.

Dynamic Neighbourhood PSO Hu and Eberhart [55] proposed this method in which, in each generation, after calculating distances to every other particles, each particle finds its new neighbourhood. Among the new neighbours each particle finds the local best particle.

The Multi-objective Particle Swarm Optimizer (MOPSO) Coello and Lechuga [24] found PSO particularly suitable for MOOP mainly because of the high speed of convergence that the PSO presents for single objective optimization problem and proposed multi-objective particle swarm optimization (MOPSO). MOPSO used two archives, one for storing globally non-dominated solutions found so far by search process, while the other for storing the individual best solutions attained by each particle. They used method inspired by Knowles and Corne [65], for maintaining diversity. An adaptive grid feature used in this method, based upon objective functions values of archive members is applied to the archive, with the goal of producing well distributed Pareto optimal front. In this method first hyperspace is divided into small hypercube and each cube is assigned weight which is inversely proportional to the number of non-dominated solutions inside the cube. Then roulette wheel selection is used to select one of the hypercubes from which the *gbest* will be picked.

The Multi-objective Particle Swarm Optimizer (MOPSO) Coello et al. [25] improved the aforementioned MOPSO by incorporating a genetic operator as mutation operator. The mutation operator boosts the exploration capability of MOPSO presented in Coello and Lechuga [24].

The Swarm Metaphor Ray and Liew [117] proposed the Pareto dominance and combining concepts of evolutionary techniques with the PSO. All non-dominated individuals which are performing better and having less constraint violations are highly ranked based on Pareto ranking and saved as a set of leaders (SOL). The selection of a group leader as *gbest* from the SOL is based on roulette wheel selection which ensures SOL members with a large crowding radius have a higher probability of being selected as a leader.

The Approach of Mostaghim and Teich [92] proposed the sigma method for finding the suitable *gbest* for each particle, they also used turbulence factor to enhance the exploration capability.

The Algorithm of Fieldsend and Singh [48] suggested an approach in which they used an unconstrained elite archive (in which a special data structure called dominated tree is adopted) to store the non-dominated individuals found along the search process. The concept of the turbulence was also incorporated by them.

Bartz-Beielstein et al. [12] proposed an idea of using elitism (through the use of external archive) into PSO. They analysed different methods of selecting and deleting particles from the archive to generate a satisfactory approximation of the Pareto optimal front.

The Non-dominated Sorting PSO Li [77] developed Non-dominated sorting particle swarm optimization (NSPSO) which incorporated the main mechanism of Non-dominated sorting genetic algorithm (NSGA-II) [34]. In his algorithm, the population of particles was combined with the personal best position and the best was selected from the new population to compose the next population.

Another Multi-objective Particle Swarm Optimization (AMOPSO) Pulido and Coello [109] further improved the performance of PSO, and proposed an MOPSO algorithm, which is called AMOPSO. Their algorithm implements the subdivision of the decision space into multiple sub-swarms via clustering techniques. Their goal was to improve the diversity of solutions on the Pareto optimal front. At some point during the search process, different sub-swarms exchange information, as each sub-swarm chooses a different leader other than its own to preserve diversity.

The Algorithm of Parsopoulos et al. [106] developed parallel vector evaluated particle swarm optimization (VEPSO), is a multi-swarm variant of PSO, which is inspired by the vector evaluated genetic algorithm (VEGA). In VEPSO, each swarm is evaluated using only one of the objective functions of the problem under consideration and the information it possesses for this objective function is communicated to other swarms through the exchange of their best experience.

The Algorithm of Sierra and Coello [133] suggested a new MOPSO, which is also known as OMOPSO. In their design, the population is divided into three sub-swarms of equal size. Each sub-swarm adapted to a different mutation operator. In doing so, the ability of exploration and exploitation was enhanced during the search process.

Multi-Objective Particle Swarm Optimization with Crowding Distance (MOPSO-CD) Raquel and Naval [112] developed another PSO based approach called MOPSO-CD, which incorporated the crowding distance into PSO and the distribution of non-dominated solutions was improved on the Pareto optimal front. The crowding distance mechanism together with a mutation operator maintains the diversity of non-dominated solutions in the external archive. Raquel and Naval [112] also showed that MOPSO-CD is highly competitive in converging towards the Pareto optimal front and generated a well-distributed set of non-dominated solutions. We discuss this approach in detail in the next chapter.

A Hybrid PSO Liu et al. [79] proposed a hybrid PSO, which combined the global search ability of PSO with a synchronous local fine-tuning and used fuzzy global best to handle the premature convergence.

Time Variant MOPSO (TV-MOPSO) Tripathi et al. [141] adapted the vital parameters in PSO, namely the inertia weight and the acceleration coefficients during the iterations.

Elitist Mutated (EM)-MOPSO Reddy and Kumar [118] proposed (EM)-MOPSO, which incorporates an efficient mutation strategy called elitist mutation to enhance exploration and exploitation in the search space.

Dynamic Population Multiple Swarm MOPSO (DMOPSO) Most recently Leong and Yen [73] proposed an algorithm DMOPSO, inspired by Pulido and Coello [109] and incorporates the following four proposed strategies: (1) cell-based rank density estimation scheme to keep track of the rank and density values of the particles; (2) population growing strategy to increase the population size to promote exploration capability; (3) population declining strategy to prevent the population size from growing excessively; and (4) adaptive local archives designed to improve the distributed solutions along the sections of the Pareto optimal front that associate with each sub-swarm.

Other than aforementioned methods several other methods [2, 13, 14, 33, 50] for MOPSO have been proposed till date. For more survey on various PSO proposals reader can refer to Reyes-Sierra and Coello [121], Fieldsend [49], Padhye et al. [97].

2 Reliability Optimization (An Overview)

Almost every one of us is acquainted with the term reliability in day-to-day life. When we assign attribute ‘reliable’ to a component or a system (a system may be consist of collection of several components) we precisely mean to say that the same will render service for a good or at least reasonable period of time. In the modern age of sciences, a high degree of reliability is being demanded from all kinds of users whether it is in general, public sectors, industries, defense and space research programmes. There is too much at stake in terms of cost, human life, and national security to take any risks with equipments which might not function properly when required. Moreover, the present day weapons used for military purposes consist of thousands of small parts, each interwoven into a complex web which constitutes the weapons. The failure of any one of these could adversely affect the operation of the weapon. Therefore, it becomes more important that each part of the complex equipment must be highly reliable so the equipment as a whole must be reliable. The concept of high reliability is equally important outside the Military field. Computers which are complex as well as expensive play a major role in industrial and scientific activities. If a Computer does not operate even for a single day even due to any software failure, it not only spells inconvenience but also cause financial loss, i.e. the software reliability testing is very important.

So the needs of obtaining highly reliable systems and components have acquired special importance with the development of the present day technology.

The theory of reliability is not very old; usually world war second in 1939 is regarded as the starting point of reliability discipline. Before world war second, in the first quarter of twentieth century a team of workers in ‘Bell Telephone Laboratories’ developed statistical methods for solving there quality control problems which is strongly linked with quality control. They provided the basis for development of statistical quality control. The American Society for Testing and Materials, The American Standard Association and The American Society for Mechanical Engineers also worked for the quality control techniques. But these technique were widely used till world war second 1939. Complexity and automation of equipments used in the war resulted in severe problems of maintenance and repairs. The equipment/component failed beyond the expectation. During this war army and navy in USA set up a joint committee known as Vacuum Tube Development Committee for the study of failure in vacuum tube which is considered to be one of the root causes of the trouble. The major committee on reliability was set up by U.S Defense Department in 1950. This was latter called the Advisory Group on Reliability of Electronic Equipment (AGREE). During 1950s Germany, Japan, and Britain also took interest in such type of study. The last 20 years have seen remarkable progress in the application of reliability in industries and in other departments of all the developed and developing countries.

