Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2016) 24: 3862 – 3876

doi:10.3906/elk-1409-53

A ring crossover genetic algorithm for the unit commitment problem

Syed Basit Ali BUKHARI*, **Aftab AHMAD**, **Syed Auon RAZA**, **Muhammad Noman SIDDIQUE**
Department of Electrical Engineering, University of Engineering and Technology, Taxila, Pakistan

Received: 09.09.2014 • **Accepted/Published Online:** 21.06.2015 • **Final Version:** 20.06.2016

Abstract: The unit commitment problem (UCP) is a nonlinear, mixed-integer, constraint optimization problem and is considered a complex problem in electrical power systems. It is the combination of two interlinked subproblems, namely the generator scheduling problem and the generation allocation problem. In large systems, the UCP turns out to be increasingly complicated due to the large number of possible ON and OFF combinations of units in the power system over a scheduling time horizon. Due to the insufficiency of conventional approaches in handling large systems, numerous metaheuristic techniques are being developed for solving this problem. The genetic algorithm (GA) is one of these metaheuristic methods. In this study, an attempt is made to solve the unit commitment problem using the GA with ring crossover and swap mutation operators. One half of the initial population is intelligently generated by focusing on load curve to obtain better convergence. Lambda iteration is used to solve the generation allocation subproblem. Tests are carried out on systems with up to 100 generators over a time horizon of 24 h. Test outcomes demonstrate the proficiency of the presented scheme when compared with previously used techniques.

Key words: Unit commitment, genetic algorithm, repair mechanism, lambda iteration method

1. Introduction

The unit commitment problem (UCP) in the power industry is very crucial, because a good unit commitment schedule can save millions of dollars annually in fuel and other costs. Unit commitment (UC) is related to the economic use of generation resources in the power industry. In brief, the UCP is how to prepare the ON/OFF schedule for generators, along with optimal dispatch of ON units, in such a way so as to minimize the total cost of the system over a specified time period, while simultaneously satisfying various plant and system constraints.

An exact UC schedule can be achieved by complete enumeration, which is not practical because it requires excessive computational time. Thus, various approaches, including dynamic programming, Lagrangian relaxation, tabu search, ant colony optimization, particle swarm optimization, and the genetic algorithm (GA), have been developed to solve this problem. A literature review of the UCP and its solution approaches was given in [1]. The techniques used to solve the UCP can be divided into three categories: deterministic approaches, metaheuristic approaches, and hybridized techniques. Extensively used conventional/deterministic techniques are the priority list approach [2], mixed integer programming [3], Lagrangian relaxation [4], dynamic programming [5], and branch-and-bound [6]. The priority list method is nimble and simple, but produces solutions that are far away from the optimum point. Dynamic programming has a problem of high dimensionality, i.e. as the problem size increases, the computational time increases drastically with the number of committed generators.

*Correspondence: s.basita41@yahoo.com

Lagrangian relaxation produces a fast solution but has a problem of local convergence. Branch-and-bound methods fail on large-scale power systems because the CPU time increases enormously.

Deterministic approaches cannot handle large systems and nonlinear minimum up time (MUT) and minimum down time (MDT) constraints; therefore, many metaheuristics are examined, including expert systems [7,8], evolutionary programming (EP) [9,10], artificial neural networks (ANNs) [11], the GA [12–18], tabu search [19,20], particle swarm optimization (PSO) [21], simulated annealing (SA) [22,23], fuzzy logic [24], differential evolution (DE) [25,26], and ant colony optimization (ACO) [27,28]. These are population-based approaches and can produce global optimum or near-optimum solutions for systems having large generating units while handling all types of complex constraints with ease. SA is based on the principle of slow heating and cooling of metals. It is a powerful stochastic method used in optimization problems, but requires large computational time. The tabu search method iteratively searches for the best solution and uses a flexible memory system to prevent entrapment at the local best minimum solution. GA, PSO, ACO, and EP are nature-inspired techniques and can find global optimal solutions. Recently, the metaheuristic approaches have been hybridized with other metaheuristics or classical techniques to solve this problem more efficiently [20,29–32]. Various approaches have been used to solve the UCP, but none of them have been accepted as the best one so far.

The GA is widely used for solving optimization problems because it is simple and function-independent. Various GA-based techniques have been used to solve the UCP. A literature survey of various GA-based techniques for the UCP was given in [17]. In this paper, a ring crossover GA along with lambda iteration is used to solve the problem. Various developed operators within the proposed scheme are employed to obtain better results and more economy with reduced computational time. The proposed scheme is implemented in systems with up to 100 generators. Finally, a comparison is made with previously developed techniques under similar operating conditions.

2. UCP formulation

The UCP is a mixed-integer problem, i.e. it involves both binary and real variables. For scheduling total generation, it is imperative to find the optimal allocation of power levels among the committed generators. Various constraints in the UCP must also be considered to obtain feasible solution. The objective function and UCP constraints are defined below.

2.1. Objective function

Minimization of the total production cost of the system over the scheduling time horizon, subject to various unit and system constraints, is the main objective of the UCP. It is mathematically formulated as:

$$Min(TPC) = \sum_{i=1}^{T}\sum_{i=1}^{N}[F_i(P_i^t)u_i^t + C_{\mathbf{s.up(i,t)}}\left(1-u_i^{t-1}\right)u_i^t + C_{\mathbf{s.down(i,t)}}\left(1-u_i^t\right)u_i^{t-1}] \tag{1}$$

Here, TPC represents the total production cost of the system and $F_i\left(P_i^t\right)$ is the fuel cost. $C_{s.up(i,t)}$ and $C_{s.down(i,t)}$ represent the start-up and shut-down costs of unit i at time t.

