

# Binary-Real Coded Genetic Algorithm Based $k$ -Means Clustering for Unit Commitment Problem

Mai A. Farag<sup>1</sup>, M. A. El-Shorbagy<sup>1</sup>, I. M. El-Desoky<sup>1</sup>, A. A. El-Sawy<sup>1,2</sup>,  
A. A. Mousa<sup>1,3</sup>

<sup>1</sup>Department of Basic Engineering Science, Faculty of Engineering, Menoufiya University, Al Minufya, Egypt

<sup>2</sup>Department of Mathematics, Faculty of Science, Qassim University, Qassim, Saudi Arabia

<sup>3</sup>Department of Mathematics, Faculty of Sciences, Taif University, Taif, Saudi Arabia

Email: [mai.farag2015@gmail.com](mailto:mai.farag2015@gmail.com), [mohammed\\_shorbagy@yahoo.com](mailto:mohammed_shorbagy@yahoo.com), [eldesokyi@yahoo.com](mailto:eldesokyi@yahoo.com),  
[a.elsawy1954@yahoo.com](mailto:a.elsawy1954@yahoo.com), [a\\_mousa15@yahoo.com](mailto:a_mousa15@yahoo.com)

Received 3 September 2015; accepted 19 October 2015; published 22 October 2015

Copyright © 2015 by authors and Scientific Research Publishing Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

---

## Abstract

This paper presents a new algorithm for solving unit commitment (UC) problems using a binary-real coded genetic algorithm based on  $k$ -means clustering technique. UC is a NP-hard nonlinear mixed-integer optimization problem, encountered as one of the toughest problems in power systems, in which some power generating units are to be scheduled in such a way that the forecasted demand is met at minimum production cost over a time horizon. In the proposed algorithm, the algorithm integrates the main features of a binary-real coded genetic algorithm (GA) and  $k$ -means clustering technique. The binary coded GA is used to obtain a feasible commitment schedule for each generating unit; while the power amounts generated by committed units are determined by using real coded GA for the feasible commitment obtained in each interval.  $k$ -means clustering algorithm divides population into a specific number of subpopulations with dynamic size. In this way, using  $k$ -means clustering algorithm allows the use of different GA operators with the whole population and avoids the local problem minima. The effectiveness of the proposed technique is validated on a test power system available in the literature. The proposed algorithm performance is found quite satisfactory in comparison with the previously reported results.

## Keywords

Unit Commitment (UC), Genetic Algorithm (GA),  $k$ -Means Clustering Technique

---

## 1. Introduction

The unit commitment (UC) problem, one of the most important tasks of operational planning of power systems, which has a significant influence on secure and economic operation of power systems [1]. An efficient commitment scheduling save millions of dollars per year in fuel and related costs [2], increases the system reliability, and maximizes the energy capability of reservoirs [3]. The UC problems involve determining on/off status as well as the real power outputs of the generating units to meet forecasted demand and reserve requirements at minimal operating cost over the planning period subject to various generator- and system-based constraints [1].

Research efforts, therefore, have concentrated on efficient and near-optimal UC algorithms which can be applied to realistic power systems and have reasonable storage and computation time requirements. Such alternative algorithms studied for the UC problem can be divided into two classes [4]: deterministic methods and meta-heuristic methods. The investigated deterministic methods include Priority List (PL) [5], Dynamic Programming (DP) [6], branch-and-bound method [7], Lagrangean Relaxation (LR) [8] and Mixed Integer Linear Programming (MILP) [9]. These methods suffer from the quality of final solution are not guaranteed, the “curse of dimensionality” if the size of a system is large, applied to small UC problems, required major assumptions that limit the solution space because it is difficult achieve a balance between the efficiency and the accuracy of the model, and may not provide feasible solutions to the relaxed problem due to the inherent non-convexity of the UC problem [10].

Various meta-heuristics are investigated, such as Artificial Neural Networks (ANN) [11], Genetic algorithm (GA) [12]-[16], Evolutionary Programming (EP) [17], Simulated Annealing (SA) [18] [19], Shuffled Frog Leaping algorithm [20], Particle Swarm Optimization (PSO) [21], Tabu Search (TS) [19] [22], Fuzzy Logic [23], harmony search algorithm (HSA) [24] and artificial bee colony algorithm (ABC) [25]. The practical advantage of meta-heuristic methods over deterministic methods lies in both their effectiveness and general applicability.

Furthermore, some hybrid methods combining meta-heuristics with deterministic methods or other meta-heuristics are also investigated in order to utilize the feature of one method to overcome the drawback of another method [26]-[33].

Genetic Algorithms are being used as powerful tools in optimization problems, especially in the non-convex problems [34]. Among the most important characteristics of these algorithms are their compatibility with nonlinear and/or discrete problems and parallel search in complicated spaces. A disadvantage of GA, though, is that they easily become trapped in local minima. This deficiency, resulting from GA’s weakness in local search, can be remedied by using clustering algorithm along with GA.

Clustering is a process of division of data into groups of similar objects. Each group, called cluster, consists of objects that are similar between themselves and dissimilar to objects of other groups [35]. Clustering techniques have been used in a wide range of disciplines such as psychiatry [36], market research [37] archaeology [38], pattern recognition, [39], medicine, [40] and engineering [41]. There are many clustering algorithms [42]. The *k*-means is possibly the most commonly-used clustering algorithm because of its simplicity and accuracy [43].

In this paper, we propose a new approach for solving UC problems using a binary-real coded genetic algorithm based on *k*-means clustering technique to integrate the main features of the both algorithms; where a binary-real coded GA, in which the binary part deals with the scheduling of units and the real part determines power output levels of committed generating units. *k*-means clustering technique is used in order to avoid the local minima problem; where the population can be divided into a specific number of subpopulations. Within each cluster, subpopulation has common features. After grouping all individuals into pre-defined number of clusters, instead of taking care of all individuals, population can regard the huge amount of individuals as just the number of groups that has been divided. In this way, different GA operators can apply to subpopulations instead of one GA operator applied to all population.

## 2. Problem Formulation

The UC problem involves determining the startup and shut down times as well as the power output levels of all generating unit at each time step during scheduling period  $T$  [1]. The formulation of the problem is described below.

### 2.1. Objective Functions

The objective function of UC is to minimize the total operating cost of the generating units, which is the sum of

fuel cost, startup cost and shut down cost during the scheduling time horizon while several constraints are satisfied.

The fuel costs  $C_{it}(p_{it})$  of thermal units are usually represented by a quadratic heat rate curve as a function of power output multiplied by the price of the selected fuel, which is frequently expressed as follows:

$$C_{it}(p_{it}) = a_i + b_i p_{it} + c_i p_{it}^2; \quad (1)$$

where  $a_i$ ,  $b_i$ ,  $c_i$  are the fuel cost coefficients of unit  $i$ .

The startup ( $ST_{it}$ ) and shut down ( $SD_{it}$ ) costs of a unit is a mixture variable of and fixed down time dependent costs. The costs will depend on for instance unit cooling constant, number of boilers and other plant components involved in the unit startup or shut down process.

The startup cost of the generator ( $ST_{it}$ ) depends on the duration time of shut down before starting up. The start-up cost function is given by two-step function as:

$$ST_{it} = \begin{cases} d_i, & \text{if } \Gamma_i^{down} \leq \tau_{it}^{off} \leq \Gamma_i^{down} + f_i \\ e_i, & \text{if } \tau_{it}^{off} > \Gamma_i^{down} + f_i \end{cases} \quad (2)$$

On the other hand, the shut-down cost ( $SD_{it}$ ) is constant and the typical value is zero in standard systems [44].

Finally, the overall objective function ( $F$ ) of the UC problem of  $N$  generating units for a scheduling time horizon  $T$  is:

$$F = \sum_{t=1}^T \sum_{i=1}^N [C_{it}(p_{it}) + ST_{it}(1 - u_{i,t-1})] u_{it} + (1 - u_{it}) SD_{it} u_{i,t-1} \quad (3)$$

## 2.2. Constraints

The optimization of the objective function is subject to a number of system and unit constraints as follows.

