3

Modified Bacterial Foraging Optimization and Application

Neeraj Kanwar
Department of Electrical Engineering
Manipal University Jaipur, Jaipur, India

Nand K. Meena
School of Engineering and Applied Science
Aston University, Birmingham, United Kingdom

Jin Yang
School of Engineering and Applied Science
Aston University, Birmingham, United Kingdom

Sonam Parashar
Department of Electrical Engineering
Malaviya National Institute of Technology, Jaipur, India

CONTENTS

3.1 Introduction

The bacterial foraging optimization (BFO) algorithm is a recently developed nature-inspired meta-heuristic optimization method that mimics the foraging behaviour of Escherichia coli (i.e. *E. coli*) bacteria. It was introduced by Passino [1] in 2002 by observing the foraging behaviour of *E. coli* bacteria. The foraging behaviour should happen in such a way that energy spent in search of food will remain less than energy gained after eating it. Bacteria obtain their nutrients optimally to get maximum intake of energy per unit time. Foraging behaviour varies from one species of animal to an other. Bacteria that have a poor foraging strategy are eliminated from nature or move to some other places. On the other hand, nature always favours the bacteria that have a worthy foraging strategy. A bacterium has 8-10 thin flagella that help them in searching of food. By sensing the nutrient present in environment, flagella can move in a clockwise or anticlockwise direction. If favourable nutrients are sensed, bacteria swim in that direction. And if a noxious environment is sensed, they move away.

The food searching behaviour found in bacteria are modelled in BFO in four steps, i.e. chemotaxis, swarming, reproduction and elimination–dispersal. In chemotaxis, bacteria move in a nutrient-rich site and avoid a harmful site. This is accomplished by two types of actions, i.e., swimming and tumbling. In swimming, bacteria will trail in the same direction as in the previous stage and in tumbling they will travel in a different direction than the former one. Swarming modelled bacteria move towards the favourable or non-favourable environment. This is a group behaviour where other bacteria can be attracted toward or repelled from a bacteria by cell-to-cell signalling. Reproduction shows the depletion of less healthy bacteria from the environment. If temperature, water flow and nutrient concentration is changed, it affects the bacteria population. The elimination-dispersal step models the elimination and dispersal of bacteria due to sudden changes in environment. It introduces some diversity in population.

This chapter is organized as follows. In Section 3.2, the standard BFO algorithm is discussed along with its algorithm. All steps of the BFO algorithm are presented. In Section 3.3, some modifications suggested in literature for improving BFO performance are presented as non-uniform elimination-dispersal probability distribution, adaptive chemotaxis step, varying population. In Section 3.4, the application of the BFO algorithm for a simultaneous DGs and SCs allocation problem of distribution system is demonstrated.

3.2 Original BFO algorithm in brief

The functioning of the BFO algorithm is based on cumulative efforts of all bacteria in a group searching for food. Each bacteria should be responsive to inducements in its foraging environment. It should be capable of exchanging information with other bacteria in the group. Each bacteria searches their food by chemotaxis procedure and this mimicking of chemotactic movement forms the basic working of this algorithm. This chemotaxis movement is a complex mixture of swimming and tumbling that retains bacteria in places where the chance of finding nutrients in large concentrations is high. It shows the foraging capabilities of bacteria in a group. Like other nature-inspired optimization techniques, this technique also follows the philosophy of "Survival of the fittest". Foraging animals change their foraging behaviour. These changes basically simulate the action and response to reduce energy consumption per unit time by following all types of constraints. Bacteria that have poor foraging behaviour are eliminated in the successive generation or it may improve to a better one. The four basic steps of the BFO algorithm namely chemotaxis, swarming, reproduction, and elimination-dispersal, are discussed in following sections.

