



Unit Commitment Solution Based on Improved Particle Swarm Optimization Method

Ali I. Abbas*, Afaneen Anwer

Electrical Engineering Dept., University of Technology-Iraq, Alsina'a street, 10066 Baghdad, Iraq.

*Corresponding author Email: eee.19.14@grad.uotechnology.edu.iq

HIGHLIGHTS

- The optimal scheduling of power generation
- The Unit Commitment Solution based on Improved Particle Swarm Optimization.
- lowest production cost during a specified period of time.
- Economic Dispatch solution with Lambda Iteration Method.
- Dynamic Programming a conventional method used for solving the unit commitment.

ABSTRACT

This paper presents an algorithm to solve the unit commitment problem using the intelligence technique based on improved Particle Swarm Optimization (IPSO) for establishing the optimal scheduling of the generating units in the electric power system with the lowest production cost during a specified time and subjected to all the constraints. The minimum production cost is calculated based on using the Lambda Iteration method. A conventional method was also used for solving the unit commitment problem using the Dynamic Programming method (DP). The two methods were tested on the 14-bus IEEE test system and the results of both methods were compared with each other and with other references. The comparison showed the effectiveness of the proposed method over other methods.

ARTICLE INFO

Handling editor: Ivan A. Hashim

Keywords:

Economic dispatch
Dynamic programming
Particle Swarm Optimization
Optimal solution
Generation scheduling

1. Introduction

The usage of electrical utility is varied during the day, week, month, and year. If enough generation is saved online throughout all time, some of the generating units operate at their minimum limit from generation during the off-peak period. This procedure is considered uneconomical. Therefore it must find an approach to turn off the unnecessary units at the off-peak period on the condition that other units meet load demand [1]. The optimal strategy to perform these requirements is unit commitment (UC). Thus, unit commitment can be realized as the problem of finding the optimal unit scheduling with the lowest production cost, which can be determined by economic load dispatch (ED). Unit scheduling means determining the status of units (ON or OFF). There are important constraints that must be taken into account and satisfied to achieve acceptable results in UC, as shown in section 2.1.

Many methods have been used to solve the UC problem as integer programming[2], branch-and-bound methods[3], mixed-integer programming[4], and Lagrangian relaxation methods (LR) [5], and priority list method [6]. Traditional methods have many drawbacks, such as long computational time or large memory [7]. Hence, there are many optimization algorithms suggested solving unit commitment. Sirote Khunkitti et al. proposed adding the sigmoid function to the hybrid (DA-PSO) to Improve DA-PSO. this method tested on 5, 6, 10, 26 units [7], Vikram Kumer Kamboj used the hybrid (PSOGWO) and tested on 10 and 5 units [8], Ashutosh Bhadoria et al. proposed sine cosine algorithm to find optimum scheduling for thermal units. They tested this algorithm on 5 and 10 units [9]. Soraphon Kigsirisin and Hajime Miyauchi proposed a novel (BAMFO) to find the best solution to unit commitment [10]. Ashutosh Bhadoria and Sanjay Marwaha proposed MFO to solve unit commitment and tested on 5 and 10 units [11]. Farsadi et al. proposed Nested PSO and tested it on ten units only [12]. Singh et al. suggested a hybrid WODEGA and tested on 6 units [13].

<http://doi.org/10.30684/etj.v39i10.2114>

Received 11 April 2021; Accepted 08 September 2021; Available online 25 October 2021

2412-0758/©Publishing rights belongs to University of Technology's Press, Baghdad, Iraq

This is an open access article under the CC BY 4.0 license (<http://creativecommons.org/licenses/by/4.0>).

This paper proposes solving the unit commitment problem using two methods, a conventional one (DP) and an intelligent one (IPSO). Both methods are tested on the the14-bus IEEE test system.

