

Unit Commitment with Renewable Energy Using Multi-agent Immune System Incorporating Evolutionary Priority List for Operation Cost Minimization

M. N. C. Othman¹, I. Z. Abidin², T. K. A. Rahman³

¹Institute for Power Engineering, Universiti Tenaga Nasional, Malaysia.

²Institute for Power Engineering, Universiti Tenaga Nasional, Malaysia.

³School of Science and Technology, Asia e-University (AeU), Kuala Lumpur, Malaysia.
 m_nazree@yahoo.com¹, izham@uniten.edu.my², titik.khawa@aeu.edu.my³

Article History: Received: 10 November 2020; Revised: 12 January 2021; Accepted: 27 January 2021;
 Published online: 05 April 2021

Abstract: Renewable energy (RE) integration into the power grid is steadily growing that effectively reducing the electricity generation cost, however, leads to additional consequences. Unit commitment (UC) problem often cited in discussing these issues. This paper proposes an approach to look at these issues by utilizing the Multi-agent Immune incorporating Evolutionary Priority List (MAI-EPL) technique for solving the UC problem with the consideration of RE resources, as an effort to reduce the operation cost without affecting the existing power system reliability. The employed test systems consider 10 thermal generating units with additional solar and wind resources in a 24-hours scheduling period. The results show that the MAI-EPL technique is capable of producing a satisfactory outcome to the UC problem with RE consideration.


Keywords: unit commitment; renewable energy; multi-agent immune system.

1. Introduction

In recent years, renewable energy (RE) is having significant growth. The integration of RE into the power grid is happening, where, its main reason is to reduce the electricity generation cost [1], [2]. This addition will for sure lead to new circumstances and challenges especially issues related to the process of deciding the best configuration of available resources to supply electrical energy or known as unit commitment (UC) [1], [3]. Aside from the usual UC constraints and considerations, which is often regarded as an NP-hard problem [4], [5], the foremost concern for this integration is the existing grid system reliability [2], [6], [7]. The presence of wind generation, for example, will introduce another uncertainty due to wind forecast error [8]. The similar response also develops due to the inclusion of solar energy. The power system is expected to be capable of compensating the random phenomenon of solar irradiance availability [9].

The current solutions to the UC problem need to be updated so it will comply with the RE consideration. It is vital in the field of UC optimization to acquire a commitment schedule with optimal qualities, especially the test system operational cost and solution consistency without jeopardizing the grid system reliability. This paper proposes the Multi-agent Immune incorporating Evolutionary Priority List (MAI-EPL) technique as an approach to resolve the UC problem with RE consideration. In sect. 2 of this paper, the UC with RE problem formulation is further deliberated. The detail explanation of the MAI-EPL algorithm is in sect. 3 and the execution process is available in sect. 4. In sects. 5 and 6, the simulation results and conclusion were discussed respectively. The analysis uses 10 generating units with solar and wind energy test system and considers common constraints correlated to the UC problem.

2. Unit Commitment with Renewable Resources

UC usually referred to a selection process to get the best configuration of available power resources to supply electricity so that the operational cost is minimum [3]. It starts with the selection of the unit on-off status, succeeded by the determination of the economic dispatch (UC-ED) [3], [10]. The unit on-off Status,  denoted by N-by-H matrix as (1) [3]:

$$U_n(t) = \begin{bmatrix} U_1^1 & U_1^2 & \dots & U_1^H \\ U_2^1 & U_2^2 & \dots & U_2^H \\ \vdots & \vdots & \vdots & \vdots \\ U_N^1 & U_N^2 & \dots & U_N^H \end{bmatrix} \quad (1)$$

Every unit $\overline{U_n(t)}$ can be on-line or off-line at any particular hour, t [1], [3]:

$$\overline{U_n(t)} \in \{0,1\} \quad (2)$$

For each unit on-off status, its associated UC-ED, $\overline{P_n(t)}$ represented by (3):

$$\overline{P_n(t)} = \begin{bmatrix} P_1^1 & P_1^2 & \dots & P_1^H \\ P_2^1 & P_2^2 & \dots & P_2^H \\ \vdots & \vdots & \vdots & \vdots \\ P_N^1 & P_N^2 & \dots & P_N^H \end{bmatrix} \quad (3)$$

The aim of solving the UC problem is to achieve minimum total operational cost (TC) [10].