The fuel cost of a thermal unit is given as:

$$F_i(P_i^t) = \alpha_i + \beta_i P_i^t + \gamma_i\left(P_i^t\right)^2 \tag{2}$$

Here α, β and γ are the cost coefficients.

The unit start-up cost depends on how long the unit remains off prior to start-up. In this study, start-up cost $C_{s.up(i,t)}$ is computed as follows:

$$C_{s,up(i,t)} = \begin{cases} \partial, & \text{if} \quad X_{i,off}^{t} \leq \tau_i \\ \sigma, & \text{if} \quad X_{i,off}^{t} \geq \tau_i \end{cases} \tag{3}$$

Here, $\partial and \sigma$ are the hot-start and cold-start costs, respectively, and τ_i represents the cold-start hours of unit i. $X_{i,on}^{t}$ and $X_{i,off}^{t}$ represent the duration of on- and off-time for unit i at time t respectively.

2.2. Constraints

2.2.1. Spinning reserve requirement

Some extra power, known as spinning reserve, should be available in the system to maintain system reliability. The summation of maximum power of all committed units must be at least equal to the sum of load demand and spinning reserve.

$$\sum_{i=1}^{N} P_i^{max} u_i^t = D_t + R_t \tag{4}$$

2.2.2. System power balance

The sum of the power generated from all committed units must meet the load demand at time t:

$$\sum_{i=1}^{N} P_i^t . u_i^t = D_t \tag{5}$$

where D_t is the load demand in MW.

2.2.3. MUT and MDT

Once a unit is committed, it should not be turned off before a specific time, called MUT.

$$For u_i^t = 1; X_{i,on}^t \geq MUT_i \tag{6}$$

Similarly, if a unit is turned off, it should not be recommitted before a specific time, known as MDT.

$$For u_i^t = 0; X_{i,off}^t \geq MDT_i \tag{7}$$

2.2.4. Unit power generation range

Power generated by each unit must be within its minimum and maximum power generation limits.

$$\forall i, t : P_i^{min} \leq P_i^t \leq P_i^{max} \tag{8}$$

Here P_i^{min} is the minimum and P_i^{max} is the maximum power of unit i.

2.2.5. Ramp rate limits

Due to the restrictions of thermal units, the rate of power changes between adjacent hours must be within certain limits:

$$P_i^t - P_i^{(t-1)} \leq UR_i \tag{9}$$

$$P_i^{t-1} - P_i^t \leq DR_i \tag{10}$$

where P_i^t and P_i^{t-1} are the power outputs of unit i at time t and $t-1$ respectively. UR_i and DR_i are the ramp-up and ramp-down rates, respectively.

3. Overview of the GA

GAs [12–16,33,34] are robust search methods inspired by natural genetics and the natural selection process in similar competitive environments. Individuals having the best adjustment to the environment tend to transfer their genetic information to the next generations. GAs are widely used to solve different optimization problems in engineering as well as in other fields.

GAs have an advantage over other optimization techniques, in that they only need an objective function for the problem. By using a GA, the solutions' search process is separated from the objective function characteristics and its associated constraints.

A GA begins with the random generation of chromosomes, or solutions, which gives the initial population. The quality of each chromosome is determined by evaluating the fitness of each individual for a given objective function. After the fitness evaluation, individuals are subjected to a process of selection in which best-fit individuals have more chances of being selected as parents. Once the parents are selected, genetic information is exchanged between these parents. This information exchange in genetics is called crossover and newly created individuals are called offspring. After crossover, some of these offspring are subjected to a random change called mutation. The mutation process introduces diversity among the chromosomes in the population. After crossover and mutation, a new population is created that replaces the old population. This process is repeated until the convergence criterion is met. Each such cycle is called a generation.

4. Proposed GA approach to solve the UCP

The proposed GA for a UCP solution consists of the following.

4.1. Solution encoding

Binary elements are adopted for UCP solution encoding; therefore, a set of bits is used to represent each UCP solution. A "1" at a certain position shows that the unit is ON at that specific hour, whereas a "0" represents that the unit is OFF at that specific hour. Hence, a chromosome to represent the status of N generators over a time horizon of T hours is an array of dimensions T × N, as shown in Figure 1.

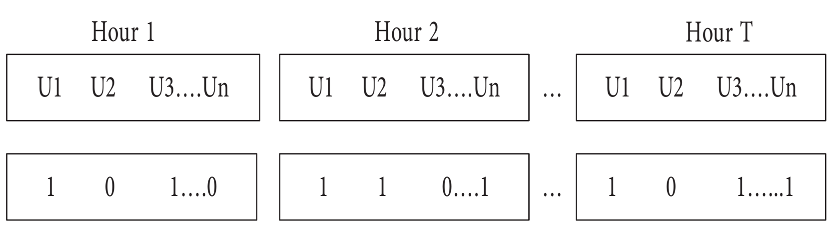

Figure 1. Solution encoding.

4.2. Creation of initial population

It is challenging to create a feasible solution when individuals in initial populations are created at random. Randomly generated solutions are usually far away from the optimum solution; as a result, convergence is slow and there is a greater chance of getting stuck in a local minimum. Therefore, the generation of individuals in the initial population is accomplished intelligently by focusing on the system load curve. Usually, more units are committed at peak load hours, whereas a small number of units are started up at light load. A full-load average production cost (FLAPC)-based priority order is used to commit the units. The FLAPCs of units are obtained by using the following expression:

$$\emptyset_i = \alpha_i / P_i^{max} + \beta_i + \gamma_i P_i^{max} \tag{11}$$

Here $\emptyset_i$ represents the FLAPC for unit i.