3.2.1 Chemotaxis

The strategies by which any bacteria will move to search of food are called chemotaxis. The movement of bacteria is modelled based on the movement of flagella that may be of two types i.e. tumbling and swimming. Random movement of bacteria is called tumbling whereas directed progression is called swimming. In tumbling, flagella turn clockwise; this shows independent movement and leads to erratic displacement. In swimming, flagella turn counterclockwise and show rotation in the same direction, thus pushing the bacterium steadily in a forward direction characterized as a 'run'. A decision system of bacteria enables it to avoid noxious substances in searching of food. Bacteria always search for an area having a greater concentration of nutrients. By following these two steps bacteria can achieve a high nutrient area and show an improvement in fitness value. If x shows the position of a bacteria, $J(x)$ is the objective function to be minimized. The position after the j-th chemotactic step is updated as

$$x_i(j+1,k,l) = x_i(j,k,l) + C(i)\frac{\Delta(i)}{\Delta^T(i)\Delta(i)} \tag{3.1}$$

where, $x_i(j,k,l)$ is the i-th bacteria at the j-th chemotactic step, k-th reproductive and l-th elimination-dispersal step. $C(i)$ is a scalar called run length unit that specifies the step size for tumble in the random directions. Δ shows a random directed unit length vector between -1 and 1.

For a maximization function, $J(x) < 0$ shows a nutrient rich area, $J(x) = 0$ shows a neutral area and $J(x) > 0$ shows a negative environment with noxious

substances. Fitness value is calculated at this new position of bacteria given by (3.1). If it is less than the previous fitness of that bacteria then the bacteria will move in that particular direction until the number of chemotactic steps are exhausted.

3.2.2 Swarming

It is always desired that if a bacteria finds a nutrient-rich area, it should send a signal to other bacteria to concentrate them in this nutrient-rich area. In BFO, this process is modelled by using swarming. Bacteria send attracting or repelling signals to other bacteria depending upon environment. The cell-to-cell attraction and repelling is represented as

$$J_{CC}(x, x^i(j,k,l)) = \sum_{i=1}^{S} \Big[- d_{attractant} \cdot \exp\Big(- \omega_{attractant} \sum_{m=1}^{d} (x_m - x_m^i)^2 \Big) \Big] + \sum_{i=1}^{S} \Big[h_{repellent} \cdot \exp\Big(- \omega_{repellent} \sum_{m=1}^{d} (x_m - x_m^i)^2 \Big) \Big] \tag{3.2}$$

where $d_{attractant}$, $w_{attractant}$, $h_{repellent}$ and $\omega_{repellent}$ are different coefficients that represent the attraction and repulsion strength of the signal. d defines the problem dimension, S represent the total swarm size, x_m is a point in d-dimensional search space, x_m^i is the position of the ith bacteria. The obtained fitness value after swarming is added in the existing fitness value, expressed as

$$J(i,j,k,l) = J(i,j,k,l) + J_{CC}(x, x^i(j,k,l)) \tag{3.3}$$

3.2.3 Reproduction

After the chemotaxis stage, it is assumed that bacteria present in a swarm having sufficient nutrients. The fitness value obtained for each bacteria represents its health. If a bacteria has good fitness value this means it got sufficient nutrients in its life cycle. This healthy bacteria can reproduce itself under a favourable environment and temperature. They increase their length and split into two parts. These parts are the replica of each other. On the other hand, if it has poor fitness value, it shows that the bacteria did not get sufficient nutrients in its life cycle and is not healthy to reproduce itself; therefore, it cannot survive further. To simulate this step, the bacterial population is sorted according to their fitness values. Half of population with poor fitness values will be liquidated while the other better bacteria split into two parts and start exploring the search space from the same location. This phenomenon is similar to the elitist-selection mechanism of the other classical evolutionary algorithms.The newly generated bacteria will replace the dead bacteria of the swarm, so swarm size remains constant after reproduction.

3.2.4 Elimination and dispersal

The environment where bacteria live may change suddenly or gradually. Due to this noxious environment some bacteria may die or some may disperse to some other locations. In general, dispersion probability is very low. Dispersed bacteria follow different paths, so this process generates a swarm having a new set of bacteria.

