**National Journal of Multidisciplinary Research and Development**
**ISSN: 2455-9040**
**Impact Factor: RJIF 5.22**
**www.nationaljournals.com**
**Volume 3; Issue 3; September 2018; Page No. 68-71**

# Optimization techniques for network reconfiguration in distribution systems: An overview

**Archna Ojha[1], Vinit Panwar[2], Prem Kant[3]**
[1-3] Department of Electrical Engineering, MITRC, Alwar, Rajasthan, India

**Abstract**
Reconfiguration of radial distribution networks is becoming a viable solution for improving the performance of distribution networks. Configurations may be varied with manual or automatic switching operations so that all of the loads are supplied and reduce power loss, increase system security, and enhance power quality. Reconfiguration also relieves the overloading of the network components. The change in the network configuration is performed by opening sectionalizing (normally closed) and closing tie (normally open) switches of the network. These switching's are performed in such a way that the cardinality of the network is maintained and all of the loads are energized. Several researchers have attempted to solve the power distribution network reconfiguration problem using various techniques. This paper is an overview of comprehensive survey on network reconfiguration to bring out a clear idea for future research.

**Keywords:** distribution systems, network reconfiguration, power loss minimization, load balancing

## 1. Introduction

The distribution system is a largest portion of network in electrical power system. It is the part of power system that distributes power to various customers in ready-to use form at their place of consumption. Hence, utilities have to ensure reliable and efficient cost effective service, while providing voltages and power within the specified range. Optimal distribution planning involves network reconfiguration for distribution loss minimization, load balancing under normal operating conditions and fast service restoration under failure conditions.

The distribution network usually operates in a radial configuration, with tie switches between circuits to provide alternate feeds. Whenever components fail, some of the switches must be operated to restore power to as many customers as possible. As loads vary with time, switch operations may reduce losses in the system and transfer of loads from heavily loaded feeder. All of these are applications of network reconfigurations.

Electricity carrying equipment has a complicated inter-connection power system. Generating stations, transmission lines, sub-stations, feeders and distribution systems are some important component of this complex interlinking. To get the maximum benefits, each component of the system can be technically monitored, examined and operated. Efficient techniques and methodology are required to operate the system efficiently. By changing the status of switches, the configuration of the radial system can be changed. It is the part of Distribution automation**.** Here the closed sectionalized switches are opened and the same numbers of the tie-switches are closed. This is called reconfiguration. To minimize the path loss for the load feeding, reconfiguration is done.

Restoring power flows immediately after any type of disturbance in the power system, called services restoration. Disturbance can be occur ant type of fault in the system. Here some sectionalized switches are off-line. Some tie-switches are closed to establish the connection. It happens in the same procedure like feeder reconfiguration.

There are 3 types of electricity customer such as commercial, industrial and residential. Electricity for more than one hour can cause the loss of raw assets in the industrial hub. The supply of electricity is provided according to the priority of the customer's requirement. There will be chance of voltage dipping if in feeder system the load point is over loaded. Overloading can reduce the capacity of the feeder line. In the system, by altering the no of switches, the configuration of the system is also changed and the power system can suffer from the bad effects of the switching surge.

## 2. Network reconfiguration

Network reconfiguration is a process of changing the status of the network topology through opening or closing tie switches to optimize the network parameters. Restructuring of specific lines leads to alternate system configurations. System reconfiguration can be accomplished by placing line interconnection switches into network. Opening and closing a switch connects or disconnect a line to the existing network. Network reconfiguration in distribution systems is performed by opening sectionalizing (normally closed) and closing tie (normally open) switches of the network. These switching are performed in such a way that the radial structure of the network is maintained and all the loads are energized. A normally open tie switch is closed to transfer a load from one feeder to another while an appropriate sectionalizing switch is opened to restore the radial structure.