To maintain diversity in the population, half of the individuals are still generated at random.

4.3. Fitness evaluation

Fitness evaluation plays a vital role in how a GA obtains an optimal solution within a large search space. An individual with a good fitness value will help the GA explore the search space more efficiently and effectively. The fitness value of each individual is evaluated by using some type of function. For the UCP, the fitness of each chromosome in the population is computed by using Eq. (1).

4.4. Tournament selection

The objective of selection in a GA is to provide higher reproductive chances to those individuals with better fitness. A selection operator is used to select individuals for mating pools. It makes a genetic copy of good chromosomes, but does not produce new individuals. This work uses tournament selection, in which two or more individuals are randomly compared and the individual with the best fitness function is copied to form a temporary population, called mating pools. This procedure is repeated until the size of the mating pool equals the dimensions of the initial population.

4.5. Ring crossover

Crossover is used to interchange genetic information between two selected parents from mating pools.

The crossover operator is implemented on mating pools to produce better offspring. In this study, ring crossover is used for exchanging genetic information. Steps to implement this operator are:

Step 1: Two parents from the mating pools population are randomly chosen for crossover, as shown in Figure 2a.

Step 2: The parents are first combined in a ring form, then a cutting point is randomly decided around the ring, as illustrated in Figure 2b.

Step 3: A point is selected in the ring in such a way that the length of the string between the randomly chosen cutting point and this point is the same as that of the parent chromosomes, as shown in Figure 2c. One offspring is produced in a counterclockwise direction and the other is created in a clockwise direction, as demonstrated in Figure 2d.

Since ring length is equal to the total length of both of the mating parents, and the offspring are created by deciding a random point in the ring, more diversity in the offspring can be obtained.

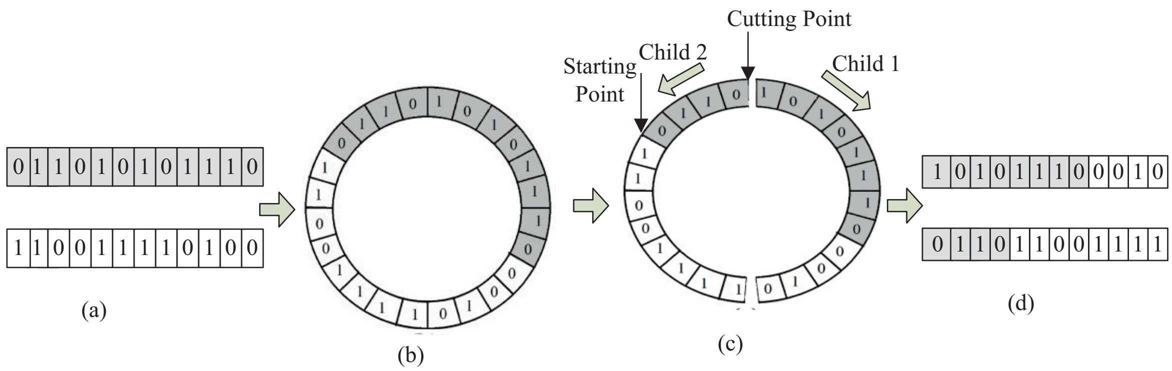

Figure 2. Ring crossover.

4.6. Swap mutation

Mutation randomly disturbs genetic information and plays a vital role in recovering lost genetic information. Mutation helps the exploration of the entire search space and provides assurance against the trapping of the algorithm at a local minimum. A probability Pm is determined for mutation. In swap mutation, two points are randomly selected in a chromosome and the order of the string between these points is reversed, as shown in Figure 3.

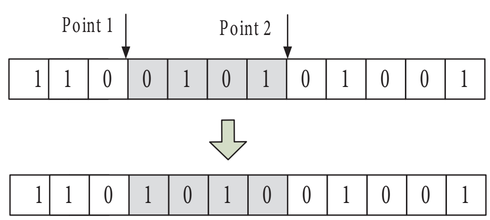

Figure 3. Swap mutation.

4.7. Elitism

To avoid genetic information loss, individuals in the parent population having the best fitness are copied in a group and transferred to the next generation. This elitist scheme increases the speed of convergence.

5. Repair mechanism

The exponential increase in the UCP search space with the increasing number of generating units to be scheduled makes the UCP a complicated combinatorial optimization problem. The algorithm search may introduce infeasibility in solutions, so a repair strategy is required that forcibly satisfies the associated constraints of the problem. While using these repair mechanisms, a penalty-less fitness function is used.

5.1. Repair mechanism for spinning reserve

The population produced using a GA may not satisfy the spinning reserve in Eq. (4) at some time instants. Thus, system spinning reserve violations must be repaired using some heuristic method [35].

The procedure for repairing spinning reserve involves the following steps:

Step 1. Set $t = 1$.

Step 2. Calculate the FLAPC for all uncommitted units using Eq. (11), where $\emptyset_i$ represents the FLAPC for unit i.

Step 3. Calculate the extra reserve at each hour by using expression given below:

$$R_{extra} = \sum_{i=1}^{N} P_i^{\max} u_i^t - D_t - R_t \tag{12}$$

If $R_{extra} \geq 0$, go to step 5;

Step 4. Else commit the uncommitted unit with the lowest $\emptyset_i$ and return to Step 3.

Step 5. If $t < T$, set $t = t + 1$ and return to Step 2. Else, stop.