- System power balance:

The sum of the unit generation output at each hour must satisfy the system load demand requirement of the corresponding hour as follows:

$$\sum_{i=1}^N u_{it} p_{it} = D_t; \quad t = 1, 2, \dots, T \quad (4)$$

- Spinning reserve constraint:

Spinning reserve requirements are necessary in the operation of a power system to deal with real-time potential sudden load increases due to unexpected demand increase or failure of any of the working units. The reserve is considered to be a prespecified amount or a given percentage of the forecasted demand.

$$\sum_{i=1}^N u_{it} p_i^{\max} \geq D_t + R_t; \quad t = 1, 2, \dots, T \quad (5)$$

- Unit Maximum/Minimum MW Limit:

The power produced by each unit must be within certain minimum and maximum limits of capacity limits, *i.e.*

$$p_i^{\min} \leq p_{it} \leq p_i^{\max} \quad (6)$$

- Unit Minimum Up and Down Times:

The unit cannot be turned on or off instantaneously once it is committed or uncommitted. The minimum up-time/downtime constraints indicate that there will be a minimum time before it is shut-down or started up, respectively.

$$\begin{aligned} (\tau_{i,t-1}^{on} - \Gamma_i^{up})(u_{i,t-1} - u_{it}) &\geq 0 \\ (\tau_{i,t-1}^{off} - \Gamma_i^{down})(u_{it} - u_{i,t-1}) &\geq 0 \end{aligned} \quad (7)$$

## 2.3. Mathematical Formulation

The UC optimization problem is formulated in general form as follows [15]:

$$\text{Min } F = \sum_{t=1}^T \sum_{i=1}^N [C_{it}(p_{it}) + ST_{it}(1 - u_{i,t-1})] u_{it} + (1 - u_{it}) SD_{it} u_{i,t-1};$$

where

$$C_{it}(p_{it}) = a_i + b_i p_{it} + c_i p_{it}^2$$

$$ST_{it} = \begin{cases} d_i, & \text{if } \Gamma_i^{\text{down}} \leq \tau_{it}^{\text{off}} \leq \Gamma_i^{\text{down}} + f_i \\ e_i, & \text{if } \tau_{it}^{\text{off}} > \Gamma_i^{\text{down}} + f_i \end{cases}$$

Subject to:

$$\sum_{i=1}^N u_{it} p_{it} = D_t; \quad t = 1, 2, \dots, T \quad (8)$$

$$\sum_{i=1}^N u_{it} p_i^{\text{max}} \geq D_t + R_t; \quad t = 1, 2, \dots, T$$

$$p_i^{\text{min}} \leq p_{it} \leq p_i^{\text{max}}$$

$$(\tau_{i,t-1}^{\text{on}} - \Gamma_i^{\text{up}})(u_{i,t-1} - u_{it}) \geq 0$$

$$(\tau_{i,t-1}^{\text{off}} - \Gamma_i^{\text{down}})(u_{it} - u_{i,t-1}) \geq 0$$

### 3. The Genetic Algorithm Approach

Genetic algorithms are general-purpose search techniques based on principles inspired from the genetic and evolution mechanisms observed in natural systems and populations of living beings. Their basic principle is the maintenance of a population of solutions to a problem (genotypes) as encoded information individuals that evolve in time [45] [46]. By simulating natural evolution, a genetic algorithm can effectively search the problem domain and easily solve complex problems. GA usually starts with a randomly generated initial population consisting of  $N_{POP}$  members called chromosomes. Chromosomes are binary or continuous encoded strings, representing potential solutions to the optimization problem. Each member becomes evaluated on the fitness function (objective function), giving a measure of the solution quality called the fitness value. Upon candidate solution selection, recombination (crossover and mutation) is being performed, ending in a new candidate solution population.

### 4. Clustering Algorithm

Clustering is process of Finding groups of objects such that the objects in a group will be similar (or related) to each other and different from (or unrelated to) the objects in other groups [35]. Several algorithms have been proposed in the literature for clustering: The Iterative Self-Organizing Data Analysis Technique (ISODATA) [47], Clustering Large Applications based up on Randomized Search (CLARANS) [48], Parallel-cluster (p-cluster) [49], Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [50] and Balanced Iterative Reducing and Clustering using Hierarchies (BIRCH) [51].  $k$ -means [43] is one of the simplest unsupervised learning algorithms that solve the well-known clustering problem. This procedure depend on the easiest way of classification of a given data set through a certain number of clusters (assume  $k$  clusters) fixed a priori. The main idea is to define  $k$  centroids, one for each cluster. These centroids should be placed in a cunning way because of different location causes different result. So, the better choice is to place them as much as possible far away from each other. The next step is to take each point belonging to a given data set and associate it to the nearest centroid. The first step will be completed and an early group is done when no pending of any point happen. At this point re-calculation of  $k$  new centroids as centers of the clusters, which resulted from the previous step should be done. After we have these  $k$  new centroids, a new binding has to be done between the same data set points and the nearest new centroid. The loop has been generated, as a result of this loop we may notice that the  $k$  centroids change their location step by step until no more changes are done. In other words centroids do not move any more. A centroid definition is the point whose coordinates are obtained by computing the average of each of the coordinates (*i.e.*, feature values) of the cluster points. Formally, the  $k$ -means clustering algorithm follows the following steps (taken from [52]):

Step 1: define a number of desired clusters,  $k$ .

Step 2: choose an initial cluster centroids randomly, which represent temporary means of the clusters.

Step 3: compute the squared Euclidean distance (sum of square error) from each object to each cluster and each object is assigned to the closest cluster as follows:

$$\text{Sum of Squire Error (SSE)} = \sum_{i=1}^k \sum_{x \in C_i} \text{dist}^2(m_i, x); \quad (9)$$

where  $x$  is a data point in cluster  $C_i$  and  $m_i$  is the centroid of cluster  $C$ .

Step 4: The new centroid is computed, which is the average coordinate of objects, for each cluster, and each centroid value is now replaced by the respective cluster centroid.

Step 5: Repeat steps 3 and 4 until no point changes its cluster.

**Figure 1** shows an illustration of  $k$ -means algorithm on a 2-dimensional dataset with three clusters.

## 5. Genetic Algorithm Based on $k$ -Means-Clustering Technique for the UC Problem

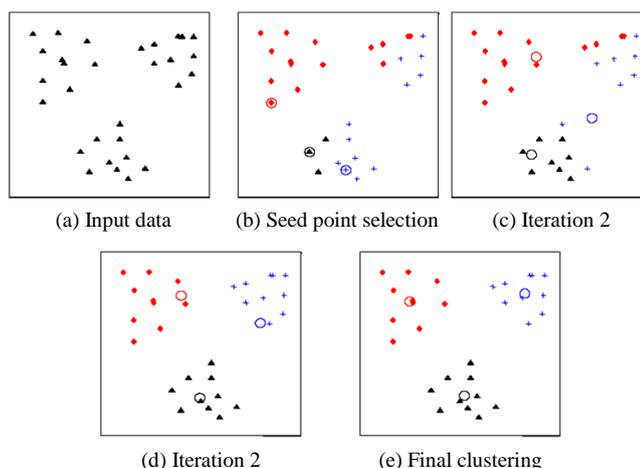
In this section, the proposed algorithm is presented. The algorithm integrates the main features of binary-real coded GA [16] and  $k$ -means clustering technique. The following subsections describe the details of proposed algorithm.

### 5.1. Chromosome Representation and Initialization

The UC problem involves both  $\{0, 1\}$  binary variables to represent the on/off status of units and real variables to represent the amounts of power to be generated by committed units. Therefore, a chromosome (solution) of the proposed algorithm is considered to be combined matrix ( $N*2T$ ), the first one ( $N*T$ )  $u_{it}$  represents the on/off status of unit  $i$  at time  $t$ , and the other one ( $N*T$ )  $p_{it}$  represents the amount of power generated by the unit at that time instant. each chromosome is initialized randomly, where  $u_{it}$  is assigned the value of 0 or 1 with equal probability and  $p_{it}$  is assigned a random real value in the range of  $[p_i^{\min}, p_i^{\max}]$  according to  $u_{it}$  matrix, *i.e.* when  $v = 1$ ,  $p_{it} = \text{random real value in the range of } [p_i^{\min}, p_i^{\max}]$  and when  $u_{it} = 0$ ,  $p_{it} = 0$ . A matrix representation of an individual in the population is shown in **Figure 2**.

