Reconfiguration of Electric Distribution Network using Modified Particle Swarm Optimization

Anoop Arya[1] Yogendra Kumar[2] Manisha Dubey[3]
[1, 2, 3] Deptt. of Electrical Engg.
Maulana Azad National Institute of Technology, BHOPAL (INDIA)

ABSTRACT
This paper presents the application of modified form of Particle Swarm Optimization as an optimization technique to the reconfiguration of electric distribution systems. The intended reconfiguration is an optimization and decision-making process which considers the maximization of the number of loads supplied associated to the minimization of the number of closed switches. A novel selection regime for the choosing of global best (*gbest*) and personal best (*pbest*) for swarm members in multi-objective particle swarm optimisation (MOPSO) without using external archives have been proposed. It means the algorithm is simple and computer coding is easy to implement to reconfiguration problem. The proposed methodology consists of use of the maximization function of the number of loads supplied and the loss minimization by the application of MOPSO. The developed algorithm has given the optimal solution in a reasonable computational time, compared to the dimension of the distribution system. Simulations for the test systems shows that the proposed MOPSO possesses better ability to finding the optimal Pareto front compared to the NSGA-II and classical PSO.

General Terms
Swarm Intelligence, Particle swarm optimization

Keywords
Reconfiguration, Distribution System, Combinatorial optimization, Multi objective particle swarm optimization

1. INTRODUCTION

Electrical Power Systems have been presenting a natural growth due to the increase in demand and consumption of electrical energy. This happens because of modern society's greater dependence on power supply. This situation is more evident in large urban centers and in regions of greater industrial concentration. The possibility of faults along the line is inherent to the system or even greater due to the rise in electrical system complexity and natural factors. Thus, after fault occurrences, it is extremely important that the supply restoration be quick to guarantee the power demand supply and the customer's satisfaction. The longer it takes, the greater the loss for the company as well as for the consumer. This situation becomes worse when the fault reaches an industrial area. The reconfiguration is a switch shifting (open/closed), loss reduction, load balancing, and service restoration process.

After occurrence of fault, some healthy components along with faulty components of distribution system remain unsupplied. To reduce the effect of fault on the customers, the electric supply is restored in de-energized healthy components due to the fault. "The process of restoring the supply in the unsupplied healthy components in distribution system by changing the switch status is called reconfiguration in distribution system." The problem of how to minimize the losses occurring due to faults in distribution system is a large-scale mixed integer-programming and the related problem of reconfiguration concerns large-scale combinatorial problems. To ensure that a power system can be rapidly restored and to ensure its reliability, distribution systems are operated on a radial configuration. The type of configuration of the network depends on status of switches already installed in the network. The switches in distribution system that are generally kept open in normal running condition are called tie switches. The tie switches are provided between the neighboring feeders. The neighboring feeders may belong to same substation or different substations. Some tie switches are also provided within the feeders. The switches which are kept closed under normal running condition in the distribution system are called sectionalizing switches.

1.1 Reconfiguration Problem in Distribution System

A typical distribution system with two substations and several switching devices are shown in Fig.1 [1]. The switches SW6, SW7 and SW10 are kept open so that the system is operating radially. Switches SW6, SW7 and SW10 are tie switches and all other switches are sectionalizing switches. After occurrence of the fault at any location between switches SW4 and SW5, all components i.e. buses B8 and B9 is de-energized along with buses B8 and B9 (faulty components), the healthy components i.e. buses B10, B11, B16, B17, B18, B20, B21 and B22 are also de-energized. The maintenance of the faulty component may take long time hours, days or even weeks. So by that time the supply can be restored in healthy de-energized components.

In case of fault considered in Fig.1, the supply can be restored in healthy de-energized components by transferring the load from de-energized area to energized area by means of reconfiguration. For supply restoration after the fault considered in Fig.1, switch SW 10 might be closed or switches SW 10 and SW6 might be closed, SW12 might be opened and there are many more options. Based on analysis of these options, the feasible and best solution should be found out. The maximum number of these options in this distribution system is 2^{12}. Since a typical distribution system

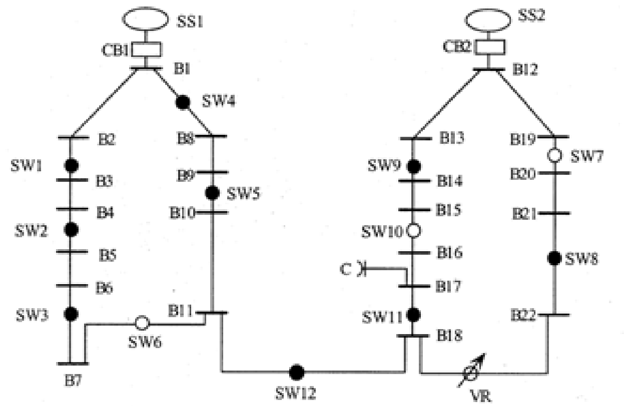

Fig 1: A Simple distribution system

would have hundreds of switches, a combinatorial analysis of the options (2^n) could easily take too long. Also the analysis and application of the constraint takes more time and complicates the use of classical optimization techniques. However, in the course of reconfiguration, several-related issues [2] must also be considered as described below:

a) To supply the electricity to as maximum customers as possible.
b) From economic point of view, there should be minimum power loss in the system after the reconfiguration is accomplished.
c) Due to various reasons such as ease of fault location, fault isolation and protective device co-ordination etc., power distribution systems are often required to operate in a radial fashion. Hence, it is important to maintain this radiality of the systems, during the reconfiguration process.
d) The software run time required by the reconfiguration algorithm should be minimized for speedier solution.

There has been considerable interest in the past in developing methods of lot of variety for reconfiguration in distribution systems. To restore the service after the fault, mainly the attempts have been made to use the knowledge base, expert systems, heuristic approach, fuzzy logic, ANN, genetic algorithms, tabu search, mathematical programming, minimal path and search technique, ant colony optimization and other methods.