5.2. Repairing the MUT and MDT

As the GA is a stochastic search process, it creates populations that may have MUT/MDT violations. At peak load, the MUT of units is violated because peak load hours are usually less than the MUT of units. Similarly, MDT violation usually occurs at off-peak load, during which the off-peak load hours are less than the MDT of units. The minimum up/down time will be repaired by using a heuristic search algorithm [35]. To check violations, first the durations of on and off time for all units are calculated by using following expressions:

$$X_{i,on}^t = \begin{cases} X_{i,on}^t + 1, & \text{if} \quad u_i^t = 1 \\ 0, & \text{if} \quad u_i^t = 0 \end{cases}, X_{i,off}^t = \begin{cases} X_{i,off}^t + 1, & \text{if} \quad u_i^t = 0 \\ 0, & \text{if} \quad u_i^t = 1 \end{cases} \tag{13}$$

Steps used for minimum up/down time repairing are given as follows:

Calculate the duration of on and off time for all units using Eq. (13).

Step 1. Set $t = 1$

Step 2. Set $i = 1$.

Step 3. If $u_i^t = 0$, $u_i^{t-1} = 1$, and $X_{i,on}^t < MUT_i$, then set $u_i^t = 1$.

Step 4. If $u_i^t = 1$, $u_i^{t-1} = 1$ and $X_{i,off}^t < MDT_i$.

4a. If $t + T_i^{down} \leq T$ then set $u_i^t = 1$.

4b. If $t + T_i^{down} > T$ and $\sum_{n=t}^{T} u_i^n > 0$ then set $u_i^t = 1$.

Step 5. Modify the on/off times duration of units using Eq. (13).

Step 6. If $i < N$, set $i = i + 1$ and go back to Step 4.

Step 7. If $t < T$, set $t = t + 1$ and return to Step 3. Else, stop.

5.3. Decommitment of excess units

Repairing MUT and MDT constraints can lead to excessive spinning reserves, which is not desirable due to the high operation cost. A heuristic method is used to turn off the excessive units one by one based on decreasing order of FLAPCs, as calculated by Eq. (11), until the system spinning reserve of Eq. (4) is just satisfied at any time. However, unit decommitment is subject to MUT/MDT satisfaction; i.e. the unit is turned off only if this decommitment does not violate the up/down time of that unit.

5.4. Lambda iteration method for economic load dispatch calculation

For each feasible unit commitment schedule, the lambda iteration method is used to solve the economic dispatch subproblem [35].

Once optimal allocation of powers is found, the total operating cost at any time T is obtained by calculating the sum of operating costs of all online units. Total startup cost at any hour T is computed by taking the sum of the startup costs of those generating units that update their states from 0 to 1.

6. Implementation of the improved GA model

To find the optimal UC schedule for a given time horizon, the proposed GA is implemented. The implementation of this model involves various steps, which are given below:

Step 1. Call unit and load-related data.

Step 2. Initialize the chromosomes in the population for the first generation, as described in Section 4.2.

Step 3. After the chromosomes initialize, the population may violate different constraints. The status of units obtained after the first generation are updated, as described in Sections 5.1 and 5.2.

Step 4. Decommit excessive generating units for all chromosomes in the population to diminish extra reserve, as in Section 5.3.

Step 5. Solve the economic dispatch problem using the lambda iteration method, as described in Section 5.4.

Step 6. Evaluate the fitness of each individual by means of the objective function given in Eq. (1). Compare the fitness values of each individual; the individual that possesses the best fitness value is considered the global best.

Step 7. Apply selection, elitism, crossover, and mutation respectively on individuals of the population to create the population for the next generation.

Step 8. If the maximum number of generations has been reached, then the optimal UC schedule is the schedule in the final population that gives the minimum total production cost. Else, increase generation number and return to Step 2.

7. Numerical results and discussions

To verify the effectiveness and feasibility of the proposed model, the proposed ring crossover GA was tested on five IEEE test systems and a practical Taipower system. The scheduling time horizon was selected as 1 day with 24 intervals of 1 h each. The C programming language was used to code the proposed algorithm. The code was executed on a personal computer with 1.8 GHz CPU, Windows 7, and 2 GB RAM.

Better convergence can be achieved by properly selecting the control parameters, namely crossover rate, mutation rate, and population size. The optimal setting of these GA parameters not only gives a better solution but also reduces computational time. In this work, crossover and mutation probabilities were selected using a self-adaptive scheme. These probabilities were evaluated by focusing on the best fitness values of current and previous generation, and 0.6 and 0.09 were selected as initial values for crossover and mutation rates, respectively. The population size varied between 40 and 150.

8. IEEE test cases

IEEE 10-, 20-, 40-, 60-, 80-, and 100-unit standard test cases were considered for UC scheduling. In all test cases, a ring crossover GA was used to obtain the ON/OFF schedule of units. The unit data and load-related profile for a 10-unit test system were taken from [12]. The scheduling time horizon was selected as 1 day with 24 time intervals of 1 h each. The spinning reserve at any hour was taken as 10% of the forecasted load demand at that hour. The data for the remaining systems were obtained by suitably scaling data from the 10-unit system. The GA is a stochastic search technique; therefore, for every test system, the solution was calculated 25 different times with a different initial population and 500 generations.