### 5.2. Handling Constraints

When using GAs to solve constrained optimization problem, the constraints must be handled because genetic operation used to manipulate the chromosome often yield infeasible solutions [54]. The existing constraint handling techniques in the literature can be classified as follow:



**Figure 1.** Illustration of  $k$ -means algorithm. (a) Two-dimensional input data with three clusters; (b) Three centroid points selected as cluster centers and initial assignment of the data points to clusters; (c) (d) Intermediate iterations updating cluster labels and their centers; (e) Final clustering obtained by  $k$ -means algorithm at convergence. (Taken from [53]).

$$\text{chromosome} = \begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1T} & \cdots & p_{11} & p_{12} & \cdots & p_{1T} \\ u_{21} & u_{22} & \cdots & u_{2T} & \cdots & p_{21} & p_{22} & \cdots & p_{2T} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ u_{N1} & u_{N2} & \cdots & u_{NT} & \cdots & p_{N1} & p_{N2} & \cdots & p_{NT} \end{bmatrix}$$

Figure 2. Representation of the chromosome.

- Rejecting technique;
- Repairing technique;
- Penalizing technique.

A Rejecting technique rejects all infeasible chromosomes and replaced by randomly drown new feasible chromosomes, while the penalty technique transforms the constrained optimization problem into unconstrained one by penalizing infeasible solution. The repairing technique means to take an infeasible chromosome and generate a feasible one through some repairing procedure [55]. For many combinatorial optimization problems, it is relatively easy to create a repairing procedure [56], there for a repairing technique is used in this paper to handle with infeasible solution.

### Repairing Mechanisms for the UC Problem

The idea of this technique is to convert any infeasible individuals to a feasible solution by repairing the sequential possible violations constraints in the UC problem. The following five repairing mechanisms taking from literature [16] [55] [57]-[59] are incorporated in the proposed algorithm.

#### 1) Spinning reserve constraint repairing:

Satisfaction of spinning reserve constraint can be accomplished by applying a heuristic algorithm [16] in which commitment of uncommitted units, in ascending order of their average full load cost, until spinning reserve requirement is met. The average full-load cost of unit  $i$  can be expressed as:

$$\alpha_i = \frac{C_{it}(P_{it})}{p_i^{\max}} = \frac{a_i}{p_i^{\max}} + b_i + c_i p_i^{\max}; \quad (10)$$

#### 2) Minimum up and down time constraints repairing:

Minimum up- and down-time constraints can be satisfied through adjusting unit status [57]. The state of a unit is evaluated starting from the first hour. If there are violation in minimum up or down time constraint at a given time “ $t$ ”, the state (on/off) of the unit at that hour is reversed and updated. The process continues until the last hour. The general heuristic procedure for handling the minimum up-time and down-time constraints is summarized according to **Algorithm 1**.

Algorithm 1: Pseudo-code of heuristic procedure for handling the minimum up-time and down-time constraints

```

for  $i = 1 : N$ 
  for  $t = 1 : T$ 
    if  $u_{it} = 1$ 
      if  $u_{i,t-1} = 0$ 
        if  $\tau_{i,t-1}^{off} < \Gamma_i^{down}$  then  $u_{it} = 0$ 
          elseif  $\tau_{i,t-1}^{off} \geq \Gamma_i^{down}$  then  $u_{it} = 1$ 
        endif
      elseif  $u_{i,t-1} = 1$  then  $u_{it} = 1$ 
      endif
    elseif  $u_{it} = 0$ 
      if  $u_{i,t-1} = 1$ 
        if  $\tau_{i,t-1}^{on} < \Gamma_i^{on}$  then  $u_{it} = 1$ 
          else if  $\tau_{i,t-1}^{on} \geq \Gamma_i^{on}$  then  $u_{it} = 0$ 
        endif
      else if  $u_{i,t-1} = 0$  then  $u_{it} = 0$ 
      endif
    endif
  endif
End
End

```

### 3) Unit de-commitment for excessive spinning reserve:

Excessive spinning reserve should be avoided because it is directly related to the high operation cost. Therefore, a heuristic algorithm is used to de-commit some units one by one [3], in descending order of their average full load costs, until the spinning reserve constraint is just satisfied at any time instant. However, such de-commitment is made subject to the up/down time constraints of a unit satisfaction, *i.e.*, a unit will be de-committed only if no up/down time constraint of the unit is violated from such de-commitment [16].

### 4) Power balance constraint repairing:

For adjusting the system power balance at time instant, the amount of divergence in generated power from the power demand at time  $t$  is obtained as:

$$E_t = \sum_{i=1}^N u_{it} p_{it} - D_t u_{it} \quad (11)$$

The repairing algorithm is applied to the following two cases [16]:

a) If  $E_t > 0$ , taking the committed units in descending order of their average full load costs given by Equation (10) and then reducing the amount of power generated by the units up to their lower limits (*i.e.*,  $p_i^{\min}$ ) until  $E_t$  becomes zero ( $E_t = 0$ ).

b) If  $E_t < 0$ , taking the committed units in ascending order of their average full load costs given by Equation (10) and then increasing the amount of power generated by the units up to their maximum limits. (*i.e.*,  $p_i^{\max}$ ) until  $E_t$  becomes zero ( $E_t = 0$ ).

## 5.3. Selection Operation

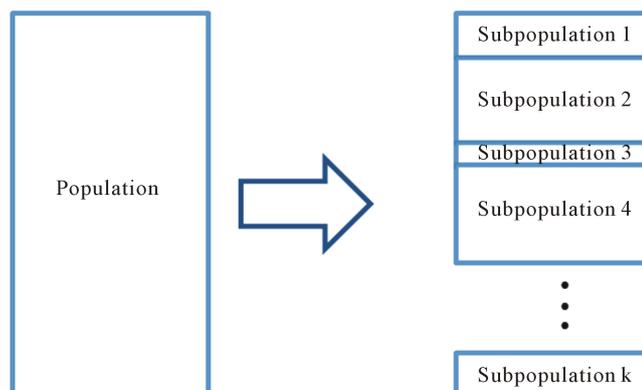
Selection operator gives the high quality chromosomes a better chance to get copied into the next generation, which led to improve the average quality of the population [60]. The selection directs GA search towards promising regions in the search space. In this paper, the binary tournament selection operator [61] is applied, in which two individuals are chosen at random and the better objective value of the two individuals is selected and copied in mating pool. The process is repeated until the size of the mating pool equals that of the original population.

## 5.4. $k$ -Means Clustering Technique

In order to keep diversity and to avoid trapping in local minima, the  $k$ -means cluster algorithm was implemented. In this step, the population in mating pool is split into  $k$  separated subpopulations with dynamic size, as illustrated in Figure 3.

## 5.5. Crossover Operator

The goal of crossover is to exchange information between two chromosomes in order to produce two new offspring for the next population [62]. In our study we used common crossover techniques in a GA. A brief explanation of these techniques is given below.



**Figure 3.** The population is split into  $k$  separated subpopulations with dynamic size.

- **Horizontal band crossover:**

In horizontal band crossover [63], two random numbers are generated, and information inside the horizontal region of the grid (matrix) determined by the numbers is exchanged between two parents to generate two off-springs based on a fixed probability. Figure 4 shows an example to illustrate how the horizontal band crossover works.

- **Uniform crossover:**

In uniform crossover [64], the bits are exchanged between the parent points to create two new offspring points by randomly generated mask. In the random mask the “1” represent bit swapping and “0” denotes bits unchanged. The scheme of uniform crossover is shown in the Figure 5.