The theory of reliability is the new scientific discipline that studies the general regularity that must be maintained under design, experimentation, manufacture, acceptance, and use of units/components in order to maximal effectiveness from their use. The need of obtaining highly reliable systems and components has acquired Special importance with the development in the present day technology.

The problem of increasing reliability of equipments/components becomes more important and urgent in connection with the complex mechanization, modern sophistication, and automation of industrial process in many fields of industry, transportation, communication, space technology, etc. The complex system, equipments, machines, etc., are not of much use if they cannot perform the work adequately for which they are intended.

A system is a combination of elements forming a planetary whole, i.e. there is a functional relationship between its components. The properties and behavior of each component ultimately affects the properties of the system. Any system has a hierarchy of components that pass through the different stages of operations which can be operational, failure, degraded or in repair. Failure does not mean that it will always be complete; it can be partial as well. But both these types affect the performance of system and hence the reliability. Majority of the systems in the industries are repairable. The performance of these systems can influence the quality of product, the cost of business, the service to the customers, and thereby the profit of enterprises directly. Modern repairable systems tend to be highly complex due to increase in convolution and automation of systems. During the last 45 years reliability concepts have been applied in various manufacturing and technological fields. Earlier researcher discussed reliability and steady state analysis of some realistic engineering systems by using different approaches. Reliability techniques have also been applied to a number of industrial and transportation problems including automobile industry. Here the study is focused on the engine assembly process of automobiles.

A high degree of reliability is also desirable from the economic point of view so as to reduce the overall costs. Sometimes the annual maintaining cost of some system in operable state is much higher than its original cost. Insufficient reliability of units engenders great loss in servicing, partial stoppage of equipment, and there may be accidents with considerable damage to the equipment and even the cost may be more serious in term of human life, national prestige and security. All these factors and many more, demanded high reliability in the design and operations of components/systems/equipments in various reliability models of practical utility, we often across with the situations of maximizing the profit. The profit earned out by an operable system besides other parameters depends upon the cost incurred against the repairmen needed to repair the failure stages of the system.

The present day theory of reliability has been developed during the last two decades by engineers and mathematicians of various countries.

Every science is based on some fundamental concepts and definitions, same is true for theory of reliability also. Some of the basic concepts on which the theory of reliability is formulated are given below.

System A system is an arbitrary device consisting of different parts components or units.

Time It is the period during which one can expect the satisfactory performance of a system.

Adequate It indicates the criteria for operations of the device to satisfactory.

Failure Mode It is the effect by which a failure is observed.

Failure Rate It is the incremental change in the number of failures per associated incremental change in time. Or the expected rate of occurrence of failure or the number of failures in a specified time period. Failure rate is typically expressed in failures per million or billion hours. For example, if your television has a failure rate of five failures per million hours, you can watch one million hour-long television shows and likely experience a failure during only five shows.

Uptime It is total time during which the system is in the acceptable operating conditions.

Downtime The total time during which the system is not in the acceptable operating conditions.

The definition of reliability of a system is usually stated in straightforward terms as “the probability that the system will not fail during delivery of service [119], or alternatively, that the overall system performance figure of merit will not enter failure mode between the time a service is requested and when that service is delivered [136]. A system can be designed for optimal reliability either by adding redundant components or by increasing the reliability of components [68].

There are several ways for improving the system’s reliability. One way of improving reliability is either to duplex some of the unit or the whole system. Other way is to provide repair and maintenance to the system at the time of need. Some important technique of reliability improvements are as under.

Redundancy In a redundant system, some additional paths are created for the proper functioning of the system. Even though one component is sufficient for successful operation of the system, we deliberately use some more components to increase probability of success, thus causing the system to become redundant. There are three types of redundancies.

Active Redundancy An active redundant system with n-units is one which operates with every one unit. Here failure of system occurs only when all the units are fails.

Standby Redundancy A standby redundant system is one in which one unit operates on line followed by a number of spare unit called standbys. On failure of the operating unit, a standby unit, if operable, is switched on to the line by perfect or imperfect switching device. Standby can be classifies as hot, warm and cold depending on how they are loaded in the standby state. Hot standbys are those which are loaded in exactly the same way as the operating unit. Warm standbys are those which are diminished load. And cold standbys are completely unloaded and never lose their operational ability and can not fail in standby state.

Partial Redundancy The redundancy where in two or more redundant items are required to perform function k-out-of: m system. The system which is good iff at least k of it m items are good.

Maintenance All recoverable systems which are used for continuous or intermittent service for some period of time are subjected to maintenance. Maintenance action can be classified in several categories, e.g. preventive, corrective, and priority maintenance.

Preventive Maintenance Preventive maintenance is such type of check which keeps the system in a condition consistent with its built in level of performance, reliability and safety.

Corrective Maintenance It deals with the system performance when the system gives wrong results. Repair/maintenance is concerned with increasing with system availability. In order to increase the system availability, failed unit are repaired to put them into operation.

Priority Maintenance A redundant system which consist of $n \geq 2$ units in which one of the units is called the priority unit (P-unit) and others are termed as non-priority units (O-units). The P-unit is the "preferred unit" for operating on line and is never used in the status of a standby. The O-units are allowed to operate on the line only when the P-unit is under failure.

Pre-emptive Priority The repair of the O-unit is interrupted and its repair is continued as soon as the repair of the P-unit is completed. The resumed repair of the O-unit can follow any one of the following rules.

Pre-emptive Resume The repair of the O-unit is continued from the point where it was left earlier.

Pre-emptive Repeat The repair of the O-unit is started as a fresh; this implies that the time for the O-unit in the repair facility before it was left from service has no influence on its service time now.

Non Pre-emptive Priority The repair of the O-unit is continued and the repair of the P-unit is entertained only when the repair of the O-unit is completed. It is also called the Head-of-line repair police.

Inspection A system requires its inspection at random epochs in order to trace out the fault in redundant, particularly in deteriorating standby system.

The reliability optimization problems can be categorized in two ways: Single objective reliability optimization problems and Multi-objective reliability optimization problems.

- **Single Objective Reliability Optimization Problems**

  Let us consider a reliability optimization problem as follows:
  Max $f_0(r_1, r_2, \ldots, r_n, x_1, x_2, x_n)$
  subject to

$$\begin{array}{lll} f_i^c(r_1, r_2, \ldots, r_n, x_1, x_2, x_n) \le b_i, & \text{for} & i = 1, 2, \ldots, m \\ l_j \le x_j \le u_j, x_j \in Z^+, & \text{for} & j = 1, 2, \ldots, n \\ r_j \in (0, 1) \subset R, & \text{for} & j = 1, 2, \ldots, n \end{array}$$

where n is the number of components with m constraints. Component reliability of jth component is denoted by r_j. x_j is the number of identical redundant components, i.e. the number of redundancies, at the jth component; f_i is the ith constraint function; b_i is the maximum allowable amount of the ith resource; f_0 is an objective function of the problem; Z^+ is the set of non-negative integers while R denote the set of real numbers. The objective of the reliability optimization problem is to find the components reliability in such a way that it maximize the overall system reliability under the given resources constraints, or minimizes the total cost under minimum system reliability and other resource limitations.