2. Research Method

Unit Commitment issues are often used to assess which units must participate in dispatch operations at particular times. Number of units, load capacity, startup cost, spinning reserve, and ramp rate are all constraints that must be considered

2.1 Constraints

To solve the UC problem, we must satisfy the constraints imposed as operational requirements. The constraints can be explained as follows [7][14]:

1. Power constraint

The sum of power generation of each generator at time t (P_{gi}^t) equal to demanded power:

$$\sum_{i=1}^{Ng} P_{gi}^t = P_D^t \tag{1}$$

Where P_D^t represents active power which demanded at time t and Ng number of generator

2. Spinning reserve constraint

This term is used to characterize the total maximum active generated power from all units in the system minus the current load demand. In other words, mathematically, we can say:

$$\sum_{i=1}^{Ng} P_{gi}^{(Max)} \geq P_D^t + P_R^t \tag{2}$$

Where $P_{ig(max)}$ represents maximum active power of the unit (i), and P_R^t is the reserve of active power at time t.

3. Power limit constraints

$$P_{gi(min)} \leq P_{gi}^t \leq P_{gi(max)} \tag{3}$$

Where $P_{gi(min)}$ is the minimum active power of unit i

4. Minimum Up Time Constraint

$$T_{i,on}^t \geq MUT_i \tag{4}$$

$T_{i,on}^t$ represents on hours of unit (i) till time t, and MUT_i is the minimum uptime of unit (i).

5. Minimum Down Time Constraint:

$$T_{i,off}^t \geq MDT_i \tag{5}$$

$T_{i,off}^t$ represents off hours of unit (i) till time t, and MDT_i is the minimum downtime of unit (i).

To solve the UC problem for any power system, ED calculation must be done first to find the cost of production for each generating unit, which is essential for scheduling the unit's commitment.

Unit Commitment issues are often used to assess which units must participate in dispatch operations at particular times. Number of units, load capacity, startup cost, spinning reserve, and ramp rate are all constraints that must be considered

2.2 Formulation of the Economic Dispatch

The input to every unit is represented as the unit cost rate (i) and denoted by (F_i). While unit output, P_{gi} , is the active power generated through that i th unit. Combined of production costs F_i of all units committed is equal total fuel cost TFC . Economic Dispatch is the process of determining the best power generation of generation plants (units) to meet power demand and spinning reserve at the time, particular with minimum cost (TFC), which is the intended total cost [15].

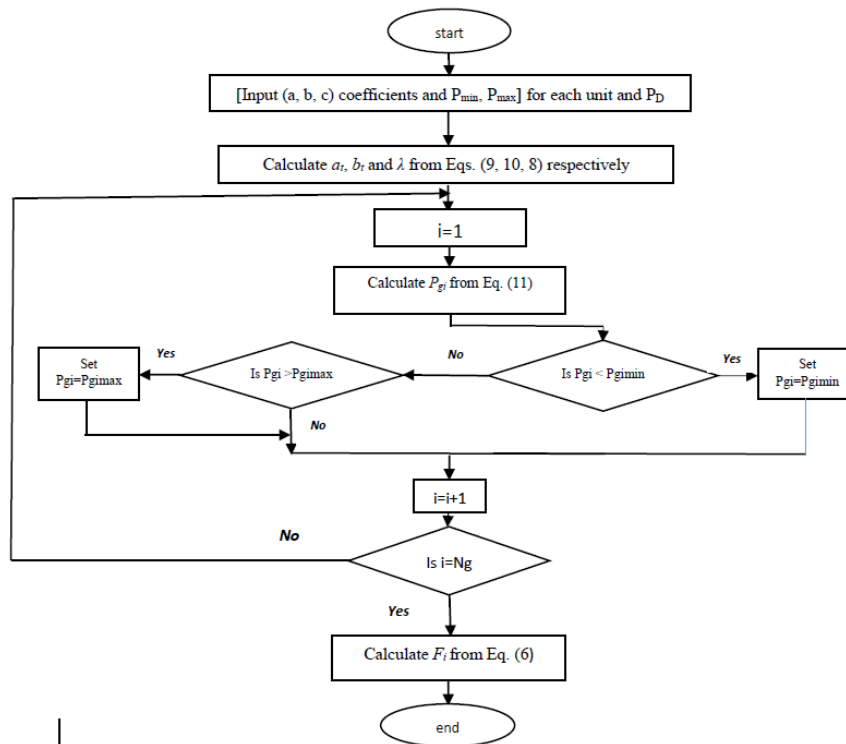


Figure 1: ED flowchart

The production cost F_i of unit (i) at any interval time is a function of the unit power output P_{gi}