- **Real part crossover:**

In the real part crossover, the information in column vectors of parents of power generated by unit are exchanged (i.e. operates on power part ( $p_{ii}$ )). The basic steps of real part crossover operation are given below [65]:

Step 1: choose two parent chromosomes randomly from mating pool. We can represent power parts ( $p_{ii}$ ) of parents by

$$\begin{aligned}
 p^{parent_1} &= [V_1^{parent_1}, V_2^{parent_1}, \dots, V_t^{parent_1}, \dots, V_T^{parent_1}] \\
 p^{parent_2} &= [V_1^{parent_2}, V_2^{parent_2}, \dots, V_t^{parent_2}, \dots, V_T^{parent_2}]
 \end{aligned}
 \tag{12}$$

where

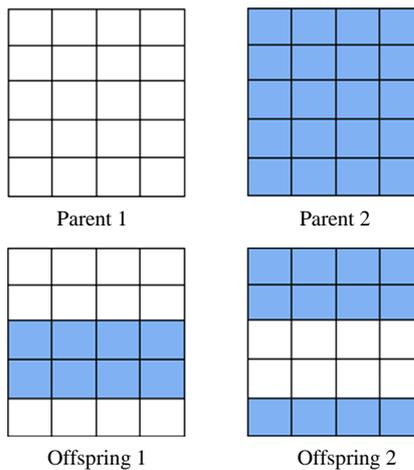


Figure 4. Horizontal band crossover operator.

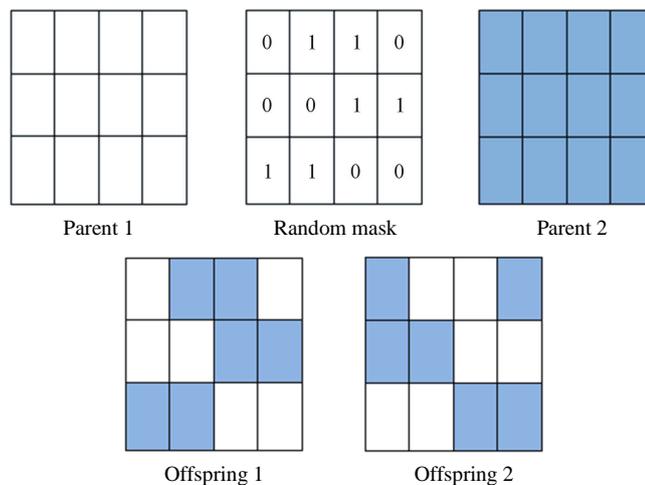


Figure 5. Uniform crossover operator.

$$V_i = [p_{1i}, p_{2i}, \dots, p_{it}, \dots, p_{Ni}]^T \tag{13}$$

Step 2: choose column vector randomly.

Step 3: Two real parts of parent chromosomes produce new real parts of offspring (offspring1 and offspring2), which are created as:

$$\begin{aligned} offspring_1 &= [V_1^{parent_1}, V_2^{parent_1}, \dots, V_{j-1}^{parent_1}, (1-\beta)V_j^{parent_1} + \beta V_j^{parent_2}, V_{j+1}^{parent_1}, \dots, V_T^{parent_1}] \\ offspring_2 &= [V_1^{parent_2}, V_2^{parent_2}, \dots, V_{j-1}^{parent_2}, \beta V_j^{parent_1} + (1-\beta)V_j^{parent_2}, V_{j+1}^{parent_2}, \dots, V_T^{parent_2}]; \end{aligned} \tag{14}$$

where  $\beta$  is the random number in range of (0, 1) and  $j$  is a random positive integer in range of [1, T].

### 5.6. Mutation Operator

Mutation is a genetic operator used to maintain genetic diversity from one generation of a population of genetic algorithm chromosomes to the next. It is analogous to biological mutation. Mutation alters one or more gene values in a chromosome from its initial state. In mutation, the solution may change entirely from the previous solution. Hence GA can come to better solution by using mutation. Mutation occurs during evolution according to a user-definable mutation probability. This probability should be set low. If it is set too high, the search will turn into a primitive random search [62]. In our study we used common mutation techniques in a GA. A brief explanation of these techniques is given below.

- **One point mutation:**

With a small probability, randomly chosen bits of the binary part ( $u_{it}$  part) of offspring genotypes change from “1” to “0” and vice versa as shown in **Figure 6**. In the same time any unit status changed from 0 to 1, the corresponding power changed from 0 to random real value in the range of  $[p_i^{min}, p_i^{max}]$  [64].

- **Intelligent mutation:**

This operator [66] looks for (01) or (10) combinations in commitment schedule. Mutation operator randomly changed the combination to 00 or 11 as shown in **Figure 7**.

### 5.7. Combination Stage

In combination stage, all subpopulations are combined together again to create a new population, as illustrated in **Figure 8**.

### 5.8. Elite-Preserving Operator

This operator helps to save a group of best solutions for the next generation. Implementation of elite-preserving operator can be done by directly copying the best 10% chromosomes from the current population to the next generation [67].

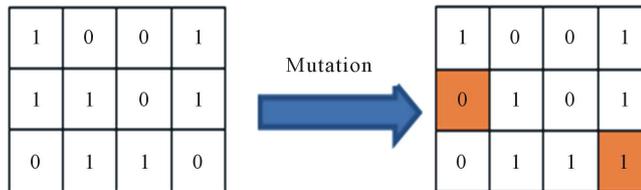


Figure 6. Mutation operator.

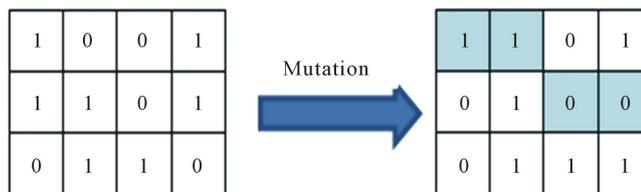


Figure 7. Intelligent mutation operator.

## 6. Simulations and Results

In this section, an experimental verification of the proposed algorithm is carried out. Tests have been performed on 10 units power systems over a 24-h time horizon taken from literature [16] [65]. The proposed algorithm is coded using MATLAB programming language. All tests have been executed on an Intel core I5, 2.6 GHz processor personal computer. The parameters setting of proposed algorithm are depicted in Table 1. The properties of the 10 units system are given in Table 2. The hourly forecast load demand  $D_i$  is presented in Table 3. At any time instant, the minimum spinning reserve requirement is considered to be 10% of the forecasted power demand. In Table 2, the initial status of unit  $i$  " $\sigma_i$ " indicates to the duration off the unit on/off prior to the start of the time horizon. A positive value means that unit  $i$  was "on" for that number of time instants prior to the starting of the time horizon, while a negative value means that the unit was off for that number of time instants prior to the starting of the time horizon.

Table 4 presents the best and worst operating costs, as well as the average and standard deviation, obtained over 10 independent runs at different number of cluster; where it is observed that the variation in the operating costs over 10 runs of a system at each cluster are not so high, which depicts the consistency of the proposed algorithm over different runs. This allows for inferring the robustness of the solution since the gaps between the best and the worst solutions are very small. Figure 9 shows the best operating cost of 10-unit of the proposed algorithm with different number of cluster; where it is observed that the operating cost of the proposed algorithm at  $k = 3$  is much smaller than those at  $k = 1$  and  $k = 2$ , therefore as the number of cluster increased, the operating cost decreased, so the increase of the number of clusters in proposed algorithm leads to lower production cost.

Table 5 presents the best solutions obtained by the proposed algorithm for the 10-unit power system. The convergence curve of the best solutions of the 10-unit power system at  $k = 1$  (without cluster),  $k = 2$  and  $k = 3$  are shown in Figure 10.