- **Muti-Objective Reliability Optimization Problems**

$$\begin{aligned} \text{Max } F = (&f_1(r_1, r_2, \ldots, r_n, x_1, x_2, \ldots, x_n),\ f_2(r_1, r_2, \ldots, r_n, x_1, x_2, \ldots, x_n), \ldots, \\ &f_K(r_1, r_2, \ldots, r_n, x_1, x_2, \ldots, x_n)) \end{aligned}$$

subject to

$$\begin{array}{lll} f_i^c(r_1, r_2, \ldots, r_n, x_1, x_2, x_n) \le b_i, & \text{for} & i = 1, 2, \ldots, m \\ l_j \le x_j \le u_j, x_j \in Z^+, & \text{for} & j = 1, 2, \ldots, n \\ r_j \in (0, 1) \subset R, & \text{for} & j = 1, 2, \ldots, n \end{array}$$

$f_k, \forall k = 1, 2, \ldots, K$ is one of the objective functions of the problem, K is total number of objective functions. In most practical situations involving reliability optimization, there are several mutually conflicting goals such as maximizing system reliability and minimizing cost, weight, volume and constraints required to be addressed simultaneously. Some main objectives can be expressed as.

***Objective* 1:** The most important objective is the maximization of system reliability (R_s). It enables the system to function satisfactorily throughout its intended service period

$$\text{Max } R_s$$

As in our approach we are considering all minimization problems. Hence, the above objective is equivalent to minimization of system unreliability $(Q_s = 1 - R_s)$, can be expressed as follows:

$$\text{Min } Q_S$$

***Objective* 2:** The addition of the redundant components increases not only the system reliability but also its overall cost (C_S). A manufacturer has to balance these conflicting objectives, keeping in view the importance of reducing the overall cost

$$\text{Min } C_S$$

***Objective* 3:** As with cost, every added redundant component increases the weight of the system. Usually, the overall weight of a system needs to be minimized along with its cost even as reliability is maximized (or unreliability is minimized)

$$\text{Min } W_S$$

Reliability optimization problems can be categorized as redundancy allocation, reliability allocation and reliability–redundancy allocation problems in accordance to the type of their decision variables. If the number of redundancies, x_j's for all j, are the only variables, the problem is called redundancy allocation problem. If component reliabilities, r_j's for all j, are the only variables, the problem is termed as reliability allocation and if the decision variables of the problem include both the component reliabilities and redundancies, the problem is called a reliability–redundancy allocation problem. From the mathematical programming point of view, redundancy allocation is a pure integer nonlinear programming problem (INLP) while reliability allocation problems can be viewed as a continuous nonlinear programming problem (NLP) and reliability–redundancy allocation can be termed as a mixed integer nonlinear programming problem (MINLP).

The suitability of a metaheuristics varies problem to problem. In other words, a metaheuristic which is giving promising results on a particular set of problem may show poor performance on different problems. Optimization of reliability of complex systems is an extremely important issue in the field of reliability engineering. Over the past three decades, reliability optimization problems have been formulated as nonlinear programming problems within either single objective or multi-objective environment.

As discussed above, reliability optimization problems are categorized into three typical problems according to the types of their decision variables: reliability allocation, redundancy allocation and reliability–redundancy allocation. A number of algorithms—also categorized as approximate, exact, or heuristic/metaheuristic have been used to find optimal solutions to these problems. Algorithms such as the surrogate worth trade-off, the Lagrange multiplier, and geometric programming methods and their variants, which are efficient for the exact solution of continuous problems of the type posed by reliability allocation optimization, can only approximate the solution in the case of redundancy or redundancy–reliability allocation optimization [93, 153]. The approximation techniques involve the use of trial and error approaches to obtain integer solutions [148, 153]. The approximation techniques were popular when exact solution algorithms were not well developed. The advent of the exact algorithms, such as integer programming (IP), branch and bound, and dynamic programming (DP) [78] have made the approximation techniques less popular for solving redundancy allocation problems. The approximation and exact algorithms, though efficient with small-to-moderate sized problems having desirable properties such as convexity or monotonicity, are deficient with complex large scale ones, such as real-life network reliability and redundancy allocation optimization problems [7, 8]. Although the heuristic/metaheuristic approaches (example GA, SA, ACO, PSO and TS) yield solutions which are not exact, they do have the ability to efficiently handle complexity [5] and thus become increasingly popular in the reliability optimization field. The redundancy and the redundancy–reliability allocation optimization problems are generally more difficult to solve than the reliability allocation ones. This is because the former belongs to the class of NP-hard problems (this phenomenon was demonstrated by Chern [19], Coit et al. [32], Coit and Konak [27]) which involve non-convex and combinatorial search spaces and require a considerable amount of computational effort to find exact optimal solutions [62]. The reliability allocation problems on the other hand involve continuous optimization with a number of classical solution algorithms based on gradient and direct search methods at their disposal. They are thus relatively easier to solve. Examples of the solution algorithms which were applied in the context of the three optimization problem types are presented in Tables 2 and 3 [142].

Tillman et al. [140] has extensively reviewed the several optimization techniques for system reliability design. However, they reviewed the application of only derivative-based optimization techniques, as metaheuristics were not applied to the reliability optimization problems by that time. Mohan and Shanker [90] applied random search technique to optimize complex system. Luus [81] optimized such problems by nonlinear integer programming procedure. Over the last decade, metaheuristics have also been applied to solve the reliability optimization problems. To list a few of them Coit and Smith [30, 31], were the first to employ a GA to solve reliability optimization problems. Ravi et al. [114] developed an improved version of nonequilibrium simulated annealing called INESA and applied it to solve a variety of reliability optimization problems. Further, Ravi et al. [115, 116] first formulated various complex system reliability optimization problems with single

Table 2 Different optimization techniques used in reliability optimization of SOOP category

Model type	Solution techniques	Algorithm description	Sources
Redundancy allocation	Approximate	Interval arithmetic optimization	Munoz and Pierre [93]
	Exact	Lagrange relaxation algorithm in conjunction with dynamic programming (DP)	Ashrafi and Berman [7]
		Integer programming (IP) algorithm	Coit and Liu [28]
		Lexicographic order (P&K-Ag)	Prasad and Kuo [108]
		Improved surrogate constraint (ISC) algorithm	Onishi et al. [96]
		IP (due to Misra)	Misra and Sharma [87]
	Heuristic–metaheuristic	Simulated annealing (SA)	Atiqullah and Rao [8]
		DETMAX algorithm	Kim and Yum [62]
		Genetic algorithm (GA)	Deeter and Smith [36]
		Heuristic algorithm	Bala and Aggarwal [9]
		GA	Coit and Smith [29]
		SA	Wattanapongsakorn and Levitan [146]
		Heuristic algorithm	You and Chen [153]
		Approximate linear programming heuristic	Prasad and Raghavachari [107]
		Tabu search (TS)	Kulturel-Konak et al. [66]
		Variable neighbourhood search algorithm	Liang and Chen [78]
		SA	Wattanapongsakorn and Levitan [145]
		GA	Coit and Smith [31]
		GA	Coit and Smith [30]
Reliability allocation	Exact	Cutting plane algorithm	Majety et al. [84]
	Heuristic–metaheuristic	Random search algorithm	Mohan and Shanker [90]
		PSO	Pant et al. [101]
		CSA	Kumar et al. [67]
Redundancy–reliability allocation	Exact	Surrogate dual problem under DP algorithm	Hikita et al. [52]
		Surrogate constraint algorithm	Hikita et al. [51]
		DP	Yalaoui et al. [149]
		Mixed integer programming (MIP) algorithm	Misra and Sharma [87]

