Generator Scheduling Model for Optimization using Genetic Algorithm with Multiparent Crossover (Ga-Mpc)

1stRudi Salman[1], 2ndIrfandi[2], 3rdArwadi[3], 4thSayuti Rahman[4]

{rudisalman@unimed.ac.id[1]}

Departement of Electrical Engineering Universitas Negeri Medan, Medan, North Sumatera[1,2,3]
Department of Information Technology, Universitas Harapan Medan, Medan,North Sumatera[4]

Abstract. Electric power systems are made and run so that they can meet the requirements of varying and growing electrical loads. The highest cost in operating an electric power system is the cost of fuel. For this reason, it is necessary to use optimization techniques to reduce these costs. Therefore, the optimization problem, namely minimizing the operating costs of the electric power system, is a significant issue. One of the efforts to reduce the operating costs of power plants is by optimizing the scheduling of power plants, in this case, generator scheduling. Generator scheduling aims to prepare a generator start-up (ON) and shut-down (OFF) schedule hourly to confer previously estimated load requirements while meeting specified constraints. Mathematically, the generator scheduling optimization problem is a very difficult nonlinear combinatorial optimization problem. Thus, to resolve this issue, one way can be to use a Genetic Algorithm with Multiparent Crossover (GA-MPC). A genetic algorithm is a method of random search that provides optimal solutions to optimization problems.
This study aims to build a generator scheduling optimization model using GA-MPC. The research was carried out at the Computer Laboratory of the Electrical Engineering Education Department, using the Matlab software version R2008 as a simulation tool. The IEEE standard electric power system with 5-Unit System was used for model testing.
The results showed that the generator scheduling model built using GA-MPC for generator scheduling optimization was successfully carried out. This situation is quite good and shows that GA-MPC can be implemented for generator scheduling optimization problems.

Keywords: Genetic Algorithm, Optimization, Generator Scheduling, Genetic Algorithm with Multiparent Crossover

1. Introduction

The electric power system is designed and operated to meet the needs of an increasing and varying electrical load on an ongoing basis [1]. To meet these load requirements, the generator units in the system must work in parallel [6]. The biggest cost that an electricity company must incur in operating a power system is the cost of generating fuel, namely the generator. So, it is necessary to use optimization techniques to reduce or reduce these costs [9]. Thus, minimizing the power systems operating costs is very important [1]. One way to reduce power system operating costs is by optimizing power generation schedules, which are referred to as generator scheduling. Generator scheduling aims to prepare the most economical generator start-up and shut-down schedule in an hourly time according to the needs of the previously estimated load and meet constraints that have been set, including minimizing fuel [2].

ICIESC 2022, October 11, Medan, Indonesia

DOI 10.4108/eai.11-10-2022.2325488

The generator scheduling optimization problem is a complex nonlinear combinatorial optimization problem from a mathematical perspective [1][2][14]. To solve these problems, several approaches have been taken, namely: a deterministic approach: branch and bound methods [3], dynamic programming [7], and integer programming [15], but these techniques require considerable computational time and computer memory. Then heuristic approaches, for example, dynamic programming modification and, Lagrangian relaxation [10], priority list [13], have been developed to reduce computational time and search space for solutions. However, this technique is still far from a globally optimal solution. Next, to get a globally optimal solution and a suitable computational time, metaheuristic techniques such as Genetic Algorithms were developed [4]., Simulated Annealing [8] and Particle Swarm Optimization [16].

This research focuses on using a Genetic Algorithm to build a power generation scheduling model. G.A. is a random search technique that is more likely to provide a near-optimal solution [2]. One of the weaknesses of G.A. is premature convergence, namely the achievement of local optimum values. This is caused when parent selection is only based on individual quality, so the best parent genetic information tends to dominate genetic characteristics in the population or lack diversity. In addition, the chromosome representation does not change during the crossover process. The crossover operator is the heart of GA [12], so the crossover operator must be able to exploit the search space information to produce new offspring.