Delfino B .et.al [3] reviewed knowledge based techniques to handle system reconfiguration problems and the experiences of system development were reported. Liu C-C et.al. [4] introduced a comprehensive knowledge based expert system for reconfiguration, with a special focus on the maximization of generation capability during power system reconfiguration. To minimize the unserved load (MWH), several important system parameters and constraints were considered.

A reconfiguration system for maximizing the restorable loads in a distribution network using three different reconfiguration schemes, a network state knowledge base (NSK) and a reconfiguration knowledge base (RKB) was described by K.Aokietal [5].The knowledge acquisition was automated through the use of a distribution system simulator. The proposed reconfiguration system was integrated as part of an existing distribution management system.

Lee et.al [6] developed the expert system using best first search method (greedy). The heuristic rules obtained from system operator are also incorporated to improve the solution procedure.

Y.Fukuyama and H. D. Chiang [7] developed parallel genetic algorithm for solving reconfiguration problem in distribution systems. Parallel genetic algorithm developed achieves the trade-off between computational speed and hardware cost. The method depends on the weighting factors required to convert the multi objective problem of reconfiguration into single objective. This method was problem dependent and optimality is also not guaranteed. In [8-10], loss reduction, voltage constraint, key customer and current constraint are considered. In [10], the comparison of three approaches based on GA, RTS and simulated annealing is made. Methods given in [11-13] applied integrated parallel tabu search and ordinal optimization to solve the network reconfiguration problem. Parallel tabu search has better performance than tabu search in terms of solution quality and computational time. Ordinal optimization is based on the probabilistic optimization method that speeds up computational time through reducing the number of solution candidates to be evaluated in a probabilistic way under guarantee of solution accuracy. A.Augugliaro et.al [14] developed the algorithm to solve the reconfiguration problem using the combination of genetic algorithm and tabu-search. This method minimizes the cost function while keeping constraints such as line power capacity, voltage drop at load point. Genetic-tabu algorithm is tabu search combined with genetic algorithm to reinforce convergence characteristics in a global solution space. The optimality is not guaranteed in this paper and the remotely controlled switches and manually controlled switches are not considered separately. Kuo et.al [15], presented an approach based on fuzzy set theory to estimate the loads in a distribution system and to devise a proper network reconfiguration plan following a fault is developed. The load of a branching point is estimated through fuzzy set operations. With the load estimated at hand, a heuristic search method is used in order to reach a reconfiguration plan. Chao-Ming Huang [16], an alternative approach using Fuzzy cause effect networks to solve the problem of network reconfiguration of distribution systems is developed. But the reconfiguration time taken was also long as there were some initial assumptions done in this process. Dariush.S. [17] discussed limitations encountered in some currently used reconfiguration techniques and a proposed improvement based on ANNs. The proposed scheme was tested on a 162-bus transmission system and compared with a breadth search reconfiguration scheme. The results indicated that the use of ANN in power system reconfiguration was a feasible option that should be considered for real-time applications.

The Non-Dominated Sorting Genetic Algorithm (NSGA-II), introduced by Srinivasan and Deb [18] implements the idea of a selection method based on classes of dominance of all the solutions. NSGA-II does not require weighting factors for conversion of such a multi-objective optimization problem into an equivalent single objective optimization problem. Based on the simulation results on four different automated distribution networks, the performance of the NSGA-II based scheme has been found significantly better than that of a conventional GA based method [19].

In all the categories of the methods to solve network reconfiguration discussed in the literature, the main observations are as follows:

(a) In most of the methods, multi-objective optimization problem of network reconfiguration have not been converted in to a single objective optimization problem, where as this conversion is there in [19, 20, 21] in which weighting factors are used to optimize the problem using GA. For every network these weights are to be tuned.

(b) For network reconfiguration problem, some issues are not considered at all and other issues are not considered together in any method.

(c) NSGA-II sometimes suffers with premature convergence and optimal solution is not guaranteed. Pareto optimal solutions are also not diversified.

The reconfiguration methodologies should meet all the issues discussed in this paper. In the existing methods, it can be seen that not a single methodology meets all the issues.Also, methods discussed in the literature, have a drawback that they are time-consuming in a large-scale problem and the obtained solution is locally optimal. In other words, more sophisticated methods are required to handle the network reconfiguration efficiently. To overcome the above problems, algorithms based on concept of mimicking swarm intelligence are popularly used in recent years. For instance, ant colony optimization (ACO) [22, 23] and particle swarm optimization (PSO) [24-26] are the algorithms that can be applied to the field of combinatorial optimization problems.

2. PARTICLE SWARM OPTIMIZATION

Swarm Intelligence is an optimization technique based on social behavior of swarming animals, such as a flock of birds or school of fish. It was developed by James Kennedy and Russel Eberhart in 1995 [24,26].Since then; much has been published about the subject, with applications in several areas like function optimization, electric power systems, the traveling salesman problem, telecommunications, among others. This has become an excellent way towards the solution to optimization- combined problems. There are many similarities between PSO and Genetic Algorithm (GA). The brief concepts of both algorithms are that they will produce an initial solution randomly at first. Through iterations of the evolution process, optimal value can be obtained. The difference between GA and PSO is that PSO have no explicit selection, crossover and mutation operations [27]. Furthermore, the concept of PSO is simple, and is easy to implement. Thus, the PSO is a powerful algorithm to aid and speed up the decision-making process for reconfiguration problem to identify the best switching strategy. However typical PSO is designed for continuous functions optimization, it is not designed for discrete functions optimization. Therefore, Kennedy and Eberhart proposed a modified version of PSO called Binary Particle Swarm Optimization (BPSO) that can be used to solve discrete function problems [28]. Wu-Chang Wu [29] presented Binary Particle Swarm Optimization (BPSO**)** concept to solve feeder reconfiguration problems. Lambet-Torreset.al [30] and A. Y. Abdelaziz et.al.[31] applied modified Particle Swarm Optimization in the service restoration problem.PSO seems particularly suitable for multi-objective optimization mainly because of the high speed of convergence that algorithm presents for single objective optimisation [32,33].