Table 3 Different optimization techniques used in reliability optimization of MOOP category

Model type	Solution techniques	Algorithm description	MOA type	Sources
Redundancy allocation	Approximate	Surrogate worth trade-off (SWT) method under dual decomposition algorithm	Scaler	Sakawa [124]
		Direct search by Min–Max algorithm		Misra and Sharma [88]
	Exact	IP due to Misra		Misra and Sharma [87]
		The weighting method in conjunction with a heuristic and an IP algorithm		Coit and Konak [27]
		Weighting method under an IP software package		Coit et al. [32]
	Heuristic–metaheuristic	GA and Monte Carlo simulation		Marseguerra et al. [85]
		Multi-objective GA		Coit and Baheranwala [26]
		Elitist non-dominated sorting GA 2 (NSGA 2)	Pareto	Wattanapongskorn and Coit [147]
		GA		Taboada and Coit [139]
		NSGA		Zhao et al. [156]
		Multi-objective ant colony		Zafiropoulos and Dialynas [154]
		Simulated annealing (SA)		Yamachi et al. [150]
		Multi-objective GA		Li and Haimes [75]
Reliability allocation	Exact	Three levels decomposition approach and the Khun Tucker multiplier method	Scaler	Salazar et al. [126]
	Heuristic–metaheuristic	NSGA 2	Pareto	Salazar et al. [126]

(continued)

Table 3 (continued)

Model type	Solution techniques	Algorithm description	MOA type	Sources
		NSGA2		Kishor et al. [63, 64]
		Ant colony (AC)		Shelokar et al. [127]
		PSO		Pant et al. [100, 102]
Redundancy–reliability allocation	Approximate	SWT	Scaler	Sakawa [123]
		Direct search technique combined with the Min–Max method		Misra and Sharma [89]
		Goal programming (GP) and goal attainment methods (GAT)		Dhingra [37]
	Heuristic/metaheuristic	Evolutionary algorithm (EA)	Pareto	Ramírez-Rosado and Bernal-Agustín [111]
		GA		Huang et al. [59]

and multi-objectives as fuzzy global optimization problems. They also developed and applied the non-combinatorial version of another metaheuristic, viz., threshold accepting to solve these problems. Recently, Shelokar et al. [127] applied the ant colony optimization (ACO) algorithm to these problems and obtained comparable results to those reported by Ravi et al. [114]. Vinod et al. [143] applied GAs to Risk Informed In-Service Inspection (RI-ISI) which aims at prioritizing the components for inspection within the permissible risk level thereby avoiding unnecessary inspections. A new fuzzy MOO method is introduced and it is used for the optimization decision-making of the series and complex system reliability with two objectives is presented by Mahapatra and Roy [82]. Mahapatra [83] considered a series-parallel system to find out optimum system reliability with an additional entropy objective function. Marseguerra et al. [86] applied GA to solve the reliability problem. Salazar et al. [125, 126] solved the system reliability optimization problem by using several EAs and MOEAs. Ravi [113] developed an extended version of the great deluge algorithm and demonstrated its effectiveness in solving the reliability optimization problems. Deep and Deepti [35] applied self-organizing migrating genetic algorithm (C-SOMGA) to optimize such type of problems. Furthermore Kuo and Prasad [70], Kuo and Wan [69] reviewed different reliability optimization and allocation techniques. More recently, Pant et al. [100, 102], Kumar et al. [67] applied PSO and cuckoos search algorithm (CSA) to solve reliability optimization problems.

3 Why Particle Swarm Approach to Reliability Optimization?

Reliability optimization problem are NP-hard in nature so it is quite difficult to achieve optimal reliability design [19]. The solution of such NP-hard optimization problems, however, is more difficult using heuristics or exact algorithms. This is because these optimization problems generate a very large search space, and searching for optimal solutions using exact methods or heuristics will necessarily be extremely time consuming. Such methods are particularly advantageous when the problem is not large. Therefore, metaheuristic algorithms, particularly cuckoos search algorithm (CSA), grey wolf optimization algorithm (GWO), ant colony optimization (ACO), genetic algorithm (GA), differential evolution (DE), particle swarm optimization (PSO), etc., are suitable for solving reliability optimization problems. The main concept of PSO is based on the food searching behavior of birds flocking or fish schooling. When PSO is adopted to solve problems, each particle has its own location and velocity, which determine the flying direction and distance, respectively. Comparing with other evolutionary approaches PSO has the following advantages [21, 53, 58, 120]:

(i) It has less parameters.
(ii) It is easy in implementation.
(iii) It has fast convergence.

These advantages are good for solving the reliability optimization problems because a population of particles in PSO can operate simultaneously so that the possibility of paralysis in the whole process can be reduced. Different PSO methods have been already successfully applied by Zavala et al. [155], Chen [18], Pandey et al. [98], Levitin et al. [74], Yeh [152], Coelho [23], Zou et al. [160], Pant and Singh [101] Pant et al. [100,102], etc., in reliability optimization problems.

References

1. **Afshinmanesh, F., Marandi, A., and Rahimi-Kian, A.,** A novel binary particle swarm optimization method using artificial immune system, in *IEEE International Conference on Computer as a Tool,* 2005, 217-220.
2. **Alatas, B. and Akin, E.,** Multi-objective rule mining using a chaotic particle swarm optimization algorithm, *Knowledge-Based Systems*, 22, 2009, 455-460.
3. **Al-kazemi, B. and Mohan, C. K.,** Multi-phase discrete particle swarm optimization, in *Fourth International Workshop on Frontiers in Evolutionary Algorithms*, 2002.
4. **AlRashidi, M. R. and El-Hawary, M. E.,** Emission-economic dispatch using a novel constraint handling particle swarm optimization strategy, in *Canadian Conference on Electrical and Computer Engineering,* 2006, 664-669.
5. **Altiparmak, F., Dengiz, B., and Smith, A. E.,** Reliability optimization of computer communication networks using genetic algorithms, in *IEEE International Conference on Systems, Man, and Cybernetics,*1998, 4676-4681.