3.2.5 Pseudo-codes of the original BFO algorithm

The pseudo-code of modified BFO (MBFO) is presented in Algorithm 4. The BFO algorithm parameters are: d=Total number of parameters to be optimized or dimension of the search space; S=Total number of bacteria in the population; Nc=total number of chemotactic steps; Ns=Maximum swimming length before tumbling; Nre=total number of reproduction steps; Ned=total number of elimination-dispersal events; Ped=probability for elimination-dispersal; $C(i)$=Step size in random direction that is specified by the tumbling process.

Algorithm 4 Pseudo-code of MBFO.

define the objective function $OF(.)$ and initialize BFO parameters
set initial value of j,k and l to zero, calculate $OF(.)$ for each bacteria x_i
Start elimination-dispersal loop
for each l-th elimination-dispersal **do**
    Start reproduction loop
    for each k-th reproduction **do**
        Start chemotactic loop
        for each j-th chemotactic step **do**
            Start bacterial loop
            for each i-th bacteria **do**
                calculate objective function, $J(x^i(j,k,l))$
                $Jlast = J(x^i(j,k,l))$
                generate a tumble angle for bacteria i, as $\frac{\Delta(i)}{\Delta^T(i)\Delta(i)}$
                evaluate new position of bacteria i, using (3.1)
                for new position, recalculate $J(x^i(j+1,k,l))$
                Start counter for swimming process
                $m = 0$
                while $m \leq Ns$ **do**
                    if $J(x^i(j+1,k,l)) < Jlast$ **then**
                        $Jlast = J(x^i(j+1,k,l))$;
                        calculate new position as
                        $x_i(j+1,k,l) = x_i(j,k,l) + C(i)\frac{\Delta(i)}{\Delta^T(i)\Delta(i)}$
                        for new position, recalculate $J(x^i(j+1,k,l))$
                    else
                        $m = Ns$

```
                    end if
                    m = m + 1
                end while
            end for
            Reproduction
            sort the bacteria according to fitness values, J(x^i)
            split the top S_r = S/2 fittest bacteria and replace poor ones
        end for
        Elimination-dispersal
        for each i-th bacteria do
            rr = rand
            if rr < P_ed then
                disperse the bacterium at random location
            end if
        end for
    end for
end for
return the best result
```

3.3 Modifications in bacterial foraging optimization

3.3.1 Non-uniform elimination-dispersal probability distribution

In elimination-dispersal step of standard BFO (see steps 35–40 in Algorithm 4), the probability of elimination for each bacteria is taken constant. It is completely independent from any ranking of bacteria in population. This constant probability may affect the convergence speed of the algorithm as in the iterative process the bacteria that are near to the optimal solution may be replaced by some other non-optimal solution. In other words, both types of fitness, best and worst, will have the same probability of elimination. This will ultimately affect the performance of the algorithm.

An improved version of BFO is proposed in [2] by suggesting a non-uniform probability distribution for the elimination and dispersal phase. It improves convergence speed due to implementation of a linear and nonlinear probability distribution to replace the constant distribution of standard BFO.

The linear elimination-dispersal probability is defined as

$$p_e(i) = \frac{(i-1)}{\left(\frac{S}{2}-1\right)} \qquad \forall i = 1, 2, \ldots, \frac{S}{2} \tag{3.4}$$

The non-uniform probability distribution is defined as

$$p_e(i) = \left[\frac{(i-1)}{\left(\frac{S}{2}-1\right)}\right]^2 \qquad \forall i = 1, 2, \ldots, \frac{S}{2} \tag{3.5}$$

Equations (3.4) and (3.5) ensure that best bacteria will be assigned a lower probability than the worst one. However, in these equations, the probability is non-linearly proportional to the bacterium index.