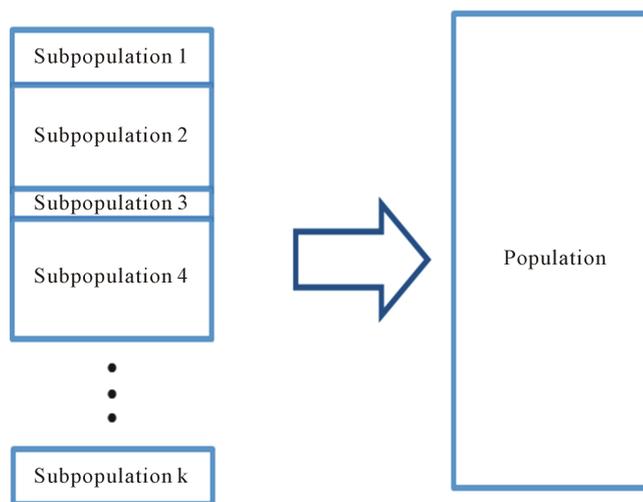


Figure 8. Commination stage.

Table 1. The proposed algorithm parameters.

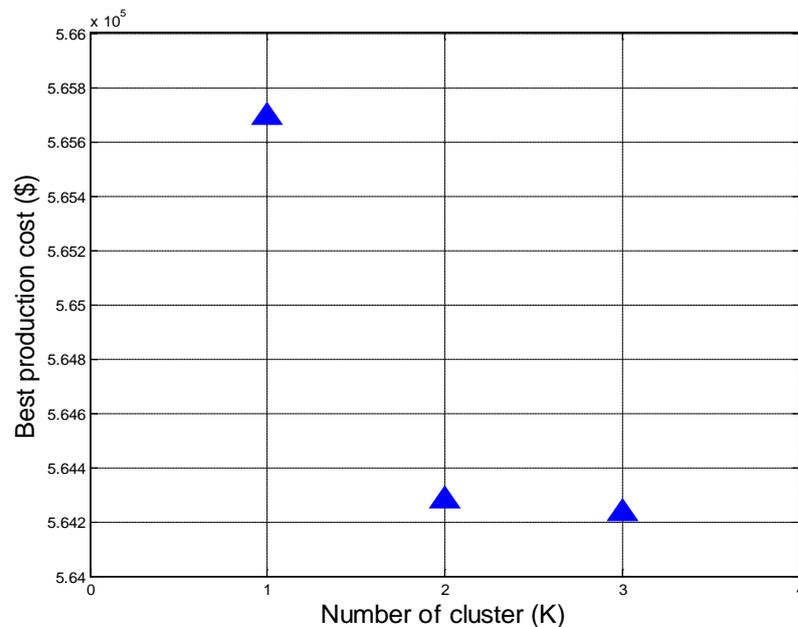
Parameter	Values
	Prolem 1
Population size	500
Crossover rate	1
Mutation rate	0.01
Iteration	100
Number of cluster ( $k$ )	1&2&3

**Table 2.** The properties of the 10 units system.

Unit	$p_i^{\max}$ (MW)	$p_i^{\min}$ (MW)	$a_i$ (\$/h)	$b_i$ (\$/MW·h)	$c_i$ (\$/MW <sup>2</sup> ·h)	$d_i$ (\$)	$e_i$ (\$)	$f_i$ (h)	$\Gamma_i^{up}$ (h)	$\Gamma_i^{down}$ (h)	$\sigma_i$ (h)
1	455	150	1000	16.19	0.00048	4500	9000	5	8	8	8
2	455	150	970	17.26	0.00031	5000	10,000	5	8	8	8
3	130	20	700	16.60	0.00200	550	1100	4	5	5	-5
4	130	20	680	16.50	0.00211	560	1120	4	5	5	-5
5	162	25	450	19.70	0.00398	900	1800	4	6	6	-6
6	80	20	370	22.26	0.00712	170	340	2	3	3	-3
7	85	25	480	27.74	0.00079	260	520	2	3	3	-3
8	55	10	660	25.92	0.00413	30	60	0	1	1	-1
9	55	10	665	27.27	0.00222	30	60	0	1	1	-1
10	55	10	670	27.79	0.00173	30	60	0	1	1	-1

**Table 3.** The hourly forecast load demand  $D_t$ .

Hour	1	2	3	4	5	6
Demand (MW)	700	750	850	950	1000	1100
Hour	7	8	9	10	11	12
Demand (MW)	1150	1200	1300	1400	1450	1500
Hour	13	14	15	16	17	18
Demand (MW)	1400	1300	1200	1050	1000	1100
Hour	19	20	21	22	23	24
Demand (MW)	1200	1400	1300	1100	900	800



**Figure 9.** Comparison of the best operating cost of 10-unit power system of the proposed algorithm with different number of cluster ( $k = 1$ ,  $k = 2$  and  $k = 3$ ).

**Table 4.** Statistical analysis of the solutions obtained from 10 independent runs for 10-unit system.

Number of unit	10-unit			
	Number of cluster	$k = 1$	$k = 2$	$k = 3$
Best cost		565,690	564,280	564,230
Worst cost		566,000	565,000	565,350
Average		565,866	564,615	564,481
Standard deviation		131.6207	251.047	103.58

**Table 5.** The best solution of the 10-unit power system (total operating cost of \$564,230).

Hour	Unit schedule	Power generated by units (MW)										Fuel cost	Startup cost	
		1	2	3	4	5	6	7	8	9	10			
1	1100000000	455	245	0	0	0	0	0	0	0	0	0	13,683.12975	0
2	1100000000	455	295	0	0	0	0	0	0	0	0	0	14,554.49975	0
3	1100100000	455	370	0	0	25	0	0	0	0	0	0	16,809.4485	900
4	1100100000	455	455	0	0	40	0	0	0	0	0	0	18,597.66775	0
5	1101100000	455	430.1015	0	89.8985	25	0	0	0	0	0	0	20,042.08517	560
6	1111100000	455	455	41.0574	123.9426	25	0	0	0	0	0	0	22,440.67775	1100
7	1111100000	455	455	85	130	25	0	0	0	0	0	0	23,284.39625	0
8	1111100000	455	455	130	130	30	0	0	0	0	0	0	24,150.34075	0
9	1111111000	455	455	130	130	85	20	25	0	0	0	0	27,251.0560	860
10	1111111100	455	455	130	130	162	33	25	10	0	0	0	30,057.5503	60
11	1111111110	455	455	130	130	162	73	25	10	10	0	0	31,916.0611	60
12	1111111111	455	455	130	130	162	80	27.4613	40.5387	10	10	0	33,893.8954	60
13	1111111100	455	455	130	130	162	33	25	10	0	0	0	30,057.5503	0
14	1111111000	455	455	130	130	85	20	25	0	0	0	0	27,251.056	0
15	1111100000	455	455	130	130	30	0	0	0	0	0	0	24,150.34075	0
16	1111100000	455	455	65.1721	49.8279	25	0	0	0	0	0	0	21,596.03802	0
17	1111100000	455	381.0243	33.2035	105.7722	25	0	0	0	0	0	0	20,704.52511	0
18	1111100000	455	443.61259	73.2606	103.12678	25	0	0	0	0	0	0	22,429.4611	0
19	1111100000	455	455	130	130	30	0	0	0	0	0	0	24,150.34075	0
20	1111111100	455	455	130	130	162	33	25	10	0	0	0	30,057.5503	490
21	1111111000	455	455	130	130	85	20	25	0	0	0	0	27,251.056	0
22	1100111000	455	455	0	0	145	20	25	0	0	0	0	22,735.521	0
23	1100010000	455	425	0	0	0	20	0	0	0	0	0	17,645.36375	0
24	1100000000	455	345	0	0	0	0	0	0	0	0	0	15,427.41975	0

The comparison between the best solutions obtained by the proposed algorithm at  $k = 1$ ,  $k = 2$  and  $k = 3$  for 10-unit system with those produced by some other meta-heuristic-based recent approaches [5] [17] [18] [20] [65] [68] [69] is presented in Table 6.

As well as, Figure 11 show the plots of the operating cost of few selective techniques for 10-unit system .The best solution obtained by the proposed algorithm at  $k = 1$  (i.e. without cluster) is better than the solution obtained by LR [12] and ESA [18]. where the proposed algorithm at  $k = 2$  gives solutions better than the solution obtained by EP [17], ICGA [69], PL [5], ESA [18], SFL [20] and GA-LR [68]. but the solution obtained by proposed algorithm at  $k = 3$  is better than the solution obtained by all techniques [5] [12] [17] [18] [20] [65] [68] [69]. The superiority of proposed algorithm at  $k = 3$  is obvious, which indicates that the proposed algorithm with large number of cluster performs better than without clustering technique. Furthermore, the proposed algorithm has the high percentage saving in the cost over these methods.