6. **Arumugam, M. S and Rao, M. V. C.,** On the improved performances of the particle swarm optimization algorithms with adaptive parameters, cross-over operators and root mean square (RMS) variants for computing optimal control of a class of hybrid systems, *Applied Soft Computing*, 8, 2008, 324-336.
7. **Ashrafi, N. and Berman, O.,** Optimization models for selection of programs, considering cost and reliability, *IEEE Transactions on Reliability*, 41, 1992, 281-287.
8. **Atiqullah, M. M. and Rao, S. S.,** Reliability optimization of communication networks using simulated annealing, *Microelectronics Reliability*, 33,1993, 1303-1319.
9. **Bala, R. and Aggarwal, K. K.,** A simple method for optimal redundancy allocation for complex networks, *Microelectronics Reliability*, 27, 1987, 835-837.
10. **Banks, A., Vincent, J., and Anyakoha, C.,** A review of particle swarm optimization. Part I: Background and Development, *Natural Computing*, 6, 2007, 467-484.
11. **Banks, A., Vincent, J., and Anyakoha, C.,** A review of particle swarm optimization. Part II: Hybridisation, combinatorial, multicriteria and constrained optimization, and indicative applications, *Natural Computing*, 7, 2008, 109-124.
12. **Bartz-Beielstein, T., Limbourg, P., Mehnen, J., Schmitt, K., Parsopoulos, K. E., and Vrahatis, M. N.,** Particle swarm optimizers for Pareto optimization with enhanced archiving techniques, in *Congress on Evolutionary Computation*, 2003, 1780-1787.
13. **Briza, A. C. and Naval Jr, P. C,** Stock trading system based on the multi-objective particle swarm optimization of technical indicators on end-of-day market data, *Applied Soft Computing*, 11, 2011, 1191-1201.
14. **Cai, J., Ma, X., Li, Q., Li, L., and Peng, H.,** A multi-objective chaotic particle swarm optimization for environmental/economic dispatch, *Energy Conversion and Management*, 50, 2009, 1318-1325.
15. **Cao, C. H., Li, W. H., Zhang, Y. J., and Yi, R. Q.,** The geometric constraint solving based on memory particle swarm algorithm, in *International Conference on Machine Learning and Cybernetics,* 2004, 2134-2139.
16. **Carlisle, A. and Dozier, G.,** Adapting particle swarm optimization to dynamic environments, in *International Conference on Artificial Intelligence*, 2000, 429-434.
17. **Chatterjee, A. and Siarry, P.,** Nonlinear inertia weight variation for dynamic adaptation in particle swarm optimization, *Computers & Operations Research*, 33, 2006, 859-871.
18. **Chen, T. C.,** Penalty guided PSO for reliability design problems, in *PRICAI 2006: Trends in Artificial Intelligence*, 2006, 777-786.
19. **Chern, M. S.,** On the computational complexity of reliability redundancy allocation in a series system, *Operations Research Letters*, 11, 1992, 309-315.
20. **Clerc, M. and Kennedy, J.,** The particle swarm-explosion, stability, and convergence in a multidimensional complex space, *IEEE Transactions on Evolutionary Computation*, 6, 2002, 58-73.
21. **Clow, B. and White, T.** An evolutionary race: A comparison of genetic algorithms and particle swarm optimization used for training neural networks, in *International Conference on Artificial Intelligence*, 2004, 582-588.
22. **Coelho, J. P., Oliviera, P. M., and Cunha, J. B.,** Non-linear concentration control system design using a new adaptive PSO, in *5th Portugese Conference on Automatic Control*, 2002.
23. **Coelho, L. S.,** An efficient particle swarm approach for mixed-integer programming in reliability-redundancy optimization applications, *Reliability Engineering & System Safety*, 94, 2009, 830-837.
24. **Coello, C. A.C. and Lechuga, M. S,** MOPSO: A proposal for multiple objective particle swarm optimization, in *Congress on Evolutionary Computation,* 2002, 1051-1056.
25. **Coello, C. A.C., Pulido, G. T., and Lechuga, M. S.,** Handling multiple objectives with particle swarm optimization, *IEEE Transactions on Evolutionary Computation,* 8, 2004, 256-279.
26. **Coit, D. W. and Baheranwala, F.,** Solution of stochastic multi-objective system reliability design problems using genetic algorithms, in *European Safety and Reliability Conference,* 2005, 391-398.

27. **Coit, D. W. and Konak, A.,** Multiple weighted objectives heuristic for the redundancy allocation problem, *IEEE Transactions on Reliability*, 55, 2006, 551-558.
28. **Coit, D. W. and Liu, J. C.,** System reliability optimization with k-out-of-n subsystems, *International Journal of Reliability Quality and Safety Engineering*, 7, 2000, 129-142.
29. **Coit, D. W. and Smith, A. E.,** Considering risk profiles in design optimization for series-parallel systems, in *Annual Reliability and Maintainability Symposium,*1997, 271-277.
30. **Coit, D. W. and Smith, A. E.,** Reliability optimization of series-parallel systems using a genetic algorithm, *IEEE Transactions on Reliability*, 45, 1996a, 254-260.
31. **Coit, D. W. and Smith, A. E.,** Penalty guided genetic search for reliability design optimization, *Computers & Industrial Engineering*, 30, 1996b, 895-904.
32. **Coit, D. W., T. Jin, T., and Wattanapongsakorn, N.,** System optimization with component reliability estimation uncertainty: A multi-criteria approach, *IEEE Transactions on Reliability*, 53, 2004, 369-380.
33. **De Carvalho, A. B., Pozo, A., and Vergilio, S. R.,** A symbolic fault-prediction model based on multiobjective particle swarm optimization, *Journal of Systems and Software*, 83, 2010, 868-882.
34. **Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T.,** A fast and elitist multiobjective genetic algorithm: NSGA-II, *IEEE Transactions on Evolutionary Computation*, 6, 2002, 182-197.
35. **Deep K. and Deepti,** Reliability Optimization of Complex Systems through C-SOMGA, Journal of Information and Computing Science, 4, 2009, 163-172.
36. **Deeter, D. L. and Smith, A. E.,** Heuristic optimization of network design considering all-terminal reliability, in *Annual Reliability and Maintainability Symposium,* 1997, 194-199.
37. **Dhingra, A. K.,** Optimal apportionment of reliability and redundancy in series systems under multiple objectives, *IEEE Transactions on Reliability*, 41, 1992, 576-582.
38. **Dian, P. R, Siti, M. S., and Siti, S. Y.,** Particle Swarm Optimization: Technique, System and Challenges, *International Journal of Computer Applications*, 14, 2011, 19-27.
39. **Du, W. and Li, B.,** Multi-strategy ensemble particle swarm optimization for dynamic optimization, *Information sciences*, 178, 2008, 3096-3109.
40. **Eberhart, R. and Shi, Y.,** Comparing inertia weights and constriction factors in particle swarm optimization, in *IEEE Congress on Evolutionary Computation,* 2000, 84-88.
41. **Eberhart, R. and Shi, Y.,** Tracking and optimizing dynamic systems with particle swarms, in *IEEE Congress on Evolutionary Computation,* 2001, 94-100.
42. **Eberhart, R., Simpson, P., and Dobbins, R.,** Computational intelligence PC tools. Academic Press Professional, Inc., USA, 1996.
43. **Engelbrecht, A. P.** Fundamentals of computational swarm intelligence, Jhon Wiley & Sons Ltd., 2005.
44. **Engelbrecht, A. P. and van Loggerenberg,** Enhancing the NichePSO, in *IEEE Congress on Evolutionary Computation,* 2007, 2297-2302.
45. **Fan, S. and Chiu, Y.,** A decreasing inertia weight particle swarm optimizer, *Engineering Optimization*, 39, 2007, 203-228.
46. **Feng, Y., Teng, G. F., Wang, A. X., and Yao, Y. M.,** Chaotic inertia weight in particle swarm optimization, in *International Conference on Innovative Computing, Information and Control*, 2007, 475-475.
47. **Feng, Y., Yao, Y. M., and Wang, A. X**., Comparing with chaotic inertia weights in particle swarm optimization, in *Conference on Machine Learning and Cybernetics, International*, 2007, 329-333.
48. **Fieldsend, J. E. and Singh, S., A** Multi-objective algorithm based upon particle swarm optimisation, an efficient data structure and turbulence., *Workshop on Computational Intelligence*, Birmingham, UK, 2002, 37–44,
49. **Fieldsend, J. E.,** Multi-objective particle swarm optimization methods, Department of Computer Science, University of Exeter, 2004.