In PSO, in each iteration, each agent is updated with reference to two "best" values: *pbest* is the best solution (in terms of fitness) the individual particle has achieved so far, while *gbest* is the best obtained globally so far by any particle in the population. Each agent seeks to modify its position using the current positions, the current velocities, the distance between the current position and *pbest*, and the distance between the current position and *gbest.* Almost all modifications vary in some way the velocity update equation as given in eq. (1):

$$v_i^{k+1} = w_i v_i^k + c_1 rand_1 * \left(pbest_i - x_i^k\right) + c_2 rand_2 * \left(gbest - x_i^k\right) \quad \text{... (1)}$$

$$x_i^{k+1} = x_i^k + v_i^{k+1} \quad \text{... (2)}$$

In eq.(1), c_1 and c_2 are positive constants, defined as acceleration coefficients, c_1 and c_2, are often equal to 2, though other settings are used in different papers, typically with $c_1 = c_2$ and in the range [1.5- 2.5];w is the inertia weight factor; $rand_1$ and $rand_2$ are two random functions in the range of [0-1]; x_i represents the i^{th} particle and $pbest_i$ the best previous position of x_i; $gbest_i$ is the best particle among the entire population; v_i is the rate of the position change (velocity) for particle x_i. Velocity changes in the eq. (1) comprise three parts, i.e. the *momentum* part, the *cognitive* part, and the *social* part. For number of particles, the typical range is 20 – 40 [26] .This combination provides a velocity getting closer to *pbest* and *gbest*. Every particle's current position is then evolved according to the eq. (2), which produces a better position in the solution space.

3. PROBLEM FORMULATION OF NETWORK RECONFIGURATION

In this paper, an attempt is made to reduce the effect of the faults and hence cost of electricity for consumers and to improve the reliability by

1) reducing time taken by software to find the configuration.
2) selecting such a configuration according to which the time taken to operate the switches is minimum.
3) supplying the electricity to as maximum customers as possible.

Reconfiguration of the network is an emergency action and can be understood as a temporary network configuration until the cause of the problem was cleared, so that the system can be returned to the normal state

Reconfiguration has two primary objectives:

(i) to provide as much service as possible to the affected customers and
(ii) to be implemented as fast as possible.

An additional objective that has to be satisfied was that the reconfiguration solution was feasible, and that during implementation it was not possible to cause further outages.

The reconfiguration problem has been formulated as a multi-objective, multi-constrained combinatorial optimization problem. The various formulations for objective functions and constraints developed in this work are described as follows:

3.1 Objective Functions

3.1.1 Minimization of the power losses:

Minimization of the real power loss over the feeders is chosen as the first objective for the feeder reconfiguration since reducing the real power loss of the distribution feeders is a main goal in feeder reconfiguration. Minimization of the total real power losses over the feeders can be calculated as:

$$f_1(X) = \sum_{i=1}^{Nbr} Ri \times |Ii|^2 \quad \text{... (3)}$$

X = [Tie_1, Tie_2. . . $\text{Tie}_{N\,\text{tie}}$, S_{w1}, S_{w2}. . . $S_{wN\,\text{tie}}$]

where R_i and I_i are resistance and actual current of the i^{th} branch, respectively. N_{br} is the number of the branches. X is the control variable vector. Tie$_i$ is the state of the i^{th} tie switch (0 and 1 correspond to open and close states, respectively). S_{wi} is the sectionalizing switch number that forms a loop with Tie$_i$. N_{tie} is the number of tie switches.

3.1.2. Minimization of the deviation of the bus voltages:

Bus voltages are one the most significant security and service quality indices, which can be described as:

$$f_2(X) = \max_i |V_i - V_{rate}|, \quad i= 1, 2, 3, \ldots, N_{bus}, \qquad \ldots(4)$$

where N_{bus} is total number of the buses. V_i and V_{rate} are the real and rated voltages on the i^{th} bus, respectively.

3.1.3. Minimizing the number of switching operation:

Minimizing the number of switching operations can be modelled as:

$$f_3(X) = \sum_{i=1}^{Ns} |S_i - S_{oi}| \qquad \ldots. (5)$$

where S_i and S_{oi} are the new and original states of the switch i, respectively. N_s is the number of switches.

3.1.4 Load balancing over the feeders:

Load balancing is one of the major objectives in feeder reconfiguration. An effective strategy to increase the loading margin of heavily loaded feeders is to transfer a part of their loads to lightly loaded feeders. Load balancing over the feeders can be described as:

$$f_4(X) = -\min i\, |I_{i,rate} - I_i|, \quad i = 1, 2, 3, \ldots, N_{br}, \qquad \ldots(6)$$

where I_i and $I_{i,rate}$ are the actual loading and the rated currents of the i_{th} branch, respectively.

3.1.5. Formulation of the distribution reconfiguration based on norm

Formulation of the multi-objective distribution feeder reconfiguration including the mentioned objective functions can be written as:

$$\max J(X) = \|f(X) - f_o\|2$$

$$= \sqrt{(f_1(X) - f_{01})^2 + (f_2(X) - f_{02})^2 + (f_3(X) - f_{03})^2 + (f_4(X) - f_{04})^2}$$

$$f(X) = \begin{bmatrix} f_1(X) \\ f_2(X) \\ f_3(X) \\ f_4(X) \end{bmatrix}, f_0 = \begin{bmatrix} f_{01} \\ f_{02} \\ f_{03} \\ f_{04} \end{bmatrix} \qquad \ldots (7)$$

where f_{01} and f_{02} are respectively the real power loss and the maximum voltage deviation before reconfiguration, f_{03} is the worst switching operation, which is 2 N_{tie}, and f_{04} is the worst load balancing before reconfiguration. $f(X)$ and f_0 are the vector value of the objective function at point X and its corresponding worst case, respectively.