$$\overline{TC} = \sum_{t=1}^H \sum_{n=1}^N [\overline{SUC_n(t)} + F_n(\overline{P_n(t)})] \cdot U_n(t) \quad (4)$$

$\overline{SUC_n(t)}$ is the start-up cost function that can be further addressed as (5).

$$\overline{SUC_n(t)} = \begin{cases} HSC_n, & \text{if } MDT_n \leq T_n^{off} \leq MDT_n + CSH_n \\ CSC_n, & \text{if } T_n^{off} > MDT_n + CSH_n \end{cases} \quad (5)$$

Where, $\overline{F_n(P_n(t))}$, the fuel cost function is shown as (6):

$$\overline{F_n(P_n(t))} = a_n + b_n \cdot P_n(t) + c_n \cdot P_n(t)^2 \quad (6)$$

$\overline{a_n}$, $\overline{b_n}$, and $\overline{c_n}$ are the fuel cost coefficients.

2.1. Unit Commitment Common Constraints

Before further deliberation of the RE effect, the conventional UC constraints are first to resemble. The constraints involve the generator capability limit, the power balance, the spinning reserve, the unit must-run constraints, and the generator minimum on-time and off-time [3], [10].

1) *Generator Capability Limit*: The generated power during on-time, $\overline{P_{n(on)}(t)}$ must fall between the maximum capacity, $\overline{P_{n(max)}(t)}$ and the minimum capacity, $\overline{P_{n(min)}(t)}$.

$$\overline{P_{n(min)}(t)} \leq \overline{P_{n(on)}(t)} \leq \overline{P_{n(max)}(t)} \quad (7)$$

2) *Power Balance Requirement*: The electricity demand and the generated power are expected to be equal at every time interval [3]:

$$\sum_{n=1}^N \overline{P_n(t)} \cdot U_n(t) = D_t(t) \quad (8)$$

3) *Spinning Reserve Requirement*: The Spinning reserve, $\overline{R_r}$ is a pre-determined value that must meet the condition of (9) [5]: generated power during on-time, $\overline{P_{n(on)}(t)}$ must fall between the maximum capacity, $\overline{P_{n(max)}(t)}$ and the minimum capacity, $\overline{P_{n(min)}(t)}$.

$$\sum_{n=1}^N \overline{P_{n(max)}(t)} \cdot U_n(t) \geq D_t(t) + R_r(t) \quad (9)$$

4) *Generator Capability Limit*: Every power plant is expected to always on-line at a particular operation period [3].

$$\overline{U_n^t} \geq U_n^{-t} \quad (10)$$

5) *Generator Minimum On-time Requirement*: It is a minimum time for every power plant should be on-line before they can be shut-down [3]:

$$\overline{T_n^{on}} \geq MUT_n \quad (11)$$

6) *Generator Minimum Off-time Requirement*: It is a minimum time required by a power plant to be off-line before it can be restarted again [3]:

$$\overline{T_n^{off}} \geq MDT_n \quad (12)$$

2.2. Renewable Energy Requirements

In addition to the UC conventional constraints, other uncertainties are expected due to wind forecast error [1] and the consideration of the random phenomenon occurrence of the solar irradiance availability [9]. Thus, the wind power function and the solar irradiance characteristic are weighed in the analysis.

1) *The wind power requirement*: The wind power function is presented by Eq. (13) below [8]:

$$P_w^t(t) = \begin{cases} 0 & v_w^t \leq v_{ci} \text{ or } v_{ci} \leq v_w^t \\ P_{wn} \left(\frac{v_w^t - v_{ci}}{v_r - v_{ci}} \right), & v_{ci} \leq v_w^t \leq v_r \\ P_{wn} & v_r \leq v_w^t \leq v_{co} \end{cases} \quad (13)$$

$\overline{P_{wn}}$ is the wind power output; $\overline{v_w^t}$, is the forecasted wind speed at hour t; $\overline{v_{ci}}$, $\overline{v_{co}}$, and $\overline{v_r}$ are the cut-in, cut-out, and rated wind turbine speed respectively.