50. **Goh, C. K., Tan, K. C., Liu, D. S., and Chiam, S. C.,** A competitive and cooperative co-evolutionary approach to multi-objective particle swarm optimization algorithm design, *European Journal of Operational Research*, 202, 2010, 42-54.
51. **Hikita, M., Nakagawa, Y., Nakashima, K., and Narihisa, H.,** Reliability optimization of systems by a surrogate-constraints algorithm, *IEEE Transactions on Reliability*, 41, 1992, 473-480.
52. **Hikita, M., Nakagawa, Y., Nakashima, K., and Yamato, K.,** Application of the surrogate constraints algorithm to optimal reliability design of systems, *Microelectronics and reliability*, 26, 1986, 35-38.
53. **Hodgson, R. J. W.** Particle swarm optimization applied to the atomic cluster optimization problem, in *Genetic and evolutionary computation conference,* 2002, 68–73.
54. **Hu, X. and Eberhart, R.,** Adaptive particle swarm optimization: Detection and response to dynamic systems, in *Congress on Evolutionary Computation,* 2002a, 1666-1670.
55. **Hu, X. and Eberhart, R.,** Multiobjective optimization using dynamic neighborhood particle swarm optimization, in *Congress on Evolutionary Computation,* 2002b, 1677-1681.
56. **Hu, X. and Eberhart, R.,** Solving constrained nonlinear optimization problems with particle swarm optimization, in *World Multiconference on Systemics, Cybernetics and Informatics*, 2002c, 203–206.
57. **Hu, X. and Eberhart, R.,** Tracking dynamic systems with PSO: Where's the cheese, in *the Workshop on Particle Swarm Optimization*, Indianapolis, 2001, 80–83.
58. **Hu, X., Y. Shi, and R. Eberhart,** Recent advances in particle swarm, in *IEEE Congress on Evolutionary Computation,* 2004, 90-97.
59. **Huang, H. Z., Qu, J., and Zuo, M. J.,** A new method of system reliability multi-objective optimization using genetic algorithms, in *Annual Reliability and Maintainability Symposium,* 2006, 278-283.
60. **Jiao, B., Lian, Z., and Gu, X.,** A dynamic inertia weight particle swarm optimization algorithm, *Chaos, Solitons & Fractals*, 37, 2008, 698-705.
61. **Kennedy, J. and Eberhart, R.,** A discrete binary version of the particle swarm algorithm, in *IEEE International Conference on Systems, Man, and Cybernetics, Computational Cybernetics and Simulation.,* 5, 1997, 4104-4108.
62. **Kim, J. H. and Yum, B. J.,** A heuristic method for solving redundancy optimization problems in complex systems, *IEEE Transactions on Reliability,* 42, 1993, 572-578.
63. **Kishor, A., Yadav, S. P., and Kumar, S.,** A Multi-objective Genetic Algorithm for Reliability Optimization Problem, *International Journal of Performability Engineering*, 5, 2009, 227–234.
64. **Kishor, A., Yadav, S. P., and Kumar, S.,** Application of a Multi-objective Genetic Algorithm to solve Reliability Optimization Problem, in International Conference on Computational Intelligence and Multimedia Applications, 2007, 458-462.
65. **Knowles, J. D. and Corne, D. W.,** Approximating the nondominated front using the Pareto archived evolution strategy, *Evolutionary computation*, 8, 2000, 149-172.
66. **Kulturel-Konak, S., Smith, A. E., and Coit, D. W.,** Efficiently solving the redundancy allocation problem using tabu search, *IIE transactions*, 35, 2003, 515-526.
67. **Kumar, A., Pant, S., and Singh, S.B.,** Reliability Optimization of Complex System by Using Cuckoos Search algorithm , *Mathematical Concepts and Applications in Mechanical Engineering and Mechatronics*, IGI Global, 2016, 95-112.
68. **Kumar, A. & Singh, S.B. (2008).** Reliability analysis of an n-unit parallel standby system under imperfect switching using copula, *Computer Modelling and New Technologies,* 12(1), 2008, 47-55.
69. **Kuo, W. and Wan, R.,** Recent advances in optimal reliability allocation, *IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans*, 37, 2007, 1-36.
70. **Kuo,W. and Prasad,V. R.,** An annotated overview of system-reliability optimization, *IEEE Transactions on Reliability*, 49, 2000, 176-187.
71. **Laskari, E. C., Parsopoulos, K. E., and Vrahatis, M. N.,** Particle swarm optimization for integer programming, in *IEEE Congress on Evolutionary Computation,* 2002, 1582-1587.

72. **Lei, K., Qiu, Y., and He, Y.,** A new adaptive well-chosen inertia weight strategy to automatically harmonize global and local search ability in particle swarm optimization, in *International Symposium on Systems and Control in Aerospace and Astronautics,* 2006, 977-980.
73. **Leong, W. F. and Yen, G. G.,** PSO-based multiobjective optimization with dynamic population size and adaptive local archives, *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics,* 38, 2008, 1270-1293.
74. **Levitin, G., Hu, X., and Dai, Y. S.,** Particle Swarm Optimization in Reliability Engineering, *Intelligence in Reliability Engineering*, 2007, 83-112.
75. **Li, D. and Haimes, Y. Y.,** A decomposition method for optimization of large-system reliability, *IEEE Transactions on Reliability*, 41, 1992, 183-188.
76. **Li, X. and Deb, K.,** Comparing lbest PSO niching algorithms using different position update rules, in *IEEE Congress on Evolutionary Computation* ,2010, 1-8.
77. **Li, X.,** A non-dominated sorting particle swarm optimizer for multiobjective optimization, in *Genetic and Evolutionary Computation,* 2003, 198-198.
78. **Liang, Y. C. and Chen, Y. C.,** Redundancy allocation of series-parallel systems using a variable neighborhood search algorithm, *Reliability Engineering & System Safety*, 92, 2007, 323-331.
79. **Liu, D., Tan, K. C., Goh, C. K., and Ho, W. K.,** A multiobjective memetic algorithm based on particle swarm optimization, *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 37, 2007, 42-50.
80. **Liu, X., Liu, H., and Duan, H.,** Particle swarm optimization based on dynamic niche technology with applications to conceptual design, *Advances in Engineering Software*, 38, 2007, 668-676.
81. **Luus, R.,** Optimization of system reliability by a new nonlinear integer programming procedure, *IEEE Transactions on Reliability,* 24, 1975, 14-16.
82. **Mahapatra, G. S. and Roy, T. K.,** Fuzzy multi-objective mathematical programming on reliability optimization model, *Applied mathematics and computation*, 174, 2006, 643-659.
83. **Mahapatra, G.S.,** Reliability optimization of entropy based series-parallel system using global criterion method*, Intelligent Information Management,* 1, 2009, 145-149.
84. **Majety, S. R.V., Dawande, M., and Rajgopal, J.,** Optimal reliability allocation with discrete cost-reliability data for components, *Operations Research*, 47, 1999, 899-906.
85. **Marseguerra, M., E. Zio, E., Podofillini, L., and Coit, D. W,** Optimal design of reliable network systems in presence of uncertainty, *IEEE Transactions on Reliability,*, 54, 2005, 243-253.
86. **Marseguerra, M., Zio, E., and Bosi, F.,** Direct Monte Carlo availability assessment of a nuclear safety system with time-dependent failure characteristics, *International Conference on Mathematical Methods in Reliability*, 2002, 429-432.
87. **Misra, K. B. and Sharma, U.,** An efficient algorithm to solve integer-programming problems arising in system-reliability design, *IEEE Transactions on Reliability,* 40, 1991a, 81-91.
88. **Misra, K. B. and Sharma, U.,** An efficient approach for multiple criteria redundancy optimization problems, *Microelectronics Reliability*, 31, 1991b, 303-321.
89. **Misra, K. B. and Sharma, U.,** Multicriteria optimization for combined reliability and redundancy allocation in systems employing mixed redundancies, *Microelectronics Reliability*, 31, 1991c, 323-335.
90. **Mohan, C. and Shanker, K.,** Reliability optimization of complex systems using random search technique, *Microelectronics Reliability*, 28, 1987, 513-518.
91. **Moore, J. and Chapman, R.,** Application of Particle Swarm to Multi-Objective Optimization: Department of Comput. Sci. Software Eng., Auburn University, 1999.
92. **Mostaghim, S. and Teich, J.,** Strategies for finding good local guides in multi-objective particle swarm optimization (MOPSO), in IEEE *Swarm Intelligence Symposium*, 2003, 26-33.