3.2 Constraints

3.2.1 Distribution line limits

$$|P^{Line}_{,ij}| < P^{Line}_{ij.\max} \qquad \ldots (8)$$

$|P^{Line}_{,ij}|$ and $P^{Line}_{ij.\max}$ are the absolute power and its corresponding maximum allowable value flowing over the distribution lines between the nodes i and j , respectively.

3.2.2 Distribution power flow equations

$$P_i = \sum_{i=1}^{N_{bus}} V_i V_j V_{ij} \cos(\theta_{ij} - \delta_i + \delta_j)$$

$$Q_i = \sum_{i=1}^{N_{bus}} V_i V_j V_{ij} \sin(\theta_{ij} - \delta_i + \delta_j) \qquad \ldots (9)$$

where, P_i and Q_i are injected active and reactive power components at the i^{th} bus on the network. V_i and δ_i are respectively the amplitude and angle of the voltage at the i^{th} bus. Y_{ij} and θ_{ij} are the respective amplitude and angle of the branch admittance between the i^{th} and j^{th} buses.

3.2.3. Objective function limit

$$f_i(X) \leq f_{0i} \quad where \;\; i = 1,2,3,4. \qquad \ldots (10)$$

3.2.4 Radial structure of the network

The main closed loops of the system are used to check the radial structure of the network. The number of main loops is calculated as:

$$N_{Fl} = N_{br} - N_{bus} + 1 \qquad \ldots (11)$$

where N_{Fl} is the number of main loops. It is noted that each main loop includes a tie switch and corresponding section switches that form a loop as shown in Fig. 2. To retain a radial network structure, when a tie switch is closed, only one section switch is opened in each loop.

For example, the vector X is defined as follows:

$$X = [Tie1, Tie2, Tie3, Sw1, Sw2, Sw3] = [1, 1, 1, 9, 6, 12] \qquad \ldots (12)$$

or

$$X = [Tie1, Tie2, Tie3, Sw1, Sw2, Sw3] = [1, 0, 1, 9, 0, 12]. \qquad \ldots (13)$$

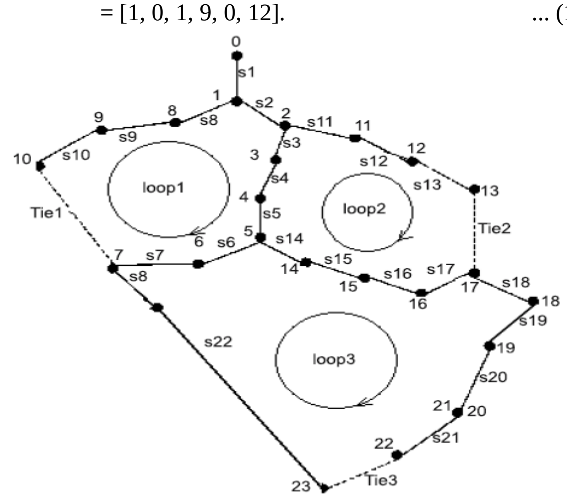

Fig 2: A Radial Distibution Network

4. SOLUTION USING PARTICLE SWARM OPTIMIZATION

4.1 Particle Swarm Implementation

PSO Technique is applied to get a reconfiguration solution based on the system functional configuration changes. It is obtained by Algorithm -1 given in Fig.3. whose purpose is to close Normally Opened Switches- NO. When the solution presents overloaded lines, the program makes use of the Algorithm-2 given in Fig.7. in Appendix, whose objective is to open Normally Closed Switches- NC in order to remove the overload. In both of the algorithms each particle represents a solution of a given problem. The solution is defined as being the switches that must have their final status changed. Each particle is represented by a matrix, where the number of lines is equal to the total number of particles established and the number of columns is equal to the number of NO switches in Algorithm-1 or NC switches in Algorithm-2. The switches may assume binary values, 0 or 1, where 0 means opened switch and 1 means closed switch. It was verified using tests accomplished empirically with the computer program that 10 particles is the ideal number to work with,in both of the algorithms.

4.2 Multi objective Particle Swarm Optimization (MOPSO):

A great number of PSO variations can be found for solving Multi objective problems. This section describes only a few of these but provides references to other approaches. The *dynamic neighborhood* MOPSO, developed by Hu and Eberhart[26], dynamically determines a new neighborhood for each particle in each iteration, based on distance in objective space. Neighborhoods are determined on the basis of the simplest objective.

4.3Multiobjective Particle Swarm Implementation

Let $f_1(x)$ be the simplest objective function, and let $f_2(x)$ be the second objective. The neighbors of a particle are determined as those particles closest to the particle with respect to the fitness values for objective $f_1(x)$. The neighborhood best particle is selected as the particle in the neighborhood with the best fitness according to the second objective, $f_1(x)$. Personal best positions are replaced only if a particle's new position dominates its current personal best solution.

The global best particle of the first swarm is used in the velocity equation of the second swarm; while the second swarm's global best particle is used in the velocity update of the first swarm. That is,

$$S_1.v_{ij}(t+1) = wS_1.v_{ij}(t) + c_1 r_{1j}(t)(S_1.y_{ij}(t) - S_1.x_{ij}(t)) + c_2 r_{2j}(t)(S_2.\hat{y}_i(t) - S_1.x_{ij}(t)) \quad \text{... (14)}$$

$$S_2.v_{ij}(t+1) = wS_2.v_{ij}(t) + c_1 r_{1j}(t)(S_2.y_{ij}(t) - S_2.x_{ij}(t)) + c_2 ri_j(t)(S_1.\hat{y}_j(t) - S.x_{2j}(t)) \quad \text{... (15)}$$

where sub-swarm S_1 evaluates individuals on the basis of objective $f_1(x)$, and subswarm S_2 uses objective $f_2(x)$. The MOPSO algorithm developed by Coello Coello and Lechuga[33] is one of the first PSO based MOO algorithms that extensively uses an archive. This algorithm is based on the Pareto archive Evolutionary Strategy, where the objective function space is separated into a number of hypercubes.