Furthermore, offspring distribution should not be very narrow compared to their parents. This will cause a loss of diversity resulting in local optimum. This is because, on the other hand, if the distribution is too wide, it will cause the computation time to be slow. Finally, to produce suitable offspring, exploitation and exploration must be appropriate [5].

1.1. Genetic Algorithm in Scheduling of Generators

The first step for applying G.A. to the generator scheduling problem is to choose binary strings (chromosomes) to represent candidate solutions to the problem [16]. In this paper, chromosomes are created in the following way: if there are n number of generators in an electric power system. The scheduling period is h hours; then, with the assumption that at every hour, a certain generator can either be ON or OFF, the chromosomes used to represent the candidate solutions of the scheduling problem are n x h bits in length.

In such a chromosome, the first n bits describe the n generators ON-OFF states, a "1" at the first bit indicates that the first unit is ON at the first hour, while a "0" indicates that the unit is OFF. In some way, a chromosome of n x habits describes the operation states of n generators at each hour over the scheduling period of h hours. The binary representation of a candidate solution is shown below :

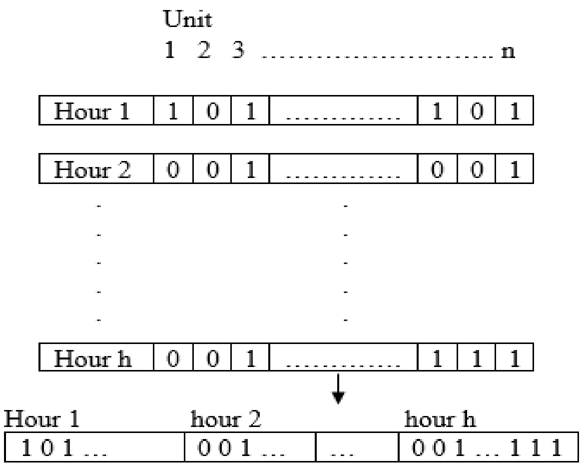

Fig.1 The binary representation of generator scheduling problem

The generators' scheduling problem aims to reduce total production costs over the scheduling period as much as possible. The creation costs comprise fuel, fire-up, and Shut-down costs. The sum of the costs associated with each generator is the total cost.
Start-up costs can be divided into two categories. The hot start cost and the cold start cost are these. The unit's hot start cost is considered if its downtime is less than or equal to its hard star hour. Otherwise, the cold start of the team is counted.

1.2 Objective function

The reduction of total costs is the goal of generator scheduling optimization. The purpose of generator scheduling optimization is to reduce startup and fuel costs :

$$\min TC \sum_{t=1}^{t} \sum_{n=1}^{n} \left(FC_i(t) + SC_i(t)\right) \tag{1}$$

The thermal unit's fuel cost is expressed as a second-order function of each output unit :

$$FC(t) = a_i + b_i P_i(t) + C_i P_i^2(t) \tag{2}$$

Start-up cost :

$$SC_i(t) = \begin{cases} SC_{h-i} : \ t_i^{off} \le X_i^{off}(t) \le \ H_i^{off} \\ \quad SC_{c-i} \ : \ X_i^{off}(t) > \ H_i^{off} \end{cases} \tag{3}$$

$$H_i^{off} = t_i^{off} + c - 8 - jam$$

1.3. Constraints

During the optimization process, the following constraints must be met. All committed units' generated power needs to meet load demand:

$$D(t) = \sum_{i=1}^{n} P_i\,(t) \tag{4}$$

A sufficient amount of spinning reverse is required to maintain system reliability:

$$\sum_{i=1}^{n} I_i(t)\, P_i^{max} \geq D(t) + R(t) \quad (5)$$

The output range of each unit is :

$$P_i^{min} \leq P_i(t) \leq P_i^{max} \quad (6)$$

A unit must be committed or decommitted at least once before it can be committed or decommitted again:

$$\begin{cases} t_i^{on} \leq X_i^{on}(t) \\ t_i^{off} \leq X_i^{off}(t) \end{cases} \quad (7)$$

where :