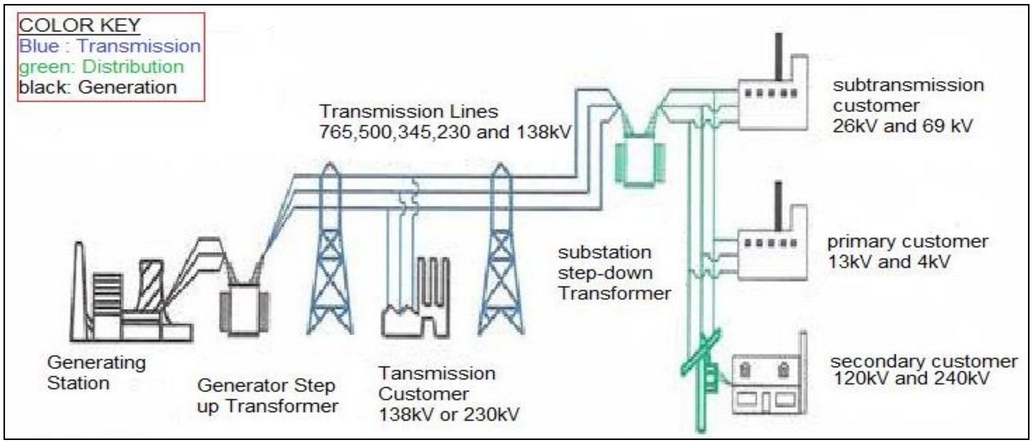


**Fig 1:** Structure of power systems

During reconfiguration of the network, the tie switch has to be closed and on the other hand, the sectionalizing switch has to be opened in the loop created, which restores radial configuration. The switch pairs are chosen through heuristics and approximate formulas for the change in losses. Branch exchange process is repeatedly applied till no more loss reductions are available. A radial distribution network can be represented by several loops. This is because, when it is connected, one tie-line makes one loop. The number of loops is equal to the number of tie-lines.

The benefits of feeder reconfiguration include:

1. Restoring power to any outage partitions of a feeder
2. Relieving overloads on feeders by shifting the load in real time to adjacent feeders.

## 3. Techniques for loss reduction

### 3.1 Particle Swarm Optimization (PSO)

Particle swarm optimization (PSO) is a population based stochastic optimization technique developed by Dr. Eberhart and Dr. Kennedy in 1995, inspired by social behavior of bird flocking or fish schooling. PSO shares many similarities with evolutionary computation techniques such as Genetic Algorithms (GA). However, unlike GA, PSO has no evolution operators such as crossover and mutation [1].

Particle Swarm Optimization (PSO) is basically a computational method that optimizes a problem by iteratively trying to improve a candidate solution with regard to a given measure of quality. It solves a problem by having a population of candidate solutions (particles), and moving these particles around in the search-space according to simple mathematical formulae over the particle's position and velocity. Each particle's movement is influenced by its local best known position, but is also guided toward the best known positions in the search-space, which are updated as better positions found by other particles. This is expected to move the swarm toward the best solutions [2]. A basic variant of the PSO algorithm works by having a population (called a swarm) of candidate solutions (called particles). These particles are moved around in the search-space according to a few simple formulae. The movements of the particles are guided by their own best known position in the search-space as well as the entire swarm's best known position. When improved positions are being discovered these will then come to guide the movements of the swarm. The process is repeated and by doing so it is hoped, but not guaranteed, that a satisfactory solution will eventually be discovered [2].