$$\left(F_i^t = c_i + b_i P_{gi} + \frac{a_i}{2} P_{gi}^2 \right) \tag{6}$$

Where a_i , b_i , and c_i are the fuel cost coefficients. To calculate P_{gi} , we need to find the incremental fuel cost which is

$$\frac{\partial f_i}{\partial P_{gi}} = \lambda_i \tag{7}$$

For the economic partition of load among units within the plant, the norm is that all units must be operating at the same incremental fuel cost (λ)

$$\text{where for units}=\{i, j, \dots, n\}, \lambda_i = \lambda_j = \dots = \lambda_n = \lambda$$

$$\lambda = a_t P_D + b_t \tag{8}$$

$$a_t = \left(\sum_{i=1}^N \frac{1}{a_i} \right)^{-1} \tag{9}$$

$$b_t = a_t \sum_{i=1}^N \frac{b_i}{a_i} \tag{10}$$

$$P_{gi}^t = \frac{\lambda - b_i}{a_i} \tag{11}$$

Consequently, the steps of calculating minimum production cost are: Firstly, from a_i , b_i , calculate a_t and b_t by Eqs. (9) and (10) respectively. Then, find incremental fuel cost from Eq. (8). Calculate real generated power for every generator at time t (P_{gi}^t) from Eq. (11). Finally, Calculate fuel production cost for each unit at time t by Eq. (6).

In this paper, a program for ED was written in MATLAB environment. Figure (1) shows the flow chart of this program.

2.3 Dynamic Programming (DP) method for solving UC

DP is one of the conventional methods to solve the commitment of the units. It is used to illustrate the problems in which decisions must be made to determine the best operation of the power system. These decisions are linked with time, so they are in stages, and each stage corresponds to one hour of operation. In other words, Dynamic Programming can be illustrated as it is a systematic procedure. Its work is based on the premise that there are many possible decisions in a multi-step problem and the systematic procedure systematically evaluates those decisions [16].

There are costs associated with every potential decision. The cost of each decision is affected by the cost of the decision made in the previous step. In contrast, other costs that can be considered are additional costs, which are the costs of transition, as they are the costs that are incurred when moving from a decision in one step to another decision in the next step [16].

Figure (2) shows the flowchart of the UC procedures using the DP method.