A truncated archive is used to store non-dominated solutions. During each iteration, if the archive is not yet full, a new particle position is added to the archive if the particle represents a non-dominated solution. However, because of the size limit of the archive, priority is given to new non-dominated solutions located in less populated areas, thereby ensuring that diversity is maintained. In the case that members of the archive have to be deleted, those members in densely populated areas have the highest probability of deletion. Deletion of particles is done during the process of separating the objective function space into hypercubes. Densely populated hypercubes are truncated if the archive exceeds its size limit. After each iteration, the number of members of the archive can be reduced further by eliminating from the archive all those solutions that are now dominated by another archive member.

For each particle, a global guide is selected to guide the particle toward less dense areas of the Pareto front. To select a guide, a hypercube is first selected. Each hypercube is assigned a selective fitness value,

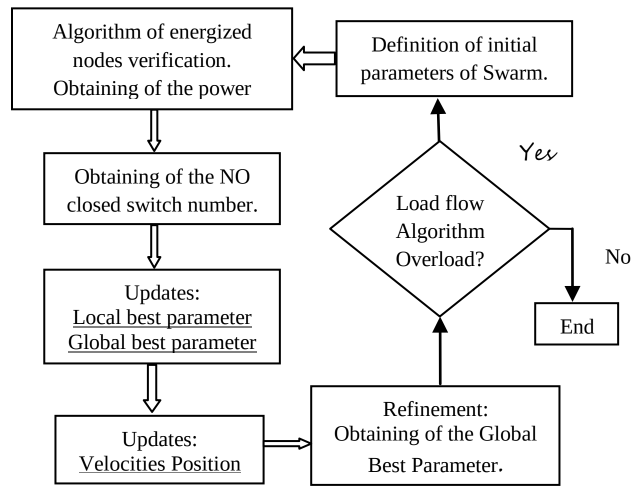

Fig 3: Flowchart of Algorithm-1

where $f_{del}(H_h) = H_h.n_s$ is the deletion fitness value of hypercube H_h; $\alpha = 10$ and $H_h.n_s$ represents the number of non-dominated solutions in hypercube H_h. More densely populated hypercubes will have a lower score. Roulette wheel selection is then used to select a hypercube, H_h, based on the selection fitness values. The global guide for particle i is selected randomly from among the members of hypercube H_h.

$$f_{sel}(H_h) = \frac{\alpha}{f_{del}(H_h)} \quad \text{… (16)}$$

Hence, particles will have different global guides. This ensures that particles are attracted to different solutions. The local guide of each particle is simply the personal best position of the particle. Personal best positions are only updated if the new position,

$$x_i(t+1) < y_i(t) \quad \text{... (17)}$$

The global guide replaces the global best, and the local guide replaces the personal best in the velocity update equation.

In addition to the normal position update, a mutation operator (also referred to as a craziness operator in the context of PSO) is applied to the particle positions. The degree of mutation decreases over time, and the probability of mutation also decrease over time. That is,

$$x_{ij}(t+1) = N(0, \sigma(t))x_{ij}(t) + v_{ij}(t+1) \quad \text{...(18)}$$

where, for example

$$\sigma(t) = \sigma(0)\, e^{-t} \quad \text{...(19)}$$

with σ(0) an initial large variance

The MOPSO algorithm is described in general as follows:

Step 1. Input data (pre-fault and post fault data) and initialize parameters. For each particle, the position and velocity vectors will be randomly initialized with the same size as the problem dimension. Create and initialize an n_x-dimensional swarm S;

Let A = Ø and $A.n_s = 0$;

Step 2. Measure the fitness (power loss) of each particle (*pbest*) and store the particle with the best fitness (*gbest*) value by running the load flow program. Evaluate all particles in the swarm;

for

all non-dominated x_i do

A = A U {xi};

End

Step3. Update velocity and position vectors according to equation (6.5), (6.6) and (6.8) for each particle.

Step 4. Check boundary constraints. Evaluate all particles in the swarm.

Step 5. Decrease the inertia weight (*w*) linearly as explained in section.

Step 6. Repeat steps 2–5 until a termination criterion is satisfied.

Parameters for MOPSO algorithm are given in appendix.

5. RESULTS AND DISCUSSION

The effectiveness of the Particle swarm optimization algorithm for reconfiguration has been studied on a IEEE Type-1 distribution systems. The details of these systems are given in Table.1. As already mentioned in the previous section, in this paper, the Multi-objective Particle swarm optimization with reduced run time complexity has been implemented for reconfiguration problem. The performance of MOPSO has also been compared with that of the conventional GA technique.

5.1 Test System

The data of test system IEEE Type-1 namely system-1 is given in Table.1.Feeder data and bus load data has been taken as given in [8] .Single line diagram of the test distribution system is given in Fig.4.

Table 1. Brief summary of the IEEE Type-1 test system.

Description	No. of Buses	No. of switches	Systems nominal voltage (KV)	Total system load	
				KW	KVAR
IEEE Type-1 Test System	10	14	13.8	5600	4080

In Test System, the switches S1, S2, S3, S7 and S12 are remote controlled automatic switches while the rest of the switches are manually controlled. Fig.4. shows the schematic diagram of the 14-switch system before reconfiguration. In this system, it has been assumed that the fault has taken place at point shown as out of service area in Fig.4 and due to this fault, the switch S4 trips for isolating the fault. As a result, the area shown inside the closed curve is left without electric power. The supply to this "out-of-service" area can be restored either by closing switch S14 (let it be called option A) or by closing switch S12 (let it be called option B). Now, switch S12 is an automatic, remote controlled switch with a typical operating time of 50 seconds whereas, switch S14 is a manually controlled switch with operating time typically in the range of 1200-1500 seconds[8]. It is to be noted that the operating time of a manually controlled switch depends on its distance from the nearest manned substation (from which an operator has to travel to operate the switch). Now, by both these options, the entire "out-of service" area can be supplied. Therefore, from the considerations of the objective function kept at first preference, both these options are equally preferable. However, as option B takes considerably less time as compared to option A to accomplish the reconfiguration task, this option is chosen and the final configuration is shown in Fig.5.