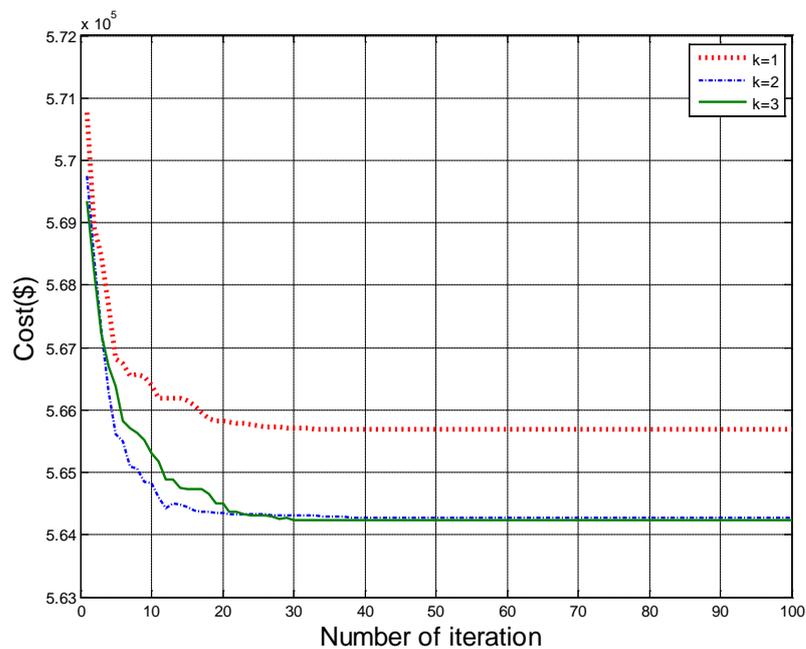


Figure 10. Convergence curve for the 10-unit system at  $k = 1$ ,  $k = 2$ ,  $k = 3$ .

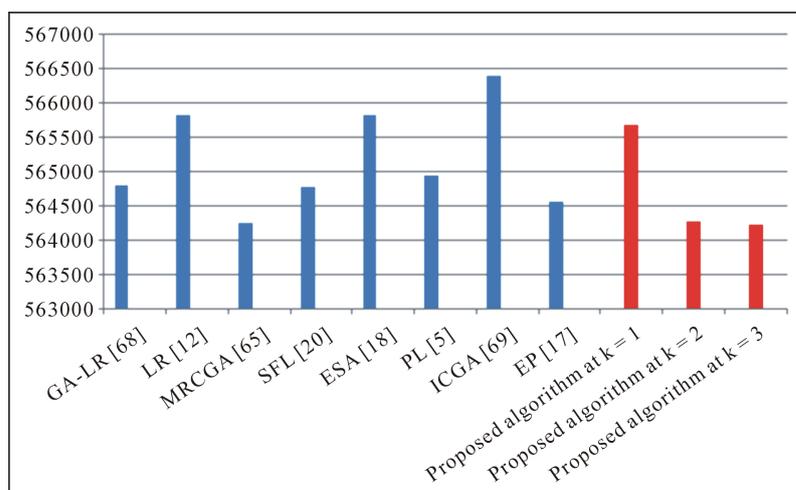


Figure 11. The comparison between the best solutions obtained by the proposed algorithm at different values of  $k$  for 10-unit system with those produced by some other meta-heuristic-based recent techniques.

**Table 6.** Comparison between the proposed algorithm and other algorithms.

Method	Production cost \$
GA-LR [68]	564,800
LR [12]	565,825
MRCGA [65]	564,244
SFL [20]	564,769
ESA [18]	565,828
PL [5]	564,950
ICGA [69]	566,404
EP [17]	564,551
Proposed algorithm at $k = 1$	565,690
Proposed algorithm at $k = 2$	564,280
Proposed algorithm at $k = 3$	564,230

## 7. Conclusions

This paper investigates the unit commitment problem by genetic algorithm based on  $k$ -mean clustering algorithm which integrates the main features of a binary-real coded genetic algorithm (GA) and  $k$ -means clustering technique. The binary coded GA is used to obtain a feasible commitment schedule for each generating unit; where the amounts of power generated by committed units are determined by using real coded GA for the feasible commitment obtained in each interval.  $k$ -means clustering algorithm divided population into a specific  $k$  of subpopulations. In this way, the different GA operators (crossover and mutation) can be applied to each subpopulation instead of one GA operator applied to the whole population. To evaluate the performance of the proposed algorithm, test power systems available in the literature are solved at different number of cluster and compared with the previous studies. A careful observation will reveal the following benefits of the proposed algorithm:

- 1) The proposed algorithm can obtain feasible and satisfactory solutions of different UC problems, regardless of the system size.
- 2) Incorporating GA with  $k$ -means clustering technique preserve, introduce diversity, and allow the algorithm to avoid local minima by preventing the population of chromosomes from becoming too similar to each other and to benefit from the advantages of both types of algorithms.
- 3) Binary-real-coded GA is investigated that a GA alone can tackle both the unit scheduling and load dispatch problems.
- 4) The tests result demonstrated that when the number of cluster increased in the proposed algorithm, the production cost decreased.
- 5) The tests result demonstrates the satisfactory performance of presented approach with respect to the quality and computational requirements with the previously reported results.

## References

- [1] Wood, A.J. and Wollenberg, B.F. (1996) Power Generation, Operation and Control. John Wiley & Sons, New York.
- [2] Guan, X.H., Luh, P.B., Yan, H. and Amalfi, J.A. (1992) An Optimization-Based Method for Unit Commitment. *International Journal of Electrical Power & Energy Systems*, **14**, 9-17. [http://dx.doi.org/10.1016/0142-0615\(92\)90003-R](http://dx.doi.org/10.1016/0142-0615(92)90003-R)
- [3] Jeong, Y.W., Park, J.B., Shin, J.R. and Lee, K.Y. (2009) A Thermal Unit Commitment Approach Using an Improved Quantum Evolutionary Algorithm. *Electric Power Components and Systems*, **37**, 770-786. <http://dx.doi.org/10.1080/15325000902762331>
- [4] Padhy, N.P. (2004) Unit Commitment—A Bibliographical Survey. *IEEE Transactions on Power Systems*, **19**, 1196-1205. <http://dx.doi.org/10.1109/TPWRS.2003.821611>