TC : Total cost
i : Index of unit (i=1,2,3,…,n)
t : Index of times (t = 1,2,3,…,t)
n : Number of unit
$FC_i(t)$: Fuel cost of i-th unit
$I_i(t)$: i-th unit status at hour t
$SC_i(t)$: Start-up cost of i-th unit
a_i, b_i, c_i : Fuel cost function coeffisients
$P_i(t)$: Output power of i-th unit at hour t
P_i^{max} : Maximum output power of i-th unit
P_i^{min} : Minimum output power of i-th unit
$D(t)$: Demand power at hour t
$R(t)$: System reverse at hour t
$X_i^{on}(t)$: Duration during which i-th unit is continuously ON
$X_i^{off}(t)$: Duration during which i-th unit is continuously OFF
t_i^{on} : Minimum up time of i-th unit
t_i^{off} : Minimum down time of i-th unit
c-8-hour : Cold start hour oh i-th unit

1.4. Genetic Algorithm with Multiparent Crossover (GA-MPC)

In GA-MPC, the initial population's population size (P.S.) is determined at random. The best m people are selected from an archive pool based on their fitness function and constraint violations. The best player is selected and placed in the selection pool following a tournament selection procedure with a size three crossover. For the hybrid activity, with a crossover rate (cr), for every three crossovers in the determination pool, three posterity are produced, as portrayed previously. To escape any local minima and reach a better region in the search space, we employ a diversity operator with a probability of p following the generation of all offspring. After that, we combine all the offspring with the people from the archive pool, and the best P.S. people are chosen to form a new population for the next generation[5]. Additionally, to increase diversity, if one person in the population is identical to another, one of them is moved within the boundary with N (0,5u, 0,25u), u [0,1].

The steps of GA-MPC are follows :

Step 1: Create an initial random population of Population size for generation t = 0. Each individual's (i) variables must fall within the following range:

$$x_i^j \ = L + \left[rand\ x\ \left(U_j - L\right)\right]$$

where : rand ε [0,1], $L_j \leq x_j \leq U_j, j = 1,2,3, \dots, D$

Step 2: The best m people should be saved in the archive pool (A) and sorted by constraint violations and objective function.

Step 3: Fill the selection pool (which includes all tournament winners) with a tournament selection with size Three Crossover (randomly 2 or 3). The size of the selection pool ought to be three times the population size.

Step 4: For each three consecutive individuals, if $u\ \epsilon\ [0,1] < cr$

a) Short these three individuals into $f(x_i) \leq f(x_{i+1}) \leq f(x_{x+2})$

b) Replace one of the selected individuals with a random individual from the selection pool if one of them is identical to another.

c) Calculate β= $N(\mu, \sigma)$,where $\mu = 0.7$, and $\sigma = 0.1$

d) Generate three offspring (o_i) :

$$o_1 = x_1 + [\beta\ x\ (x_2 - x_3)]$$
$$o_2 = x_2 + [\beta\ x\ (x_3 - x_1)]$$
$$o_3 = x_3 + [\beta\ x\ (x_1 - x_2)]$$

Step 5**:** For each $o_i'^j$ generate a random number ϵ [0,1], If $u\ \epsilon\ [0,1] < p$, then $o_i'^j \ = \ x_{arch}^i$, where integer arch $\epsilon\ [1, m]$.

Step 6 : If there is more than one individual, then

$$x_{i,jk} = \ x_{i,jk} + N(0{,}5\ x\ u, 0{,}25 * u), where\ u\ \epsilon[0,1]$$

Step 7: If the criteria for termination are met, stop; **else** set $t = t + 1$ and go to Step 2

1.5 The Diversity Operator

A diversity operator was developed in [5] to boost population diversity. They were utilized in this work. The new chromosome is chosen at random by the diversity operator from the selection pool.