A large value of elimination probability may avoid local trapping by relocating the bacteria in a better position and give exploration of the search space. On the contrary, a small value of elimination probability strengthens the search in local area and may stuck in local optima. Therefore, elimination probability should be selected in a proper fashion. During initial iterations, the whole search space should be explored by the algorithm and during anaphase the exploitation rate should be high. It can be achieved by assigning a high value of p_e during exploration and a low value during exploitation. Therefore, (3.4) and (3.5) are proposed in a more generalised way than (3.6) and (3.7) for linear and non-linear distributions [2].

Linear:

$$p_e(i,l) = P_e + \left[i - \left(\frac{S}{4} + 0.5 \right) \right] \frac{(l-1)}{(N_{ed}-1)\left(\frac{S}{2}-1\right)} \qquad \forall i, l \tag{3.6}$$

Non-linear:

$$p_e(i,l) = P_e \left(\frac{l-1}{N_{ed}-1} \right) \left[2 \times \left(\frac{i-1}{\frac{S}{2}-1} \right)^2 - 1 \right] + P_e \qquad \forall i, l \tag{3.7}$$

where, i and l are population and elimination index respectively. P_e shows constant probability.

3.3.2 Adaptive chemotaxis step

Run-length unit parameter $C(i)$ plays a crucial role in deciding the performance of BFO. A large value can increase exploration of the search space. It benefited the BFO to escape local optima very easily but sometimes global optima may be missed due to this large step size. On the contrary, a bacteria with small run-length parameter can only search the region which is near the starting point. So there is a fair chance that it may be stuck in the local region. Therefore, run-length unit parameter can be made adaptive with the chemotaxis step. In [3], an adaptive BFO algorithm is proposed to maintain a proper balance between exploration and exploitation of the search space using the self-adaptive chemotaxis step. Two factors are considered that guide the algorithm for switching in exploration or exploitation. These factors depend on the improvement in fitness value. When there is an improvement in fitness value beyond a certain limit from the last to the current chemotactic generation, it means that bacteria have searched a new promising region and now there is a requirement to exploit this area further to obtain a better solution. Therefore run-length unit parameter should be small compared to the previous generation. So bacteria should be self-adaptive with the exploitation state. If there is no improvement in fitness value from the previous generation to the current generation continuously for a predefined time, it shows that it is not

fruitful to search that area any more; it requires a large change in run-length unit so that bacteria can jump to a new search area. It helps to escape local trapping. So bacteria should be self-adaptive with the exploration state under such situation.

3.3.3 Varying population

The iterative process of all evolutionary algorithms (EAs) in general starts with a fixed population size. So search space is not fully explored and it also increases the computational time. An improved version of BFO i.e. bacterial foraging algorithm with varying population (BFAVP) is proposed in [4]. Bacterial energy and bacterial age indexes have been introduced to measure the search ability and life of each bacteria. It simulates three behaviours of each bacteria, i.e. metabolism, proliferation and elimination. The better position of bacteria than the previous step is assured by incorporating these. Diversity is controlled by combining chemotaxis behavior with the phenomena of quorum sensing. It also speeds up the convergence and avoids local trapping. In [5], BFAVP is used to solve an optimal power flow problem.

3.4 Application of BFO for optimal DER allocation in distribution systems

3.4.1 Problem description

Reactive power compensation in distribution systems using SCs is typically an old classical problem of power systems. The capacitors must be allocating optimally; otherwise line losses may increase and develop over-voltages during light load hours. The optimal capacitor placement problem involves the determination of their optimal number, location and capacity. In an electric distribution system with specified structure, the value of the active component of losses cannot be reduced by the use of the SCs alone. Therefore, to supply active loads of the distribution system, local generation should be privileged by DGs. DGs refer to small generating units typically connected to the utility grid in parallel near load centres. The DG allocation problem also involves the determination of the optimal number, location and capacity of DG units to achieve certain objectives under specified constraints. The optimal generation of active and reactive power from these devices reduces power import from the substation and thus regulates feeder power flows. Optimal capacitor placement achieves this goal by regulating reactive power flow, whereas optimal DG placement does the same by regulating active power flow in the system. These key technologies may be coordinated together to get better solutions so that distribution systems can achieve optimum performance. The simultaneous allocation problem of DGs and SCs in distribution systems is modelled to reduce annual energy losses and to maintain better node voltage profiles while considering a piece-wise multi-level annual load profile.