- [5] Senjyu, T., Miyagi, T., Saber, A.Y., Urasaki, N. and Funabashi, T. (2006) Emerging Solution of Large-Scale Unit-commitment Problem by Stochastic Priority List. *Electrical Power and Energy Systems*, **76**, 283-292. <http://dx.doi.org/10.1016/j.epsr.2005.07.002>
- [6] Rong, A., Hakonen, H. and Lahdelma, R. (2009) A Dynamic Regrouping Based Sequential Dynamic Programming Algorithm for Unit Commitment of Combined Heat and Power Systems. *Energy Conversion and Management*, **50**, 1108-1115. <http://dx.doi.org/10.1016/j.enconman.2008.12.003>
- [7] Chen, C.L. and Wang, S.C. (1993) Branch-and-Bound Scheduling for Thermal Generating Units. *IEEE Transactions on Energy Conversion*, **8**, 184-189.
- [8] Singhal, P.K. (2011) Generation Scheduling Methodology for Thermal Units Using Lagrangian Relaxation. *Proceedings of the 2nd IEEE International Conference on Current Trends in Technology*, 1-6.
- [9] Frangioni, A., Gentile, C. and Lacalandra, F. (2009) Tighter Approximated MILP Formulations for Unit Commitment Problems. *IEEE Transactions on Power Systems*, **24**, 105-113. <http://dx.doi.org/10.1109/TPWRS.2008.2004744>
- [10] Sheble, G.B. and Fahd, G.N. (1993) Unit Commitment Literature Synopsis. *IEEE Transactions on Power Systems*, **9**, 128-135. <http://dx.doi.org/10.1109/59.317549>
- [11] Swarup, K. and Simi, P. (2006) Neural Computation Using Discrete and Continuous Hopfield Networks for Power System Economic Dispatch and Unit Commitment. *Neurocomputing*, **70**, 119-129. <http://dx.doi.org/10.1016/j.neucom.2006.05.002>
- [12] Kazarlis, S.A., Bakirtzis, A.G. and Petridis, V. (1996) A Genetic Algorithm Solution to the Unit Commitment Problem. *IEEE Transactions on Power Systems*, **11**, 1129-1136.
- [13] Arroyo, J.M. and Conejo, A.J. (2002) A Parallel Repair Genetic Algorithm to Solve the Unit Commitment Problem. *IEEE Electrical Power Energy Systems*, **17**, 1216-1224. <http://dx.doi.org/10.1109/tpwrs.2002.804953>
- [14] Dang, C. and Li, M. (2007) A Floating-Point Genetic Algorithm for Solving the Unit Commitment Problem. *European Journal of Operational Research*, **181**, 1370-1395. <http://dx.doi.org/10.1016/j.ejor.2005.10.071>
- [15] Sundararajan, D., Subramanian, B., Subramanian, K. and Krishnan, M. (2013) Generation Scheduling Problem by Intelligent Genetic Algorithm. *Computers and Electrical Engineering*, **39**, 79-88.
- [16] Datta, D. (2013) Unit Commitment Problem with Ramp Rate Constraint Using a Binary-Real-Coded Genetic Algorithm. *Applied Soft Computing*, **13**, 3873-3883.
- [17] Juste, K.A., Kita, H., Tanaka, E. and Hasegawa, J. (1999) An Evolutionary Programming Solution to the Unit Commitment Problem. *IEEE Transactions on Power Systems*, **14**, 1452-1459. <http://dx.doi.org/10.1109/59.801925>
- [18] Simopoulos, N., Kavatza, S. and Vournas, C. (2006) Unit Commitment by an Enhanced Simulated Annealing Algorithm. *IEEE Transactions on Power Systems*, **21**, 68-76. <http://dx.doi.org/10.1109/TPWRS.2005.860922>
- [19] Mantawy, A.H., Abdel-Magid, Y.L. and Selim, S.Z. (1999) Integrating Genetic Algorithms, Tabu Search, and Simulated Annealing for the Unit Commitment Problem. *IEEE Transactions on Power Systems*, **14**, 829-836. <http://dx.doi.org/10.1109/59.780892>
- [20] Ebrahimi, J., Hosseini, S.H. and Gharehpetian, G.B. (2011) Unit Commitment Problem Solution Using Shuffled Frog Leaping Algorithm. *IEEE Transactions on Power Systems*, **26**, 573-581.
- [21] Logenthiran, T. and Srinivasan, D. (2010) Particle Swarm Optimization for Unit Commitment Problem. *Proceedings of the IEEE International Conference on Probabilistic Methods Applied to Power Systems*, Singapore, 14-17 June 2010, 642-647. <http://dx.doi.org/10.1109/PMAPS.2010.5528899>
- [22] Mantawy, A.H., Abdel-Magid, Y.L. and Selim, S.Z. (1998) Unit Commitment by Tabu Search. *IEE Proceedings—Generation, Transmission and Distribution*, **145**, 56-64. <http://dx.doi.org/10.1049/ip-gtd:19981681>
- [23] El-Saadawi, M.M., Tantawi, M.A. and Tawfik, E. (2004) A Fuzzy Optimization-Based Approach to Large Scale Thermal Unit Commitment. *Electric Power Systems Research*, **72**, 245-252. <http://dx.doi.org/10.1016/j.epsr.2004.04.009>
- [24] Pourjamal, Y. and Ravadanegh, S.N. (2013) HSA Based Solution to the UC Problem. *International Journal of Electrical Power & Energy Systems*, **46**, 211-220. <http://dx.doi.org/10.1016/j.ijepes.2012.10.042>
- [25] Chandrasekaran, K., Hemamalini, S., Simon, S.P. and Padhy, N.P. (2012) Thermal Unit Commitment Using Binary/Real Coded Artificial Bee Colony Algorithm. *Electric Power Systems Research*, **84**, 109-119.
- [26] Mousa, A.A., El-Shorbagy, M.A. and Abd El-Wahed, W.F. (2012) Local Search Based Hybrid Particle Swarm Optimization for Multiobjective Optimization. *International Journal of Swarm and Evolutionary Computation*, **3**, 1-14.
- [27] Farag, M.A., El-Shorbagy, M.A., El-Desoky, I.M., El-Sawy, A.A. and Mousa, A.A. (2015) Genetic Algorithm Based on K-Means-Clustering Technique for Multi-Objective Resource Allocation Problems. *British Journal of Applied Science & Technology*, **8**, 80-96. <http://dx.doi.org/10.9734/BJAST/2015/16570>
- [28] El-Shorbagy, M.A., Mousa, A.A. and Abd-El-Wahed, W.F. (2011) Hybrid Particle Swarm Optimization Algorithm for

Multi-Objective Optimization. Lambert Academic Publishing GmbH & Co. KG, Berlin.

- [29] El-Shorbagy, M.A., El-Sawy, A.A. and Hendawy, Z.M. (2013) Numerical Optimization & Swarm Intelligence for Optimization: Trust Region Algorithm & Particle Swarm Optimization. Lambert Academic Publishing GmbH & Co. KG, Berlin.
- [30] Mousa, A.A. and El-Desoky, I.M. (2008) GENLS: Co-Evolutionary Algorithm for Nonlinear System of Equations. *Applied Mathematics and Computation*, **197**, 633-642.
- [31] Mousa, A.A. and El-Desoky, I.M. (2013) Stability of Pareto Optimal Allocation of Land Reclamation by Multistage Decision-Based Multipheromone Ant Colony Optimization. *Swarm and Evolutionary Computation*, **13**, 13-21
- [32] Abd El-Wahed, W.F., Mousa, A.A. and El-Shorbagy, M.A. (2011) Integrating Particle Swarm Optimization with Genetic Algorithms for Solving Nonlinear Optimization Problems. *Journal of Computational and Applied Mathematics*, **235**, 1446-1453.
- [33] Mousa, A.A. and El-Shorbagy, M.A. (2012) Enhanced Particle Swarm Optimization Based Local Search for Reactive Power Compensation Problem. *Applied Mathematics*, **3**, 1276-1284. <http://dx.doi.org/10.4236/am.2012.330184>
- [34] Michalewicz, Z. (1996) Genetic Algorithms + Data Structures = Evolution Programs. Third Edition, Springer-Verlag, New York.
- [35] Verma, M., Srivastava, M., Chack, N., Diswar, A.K. and Gupta, N. (2012) A Comparative Study of Various Clustering Algorithms in Data Mining. *International Journal of Engineering Research and Applications (IJERA)*, **2**, 1379-1384.
- [36] Levine, S., Pilowski, I. and Boulton, D.M. (1969) The Classification of Depression by Numerical Taxonomy. *The British Journal of Psychiatry*, **115**, 937-945. <http://dx.doi.org/10.1192/bjp.115.525.937>
- [37] Dolnicar, S. (2003) Using Cluster Analysis for Market Segmentation-Typical Misconceptions, Established Methodological Weaknesses and Some Recommendations for Improvement. *Australasian Journal of Market Research*, **11**, 5-12.
- [38] Hodson, F.R. (1971) Numerical Typology and Prehistoric Archaeology. In: Hodson, F.R., Kendall, D.G. and Tautu, P.A., Eds., *Mathematics in the Archaeological and Historical Sciences*, University Press, Edinburgh.
- [39] Pratima, D., Nimmakanti, N. and Recognition, P. (2011) Algorithms for Cluster Identification Problem. *Special Issue of International Journal of Computer Science & Informatics (IJCSI)*, **2**, 2231-5292.
- [40] Doreswamy and Hemanth, K.S. (2012) Similarity Based Cluster Analysis on Engineering Materials Data Sets. *Advances in Intelligent Systems and Computing*, **167**, 161-168.
- [41] Dilts, D., Khamalah, J. and Plotkin, A. (1995) Using Cluster Analysis for Medical Resource Decision Making. *Cluster Analysis for Resource Decisions*, **15**, 333-347.
- [42] Aggarwal, C.C. and Reddy, C.K. (2014) Data Clustering Algorithms and Applications. CRC Press, New York.
- [43] MacQueen, J.B. (1967) Some Methods for Classification and Analysis of Multivariate Observation. In: *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability*, University of California Press, Berkeley, 281-297.
- [44] Priya, R.P., Sivaraj, N. and Muruganandam, M. (2015) A Solution to Unit Commitment Problem with V2G Using Harmony Search Algorithm. *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, **4**, 1208-1214.
- [45] Goldberg, D.E. (1989) Genetic Algorithms in Search, Optimization and Machine Learning. Addison Wesley, Reading.
- [46] Grefenstette, J.J. (1986) Optimization of Control Parameters for Genetic Algorithms. *IEEE Transactions on Systems, Man and Cybernetics*, **16**, 122-128. <http://dx.doi.org/10.1109/TSMC.1986.289288>
- [47] Arai, K. and Bu, X.Q. (2007) ISODATA Clustering with Parameter (Threshold for Merge and Split) Estimation Based on GA: Genetic Algorithm. *Reports of the Faculty of Science and Engineering, Saga University*, **36**, 17-23.
- [48] Ng, R.T. and Han, J. (2002) CLARANS: A Method for Clustering Objects for Spatial Data Mining. *IEEE Transactions on Knowledge Data Engineering*, **14**, 1003-1016.
- [49] Judd, D., McKinley, P.K. and Jain, A.K. (1998) Large-Scale Parallel Data Clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **20**, 871-876.
- [50] Edlaa, D.R. and Janaa, P.K. (2012) A Prototype-Based Modified DBSCAN for Gene Clustering. *Procedia Technology*, **6**, 485-492.
- [51] Rani, Y., Manju and Rohil, H. (2014) Comparative Analysis of BIRCH and CURE Hierarchical Clustering Algorithm Using WEKA 3.6.9. *The SIJ Transactions on Computer Science Engineering & Its Applications (CSEA)*, **2**, 25-29.
- [52] Sakthi, M. and Thanamani, A.S. (2011) An Effective Determination of Initial Centroids in K-Means Clustering Using Kernel PCA. *International Journal of Computer Science and Information Technologies*, **2**, 955-959.
- [53] Jain, A. (2010) Data Clustering: 50 Years beyond K-Means. *Pattern Recognition Letters*, **31**, 651-666.