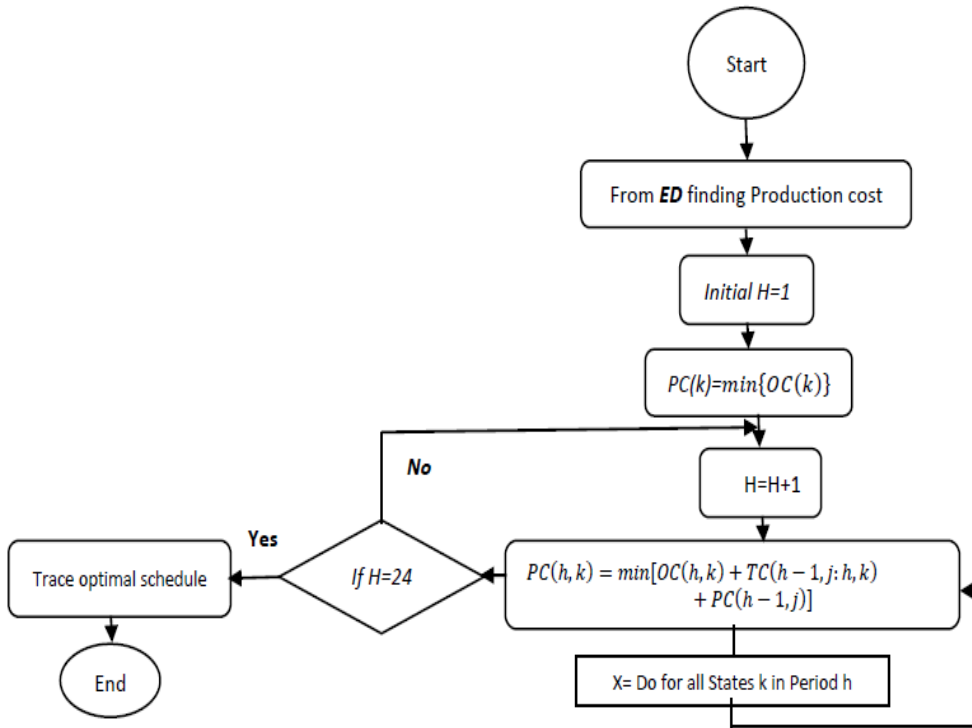


Figure 2: The flowchart of UC using DP

The recursive algorithm is shown below to compute the minimum cost in h^{th} hour with k^{th} combination [16][17]:

$$PC(h, k) = \min[OC(h, k) + TC(h - 1, j: h, k) + PC(h - 1, j)] \tag{12}$$

Where:

$PC(h, k)$: is the minimum cumulative cost to arrive at state (h, k) , which is also called return function.

$OC(h, k)$: operating cost for the state (h, k) .

$TC(h-1, j: h, k)$: Transition cost from the state $(h-1, j)$ to state (h, k) .

$PC(h-1, j)$: the production cost of combination j^{th} at the time $(h-1)$.

In this paper, a program was written in a Matlab environment for UC using the DP method. The structure of the program is shown as follows:

- 1) Read the inputs and form (2^N) number of combinations for each sub-period.
- 2) Run the economic dispatch program to determine the production cost of each state in each subinterval.
- 3) Calculate start-up and shut down the cost for each state from the previous sub-interval.
- 4) Calculate the return function for each state from the recursive equation and obtain the best path to arrive current combination from the previous sub-interval.
- 5) Find an optimal path using backtracing then trace the best path to reach that state from the previous state.
- 6) Finally, the total cost is calculated.

2.4 The Proposed Method for Solving UC Using IPSO

2.4.1 Overview of PSO

PSO can be understood as a technique intended to search for specific content to improve the results. PSO relies on the behavior and intelligence of organisms like many other algorithms presented by James Kennedy and Russell Eberhart [18]. This theory was developed relying mainly on the intelligence of the swarm to do this advanced computational procedure. The main idea of this behavior was taken from the behavior of fish and birds in the food search.

The members of the population in PSO are named particles. In a swarm, each particle is considered a feasible solution to the optimization problem. The movement of particles in space depends on the rules of flow. In the case of one of the particles discovering or inventing a solution, the other particles will approach it. The population will entities shift slowly into a better area. For all individuals, the fitness values are estimated and optimized through the function of fitness [18].

As a PSO is a tool of optimization, It provides an exploration process in which residents constantly change their locations over time. The movement of each particle controls its velocity .Through investigation, each particle maintains the best position it found, this position is called the personal best, symbolized by P_{best} . Besides, amongst the P_{best} values gained, the better fitness is specific by one particle, known as the global best, denoted by G_{best} [19].

The particle position and velocity must be updated. Based on the reconstituted individuals' fitness values, the personal location P_{best} and global location G_{best} are reconstituted.

The equations of update for the position and velocity are:

$$V_i^{t+1} = V_i^t + c_1 r_1^t (P_{best}^t - x_i^t) + c_2 r_2^t (G_{best}^t - x_i^t) \tag{13}$$

$$x_i^{t+1} = x_i^t + V_i^{t+1} \tag{14}$$

Where: c_1, c_2 : acceleration coefficients

x : the location of the particle

V_i^t : The particle's rapidity (velocity) in i^{th} dimension

r_1, r_2 are two independently engendered evenly dispersed random numbers between 0 and 1

2.4.2 Improved PSO

The proposed algorithm that was used in this work is the PSO which has been improved by the following improvements to obtain more accurate results that mean obtaining the best (lowest) production cost:

1. Introducing inertia weight factor (w) to increase the optimization [20].

$$w^t = w_{max} - \frac{w_{max} - w_{min}}{Iter_{max}} \times Iter \quad (15)$$

Where w^t is inertia weight factor at time t . w_{max} and w_{min} are the initial and final weights, respectively, and $iter_{max}$ is the maximum iteration number. Typically $w_{max} = 0.9$ and $w_{min} = 0.4$

Hence the update of velocity for equation (14) will be as:

$$V_i^{t+1} = w^t V_i^t + c_1 r_1^t (P_{best}^t - x_i^t) + c_2 r_2^t (G_{best}^t - x_i^t) \quad (16)$$

2. Another improvement has been applied to PSO where the two random numbers will engender as follows [8]:

$r_1 = (\text{fitness} < P_{\text{fitness}})$, and $r_2 = (\text{fitness} < G_{\text{fitness}})$.

So the selection of r_1 and r_2 will be based on fitness value, meaning that their choice is more appropriate with each iteration to give results with high accuracy.

2.4.3 The Program of Unit Commitment Using PSO

In this paper, a program was written for UC using the proposed algorithm. The structure of the program is :

- 1- Initially, configured particles randomly and input other parameters as iteration number, c_1 , c_2 , w , etc., within acceptable limits of generated power (p_{min} and p_{max}).
- 2- Calculate production cost (F_i) and all ED requirements given in section (2.3).
- 3- Compare the current (F_i) for each particle with P_{best} . If the current (F_i) is better than P_{best} , this value is P_{best} otherwise, P_{best} same.
- 4- The particle with the best fitness value is determined. This value will be G_{best} value.
- 5- Update the speed of each individual from equation (16).
- 6- Revise the position of individual x_i^t using equation (14).
- 7- If the iterations number arrives at the maximum number, go to step 8, else go to step 3.
- 8- Determine the total cost of all combinations, power distribution between the units, and trace the units' scheduling.

Determine the best value of G_{best} means that it is the best power generated from each unit with the lowest total generation cost. The flowchart of the program is shown in figure (3)