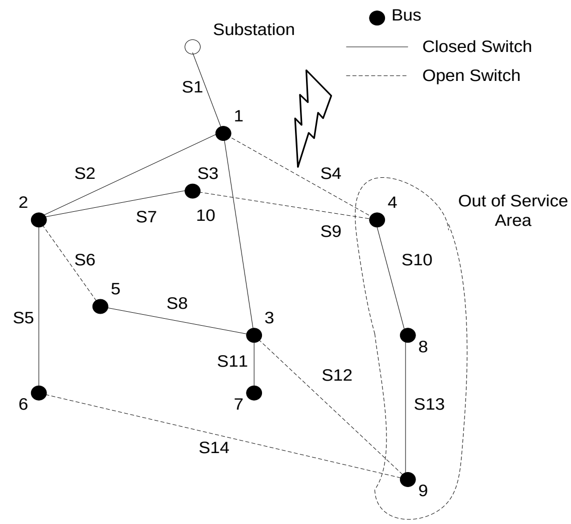

Fig 4: Network before reconfiguration

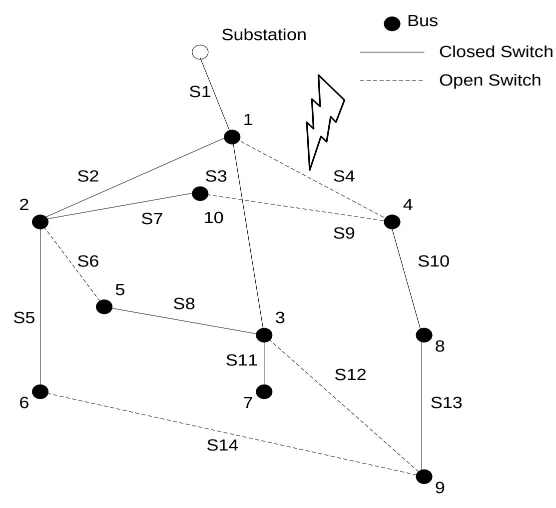

Fig 5: Network after reconfiguration

It is to be noted that for the results in Table.2, only single fault cases have been assumed for all the systems. Also, in Table.2, methods 'M1', M2' and 'M3' denote the Multi-Objective Particle Swarm Optimization Technique, Particle Swarm Optimization Technique [30], the traditional GA technique [19] respectively: It is also to be noted that in Table.2 and the other subsequent table, the entries corresponding to the "out-of-service" area has been given in terms of the total amount of load (in kW) left un-restored. For example, from Table.2 it can be concluded that

800kW of total load is always left unsupplied in system; irrespective of whichever solution methodology is used. Actually, the faulted zones in Test System contain 800kW of load that cannot be restored by any of the methods.

Closer observation of Table.2 reveals that for both systems considered, both the methods 'M1' and 'M2' produce identical results. However, because of superior implementation, the run-time taken by 'M1' is less than that taken by 'M2' to achieve the better results. On the other hand, method 'M3' can be observed to be inferior to both 'M1' and 'M2' for each system. Similarly, for all the systems, the number of manual switch operation suggested by 'M3' is more than that suggested by 'M1' or 'M2'. Also, the run-time needed by the algorithm 'M3' is always more than that needed by 'ml' and 'M2'. Therefore, from the observation of table .3, it can be concluded that methods 'M1', and 'M2' give the solution inferior to method 'M3', and method 'M1' gives the solution faster than method 'M2', and 'M3' for single fault full service restoration.

Table 2. Single fault full service restoration

		Test System
Un-Restored Load (KW)	M1	800
	M2	800
	M3	800
No. of manual switch operation	M1	1
	M2	1
	M3	2
No. of automatic switch operation	M1	1
	M2	1
	M3	0
Losses (KW)	M1	710.25
	M2	782.40
	M3	779.00
Run-time of The algorithm	M1	18.774
	M2	30.121
	M3	64.910

Fig.5 shows the iteration wise variation of objective function. From the fig.5, it can be observed that sometimes the objective function improves drastically and sometimes the objective function does not change for long time. After achieving the best point at iteration 10, the software runs for 25 iterations more just to achieve the convergence criteria. Therefore, there is a long gap between time to achieve the best solution and time to achieve the convergence criteria. Table 3. shows the time needed to reach the best solution is lesser than the time needed for convergence and in some cases; time for best solution is considerably smaller than time for convergence.

Hence, in all the cases, the software continues the execution even after it reaches the best solution, which in turn, increases the time for reconfiguration. Hence, for enhancing the speed of reconfiguration further, this time gap between best solution and convergence must be reduced (ideally should be made zero). This gap is more in conventional GA in comparison to PSO optimization because the elitism is retained for the next iteration. In presence of elitism the convergence is achieved faster. Table 3. also show the effect of inclusion of pre-fault configuration (PFC) in initial population of conventional GA and PSO optimization improved. The solution is achieved in lesser time by starting the search from PFC if both conventional GA and PSO optimization improved.