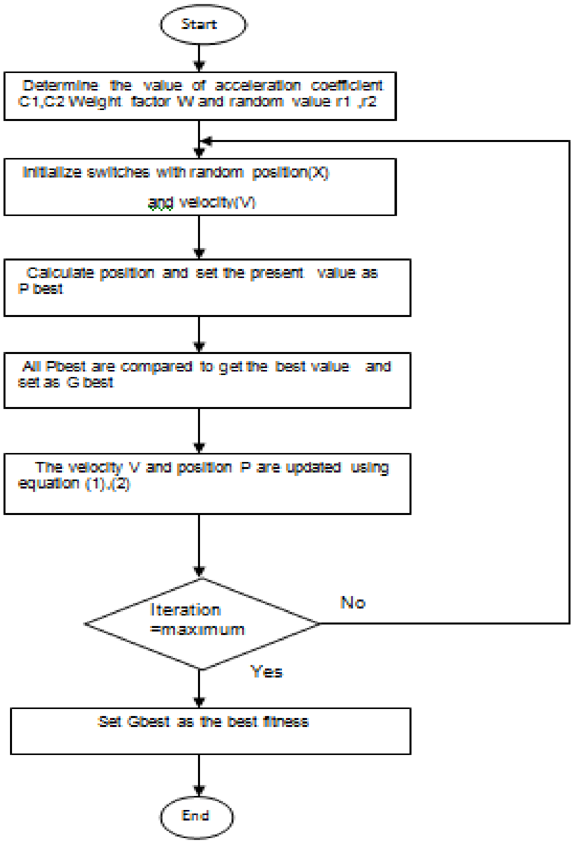


**Fig 2:** Flow Chart of Particle Swarm Optimization

In general the PSO steps can be summaries as follows:

1. Initialization bus data and line data.
2. Calculate power loss
3. Find Pbest and Gbest for all switches.
4. Find new velocity and new position
5. Repeat step (ii)

### 3.2 Gravitational Search Algorithm

GSA was introduced by Rashedi in 2009 and is intended to solve optimization problems. The population-based heuristic algorithm is based on the law of gravity and mass interactions. The algorithm is comprised of collection of searcher agents that interact with each other through the gravity force. The agents are considered as objects and their performance is measured by their masses. The gravity force causes a global movement where all objects move towards other objects with heavier masses. The slow movement of heavier masses guarantees the exploitation step of the algorithm and corresponds to good solutions. The masses are actually obeying the law of gravity as shown in Equation (1) and the law of motion in Equation (2).

$$F = G (M1M2 / R2) \quad (1)$$

$$a= F/M \quad (2)$$

Where

F = magnitude of the gravitational force,
G = gravitational constant,
M1, M2 = mass of the first and second objects and
R = distance between the two objects.

Equation (1) shows that in the Newton law of gravity, the gravitational force between two objects is directly proportional to the product of their masses and inversely proportional to the square of the distance between the objects.

While for Equation (2), Newton’s second law shows that when a force, F, is applied to an object, its acceleration a, depends on the force and its mass, M.

In GSA, the agent has four parameters which are position, inertial mass, active gravitational mass, and passive gravitational mass. The position of the mass represents the solution of the problem, where the gravitational and inertial masses are determined using a fitness function. The algorithm is navigated by adjusting the gravitational and inertia masses, whereas each mass presents a solution. Masses are attracted by the heaviest mass. Hence, the heaviest mass presents an optimum solution in the search space [5].

### 3.3 Ant Colony Optimization (ACO) algorithm

The ant colony optimization algorithm (ACO) is a probabilistic technique for solving computational problems which can be reduced to finding good paths through graphs. Artificial Ants stand for multi-agent methods inspired by the behavior of real ants. The pheromone-based communication of biological ants is often the predominant paradigm used.[2] Combinations of Artificial Ants and local search algorithms have become a method of choice for numerous optimization tasks involving some sort of graph, e.g., vehicle routing and internet routing. The burgeoning activity in this field has led to conferences dedicated solely to Artificial Ants, and to numerous commercial applications by specialized companies such as Ant Optima [6].