2) *Solar irradiance*: The solar irradiance modeling to describe the random phenomenon of the irradiance data is set out as (14) [9]:

$$f_b(s) = \begin{cases} 0 & 0 \leq s \leq 1, \\ \frac{\Gamma(\alpha_s + \beta_s)}{\Gamma(\alpha_s)\Gamma(\beta_s)} (s^{\alpha_s-1})(1-s)^{\beta_s-1}, & \text{for } \alpha_s \geq 0, \\ 0 & \beta_s \geq 0 \\ & \text{else} \end{cases} \quad (14)$$

Where

$$\overline{\beta_s} = (1 - \mu) \left(\frac{\mu^{\alpha_s + \mu}}{\mu^{\alpha_s} - 1} - 1 \right) \quad (15)$$

$$\overline{\alpha_s} = \frac{\mu^{\beta_s}}{\mu^{\beta_s} - 1} \quad (16)$$

\overline{s} is the solar irradiance in kW/m²; $\overline{f_b(s)}$ is the beta distribution function of \overline{s} ; $\overline{\alpha_s}$ and $\overline{\beta_s}$ are parameters of the Beta distribution function.

To compensate for the possible uncertainties that might emerge, the forecasted RE capabilities are set to be less or equal to the amount of the spinning reserve of the test system [11]. This strategy will allow the system to respond immediately in the event of contingencies without affecting the system stability and security.

3. MAI-EPL Optimization Technique

MAI-EPL is a hybrid computational system that applies the simultaneous cooperation and competition (SCC) as its basis. The formulated learning mechanism employs the act of remembering (also known as immune memory) of the AIS algorithm [12], [13] together with the essence of the Multi-agent System (MAS) [14] characteristic. This essential configuration then enhanced with the application of the refined stochastic evolutionary-based algorithm assimilated using the Evolutionary Programming (EP) [10], [15] and the Priority List (PL) [16] techniques.

The MAI-EPL basic procedure consists of several steps starting with the initialization (production of agents), lattice creation, lattice duplication, and finally, the application of the SCC. These fundamental steps are illustrated in Fig. 1.

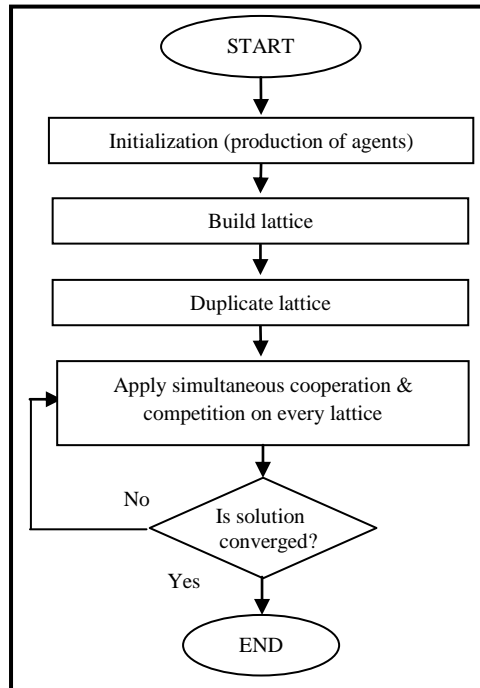


Figure 1. MAI-EPL basic procedure

3.1. Initialization

Initialization is a process to generate agents randomly and according to PL rules of priority [17]. In this analysis, an agent is an independent entity that observes and may learn using the collected knowledge to achieve the objective function [18]. Each agent comprises the unit on-off status $U_n(t)$ and its related UC-ED $P_n(t)$ that match the (1) and (3) respectively. The random agent is generated using (17) [10];

$$\overline{P_n(t)} = P_{min}(t) + rand \cdot (P_{max}(t) - P_{min}(t)) \quad (17)$$

where, $rand$ is a random number ranging from 0 to 1. The $\overline{P_n(t)}$ value is generated within the minimum and maximum generator boundaries to effectively produce random solution with minimum effect on the computational time. Then, other sets of agents are generated using two rules of priority; Rule 1 and Rule 2.

1) *Rule 1; Priority list considering minimum cost per mega-watt (MC)*: The MC cost function is decided using (18) [17]. The list by using Rule 1 is in Table II, where, the 10 generating units data are available in Table I [10]. Generator with the lowest cost per Mega-watt (MW) is the first to commit power. If the electricity demand is exceeding the capacity of the selected generator, another unit from the priority order will be turned on. The process resumes until electricity demand and system constraints are satisfied [10], [16].

$$MC = \frac{F_n(P_{n(max)}(t))}{P_{n(max)}(t)} \quad (18)$$

2) *Rule 2; Priority list considering minimum cost and generation capability*: Rule 2 is the priority order based on the value of the MC cost function and considering generator maximum power capacity [16], [17]. It is the extended version of the Rule 1 [17]. Generator with the highest power capacity are placed at the top priority. If two units have the same $\overline{P_{n(max)}(t)}$ value, the unit with a lower cost per MW value will be at higher priority. The priority order using Rule 2 available in Table III.