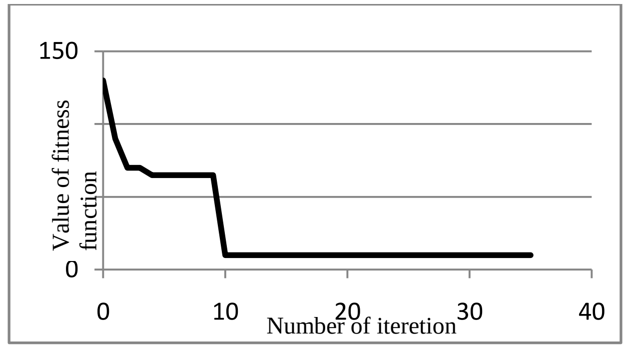

Fig 6: Objective function versus iteration

Table 3. Computational time and effect of PFC for Test System

Optimization Technique with/out Pre fault configuration	Time taken in Single fault case(sec)		
	For Best Solution	For Convergence	GAP
MOPSO without PFC	34.2785	62.3835	28.105
MOPSO with PFC	12.6453	18.774	6.1287
PSO without PFC	38.7654	73.5736	34.746
PSO with PFC	23.7463	25.7356	15.736
GA without PFC	43.5476	88.9387	45.391
GA with PFC	38.9478	64.910	25.962

6. CONCLUSION

This paper presents a novel strategy for the reconfiguration of a distribution system through the use of an optimized reconfiguration using MOPSO. The proposed approach changes the logical status of a switch which is normally open. Moreover, it considers the maximization of number of loads supplied and the minimization of the number of switched on lines. The strategy disregards a total sweeping in the databank. Thus, less computational time is necessary to find the solution. This is essential for large scale systems, with greater possibilities of configurations.

The proposed algorithms have many advanced features that are different from other conventional optimization methods:

(1) it is easy to deploy and implement to solve many optimal problems,

(2) it is very flexible for many problem formulations and

(3) it has the ability to escape the local optimal solution and achieve the global optimal solution. Based on a large number of simulation studies it has been found that the PSO based approach performs better than the GA based approach in solving the reconfiguration problem.

7. APPENDIX

Parameters for MOPSO:

Termination Options (Multiple options are allowed)

Terminate _Iters-----> Use Vars.MaxIt as a termination criterion. 1=Yes, 0=No
Terminate _Err-------> Use Vars.ErrGoal as a termination criterion.1=Yes, 0=No
Parameters common across all functions:
SParams_c1 = 2, SParams_c2 = 2
SParams_w_start = 0.9, SParams_w_end = 0.4
SParams_w_varyfor = 1
Flags_ShowViz = 0, Flags_Neighbor = 0
Run experiments for the three complex functions:
Obj_f2eval = 'Rastringrin'
Obj_lb = 2.56, Obj_ub = 5.12
SParams_Vmax = 10
Obj_f2eval = 'Griewank'
Obj_lb = 300, Obj_ub = 600
SParams_Vmax = 600
Obj_f2eval = 'Rosenbrock'
Obj_lb = 15, Obj_ub = 30
SParams_Vmax = 100

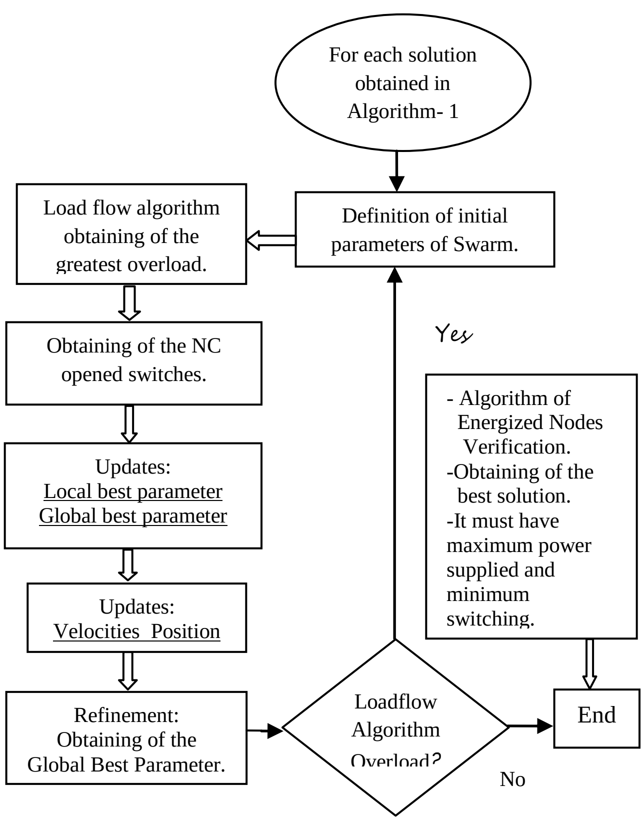

Fig 7: Flowchart for Algorithm-2

8. REFERENCES

[1] S. S. Venkata, A Pahwa, R. E. Brown and R. D. Christie, "What future distribution engineers need to learn", IEEE Transactions on Power Systems, vol. 19, no. 1, pp. 1723, February 2004.

[2] M. A. Kashem, V. Ganapathy and G. B. Jasmon, "Network reconfiguration for load balancing in distribution networks", lEEE Transactions on Generation, Transmission and Distribution, vol. 147, no. 3, pp. 171 - 175, May 2000.

[3] Delfino B, Invernizzi M, Morini A, "Knowledge-based restoration guidelines." IEEE Transactions on Computer Applications in Power1992; 5:54–59.

[4] Liu C-C, Liou K-L, Chu RF, Holen AT," Generation capability dispatch for bulk power system restoration: a knowledge-based approach". IEEE Transactions on Power Systems 1993; 8(1):316–325.

[5] K.Aokietal., “Voltage drop constrained restoration of supply by switch operation in distribution systems,” IEEE Trans. on Power Delivery, vol. 3, pp.1267–1274, July 1988

[6] H. J. Lee and Y. M. Park, “A restoration aid expert system for distribution substations,” IEEE Transactions on Power Del., vol. 11, no. 4, pp.1765–1769, Oct. 1996.

[7] Y. Fukuyama and H. D. Chiang, “A parallel genetic algorithm for service restoration in electric power distribution systems,” in Proc. IEEE FUZZ/IFES Conference, Yokohama, Japan, Mar. 1995.