As an example, Ant colony optimization [3] is a class of optimization algorithms modeled on the actions of an ant colony. Artificial 'ants' (e.g. simulation agents) locate optimal solutions by moving through a parameter space representing all possible solutions. Real ants lay down pheromones directing each other to resources while exploring their environment. The simulated 'ants' similarly record their positions and the quality of their solutions, so that in later simulation iterations more ants locate better solutions.[4] One variation on this approach is the bees algorithm, which is more analogous to the foraging patterns of the honey bee, another social insect.

This algorithm is a member of the ant colony algorithms family, in swarm intelligence methods, and it constitutes some metaheuristic optimizations. Initially proposed by Marco Dorigo in 1992 in his PhD thesis [5, 6] the first algorithm was aiming to search for an optimal path in a graph, based on the behavior of ants seeking a path between their colony and a source of food. The original idea has since diversified to solve a wider class of numerical problems, and as a result, several problems have emerged, drawing on various aspects of the behavior of ants. From a broader perspective, ACO performs a model-based search [7] and shares some similarities with estimation of distribution algorithms [7].

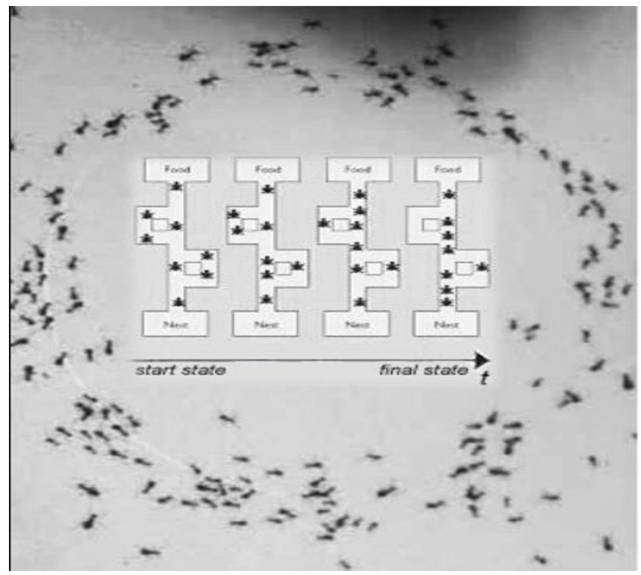


**Fig 3:** Representation of ant colony method

### 3.4 Genetic algorithm

Genetic Algorithm has been used extensively "as a powerful tool to solve various optimization problems such as integer nonlinear problems (INLP)"[3]. "GA is one of the meta-heuristics that has been frequently utilized to find near-optimum solutions of many combinational problems" [4]. In a genetic algorithm, a population of candidate solutions (called individuals, creatures, or phenotypes) to an optimization problem is evolved toward better solutions. Each candidate solution has a set of properties (its chromosomes or genotype) which can be mutated and altered; traditionally, solutions are represented in binary as strings of 0s and 1s, but other encodings are also possible [8].

The evolution usually starts from a population of randomly generated individuals, and is an iterative process, with the population in each iteration called a *generation*. In each generation, the fitness of every individual in the population is evaluated; the fitness is usually the value of the objective function in the optimization problem being solved. The more fit individuals are stochastically selected from the current population, and each individual's genome is modified (recombined and possibly randomly mutated) to form a new generation. The new generation of candidate solutions is then used in the next iteration of the algorithm. Commonly, the algorithm terminates when either a maximum number of generations has been produced, or a satisfactory fitness level has been reached for the population.

A typical genetic algorithm requires:

1. A genetic representation of the solution domain,
2. A fitness function to evaluate the solution domain.

A standard representation of each candidate solution is as an array of bits [5]. Arrays of other types and structures can be used in essentially the same way. The main property that makes these genetic representations convenient is that their parts are easily aligned due to their fixed size, which facilitates simple crossover operations. Variable length representations may also be used, but crossover implementation is more complex in this case. Tree-like representations are explored in genetic programming and graph-form representations are explored in evolutionary programming; a mix of both linear chromosomes and trees is explored in gene expression programming [9].