TABLE 1. 10 Generating Units Test System Data

	Unit 1	Unit 2	Unit 3	Unit 4	Unit 5
Pmax (MW)	455	455	130	130	162
Pmin (MW)	150	150	20	20	25
a (\$/h)	1000	970	700	680	450
b (\$/MWh)	16.19	17.26	16.60	16.50	19.70
c (\$/MWh ²)	0.00048	0.00031	0.002	0.00211	0.00398
Hot Start Cost (\$)	4500	5000	550	560	900
Cold Start Cost (\$)	9000	10000	1100	1120	1800
Min Up (h)	8	8	5	5	6
Min down (h)	8	8	5	5	6
Cold Start hrs.(h)	5	5	4	4	4
Initial status (h)	8	8	-5	-5	-6
	Unit 6	Unit 7	Unit 8	Unit 9	Unit 10
Pmax (MW)	80	85	55	55	55
Pmin (MW)	20	25	10	10	10
a (\$/h)	370	480	660	665	670
b (\$/MWh)	22.26	27.74	25.92	27.27	27.79
c (\$/MWh ²)	0.00712	0.00079	0.00413	0.00222	0.00173
Hot Start Cost (\$)	170	260	30	30	30
Cold Start Cost (\$)	340	520	60	60	60
Min Up (h)	3	3	1	1	1
Min down (h)	3	3	1	1	1
Cold Start hrs.(h)	2	2	0	0	0
Initial status (h)	-3	-3	-1	-1	-1

Table 2. Priority order according to rule 1

Unit	MC (\$/MW)	Priority order	Unit	MC (\$/MW)	Priority order
1	18.6	1	6	27.5	6
2	19.5	2	7	33.5	7
3	22.2	4	8	38.1	8
4	22.0	3	9	39.5	9
5	23.1	5	10	40.1	10

Table 3. Priority Order According to Rule 2

Unit	<input checked="" type="checkbox"/>	MC (\$/MW)	Priority order	Unit	<input checked="" type="checkbox"/>	MC (\$/MW)	Priority order
1		455	18.6	1		80	27.5
2		455	19.5	2		85	33.5
3		130	22.2	5		50	38.1
4		130	22.0	4		50	39.5
5		162	23.1	3		50	40.1

3.2. Building the Lattice

The next step is to build the lattice using the best agents from the initialization process. The lattice is required to be in the formation of two by two matrices, three by three, and so on [14], [18]. The lattice illustration is available in Fig. 2. Every circle represent an agent with the designated coordinate (i, j) as $i, j = 1, 2, \dots, L_{size}$ [14], [18].

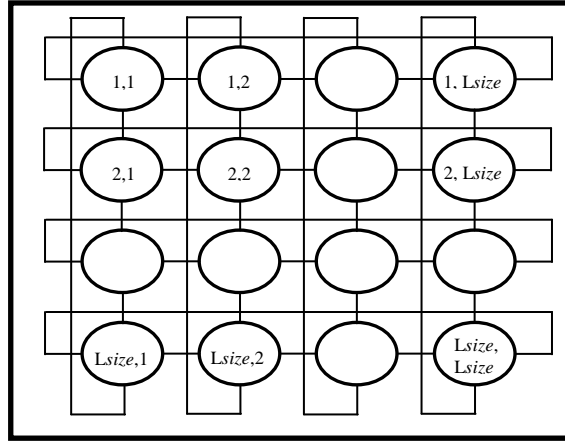


Figure 2. Model of the lattice

3.3. The Lattice duplication

Based on the generated lattice, cream agents are selected, duplicated, and stored inline among them. The duplication number will be decided to deviate from 2 to 1000 depending on the requirement of the test system. This procedure yielded as an effort to initiate simultaneous optimization procedure based on the act of remembering from the Artificial Immune System (AIS) algorithm [12], [13]. Individual lattice will have different optimization path using the SCC procedure. At the end of the process, the solutions of each lattice will undergo the convergence confirmation.