93. **Munoz, H. and Pierre, E.,** Interval arithmetic optimization technique for system reliability with redundancy, in *International Conference on Probabilistic Methods Applied to Power Systems,* 2004, 227-231.
94. **Nickabadi, A., Ebadzadeh, M. M., and Safabakhsh, R.,** A novel particle swarm optimization algorithm with adaptive inertia weight, Applied *Soft Computing*, 11, 2011, 3658-3670.
95. **Nickabadi, A., Ebadzadeh, M. M., and Safabakhsh, R.,** DNPSO: A dynamic niching particle swarm optimizer for multi-modal optimization, in. *IEEE Congress on Evolutionary Computation*, 2008, 26-32.
96. **Onishi, J., Kimura, S., James, R. J.W., and Nakagawa, Y.,** Solving the redundancy allocation problem with a mix of components using the improved surrogate constraint method, *IEEE Transactions on Reliability*, 56, 2007, 94-101.
97. **Padhye, N., Branke, J., and Mostaghim, S.,** Empirical comparison of MOPSO methods-guide selection and diversity preservation, in *IEEE Congress on Evolutionary Computation, ,* 2009, 2516-2523.
98. **Pandey, M. K., Tiwari, M. K., and Zuo, M. J.,** Interactive enhanced particle swarm optimization: A multi-objective reliability application, in *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability*, 221, 177-191, 2007.
99. **Panigrahi, B. K., Ravikumar Pandi, V., and Das, S.,** Adaptive particle swarm optimization approach for static and dynamic economic load dispatch, *Energy conversion and management*, 49, 2008, 1407-1415.
100. **Pant, S., Anand, D., Kishor, A., & Singh, S. B.**, A Particle Swarm Algorithm for Optimization of Complex System Reliability, *International Journal of Performability Engineering,* 11(1), 2015, 33-42.
101. **Pant, S., Singh, S. B.,** Particle Swarm Optimization to Reliability Optimization in Complex System, In the proceeding of *IEEE Int. Conf. on Quality and Reliability,* Bangkok, Thailand, 2011, 211-215.
102. **Pant, S., Kumar, A., Kishor, A., Anand, D., and Singh, S.B.**, Application of a Multi-Objective Particle Swarm optimization Technique to Solve Reliability Optimization Problem, In the proceeding of *IEEE Int. Conf. on Next generation Computing Technologies*, 2015, 1004-1007.
103. **Parsopoulos, K. E. and Vrahatis, M. N.,** Particle swarm optimization method for constrained optimization problems, *Intelligent technologies–theory and application: New trends in intelligent technologies*, 2002a, 214–220.
104. **Parsopoulos, K. E. and Vrahatis, M. N.,** Recent approaches to global optimization problems through particle swarm optimization, *Natural computing*, 1, 2002b, 235-306.
105. **Parsopoulos, K. E. and Vrahatis, M. N.,** Unified particle swarm optimization for tackling operations research problems, in *IEEE Swarm Intelligence Symposium,* 2005, 53-59.
106. **Parsopoulos, K. E., Tasoulis, D. K., and Vrahatis, M. N.,** Multiobjective optimization using parallel vector evaluated particle swarm optimization, in *International conference on artificial intelligence and applications,* 2004, 2, 823-828.
107. **Prasad, R. and Raghavachari, M.,** Optimal allocation of interchangeable components in a series-parallel system, *IEEE Transactions on Reliability*, 47,1998, 255-260.
108. **Prasad,V. R. and Kuo, W.,** Reliability optimization of coherent systems, *IEEE Transactions on Reliability*, 49, 2000, 323-330.
109. **Pulido, G. T. and Coello C.A.C.,** Using clustering techniques to improve the performance of a multi-objective particle swarm optimizer, in *Genetic and Evolutionary Computation Conference ,* 2004, 225-237.
110. **Qin, Z., Yu, F., Shi, Z., and Wang, Y.,** Adaptive inertia weight particle swarm optimization, in *International conference on Artificial Intelligence and Soft Computing,* 2006, 450-459.
111. **Ramírez-Rosado, I. J. and Bernal-Agustín, J. L.,** Reliability and costs optimization for distribution networks expansion using an evolutionary algorithm, *IEEE Transactions on Power Systems*, 16, 2001, 111-118.

112. **Raquel, C. R. and Naval Jr, P. C.,** An effective use of crowding distance in multiobjective particle swarm optimization, in *Genetic and evolutionary computation conference*, 2005, 257-264.
113. **Ravi, V.,** Modified great deluge algorithm versus other metaheuristics in reliability optimization, *Computational Intelligence in Reliability Engineering*, 40, 2007, 21-36.
114. **Ravi, V., Murty, B. S. N., and J. Reddy,** Nonequilibrium simulated-annealing algorithm applied to reliability optimization of complex systems, *IEEE Transactions on Reliability*, 46, 1997, 233-239.
115. **Ravi, V.,** Optimization of complex system reliability by a modified great deluge algorithm, *Asia-Pacific Journal of Operational Research*, 21, 2004, 487–497.
116. **Ravi, V., Reddy, P. J., and Zimmermann, H. J.,** Fuzzy global optimization of complex system reliability, *IEEE Transactions on Fuzzy Systems*, 8, 2000, 241-248.
117. **Ray, T. and Liew, K. M.,** A swarm metaphor for multiobjective design optimization, *Engineering Optimization*, 34, 2002, 141-153.
118. **Reddy, M. J. and Kumar, D. N.,** An efficient multi-objective optimization algorithm based on swarm intelligence for engineering design, *Engineering Optimization*, 39, 2007, 49-68.
119. **Reibman, A. L. and Veeraraghavan, M.,** Reliability modeling: An overview for system designers, *Computer*, 24, 1991, 49-57.
120. **Reklaitis, G. V., Ravindran, A. and Ragsdell, K. M.,** Engineering optimization, methods and applications. John Wiley & Sons, 1983.
121. **Reyes-Sierra, M. and Coello, C. A.C.,** Multi-objective particle swarm optimizers: A survey of the state-of-the-art, *International Journal of Computational Intelligence Research*, 2, 2006, 287-308.
122. **Saber, A. Y., Senjyu, T., Yona, A., and Funabashi, T.,** Unit commitment computation by fuzzy adaptive particle swarm optimisation, *Generation, Transmission & Distribution, IET*, 1, 2007, 456-465.
123. **Sakawa, M.,** Multiobjective reliability and redundancy optimization of a series-parallel system by the Surrogate Worth Trade-off method, *Microelectronics and Reliability*, 17, 1978, 465-467.
124. **Sakawa, M.,** Optimal reliability-design of a series-parallel system by a large-scale multiobjective optimization method, *IEEE Transactions on Reliability,* 30, 1981, 173-174.
125. **Salazar, D, E., Rocco, S., and Claudio, M.,** Solving advanced multi-objective robust designs by means of multiple objective evolutionary algorithms (MOEA): A reliability application, *Reliability Engineering & System Safety*, 92, 2007, 697-706.
126. **Salazar, D., Rocco, C. M., and Galván, B. J.,** Optimization of constrained multiple-objective reliability problems using evolutionary algorithms, *Reliability Engineering & System Safety*, 91, 2006, 1057-1070.
127. **Shelokar, P. S., Jayaraman, V. K., and Kulkarni, B. D.,** Ant algorithm for single and multiobjective reliability optimization problems, *Quality and Reliability Engineering International*, 18, 2002, 497-514.
128. **Shi, Y. and Eberhart, R.,** A modified particle swarm optimizer, in *IEEE World Congress on Evolutionary Computational,* 1998, 69-73.
129. **Shi, Y. and Eberhart, R.,** Empirical study of particle swarm optimization, in *Congress on Evolutionary Computation,* 3, 1999a, 1945- 1950.
130. **Shi, Y. and Eberhart, R.,** Experimental study of particle swarm optimization, in *World Multiconf. Systematica, Cybernatics and Informatics*, 2000.
131. **Shi, Y. and Eberhart, R.,** Fuzzy adaptive particle swarm optimization, in *Congress on Evolutionary Computation,* 2001, 101-106.
132. **Shi, Y. and Eberhart, R.,** Parameter selection in particle swarm optimization, in *Annual Conference on Evolutionary Programming,* 1998b, 25-27.
133. **Sierra, M. R. and Coello, C. A.C.,** Improving PSO-based multi-objective optimization using crowding, mutation and e-dominance, in *International Conference on Evolutionary Multi-Criterion Optimization,* 2005, 505-519.