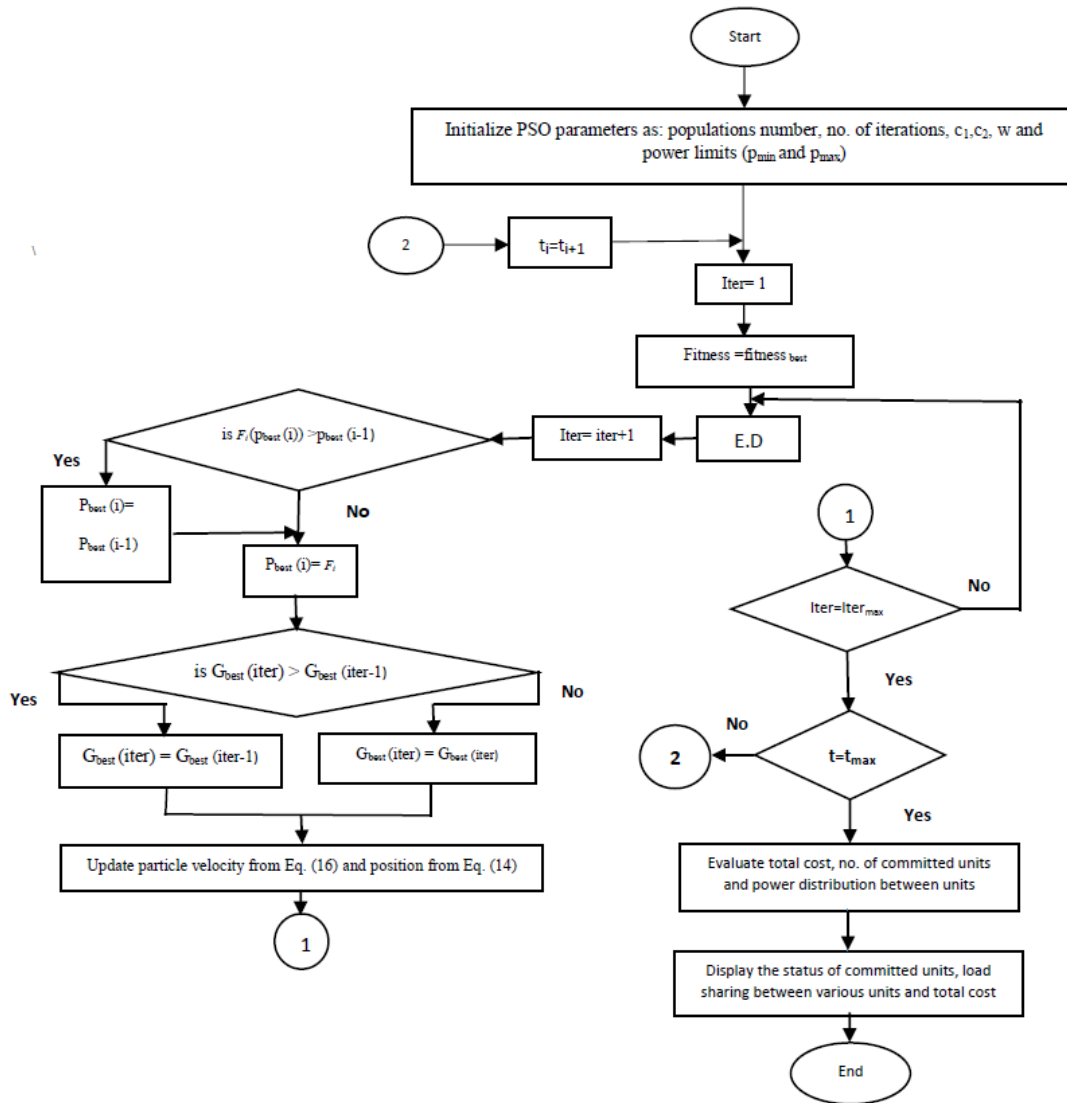


Figure 3: The flowchart of UC using IPSO

3. Results and Discussions

The proposed two algorithms used in this paper are applied to a standard test system: (14-bus IEEE test system). The configuration of the system under consideration is shown in Figure (1) [21]. The data of the system and load demand for 24 hours are in Tables (1) and (2) respectively in the appendix [13]. Both algorithms were written using MATLAB tools. The parameters of IPSO used to obtain the best result are $c_1=1.2$, $c_2=2$, $w_{min}=0.4$, $w_{max}=0.9$, and the number of populations is 50 with spinning reserve is 0.1 from load demand. Table (I) shows the scheduling of the units to operate five units system in 24 hours by improved PSO, while Table (2) represents the operation of the same units by DP. The sky-blue color refers to the units that were committed.

Table 1: Scheduling of 5 units system by DP

Hour	U1	U2	U3	U4	U5
1	1	0	0	0	0
2	1	0	0	0	0
3	1	0	0	0	0
4	1	0	0	1	0
5	1	0	0	1	0
6	1	0	0	1	0
7	1	0	0	0	0
8	1	0	0	0	0
9	1	0	0	0	0
10	1	0	0	0	0
11	1	0	0	0	0
12	1	0	0	0	0
13	1	0	0	0	0
14	1	0	0	0	0
15	1	0	0	0	0
16	1	0	0	0	0
17	1	1	0	0	0
18	1	1	0	0	0
19	1	0	0	1	0
20	1	0	0	1	0
21	1	0	0	0	0
22	1	0	0	0	0
23	1	0	0	0	0
24	1	0	0	0	0
Total cost: 11020 \$					

Table 2: Scheduling of 5 units system by IPSO

Hour	U1	U2	U3	U4	U5
1	1	0	0	0	0
2	1	1	1	0	0
3	1	1	1	0	0
4	1	1	1	0	0
5	1	1	1	0	0
6	1	1	1	0	0
7	1	1	1	0	0
8	1	1	1	0	0
9	1	1	1	0	0
10	1	1	1	0	0
11	1	1	1	0	0
12	1	1	1	0	0
13	1	1	1	0	0
14	1	1	1	0	0
15	1	1	1	0	0
16	1	1	1	0	0
17	1	1	1	0	0
18	1	1	1	0	0
19	1	1	1	0	0
20	1	1	1	0	0
21	1	1	1	0	0
22	1	1	1	0	0
23	1	1	1	0	0
24	1	1	1	0	0
Total cost: 11159 \$					