Under the selected parameters, the ring crossover GA was first implemented for the 10-unit test system and 25 independent runs were made with different initial populations, in which the best schedule had a production cost of $563,937. The detailed schedule of the 10-unit test, along with the optimal dispatch of each unit, is given in Table 1. As the data for other systems were produced by suitably scaling the 10-unit system data, the obtained schedule for the 10-unit system was also scaled properly and considered as one of the primary solutions during each run of the other five systems. Table 2 shows different results for all test systems, which were obtained from 25 independent runs for each system. From Table 2, it is clear that the variation in average production costs is small and the standard deviations are small and tolerable. The convergence processes of the best solution in the 25 runs for different systems are shown in Figures 4–9. These figures show that the ring crossover GA has satisfactory convergence characteristics and the algorithm does not trap at any local minima.

Table 1. UC schedule along with power dispatch of each unit for the 10-unit test system.

Hour	Unit status	Power generated in MW										Total 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	13,683	0
2	1100000000	455	295	0	0	0	0	0	0	0	0	14,554	0
3	1100100000	455	370	0	0	25	0	0	0	0	0	16,809	900
4	1100100000	455	455	0	0	40	0	0	0	0	0	18,598	0
5	1101100000	455	390	0	130	25	0	0	0	0	0	20,020	560
6	1111100000	455	360	130	130	25	0	0	0	0	0	22,387	1100
7	1111100000	455	410	130	130	25	0	0	0	0	0	23,262	0
8	1111100000	455	455	130	130	30	0	0	0	0	0	24,150	0
9	1111111000	455	455	130	130	85	20	25	0	0	0	27,251	860
10	1111111100	455	455	130	130	162	33	25	10	0	0	30,058	60
11	1111111110	455	455	130	130	162	73	25	10	10	0	31,916	60
12	1111111111	455	455	130	130	162	80	25	43	10	10	33,890	60
13	1111111100	455	455	130	130	162	33	25	10	0	0	30,058	0
14	1111111000	455	455	130	130	85	20	25	0	0	0	27,251	0
15	1111100000	455	455	130	130	30	0	0	0	0	0	24,150	0
16	1111100000	455	310	130	130	25	0	0	0	0	0	21,514	0
17	1111100000	455	260	130	130	25	0	0	0	0	0	20,642	0
18	1111100000	455	360	130	130	25	0	0	0	0	0	22,387	0
19	1111100000	455	455	130	130	30	0	0	0	0	0	24,150	0
20	1111111100	455	455	130	130	162	33	25	10	0	0	30,058	490
21	1111111000	455	455	130	130	85	20	25	0	0	0	27,251	0
22	1111111000	455	455	0	0	145	20	25	0	0	0	22,736	0
23	1100010000	455	425	0	0	0	20	0	0	0	0	17,645	0
24	1100000000	455	345	0	0	0	0	0	0	0	0	15,427	0

Table 2. Statistical analysis of results obtained from 25 independent runs for each system.

Number of units	Best cost ($)	Worst cost ($)	Average cost ($)	Standard deviation ($)
10	563,937	564,219	564,019	17
20	1,123,297	1,124,537	1,123,851	48
40	2,242,887	2,244,117	2,243,569	75
60	3,365,337	3,366,873	3,366,052	84
80	4,486,991	4,487,949	4,487,476	98
100	5,606,663	5,607,850	5,607,088	113

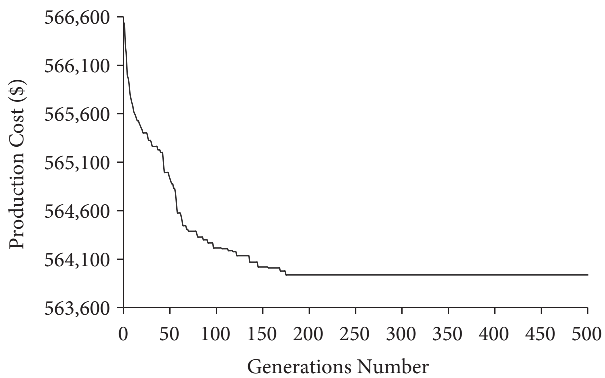

Figure 4. Convergence characteristics for a 10-unit system.

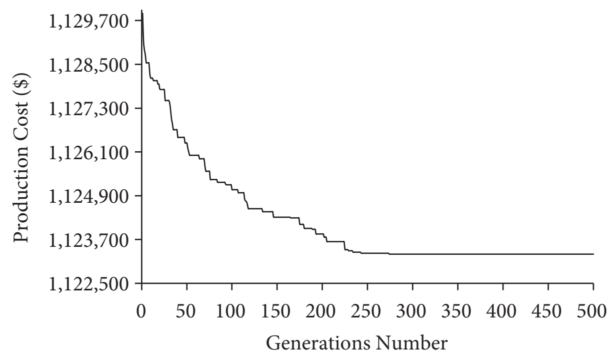

Figure 5. Convergence characteristics for a 20-unit system.

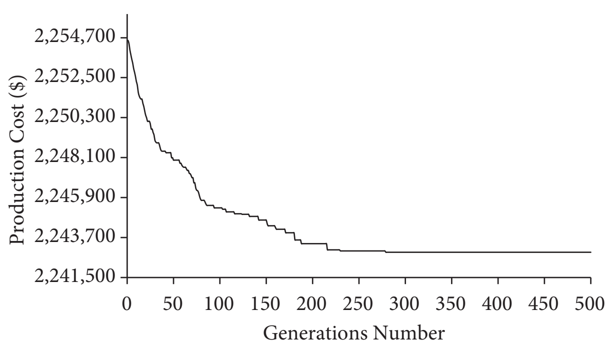

Figure 6. Convergence characteristics for a 40-unit system.