3.4.2 Individual bacteria structure for this problem

For a simultaneous DG and SC placement problem, each bacteria is structured as shown in Fig. 3.1 which is composed of candidate location and capacity for the respective candidate DGs and SCs. Here, l^{DG} and l^{SC} represents location, and, P^{DG} and Q^{SC} represents capacity of these components, respectively. The candidate nodes are allocated randomly between defined limits. Repetition of nodes is avoided using non-repeatability constraints as given in the next section. The capacity of the candidate DGs and SCs is selected randomly within their respective predefined bounds. However, the number of locations for DGs and SCs is set more than what should be required and the lower bound of the capacity for DGs and SCs is set equal to zero.

$$\underbrace{l_1^{DG}, l_2^{DG}, \ldots, l_{T_{DG}}^{DG}}_{\text{DG Locations}} \underbrace{P_1^{DG}, P_2^{DG}, \ldots, P_{T_{DG}}^{DG}}_{\text{DG Capacities}} \underbrace{l_1^{SC}, l_2^{SC}, \ldots, l_{T_{SC}}^{SC}}_{\text{SC Locations}} \underbrace{Q_1^{SC}, Q_2^{SC}, \ldots, Q_{T_{SC}}^{SC}}_{\text{SC Capacities}}$$

FIGURE 3.1
Individual structure of each bacteria.

3.4.3 How can the BFO algorithm be used for this problem?

We want to generate an optimization method to give optimal allocation of DGs and SCs in a distribution system. The optimal allocation is represented by optimal number, location and capacity of these components. Here, the BFO algorithm consists of mainly five steps to find out the optimal solution for any optimization problem. Initially we define the different control parameters of the BFO algorithm, i.e. dimension of the search space (d), total number of bacteria in the population (S), total number of chemotactic steps (N_c), maximum swimming length before tumbling (N_s), total number of reproduction steps (N_{re}), total number of elimination–dispersal events (N_{ed}), probability for elimination-dispersal (Ped), step size for tumbling process ($C(i)$), maximum number of iterations, maximum and minimum limit of DGs and SCs at each candidate node, total compensation limit at all candidate nodes, minimum and maximum voltage limit at each node.

After defining all problem and algorithm specific control parameters of BFO, the first step is the initialization of the initial swarm. It consists of S bacteria (each bacteria is a d-dimensional vector). The dimensions depend upon design variables considered for optimal allocation of DGs and SCs in the distribution system. Simultaneous optimal allocation of DGs and SCs is a complex, combinatorial problem as so many combinations of siting and sizing of DGs and SCs are possible. The initial swarm of bacteria is randomly generated. Each bacteria in this swarm has a defined position in the swarm i.e. x^i and structured as in Fig. 3.1. Each individual bacteria represents an allocation of DGs and SCs simultaneously and corresponds to some fitness

value of objective function. Here objective function is formulated for annual energy loss reduction. So each bacteria will represent annual energy loss for that allocation. The fitness value of each bacteria in the swarm is calculated and saved.

The second step is chemotaxis operation of bacteria. Bacteria can move in one direction or multiple directions in searching fore food. These movements basically depend upon environmental condition. A bacteria starts tumbling to change its present location with small steps. If they find sufficient nutrients at this new position, they continuously swim in the same direction using (3.1), until the maximum swimming length (N_s) is exhausted. This process is done for all bacteria in the swarm until the maximum chemotactic steps (N_c) are reached. After the chemotaxis step, bacteria will have a different combination of siting and sizing of DGs and SCs. Fitness value is calculated by placing the given capacity at a given location for each bacteria.