Table 3: A comparison between IPSO and the related works

No.	Algorithm	Total production cost (\$)	Percentage of production cost for IPSO to other algorithms
1	GWO-PSO	12281 [8]	10.267 %
2	BMFO	11980 [10]	8.013 %
3	BAMFO	11980 [10]	8.013 %
4	MFO	11980 [11]	8.013 %
5	Sine Cosine algorithm	11979.726 [9]	8.011 %
6	iDA-PSO	11830.94 [7]	6.854 %
7	The proposed algorithm (IPSO)	11020	-

The results in tables (1) and (2) show that the total cost using IPSO for the 14-bus IEEE test system is (11020 \$), while the total cost using DP is (11159 \$). The reduction of the cost of production is 1.25 %.

Table (3) represents a comparison between IPSO and the related works to show the activity of IPSO. From the observation of this table, the reduction percentage of production cost of IPSO with respect to least works are in range (10.267 – 6.854) %, which clarifies the efficiency of the IPSO algorithm on other works in solving of unit commitment.

4. Conclusion

In this paper, two methods were adopted for solving the unit commitment problem those are. The intelligent technique improved Particle Swarm Optimization (IPSO) and the DP conventional method. After simulating both methods on the 14-bus IEEE standard test system, the obtained results of the IPSO proposed method are compared with other related work to prove its superiority and efficiency. The comparison shows that the IPSO overcame the other methods in terms of minimizing the production cost also in terms of scheduling the units.

Author contribution

All authors contributed equally to this work.

Funding

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Data availability statement

The data that support the findings of this study are available on request from the corresponding author.

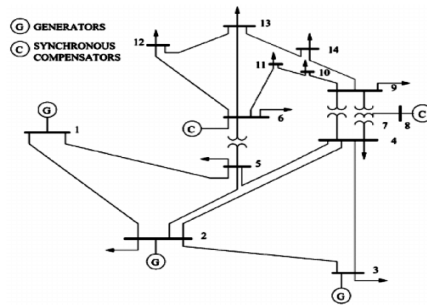
Conflicts of interest

The authors declare that there is no conflict of interest.

References

[1] S. Sivanagaraju, *Power system operation and control*. Pearson Education India, (2009).