3.4. Simultaneous Cooperation and Competition

The Simultaneous Cooperation and Competition (SCC) is a learning mechanism that applies an effective simultaneous learning procedure instead of the conventional single directional learning process at a time. The SCC process starts with the Parent Pool creation of every duplicated lattice, which is the selection of an agent and its four neighbors. An agent, $\overline{P_{n(i,j)}}(t)$ can only have four neighbors, $\overline{Nb_{n(i,j)}}$ as represented by (19) [15], [18]:

$$\overline{Nb_{n(i,j)}} = \{L_{i',j}, L_{i,j'}, L_{i'',j}, L_{i,j''}\} \quad (19)$$

Where,

$$i' = \begin{cases} i-1, & \text{if } i \neq 1 \\ L_{size}, & \text{if } i = 1 \end{cases}, j' = \begin{cases} j-1, & \text{if } j \neq 1 \\ L_{size}, & \text{if } j = 1 \end{cases}$$

$$i'' = \begin{cases} i+1, & \text{if } i \neq L_{size} \\ 1, & \text{if } i = L_{size} \end{cases}, j'' = \begin{cases} j+1, & \text{if } j \neq L_{size} \\ 1, & \text{if } j = L_{size} \end{cases}$$

The cooperation process then applied toward the Parent Pool simultaneously on every lattice. The cooperation process is the interaction among agent and the four neighbors so that the valuable information of individual agent will be shared to the whole lattice. By using a similar principle, the knowledge within the lattice, later on, will be diffused to the other lattices through the simultaneous cooperation. This cooperation process then being refined with the assimilation of the mutation procedure of the EP algorithm. Avery members of the Parent pool will go through the mutation process to produce the Offspring pool using the following formula [15], [19]:

$$\overline{P_n'} = P_n + \alpha \cdot \beta_n \quad (20)$$

Where,

$$\alpha = \eta \cdot \exp(\tau' \cdot \beta_n + \tau \cdot \beta_n) \quad (21)$$

$$\beta = N(\mu, \sigma^2) \quad (22)$$

and,

$$\tau = \left((2(m)^{1/2})^{1/2} \right)^{-1} \quad (23)$$

$$\tau' = (2m)^{1/2})^{-1} \quad (24)$$

\overline{P}_n' is the offspring of each agent after the mutation process, $\overline{\alpha}$ is the mutation factor, and $\overline{\beta}$ is the Gaussian random variable. β_n and β_n are the initial random variable and the n -th element of Gaussian random variable respectively [10], [19]. m describes the decision variable of individual generating unit.

The Offspring pool and the Parent pool are later combined before arranged according to their fitness. Then, the competition procedure is applied to formulate a better succeeding agent. It is performed toward every agent, $\overline{P}_{n(i,j)}(t)$ at the selected lattice point, (i, j) . If the fittest agent, $\overline{P}_{n(best)}(t)$ meet (25), it can fill the particular lattice point, (i, j) , and the other agents omitted for further processing.

$$\overline{fitness(Best_{i,j})} < \overline{fitness(L_{i,j})} \quad (25)$$

All the process will end if the stopping criterion is satisfied, which usually happens whenever every total operational cost (TC) value of agents in the lattices are equal.

4. Solving UC Problem with RE using MAIEP-PL Technique

The procedure of the proposed MAI-EPL technique to solve the UC with RE problem is explained in the following steps, and the corresponding flowchart is provided in Fig. 3.

Step 1: Initialization (random).

Optimization procedures start with the initialization. An initial population that contains a set of on-off generators combination and UC-ED will be generated randomly within the minimum and maximum power limits of (7). The Initial population size is in the form of N-by-H matrix, where N is the total number of generating unit and H is the total number of time intervals. Then, the generated real power will be tested to comply other test system requirements as in (7) – (14). If all test system requirements have been met, the on-off units combination and its UC-ED will be kept in the Random pool. The process will back to earlier step if any specified test system requirements being violated. Then, another set of the on-off units combination and its UC-ED will be generated until the Random pool is equal to P.

Step 2: Initialization using PL operator.