The third step of BFO is swarming. It basically simulates the singling process of bacteria from one cell to other. Bacteria inform other bacteria in the swarm about favorable (nutrient-rich) or non-favorable (noxious) environment. For that they release some chemicals for attracting or repelling the other. This step is completed using (3.2). This equation gives a cell-to-cell communication value (J_{cc}) for each bacteria. This value is added in the obtained fitness value of each bacteria after the chemotaxis step using (3.3).

The fourth step of BFO is reproduction. This step shows that unhealthy bacteria cannot survive in nature while a healthy bacteria can reproduce itself in a suitable environment. To simulate it all bacteria present in swarm are sorted according to their fitness value. Generally, half the bacteria of high fitness values (means unhealthy) are removed and the other half of the bacteria are split into two parts that are replicas of each other. They replace the position of unhealthy bacteria in the swarm. By following this step all individuals having poor allocation for DGs and SCs are removed from the swarm.

The fifth step is elimination and dispersal of bacteria. It generates a new set of swarm having inclusion of some new bacteria and deletion of some bacteria. After this last step, the best/healthiest bacteria having minimum fitness value (J_{cc}) and its position are saved. For the simultaneous optimal allocation of DGs and SCs, this bacteria will generate optimal siting and sizing of these components. When this solution is implemented on the network, it will give maximum energy loss reduction ($1/J_{cc}$). This solution shows optimal allocation of these components after the first iteration. The iterative process continues until maximum iteration counts are exhausted.

3.4.4 Description of experiments

The integration of DGs and SCs alters power flow in distribution feeders. This causes reduction in annual energy losses and node voltage deviations. Therefore, the simultaneous DGs and SCs allocation problem is formulated to maximize the annual energy loss reduction while maintaining better node voltage profiles. In order to limit the voltage deviation at different nodes, a hard

voltage constraint is used as desired. Similarly, to check the current carrying capacities of distribution feeders, a feeder ampacity constraint is essential. A multi-level piece-wise linearized annual load duration profile of the system is considered to evaluate annual energy losses of distribution feeders. The DR allocation problem is structured as a single-objective constrained optimization problem where optimal number, size and location of DGs and SCs are determined simultaneously. The problem objective is formulated as

$$\max F = \sum_{ll=1}^{N_{Total}} \left(ELOSS_{bj} - ELOSS_{aj}\right); \qquad \forall\, ll \epsilon L \tag{3.8}$$

OR

$$\min J_{cc} = \frac{1}{(1+F)} \tag{3.9}$$

This objective function is solved using the BFO technique, subject to the power flow and node voltage limit constraints. Node voltages $V_{n,ll}$ of all system buses must be kept within the minimum and maximum permissible limits, i.e., V_{min} and V_{max}, respectively, during the optimization process as defined by (3.10). The current flow in each branch must satisfy the rated ampacity of each branch as defined by (3.11). The active power injected by DG at each bus must be within their permissible range as given by (3.12), where, $P_{min}^{DG}$and P_{max}^{DG} are the minimum and maximum active power generation limit at a bus, respectively. The reactive power injected by SC at each bus must be within their permissible range as defined by (3.13), where, $Q_{min}^{SC}$and Q_{max}^{SC} are the minimum and maximum reactive power generation limit at a bus, respectively. The sum of active power injected by DGs at all candidate nodes should be less than nominal active power demand (P_D) of the distribution system as given in (3.14). Similarly, the sum of reactive power injected by SCs at all candidate nodes should be less than nominal reactive power demand (Q_D) of the distribution system as given in (3.15). Equations (3.16) and (3.17) prohibit the repetition of candidate sites for DGs and SCs, respectively.