### 3.5 Differential Evolution (DE)

In evolutionary computation, differential evolution (DE) is a method that optimizes a problem by iteratively trying to improve a candidate solution with regard to a given measure of quality. Such methods are commonly known as met heuristics as they make few or no assumptions about the problem being optimized and can search very large spaces of candidate solutions. However, met heuristics such as DE do not guarantee an optimal solution is ever found.

DE is used for multidimensional real-valued functions but does not use the gradient of the problem being optimized, which means DE does not require the optimization problem to be differentiable, as is required by classic optimization methods such as gradient descent and quasi-newton methods. DE can therefore also be used on optimization problems that are not even continuous, are noisy, change over time, etc.[1]

DE optimizes a problem by maintaining a population of candidate solutions and creating new candidate solutions by combining existing ones according to its simple formulae, and then keeping whichever candidate solution has the best score or fitness on the optimization problem at hand. In this way the optimization problem is treated as a black box that merely provides a measure of quality given a candidate solution and the gradient is therefore not needed [10].

## 4. Conclusion

In distribution systems, network reconfiguration is done to minimize the power losses by using various techniques such as particle swarm optimization (PSO), Gravitational search algorithm (GSA), Genetic algorithm (GA), Differential evolution (DE), Ant colony optimization techniques etc.

All these optimization techniques are used individually for power loss reduction in three phase distribution systems. A comparative study of above techniques reveals that the PSO optimization techniques is better and can be hybrid with other algorithms for further more better results.

## 5. References


1. Mirjalili S, Hashim SZM. A new hybrid PSOGSA algorithm for function optimization. International Conference on Computer and Information Application, Tianjin. 2010; pp.374-377. doi:10.1109/ICCIA.2010.6141614,2010.
2. Reddy AVS, Reddy MD. Optimization of network reconfiguration by using Particle swarm optimization. IEEE 1st International Conference on Power Electronics, Intelligent Control and Energy Systems (ICPEICES), Delhi. 2016; pp.1-6.
3. Zazou AA, Grolleau E, Chevrier E, Richard P, Gaubert JP, Bellatreche L. Power distribution network reconfiguration based on min-cost flow problem. IEEE 14th International Conference on Industrial Informatics (INDIN), Poitiers. 2016; pp. 286-291.
4. Herazo E, Quintero M, Candelo J, Soto J, Guerrero J. Optimal power distribution network reconfiguration using Cuckoo Search. 4th International Conference on Electric Power and Energy Conversion Systems (EPECS), Sharjah. 2015; pp.1-6.
5. Silva, Guimarães FG. Reconfiguration of Power Distribution Networks by Multi-agent Systems. BRICS Congress on Computational Intelligence and 11th Brazilian Congress on Computational Intelligence, Ipojuca. 2013; pp. 546-550.
6. Van Westering W, Van Der Meulen M, Bosma W. Evaluating electricity distribution network reconfiguration to minimize power loss on existing networks. CIRED Workshop, Helsinki. 2016; pp. 1-4.
7. Tuladhar SR, Singh JG, Ongsakul W. Multi-objective approach for distribution network reconfiguration with optimal DG power factor using NSPSO. IET Generation, Transmission & Distribution. 2016; 10(12):2842-2851.
8. Abbasi RS, Mehmood T. Feeder reconfiguration techniques: A review. International Conference on Energy Systems and Policies (ICESP), Islamabad. 2015; pp. 1-5.
9. Hernando-Gil I, Ilie IS, Djokic SZ. Reliability planning of active distribution systems incorporating regulatory requirements and network-reliability equivalents, in IET Generation, Transmission & Distribution. 2016; 10(1):93-106.
10. Seyedi Y. Karimi H. Coordinated Protection and Control Based on Synchrophasor Data Processing in Smart Distribution Networks, in IEEE Transactions on Power Systems. 2017; 99:1-1 doi:10.1109/TPWRS.2708662,2017.