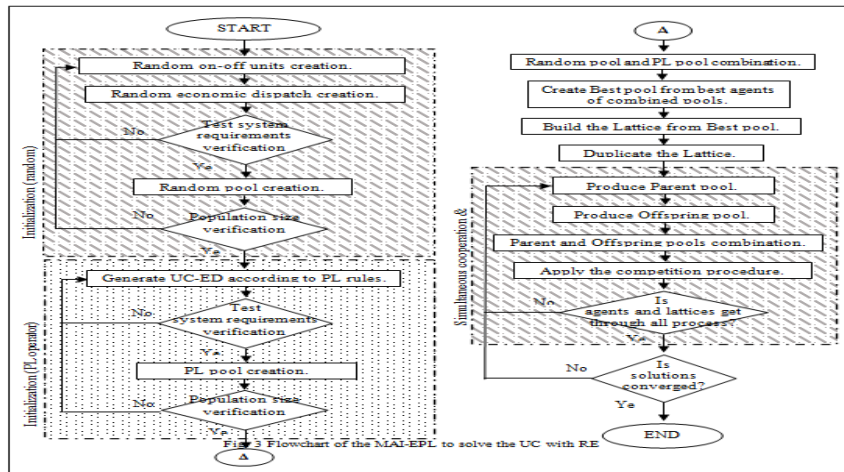


Figure 3. Flowchart of the MAI-EPL to solve the UC with RE

The PL pool is created using the PL Rules 1 and 2. The PL pool contains the on-off generating units combination and its UC-ED that selected from the best result of the PL Rules 1 and 2. The selected PL pool solution also has to undergo all the test system requirements of (7) – (14).

Step 3: Random and PL pools combination.

The Random and PL pools are merged. The total operating cost then calculated using (4) – (6) and ranked accordingly.

Step 4: Create Best pool.

The solutions with top fitness (minimum total operation cost) of the combined pools were selected to form the Best pool.

Step 5: Build the lattice.

The Best pool members then arranged in the $\sqrt{L_{size} \times L_{size}}$ lattice-like environment to form the lattice.

Step 6: Duplicate the lattice:

The lattice then duplicated depends on the requirement of the analysis ranging from 2 to 100 for this study.

Step 7: Produce Parent pool.

An agent and its neighbors of every lattice later selected to form the Parent pool, which altogether consists of five members.

Step 8: Produce Offspring pool.

Every member of the Parent pool then will go through the mutation process to produce the Offspring pool. Its size is equal to the Parents pool and formulated using (20).

Step 9: Parent and Offspring pools combination.

The Parent and Offspring pools then merged. Its' fitness are calculated using (4) – (6).

Step 10: Apply the competition process.

Every member of the combined Parent pool and Offspring pool are ranked using the calculated fitness. The best agent with minimum TC value is selected and being compared with the first elected agent (from step 7). Then, the best of it will occupy the lattice location (from step 7).

Step 11: The SCC application.

Steps 7 to 10 are repeated toward all agents of the lattice simultaneously with other duplicated lattices.

Step 12: Convergence test:

The processes stop if the solutions have converged. Otherwise, steps 7 to 11 will be repeated. The solutions are considered converged if all agents' TC value in all lattice is same.

5. Results and Discussion

The proposed MAI-EPL is developed and executed using MATLAB programming language on a computer with Intel i7 processor of 3.40 GHz CPU speed and 3 GB RAM. The MAI-EPL technique has been applied to the 10 units test system with the addition of the forecasted wind and solar resources [11]. The units' specifications are available in Table I, while, the 24-hour forecasted loading demand is in Table IV. The spinning reserve rated at 10% of the forecasted electricity demand. The forecasted wind and solar data are tabulated in Table V [11]. Table VI shows the modified wind and solar data based on the forecasted data of Table V for comparison purpose. The solar farm estimated to be capable of providing a total of 40MW electrical energy with 16% photovoltaic panel efficiency and a total 25.5 MW of wind capacity [11].

The simulations outcomes are available in two sections. The first section discusses the proposed technique parameters determination. Then, a comparison of the proposed MAI-EPL technique for solving UC Problem and UC problem with the addition of RE resources were reviewed in the next section.

Table 4. 24-hour forecasted loading demand

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

Table 5. 24-Hour Forecasted Solar and Wind Power

<i>Hours</i>	<i>Solar (MW)</i>	<i>Wind (MW)</i>	<i>Hours</i>	<i>Solar (MW)</i>	<i>Wind (MW)</i>
1	0	10.54	13	36.78	25.5
2	0	22.27	14	31.59	24.82
3	0	25.5	15	9.70	20.74
4	0	25.5	16	12.92	14.62
5	0	25.5	17	0	25.5
6	0	25.5	18	0	19.04
7	0.09	25.5	19	0	25.5
8	17.46	25.5	20	0	18.02
9	31.45	25.5	21	0	25.5
10	36.01	25.5	22	0	21.42
11	38.06	25.5	23	0	0
12	35.93	25.5	24	0	2.55