$$V_{min} \leq V_{n,ll} \leq V_{max} \qquad \forall\, n, ll \tag{3.10}$$

$$I_{n,ll} \leq I_n^{max} \qquad \forall\, n, ll \tag{3.11}$$

$$P_{min}^{DG} \leq P_{n,ll}^{DG} \leq P_{max}^{DG} \qquad \forall\, n \tag{3.12}$$

$$Q_{min}^{SC} \leq Q_{n,ll}^{SC} \leq Q_{max}^{SC} \qquad \forall\, n \tag{3.13}$$

$$\sum_{n=1}^{T_{DG}} P_n^{DG} \leq P_D \qquad \forall\, n \tag{3.14}$$

$$\sum_{n=1}^{T_{SC}} Q_n^{SC} \leq Q_D \qquad \forall\, n \tag{3.15}$$

$$l_a^{DG} \neq l_b^{DG} \qquad \forall\, a, b \epsilon N \tag{3.16}$$

$$l_a^{SC} \neq l_b^{SC} \qquad \forall\, a, b \epsilon N \tag{3.17}$$

3.4.5 Results obtained

The simultaneous DGs and SCs allocation problem is solved using standard BFO algorithm for benchmark 33-bus [6] test distribution systems. The population size and maximum iterations are set at 10 and 200, respectively. The best result obtained after 100 independent trials of MBFO is presented in Table 3.1. The table shows the optimal DG and SC capacity as well as their obtained optimal locations. The annual load profile is segmented into three different load levels, i.e., light, nominal and peak to show 50%, 100% and 160% of the nominal system loading, respectively. The corresponding load durations are taken as 2000, 5260 and 1500 hours, respectively.

TABLE 3.1
Optimal solution of DGs and SCs using MBFO.

Parameter	DGs	SCs
Location (nodes)	14, 25, 32	14, 24, 30
Individual node capacity (kW/kVAr)	898, 944, 934	600, 300, 900
Total capacity (kW/kVAr)	2776	1800

TABLE 3.2
Comparison of power and energy loss with and without DRs.

Particular	Power loss (kW)			Annual energy loss (kWh)	% loss reduction
	Light	Nominal	Peak		
Without compensation	47.07	202.50	575.39	2022398.44	–
With compensation	3.53	15.12	93.40	226700.42	88.79

The improvement in network performance is evaluated with and without compensation. The comparison results are presented in Table 3.2. The table shows the power loss obtained at each load level. The table also shows that an annual energy loss reduction of about 89% is achieved after allocating the optimal solution of Table 3.1 in the distribution network.

3.5 Conclusions

In this chapter, the BFO algorithm was presented in detail. Some modifications of the BFO algorithm were demonstrated and discussed. Finally, the

application of the BFO algorithm to a complex combinatorial problem was shown. The BFO algorithm was used for a simultaneous optimal allocation problem of DGs and SCs in a distribution system. The results of the BFO algorithm show that there is a significant improvement in the desired objectives.

Acknowledgement

This work was supported by the Engineering and Physical Sciences Research Council (EPSRC) of United Kingdom (Reference Nos.: EP/R001456/1 and EP/S001778/1).

References

1. K.M. Passino, "Biomimicry of bacterial foraging for distributed optimization and control", IEEE Control Systems Magazine (2002) pp. 52-67.
2. Mouayad A. Sahib, Ahmed R. Abdulnabi, and Marwan A. Mohammed, "Improving bacterial foraging algorithm using non-uniform elimination-dispersal probability distribution". Alexandria Engineering Journal (2018) 57, pp. 3341-3349.
3. Nandita Sanyal, Amitava Chatterjee, Sugata Munshi. "An adaptive bacterial foraging algorithm for fuzzy entropy based image segmentation". Expert Systems with Applications 38 (2011) pp. 15489-15498.
4. M.S. Li, T.Y. Ji, W.J. Tang, Q.H. Wu, J.R. Saunders. "Bacterial foraging algorithm with varying population". BioSystems 100 (2010) pp. 185-197.
5. M. S. Li, W. J. Tang, W. H. Tang, Q. H. Wu, and J. R. Saunders. "Bacteria foraging algorithm with varying population for optimal power flow" in Proc. Evol. Workshops 2007, LNCS vol. 4448. pp. 32-41.
6. M. E. Baran and F. F. Wu. "Network reconfiguration in distribution systems for loss reduction and load balancing". IEEE Transactions on Power Delivery, vol. 4(2), 1989, pp. 1401-1407.