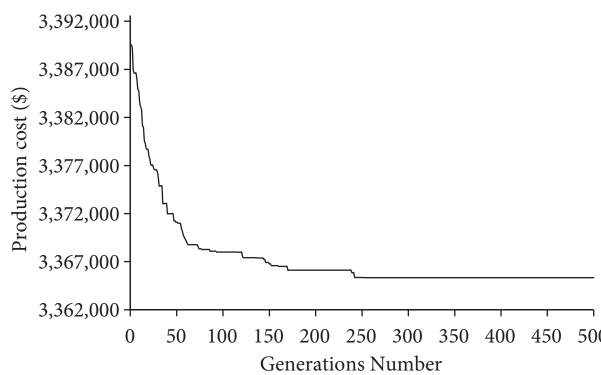

Figure 7. Convergence characteristics for a 60-unit system.

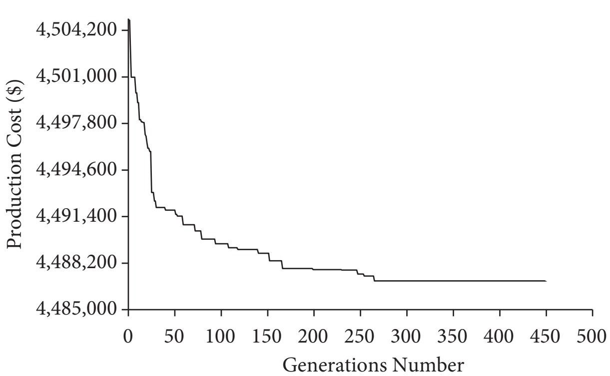

Figure 8. Convergence characteristics for an 80-unit system.

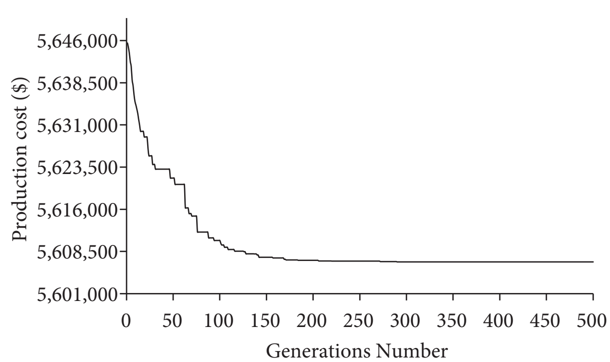

Figure 9. Convergence characteristics for a 100-unit system.

To validate the performance of the improved GA, the best solution of each system is compared with other metaheuristic approaches. The best results obtained from the proposed algorithm were compared with UCGA [36], DE [25], FPGA [33], IPSO [37], GA [38], ICA [39], and BRGA [12]. The comparisons are given in Table 3. From this table, it is clear that the proposed ring crossover GA produces better results in terms of production cost for small as well as large systems.

Table 3. Comparison of the best solutions with other metaheuristic approaches.

Method	10-unit	20-unit	40-unit	60-unit	80-unit	100-unit
UCGA [40]	563,977	1,125,516	-	-	-	-
DE [25]	563,938	1,124,291	2,246,274	3,365,784	4,488,450	5,607,900
FPGA [33]	564,094	1,124,998	2,248,235	3,368375	4,491,169	5,614357
IPSO [41]	563,954	1,125,279	2,248,163	3,370,979	4,495,032	5,619,284
GA [42]	564,368	1,124,893	2,245,827	3,368,537	4,497,871	5,622,746
ICA [39]	563,938	1,124,274	2,247,078	3,371,722	4,497,919	5,617,913
BRGA [43]	563,937	1,124,290	2,246,165	3,365,431	4,487,766	5,606,811
Proposed GA	563,937	1,123,297	2,242,887	3,365,337	4,486,991	5,606,663

9. The 38-unit Taipower system

The generator data and load profile for this system were taken from [14]. The system was executed under same conditions taken by [14]; i.e. the spinning reserve at any hour was taken as 11% of the forecasted load demand at that hour. Ramp rate limits were also taken into account for this system. The program was executed 25 times independently, with different initial populations, for 500 generations. Out of 25 independent runs, the best operating cost achieved for this system over 24 h was $197,065,470. The detailed schedule, along with total operating cost, is shown in Table 4. To verify the performance of the proposed algorithm for this system, the best cost obtained from the ring crossover GA was compared with other techniques. This comparison is shown in Table 5. It is evident that the proposed algorithm produces better results than the other techniques. The convergence process of the best solution in 25 runs is shown in Figure 10.

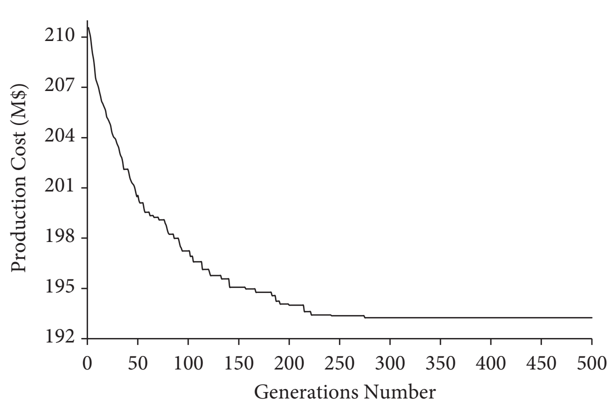

Figure 10. Convergence characteristics of a 38-unit Taipower system.

Table 4. UC schedule along with cost for the 38-unit system.