2. Method

The following is a flowchart for scheduling simulation using a genetic algorithm with multiparent crossover. From the flowchart, the initial population generation is the first step. The second step is determining how to fit each individual in the population. The third step is to select the parent chromosome. The fourth step is to cross and mutate the parent chromosome. Finally, do the child's chromosomes meet the desired criteria in the fifth step? If yes, the process is complete. Otherwise, the process will return to the second step.

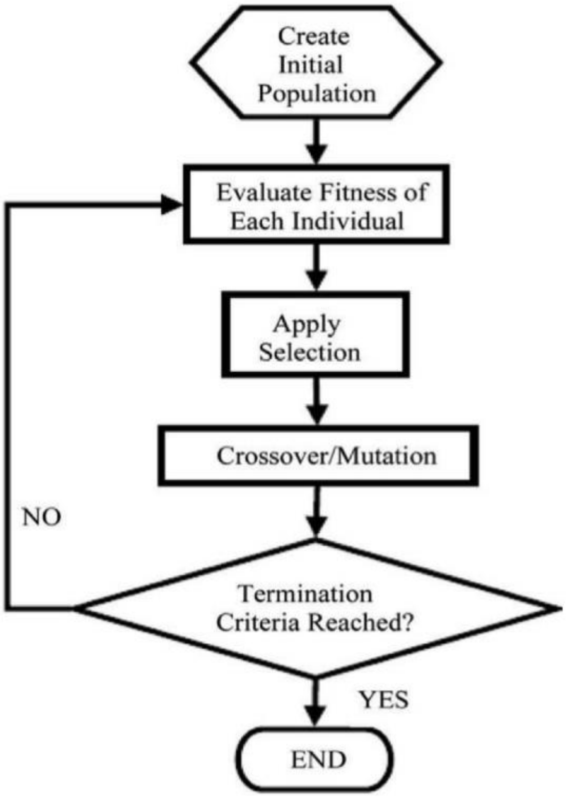

Fig.2 Generator scheduling flowchart using genetic algorithm with multiparent crossover

3. Results

To minimize total fuel costs, the proposed GA-MPC method is put through its paces with 5-unit systems. MATLAB R2008 was used to write the proposed code, which runs on a computer with an i5 core processor running at 2.3 GHz.The number of generations and chromosomes is thought to be 40 and 200, respectively. The crossover probability (Pc) has lower and higher values of 0.1 and 0.9, respectively. There are five generators and a load of 259 MW in the current test system [5]. According to Fig. 2, the GA-MPC is the most affordable option.

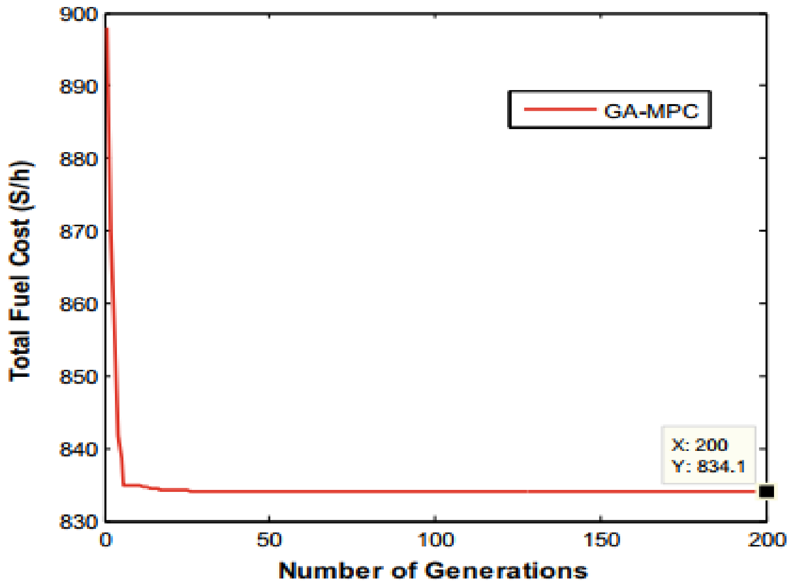

Fig.2.Variation in Total Fuel Cost for a 5-unit System Based on Iteration Number

4. Conclusion

A genetic algorithm with a crossover between multiple parents has been used to solve generator scheduling optimization. With the help of diversity and multiparent crossover factors, this algorithm finds the best solutions. On 5-generator systems, the GA-MPC is validated by minimizing Total Fuel Cost.