- [2] K. W. Edwin, H. D. Kochs, and R. J. Taud, Integer programming approach to the problem of optimal unit commitment with probabilistic reserve determination, *IEEE Trans. Power Appar. Syst.*, PAS-97, 6, (1978), 2154–2166. doi: 10.1109/TPAS.1978.354719.
- [3] A. I. Cohen and M. Yoshimura, A branch-and-bound algorithm for unit commitment, *IEEE Trans. Power Appar. Syst.*, 2, (1983), 444–451.
- [4] J. A. Muckstadt and R. C. Wilson, An Application of Mixed-Integer Programming Duality to Scheduling Thermal Generating Systems, *IEEE Trans. Power Appar. Syst.*, 87, 12, (1968), 1968–1978. doi: 10.1109/TPAS.1968.292156.
- [5] F. Zhuang and F. D. Galiana, Towards a more rigorous and practical unit commitment by Lagrangian relaxation, *IEEE Trans. Power Syst.*, 3, 2, (1988), 2, 763–77.
- [6] G. B. Sheble, Solution of the unit commitment problem by the method of unit periods, *IEEE Trans. Power Syst.*, 5, 1, (1990), 257–260. doi: 10.1109/59.49114.
- [7] S. Khunkitti, N. R. Watson, R. Chatthaworn, S. Premrudeepreechacharn, and A. Siritaratiwat, An improved DA-PSO optimization approach for unit commitment problem, *Energies*, 12, 12, (2019), 2335.
- [8] V. K. Kamboj, A novel hybrid PSO–GWO approach for unit commitment problem, *Neural Comput. Appl.*, 27, 6, (2016), 1643–1655.
- [9] A. Bhadoria, S. Marwaha, and V. K. Kamboj, An optimum forceful generation scheduling and unit commitment of thermal power system using sine cosine algorithm, *Neural Comput. Appl.*, (2019), 1–30.
- [10] S. Kigsirisin and H. Miyauchi, Short-Term Operational Scheduling of Unit Commitment Using Binary Alternative Moth-Flame Optimization, *IEEE Access*, 9, (2021), 12267–12281.
- [11] A. Bhadoria and S. Marwaha, Moth flame optimizer-based solution approach for unit commitment and generation scheduling problem of electric power system, *J. Comput. Des. Eng.*, 7, 5, (2020), 668–683.
- [12] M. Farsadi, H. Hosseinnejad, and T. S. Dizaji, Solving unit commitment and economic dispatch simultaneously considering generator constraints by using nested PSO, *ELECO 2015 - 9th Int. Conf. Electr. Electron. Eng.*, (2016), 493–499. doi: 10.1109/ELECO.2015.7394478.
- [13] A. Singh and A. Khamparia, A hybrid whale optimization-differential evolution and genetic algorithm based approach to solve unit commitment scheduling problem: WODEGA, *Sustain. Comput. Informatics Syst.*, 28, (2020), 100442. doi: 10.1016/j.suscom.2020.100442.
- [14] Y. Zhai, X. Liao, N. Mu, and J. Le, A two-layer algorithm based on PSO for solving unit commitment problem, *Soft Comput.*, 24, 12, (2020), 9161–9178.
- [15] A. J. Wood, B. F. Wollenberg, and G. B. Sheblé, *Power generation, operation, and control*. John Wiley & Sons, (2013).
- [16] P. K. Singhal and R. N. Sharma, Dynamic programming approach for solving power generating unit commitment problem, in *2011 2nd International Conference on Computer and Communication Technology (ICCT-2011)*, (2011), 298–303.
- [17] S. U. Rani and C. H. P. Raju, A Solution to Unit Commitment Problem via Dynamic Programming and Particle Swarm Optimization, *Int. J. Curr. Eng. Technol.*, 3, 4, (2013).
- [18] J. Kennedy and R. Eberhart, Particle swarm optimization, in *Proceedings of ICNN'95-international conference on neural networks*, 4, (1995), 1942–1948.
- [19] P. Sriyanyong and Y. H. Song, Unit commitment using particle swarm optimization combined with Lagrange relaxation, in *IEEE Power Engineering Society General Meeting*, (2005), 2752–2759.
- [20] Y. Shi and R. Eberhart, A modified particle swarm optimizer, in *1998 IEEE international conference on evolutionary computation proceedings. IEEE world congress on computational intelligence (Cat. No. 98TH8360)*, (1998), 69–73.
- [21] J. A. Boudreaux, Design, Simulation, and Construction of an IEEE 14-Bus Power System, 42, (2018).



Appendix:

Table 4: Data of 5 units, 14-bus IEEE system

Unit No.	P_{gimax}	P_{gimin}	$a(\$/MW^2)$	$b(\$/MW)$	$c(\$/h)$	MUT i	MDT i	HSCi	CSCi	CSHi	ISi
U1	250	10	0.00315	2	0	1	1	70	176	2	1
U2	140	20	0.0175	1.75	0	2	1	74	187	2	-3
U3	100	15	0.0625	1	0	1	1	50	113	1	-2
U4	120	10	0.00834	3.25	0	2	2	110	267	1	-3
U5	45	10	0.025	3	0	1	1	72	180	1	-2

Table 5: Load demand for 14-bus IEEE system

Hour	1	2	3	4	5	6	7	8	9	10	11	12
Demand	14	17	22	24	25	24	22	20	17	13	10	13
Hour	8	3	0	4	9	8	7	2	2	4	0	0
Demand	13	14	15	16	17	18	17	20	21	22	23	24
Demand	15	16	19	22	24	24	23	21	17	15	13	10
Demand	7	8	5	5	4	1	0	0	6	7	8	3