<http://dx.doi.org/10.1016/j.patrec.2009.09.011>

- [54] Kumar, A. and Jani, N.N. (2010) Network Design Problem Using Genetic Algorithm—An Empirical Study on Selection Operator. *International Journal of Computer Science and Applications (IJCSA)*, **3**, 48-52.
- [55] Xing, W. and Wu, F.F. (2002) Genetic Algorithm Based Unit Commitment with Energy Contracts. *International Journal of Electrical Power & Energy Systems*, **24**, 329-336. [http://dx.doi.org/10.1016/S0142-0615\(01\)00048-5](http://dx.doi.org/10.1016/S0142-0615(01)00048-5)
- [56] Gen, M., Cheng, R. and Lin, L. (2008) Network Models and Optimization: Multi-objective Genetic Algorithm Approach. Springer Science & Business Media, New York.
- [57] Jeong, Y.W., Lee, W.N., Kim, H.H., Park, J.B. and Shin, J.R. (2009) Thermal Unit Commitment Using Binary Differential Evolution. *Journal of Electrical Engineering & Technology*, **4**, 323-329.
- [58] Uyar, A.S., Türkay, B. and Keles, A. (2011) A Novel Differential Evolution Application to Short-Term Electrical Power-Generation Scheduling. *Electrical Power and Energy Systems*, **33**, 1236-1242.
- [59] Yuan, X., Su, A., Nie, H., Yuan, Y. and Wang, L. (2009) Application of Enhanced Discrete Differential Evolution Approach to Unit Commitment Problem. *Energy Conversion & Management*, **50**, 2449-2456.
- [60] Mohd, R.N. and Geraghty, J. (2011) Genetic Algorithm Performance with Different Selection Strategies in Solving TSP. *Proceedings of the World Congress on Engineering II*, London, 6-8 July 2011.
- [61] Sivaraj, R. and Ravichandran, T. (2011) A Review of Selection Methods in Genetic Algorithm. *International Journal of Engineering Science and Technology*, **3**, 3792-3797.
- [62] Tzeng, G.H. and Huang, J.J. (2013) Fuzzy Multiple Objective Decision Making. Chapman and Hall/CRC Press, London/Boca Raton.
- [63] Anderson, C.A., Jones, K.F. and Ryan, J. (1991) A Two-Dimensional Genetic Algorithm for the Ising Problem. *Complex Systems*, **5**, 327-333.
- [64] Cheng, C.P. and Liu, C.W. (2002) Unit Commitment by Annealing Genetic Algorithm. *International Journal of Electrical Power & Energy Systems*, **24**, 149-158.
- [65] Sun, L., Zhang, Y. and Jiang, C. (2006) A Matrix Real-Coded Genetic Algorithm to the Unit Commitment Problem. *Electric Power Systems Research*, **76**, 716-728.
- [66] Senjyu, T., Yamashiro, H., Uezato, K. and Funabashi, T. (2002) A Unit Commitment Problem by Using Genetic Algorithm Based on Unit Characteristic Classification. *Proceedings of the IEEE Power Engineering Society Winter Meeting*, **1**, 58-63.
- [67] Kumar, S. and Naresh, R. (2007) Efficient Real Coded Genetic Algorithm to Solve the Non-Convex Hydrothermal Scheduling Problem. *Electrical Power and Energy Systems*, **29**, 738-747.
- [68] Cheng, C.-P., Liu, C.-W. and Liu, C.-C. (2000) Unit Commitment by Lagrangian Relaxation and Genetic Algorithms. *IEEE Transactions on Power Systems*, **15**, 707-714.
- [69] Damousis, I., Bakirtzis, A. and Dokopoulos, P. (2004) A Solution to the Unit-Commitment Problem Using Integer-Coded Genetic Algorithm. *IEEE Transactions on Power Systems*, **19**, 1165-1172. <http://dx.doi.org/10.1109/TPWRS.2003.821625>

## List of Symbols

$p_{it}$	Power output of unit $i$ at hour $t$ , in MW
$u_{it}$	On/off status of unit $i$ at hour $t$ (on = 1, off = 0)
$D_t$	Load demand at hour $t$ , in MW
$N$	Number of units
$p_i^{\max}$	Maximum capacity of unit $i$ , in MW
$p_i^{\min}$	Minimum capacity of unit $i$ , in MW
$R_t$	Spinning reserve at hour $t$ , in MW
$e_i$	Cold startup cost of unit $i$ , in \$
$d_i$	Hot startup cost of unit $i$ , in \$
$f_i$	Cold start hours of unit $i$ , in $h$ (hour)
$SD_{it}$	Shut down cost of unit $i$ at hour $t$ , in \$
$C_{it}(p_{it})$	Fuel cost of unit $i$ at hour $t$ , in \$
$ST_{it}$	Startup cost of unit $i$ at hour $t$ , in \$
$T$	Number of hours, e.g. 24 hour
$\Gamma_i^{\text{down}}$	Minimum down time of unit $i$ , in $h$ (hour)
$\Gamma_i^{\text{up}}$	Minimum up time of unit $i$ , in hour
$\tau_{it}^{\text{off}}$	Continuously off time of unit $i$ up to time $t$
$\tau_{it}^{\text{on}}$	Continuously on time of unit $i$ up to time $t$
$N_{\text{pop}}$	Number of chromosomes in population (population size)
$\sigma_i$	The initial status of the unit $i$