134. **Sivasubramani, S. and Swarup, K.,** Multiagent based particle swarm optimization approach to economic dispatch with security constraints, in *International Conference on Power Systems,* 2009, 1-6.
135. **Sun, C., Liang, H., Li, L., and Liu, D.,** Clustering with a Weighted Sum Validity Function Using a Niching PSO Algorithm, in *IEEE International Conference on*, *Networking, Sensing and Control,* 2007, 368-373.
136. **Sun, H., Han, J. J. and Levendel, H.,** A generic availability model for clustered computing systems, in *Pacific Rim International Symposium on Dependable Computing*, 2001, 241-248.
137. **Sun, L. and Gao, X.,** Improved chaos-particle swarm optimization algorithm for geometric constraint solving, in *International Conference on Computer Science and Software Engineering, 2008*, 992-995.
138. **Suresh, K., Ghosh, S., Kundu, D., Sen, A., Das, S., and Abraham, A.,** Inertia-adaptive particle swarm optimizer for improved global search, in *International Conference on Intelligent Systems Design and Applications*, 2008, 253-258.
139. **Taboada, H. and Coit, D. W.,** Data clustering of solutions for multiple objective system reliability optimization problems, *Quality Technology & Quantitative Management Journal*, 4, 2007, 35-54.
140. **Tillman, F. A., Hwang, C. L., and Kuo,W.,** Optimization of systems reliability, Marcel Dekker Inc., 1980.
141. **Tripathi, P. K., Bandyopadhyay, S., and Pal, S. K.,** Multi-objective particle swarm optimization with time variant inertia and acceleration coefficients, *Information Sciences*, 177, , 2007, 5033-5049.
142. **Twum, S. B.,** Multicriteria optimisation in design for reliability, Ph.D. Thesis, University of Birmingham, 2009.
143. **Vinod, G., Kushwaha, H. S., Verma, A. K., and Srividya, A.,** Optimisation of ISI interval using genetic algorithms for risk informed in-service inspection, *Reliability Engineering & System Safety*, 86, 2004, 307-316.
144. **Wang, J., Liu, D., and Shang, H.,** Hill valley function based niching particle swarm optimization for multimodal functions, in *International Conference on Artificial Intelligence and Computational Intelligence*, 2009, 139-144.
145. **Wattanapongsakorn, N. and Levitan, S. P.,** Reliability optimization models for embedded systems with multiple applications, *IEEE Transactions on Reliability,* 53, 2004, 406-416.
146. **Wattanapongsakorn, N. and Levitan, S.,** Reliability optimization models for fault-tolerant distributed systems, in *Reliability and Maintainability Symposium,* 2001,193-199.
147. **Wattanapongskorn, N. and Coit, D. W,** Fault-tolerant embedded system design and optimization considering reliability estimation uncertainty, *Reliability Engineering & System Safety*, 92, 2007, 395-407.
148. **Xu, Z., Kuo, W., and Lin, H. H.,** Optimization limits in improving system reliability, *IEEE Transactions on Reliability*, 39, 1990, 51-60.
149. **Yalaoui, A., Châtelet, E., and Chu, C.,** A new dynamic programming method for reliability & redundancy allocation in a parallel-series system, *IEEE Transactions on Reliability*, 54, 2005, 254-261.
150. **Yamachi, H., Tsujimura, Y., Kambayashi, Y., and Yamamoto, H.,** Multi-objective genetic algorithm for solving N-version program design problem, *Reliability Engineering & System Safety*, 91, 2006, 1083-1094.
151. **Yang, X., Yuan, J., Yuan, J., and Mao, H.,** A modified particle swarm optimizer with dynamic adaptation, *Applied Mathematics and Computation*, 189, 2007, 1205-1213.
152. **Yeh, W. C.,** A two-stage discrete particle swarm optimization for the problem of multiple multi-level redundancy allocation in series systems, *Expert Systems with Applications*, 36, 2009, 9192-9200.
153. **You, P. S. and Chen, T. C.,** An efficient heuristic for series-parallel redundant reliability problems, *Computers & Operations research*, 32, 2005, 2117-2127.

154. **Zafiropoulos, E. P. and Dialynas, E. N.,** Methodology for the optimal component selection of electronic devices under reliability and cost constraints, *Quality and Reliability Engineering International*, 23, 2007, 885-897.
155. **Zavala, A. E.M., Diharce, E. R.V., and Aguirre, A. H.,** Particle evolutionary swarm for design reliability optimization, in Evolutionary multi-criterion optimization. Third international conference, EMO 2005. Lecture notes in computer science, Coello Coello CA, Aguirre AH, Zitzler E (eds) , Springer, Guanajuato, Mexico, 3410, 2005, 856-869.
156. **Zhao, J. H., Liu, Z., and Dao, M. T.,** Reliability optimization using multiobjective ant colony system approaches, *Reliability Engineering & System Safety*, 92, 2007, 109-120.
157. **Zhao, S. Z., Liang, J. J., Suganthan, P. N., and Tasgetiren, M. F.,** Dynamic multi-swarm particle swarm optimizer with local search for large scale global optimization, in *IEEE Congress on Evolutionary Computation,* 2008, 3845-3852.
158. **Zheng, Y., Ma, L., Zhang, L. and Qian, J.,** On the convergence analysis and parameter selection in particle swarm optimization, in *International Conference on Machine Learning and Cybernetics,* 2003b, 1802-1807.
159. **Zheng, Y., Ma, L., Zhang, L., and Qian, J.,** Empirical study of particle swarm optimizer with an increasing inertia weight, in *IEEE Congress on* Evolutionary *Computation,* 2003a, 221-226.
160. **Zou, D., Wu, J., Gao, L., and Wang, X.,** A modified particle swarm optimization algorithm for reliability problems, in *IEEE Fifth International Conference on Bio-Inspired Computing: Theories and Applications (BIC-TA),* 2010, 1098-1105.