Hour	Unit status 1 ... 38	Total operating cost ($)
1	11111111010010000001111011000000000010	6,452,322.2
2	11111111010010000001111011000000000010	6,091,785.8
3	11111111010010000001111011000000000010	5,818,991
4	11111111010010000001111011000000000010	5,494,396.9
5	11111111010010000001111011000000000010	5,602,777.5
6	11111111010010000001111011000000000010	5,445,769.1
7	11111111010010000001111011000000000010	5,495,104.4
8	11111111010010100001111011000000000010	6,775,942
9	11111111111111110001111011000000000010	10,754,824
10	11111111111111111001111011000000000010	10,036,950
11	11111111111111111001111011000000000010	10,190,996
12	11111111111111111001111011000000000010	10,337,713
13	11111111111111111001111011000000000010	8,667,803.6
14	11111111111111111011111011000000000010	10,722,450
15	11111111111111111001111011000000000010	10,569,153
16	11111111111111111001111011000000000010	10,193,487
17	11111111111111111001111011000000000010	9,885,858.5
18	11111111110111111001111011000000000010	8,812,379.7
19	11111111110111101001111011000000000010	8,325,098.5
20	11111111110111101001111011000000000010	9,019,449.3
21	11111111110110101001111011000000000010	8,679,161.1
22	11111111110110001001111011000000000010	8,217,181.7
23	11111111110110001001111011000000000010	7,859,428.2
24	11111111010110001001111011000000000010	7,633,658.3
	Total Cost($)	197,081,350

Table 5. Cost comparison of results with the 38-unit system.

Technique	Operating cost (M$)
MRCGA [14]	206.70
MACO [44]	203.32
FAPSO [45]	199.07
ASSA [46]	198.84
HSSA [47]	197.41
Proposed GA	197.081

10. Conclusions

The UCP is critical in the operational planning of a power system. A good UC schedule has the ability to save millions of dollars in fuel and other related costs. Due to the high dimensionality and combinatorial nature of the UCP, it is challenging to develop any rigorous optimization method that has the ability to solve the whole problem for any real-sized system. Various solution techniques are being examined in order to solve this problem. In this study, a GA with ring crossover and swap mutation was presented. The chromosomes in the initial population were produced based on the system load curve to attain better convergence. Ring crossover and swap mutation introduce more diversity among chromosomes of a population and therefore prevent the trapping of the algorithm at local optima. The proposed algorithm was implemented on six test cases (up to

100 generators) and on a practical 38-unit system over a scheduling horizon of 24 h, and it produced noteworthy results as compared to other approaches.

References

[1] Saravanan B, Das S, Sikri S, Kothari D. A solution to the unit commitment problem—a review. Frontiers in Energy 2013; 7: 223-236.

[2] Senjyu T, Miyagi T, Saber AY, Urasaki N, Funabashi T. Emerging solution of large-scale unit commitment problem by stochastic priority list. Electr Pow Syst Res 2006; 76: 283-292.

[3] Delarue E, D'Haeseleer W. Adaptive mixed-integer programming unit commitment strategy for determining the value of forecasting. Appl Energ 2008; 85: 171-181.

[4] Ongsakul W, Petcharaks N. Unit commitment by enhanced adaptive Lagrangian relaxation. IEEE T Power Syst 2004; 19: 620-628.

[5] Singhal PK, Sharma RN. Dynamic programming approach for large scale unit commitment problem. In: IEEE 2011 International Conference on Communication Systems and Network Technologies; 3–5 June 2011; Katra, Jammu, India. New York, NY, USA: IEEE. pp. 714-717.

[6] Chen CL, Wang SC. Branch-and-bound scheduling for thermal generating units. IEEE T Power Syst 1993; 8: 184-189.

[7] Ouyang Z, Shahidehpour S. Short-term unit commitment expert system. Electr Pow Syst Res 1990; 20: 1-13.

[8] Kothari D, Ahmad A. An expert system approach to the unit commitment problem. Energ Convers Manage 1995; 36: 257-261.

[9] Rajan CCA, Mohan M. An evolutionary programming-based tabu search method for solving the unit commitment problem. IEEE T Power Syst 2004; 19: 577-585.

[10] Juste K, Kita H, Tanaka E, Hasegawa J. An evolutionary programming solution to the unit commitment problem. IEEE T Power Syst 1999; 14: 1452-1459.

[11] Wong MH, Chung TS, Wong YK. An evolving neural network approach in unit commitment solution. Microprocess Microsy 2000; 24: 251-262.

[12] Datta D. Unit commitment problem with ramp rate constraint using a binary-real-coded genetic algorithm. Appl Soft Comput 2013; 13: 3873-3883.

[13] Dudek G. Genetic algorithm with binary representation of generating unit start-up and shut-down times for the unit commitment problem. Expert Syst Appl 2013; 40: 6080-6086.

[14] Sun L, Zhang Y, Jiang C. A matrix real-coded genetic algorithm to the unit commitment problem. Electr Pow Syst Res 2006; 76: 716-728.

[15] Dudek G. Unit commitment by genetic algorithm with specialized search operators. Electr Pow Syst Res 2004; 72: 299-308.

[16] Swarup K, Yamashiro S. Unit commitment solution methodology using genetic algorithm. IEEE T Power Syst 2002; 17: 87-91.

[17] Bukhari SBA, Ahmad A, Raza SA, Zaki AUM. Genetic algorithm based generator scheduling - a review. World Applied Sciences Journal 2014; 30: 1826-1833.

[18] Ahmad A, Malik T. Unit commitment problem of thermal generation units for short term operational planning using simple genetic algorithm. Journal of the Institution of Electrical and Electronic Engineers Pakistan 2006; 52: 22-26.