References

[1] Agarwal, A and Pal, K, Optimization of Unit Commitment Problem using Genetic Algorithm, International Journal of System Dynamics Applications 10 (3): July-September, pp.21-37 (2021)

[2] Ali Bukhari, S.B, Ahmad, A., Raza, S.A., Zaki, A, M., Genetic Algorithm Based Generator Scheduling-A Review, World Applied Sciences Journal 30 (12): pp.1826-1838 (2014)

[3] Chen, C.L. and S.C.Wang., Branch-and-bound scheduling for thermal generating units. IEEE Transactions on Energy Conversion, 8(2): pp.184-189 (1993).

[4] Dasgupta, D. and D.R. McGregor, Short term unit-commitment using genetic algorithms. Tools with Artificial Intelligence. Proceedings., Fifth International Conference on, pp: 240-247 (1993).

[5] Elsayed, S.M, R.A. Sarker, D.L. Essam, GA with a New Multi-parent Crossover for Constrained Optimization, Evolutionary Computation (CEC), IEEE Congress, pp.857–864 (2011).

[6] Joshi, G.K, and Mathur, S, Optimum Scheduling of Generators Using Genetic Algorithm, International Journal of Computer Science and Network Security, 8 (6), pp.61-66 (2008).

[7] Lowery, P., Generating unit commitment by dynamic programming. IEEE Transactions on Power Apparatus and Systems,5, pp. 422-426 (1966).

[8] Mantawy, A., Y.L. Magid, A and Selim, S.Z., A simulated annealing algorithm for unit commitment IEEE Transactions on Power Systems, IEEE Transactions on, 13 (1): pp.197-204 (1998).

[9] Marsudi,D., Operasi Sistem Tenaga Listrik, Balai Penerbit dan Humas ISTN, Bhumi Srengseng Indah, Jakarta Selatan (1990).

[10] Ongsakul, W. and N. Petcharaks, Unit commitment by enhanced adaptive Lagrangian relaxation. Power Systems, IEEE Transactions on Power Systems 19(1): pp. 620-628 (2004).
[11] Ouyang, Z. and Shahidehpour, S., An intelligent dynamic programming for unit commitment application. Power Systems, IEEE Transactions on Power Systems, 6(3): pp.1203-1209 (1991).
[12] Pulluri.,H,et. 2020, Genetic Algorithm with Multiparent Crossover Solution for Economic Dispatch with Valve Point Loading Effects, Springer Nature Singapore Pte, Ltd (2020).
[13] Senjyu, T., K. Shimabukuro, K. Uezato and T.Funabashi, 2003. A fast technique for unit commitment problems by extended priority list. IEEE Transactions on Power Systems, 18(2): pp.882-888 (2003).
[14] Sundararajan., D.Subramanian., B.Subramanian., K and Krishnan., M, Generation scheduling problem by intelligent genetic algorithm. Computers and Electrical Engineering, 39(1): pp.79-88 (2013).
[15] Takriti, S. and Birge, J.R. Using integer programming to refine Lagrangian-based unit commitment solutions, IEEE Transactions on Power Systems 15(1): 151-156 (2000).
[16] Wong, Y., K. Cung., T.S. and Tuen., T.W, Genetic Algorithm Approach to Scheduling of Generator Unit. Proc. of the 5th Int.Conf. on Advance in Power System Control, Operation and Management, APSCOM, Hongkong (2000).
[17] Zhao, B., C. Guo, B. Bai and Y. Cao, An improved particle swarm optimization algorithm for unit commitment. International Journal of Electrical Power & Energy Systems, 28(7): pp.482-490 (2006).