[8] Yogendra Kumar, Biswarup Das, Jaydev Sharma,” Genetic algorithm for supply restoration in distribution system with priority customers”,Int. Conference on Probabilistic Methods Applied to Power Systems KTH, Stockholm, Sweden, June 11-15, 2006-05-20

[9] K.Nara et al., “Implementation of genetic algorithm for distribution Systems loss minimum re-configuration,” IEEE Transactions on Power System., vol.7, pp. 1044–1051, Aug. 1992.

[10] S. Toune et al., “Comparative study of modern heuristic algorithms to service restoration in distribution systems,” IEEE Transactions on Power Del.,vol. 17, no. 1, pp. 173–181, Jan. 2002.

[11] Marcos A. N. Guimaraes, Jorge F.C. Lorenzeti, Carlos A. Castro, “Reconfiguration of Distribution Systems for voltage stability margin enhancement using Tabu Search”, International Conference on Power System Technology POWERCON 2004, Singapore, pp.-1556-1561, 2004.

[12] S.Toune, H. Fudo, T.Genji, Y.Fukuyama, “A reactive Tabu search for service restoration in electric power system”, Int. conf. on Evolutionary computaion, pp 763-768, May 1998.

[13] H. Mori and H.Tani, “A hybrid method of PTS and ordinal optimization for distribution system service restoration”, IEEE Int. conf. on systems, Man and cybernatics, pp 3476-3483, Oct. 2003.

[14] A.Augugliaro, L.Dusonchet and R.Sanseverino, “Service restoration in compensated distribution networks using a hybrid genetic algorithm”, Electric Power System Research, vol.46, no.1, pp59-66, July 1998.

[15] H.C.Kuo and Y.Y.Hsu,“ Distribution system load estimation and service restoration using a fuzzy set approach,” IEEE Transactions on Power Del., vol. 8, no. 4, pp. 1950–1957, Oct. 1993.

[16] Chao-Ming Huang,” Multiobjective Service Restoration of Distribution Systems Using Fuzzy Cause-Effect Networks”, IEEE Transactions on Power Systems, Vol. 18, No. 2, May 2003

[17] Dariush S," Service restoration in distribution networks via network reconfiguration" IEEE Transactions on Power Delivery 1992; 7(2):952–958.

[18] K.Deb, Amrit Pratap, Sameer Agrawal, “A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II,IEEE Transactions on Evolutionary Computation.vol.6, no.2.April 2002.

[19] Yogendra Kumar, Biswarup Das, Jaydev Sharma, “Multiobjective, multiconstraint service restoration of electrical power distribution system with priority customers” IEEE Transactions on Power Delivery, vol23, No.1, pp.261-270, 2008.

[20] Augugliaro, Luigi Dusonchet ,"Evolving non-dominated solutions in multiobjective service restoration for automated distribution networks" Antonino 1, Eleonora Riva Sanseverino 25 July 2001

[21] Huang Xianchao, G.A. Taylor,”Service Restoration of Distribution Systems Based on NSGA-II”, 2010.

[22] Indira Mohanty, Jugal Kalita, Sanjoy Das, Anil Pahwa, Erik Buehler, “Ant Algorithm for the optimal restoration of Distribution feeders during cold load pickup’, IEEE Transactions on Power Delivery,pp.132-137, 2007.

[23] Isamu Watanabe, “An ACO Algorithm for service Restoration in Power Distribution Systems”,IEEE Transactions on Power delivery, vol.4, no.3 pp.2864-2871, 2006.

[24] Kennedy. J. and Eberhart. R, “Particle Swarm Optimization”, Proceedings of the Fourth IEEE Int.Conference on Neural Networks, Perth, Australia. 1995.

[25] Reynolds, C.W.: Flocks, herds and schools: a distributed behavioural model, Computer Graphics, 21(4), p.25-34, 1987.

[26] X. Hu, R.C. Eberhart, and Y. Shi, “ Particle Swarm with Extended Memory for Multi-objective Optimization” , In Proceedings of the IEEE Swarm Intelligence Symposium, pages 193–197,2003.

[27] Rania Hassan, Babak Cohanim,Olivier de Weck,“ A comparison of particle swarm optimisation and Genetic Algorithm” , Journal of American Institute of Aeronautics and Astronautics (AAIA),2005

[28] J. Kennedy and R.C. Eberhart , “ A Discrete Binary Version of the Particle Swarm Algorithm” In Proceedings of the World Multiconference on Systemics, Cybernetics and Informatics, pp 4104– 4109, 1997.

[29] Wu-Chang Wu, Men-Shen Tsai and Fu-Yuan Hsu, “A New Binary Coding Particle Swarm Optimization for Feeder Reconfiguration” IEEE Transactions on Power Systems, vol 2.no. 4 pp 243-249, 2009

[30] Lambert-Torres, G., Martins, H. G., Coutinho, M. P., Salomon, C. P.,Filgueiras, L. S., “Particle Swarm Optimization Applied to Restoration of Electrical Energy Distribution Systems”, in Proc. of The 3rd International Symposium on Intelligence Computational Applications, ISICA2008, Wuhan,China, 2008.

[31] A. Y. Abdelaziz,S. F. Mekhamer,F. M. Mohammed,M. A. L. Badr, “A Modified Particle Swarm Technique for Distribution Systems Reconfiguration” Online Journal on Electronics and Electrical Engineering (OJEEE) Vol. (1) – No. (2), 2010

[32] C.A. CoelloCoello and M.S. Lechuga. “MOPSO: A Proposal for Multiple Objective Particle Swarm Optimization” in Proceedings of the IEEE Congress on Evolutionary Computation, volume 2, pages 1051–1056, 2002.

[33] C.A. CoelloCoello, G. Toscano Pulido, and M. Salazar Lechuga, “Handling Multiple Objectives with Particle Swarm Optimization” in IEEE Transactions on Evolutionary Computation, vol.8, no.3 2004.