[19] Rajan C, Mohar R. An evolutionary programming-based tabu search method for solving the unit commitment problem. IEEE T Power Syst 2004; 19: 557-585.

[20] Mantawy AH, Abdel-Magid YL, Selim SZ. A new genetic-based tabu search algorithm for unit commitment problem. Electr Pow Syst Res 1999; 49: 71-78.

[21] Ting T, Rao M, Loo C. A novel approach for unit commitment problem via an effective hybrid particle swarm optimization. IEEE T Power Syst 2006; 21: 411-418.

[22] Mantawy A, Abdel-Magid YL, Selim SZ. A simulated annealing algorithm for unit commitment. IEEE T Power Syst 1998; 13: 197-204.

[23] Simopoulos DN, Kavatza SD, Vournas CD. Unit commitment by an enhanced simulated annealing algorithm. IEEE T Power Syst 2006; 21: 68-76.

[24] Saneifard S, Prasad NR, Smolleck HA. A fuzzy logic approach to unit commitment. IEEE T Power Syst 1997; 12: 988-995.

[25] Datta D, Dutta S. A binary-real-coded differential evolution for unit commitment problem. Int J Elec Power 2012; 42: 517-524.

[26] Uyar AŞ, Türkay B, Keleş A. A novel differential evolution application to short-term electrical power generation scheduling. Int J Elec Power 2011; 33: 1236-1242.

[27] Simon SP, Padhy NP, Anand R. An ant colony system approach for unit commitment problem. Int J Elec Power 2006; 28: 315-323.

[28] Vaisakh K, Srinivas LR. Evolving ant colony optimization based unit commitment. Appl Soft Comput 2011; 11: 2863-2870.

[29] Nayak R, Sharma JD. A hybrid neural network and simulated annealing approach to the unit commitment problem. Comput Electr Eng 2000; 26: 461-477.

[30] Sadati N, Hajian M, Zamani M. Unit commitment using particle swarm-based-simulated annealing optimization approach. In: IEEE 2007 Swarm Intelligence Symposium; 1–5 April 2007; Honolulu, HI, USA: IEEE. pp. 297-302.

[31] Patra S, Goswami S, Goswami B. Fuzzy and simulated annealing based dynamic programming for the unit commitment problem. Expert Syst Appl 2009; 36: 5081-5086.

[32] Kumar SS, Palanisamy V. A dynamic programming based fast computation Hopfield neural network for unit commitment and economic dispatch. Electr Pow Syst Res 2007; 77: 917-925.

[33] Dang C, Li M. A floating-point genetic algorithm for solving the unit commitment problem. Eur J Oper Res 2007; 181: 1370-1395.

[34] Pavez-Lazo B, Soto-Cartes J. A deterministic annular crossover genetic algorithm optimisation for the unit commitment problem. Expert Syst Appl 2011; 38: 6523-6529.

[35] Yuan X, Su A, Nie H, Yuan Y, Wang L. Application of enhanced discrete differential evolution approach to unit commitment problem. Energ Convers Manage 2009; 50: 2449-2456.

[36] Senjyu T, Yamashiro H, Uezato K, Funabashi T. A unit commitment problem by using genetic algorithm based on unit characteristic classification. In: IEEE 2002 Power Engineering Society Winter Meeting; 27–31 January 2002. New York, NY, USA: IEEE. pp. 58-63.

[37] Zhao B, Guo CX, Bai BR, Cao YJ. An improved particle swarm optimization algorithm for unit commitment. Int J Elec Power 2006; 28: 482-490.

[38] Senthil Kumar V, Mohan MR. Solution to security constrained unit commitment problem using genetic algorithm. Int J Elec Power 2010; 32: 117-125.

[39] Moghimi Hadji M, Vahidi B. A solution to the unit commitment problem using imperialistic competition algorithm. IEEE T Power Syst 2012; 27: 117-124.

[40] Senjyu T, Yamashiro H, Shimabukuro K, Uezato K, Finabashi T. A fast solution technique for large scale unit commitment problem using genetic algorithm. In: IEEE 2002 Transmission and Distribution Conference and Exhibition; 6–10 October 2002. New York, NY, USA: IEEE. pp. 1611-1616.

[41] Zhao B, Guo C, Bai B, Cao Y. An improved particle swarm optimization algorithm for unit commitment. Int J Elec Power 2006; 28: 482-490.

[42] Senthil Kumar V, Mohan M. Solution to security constrained unit commitment problem using genetic algorithm. Int J Elec Power 2010; 32: 117-125.

[43] Datta D. Unit commitment problem with ramp rate constraint using a binary-real-coded genetic algorithm. Appl Soft Comput 2013; 13: 3873-3883.

[44] Saber AY, Alshareef AM. Scalable unit commitment by memory-bounded ant colony optimization with a local search. Int J Elec Power 2008; 30: 403-414.

[45] Saber A, Senjyu T, Yona A, Funabashi T. Unit commitment computation by fuzzy adaptive particle swarm optimisation. IET Gener Transm Dis 2007; 1: 456-465.

[46] Saber AY, Senjyu T, Miyagi T, Urasaki N, Funabashi T. Fuzzy unit commitment scheduling using absolutely stochastic simulated annealing. IEEE T Power Syst 2006; 21: 955-964.

[47] Saber A, Senjyu T, Miyagi T, Urasaki N, Funabashi T. Unit commitment by heuristics and absolutely stochastic simulated annealing. IET Gener Transm Dis 2007; 1: 234-243.