Table 6. Modified 24-Hour Forecasted Solar and Wind Power

<i>Hours</i>	<i>Solar (MW)</i>	<i>Wind (MW)</i>	<i>Hours</i>	<i>Solar (MW)</i>	<i>Wind (MW)</i>
1	0	10	13	36	25
2	0	22	14	31	24
3	0	25	15	9	20
4	0	25	16	12	14
5	0	25	17	0	25
6	0	25	18	0	19
7	0	25	19	0	25
8	17	25	20	0	18
9	31	25	21	0	25
10	36	25	22	0	21
11	38	25	23	0	0
12	35	25	24	0	2

5.1. Parameters determination

Most of the evolutionary algorithm (EA) techniques require optimum population members to produce good optimization results, as an unfit number would commence to premature convergence [10], [16] – [18]. Since the MAI-EPL is a hybrid technique based on the EA, having the right sample size is crucial. The search for the right population size starts with a review of other available approaches. Other techniques use 9 up to 36 population members and successfully providing reliable performance in term of linear systems approximation and function optimizations using specified test system [1], [20], [21]. For MAI-EPL technique, agent and duplication numbers are the parameters that will define the population size. Thus, 9 and 16 agents number will be considered. It is then being matched with the different duplication values ranging from 2 to 10. The results of this analysis are available in Table VII. The performance of the MAI-EPL for solving the UC and the UC with the integration of RE problems are also presented in Table VII.

The MAI-EPL technique performs best at the agents’ value of 9 and the duplication number of 3, in which out of 100 trials for each combination, resulted in the optimum total operational cost (TC) and minimum execution time. The best TC for the UC problem is \$563937.70, by using 27 population members (9 agent value and 3 duplication number), while the best execution times is 0.57 seconds. The execution times vary, but the best, average, and worst TC values converged to the same lowest point when the population sizes are larger than 27. The similar pattern also happens with the UC with RE resources. Thus, 27 population members seem to be adequate to produce optimum results for the test systems. Comparable population size also being used by other techniques in the field of Evolutionary Algorithm (EA) [10], [17], [20].

5.2. The performance of the MAI-EPL for solving the UC and the UC with the integration of RE resources

The objective of solving the UC problem is to obtain a schedule of available generating units that produce minimum operational cost and comply with all test system requirement. Different test systems would require distinct combination so the result would be optimum. Before further analysis of the RE integration, the proposed MAI-EPL technique held probed on the conventional UC problem. The plot of the MAI-EPL search process for solving the UC problem starting from the best-selected lattices until convergence is available in Fig. 4. Each

lattice consists of 9 agents. Every lattice will then duplicated three times, producing 27 population members. This three-duplication number represents three different optimization path of the SCC learning process that occurs simultaneously at the same time for every agent. Example of this optimization plot is available in Fig. 5, where, the lattice with the best initial TC of \$564199.80 converged to \$563937.70.

Based on Fig. 4, the \$564199.80 is indeed the best value among selected 9 initial agents where most of the other agents revolve around TC of \$564419.90. However, this gap certainly reduced for the next iteration, where all of the agents turn around almost at the same point, before the following procedure. This characteristic is the essential criteria of the MAI-EPL with the use of the SCC learning algorithm, where the accuracy of the result stays secured with a less necessary search process and reducing the computational burden.

The outcome of integrating the additional RE resources to the existing test system is available in Fig. 6. The TC value reduced to \$548799.32 from \$563937.70, with a total saving of \$15138.38. Contrarily, the execution times increased to 0.83 seconds from 0.57 seconds. The execution time extended due to the amplified complexity of the test system as more constraints and requirements to be satisfied with the addition of RE resources. However, the consistency of the solutions remains even with the consideration of RE resources. The MAI-EPL technique successfully produces 100% consistency out of 100 trials for both UC and UC with RE test systems without jeopardizing the system reliability.

The effect of RE integration to the existing thermal system continues investigated with the use of modified RE resources of Table VI. The forecasted energy of the solar and wind resources from Table V are simplified and roundup to the lower value of the initial, resulting the supplied RE resources to be less than the previous analysis.

Fig. 6 shows the results of the UC problem and UC with RE resources using the data of both Table V and Table VI. With the use of modified data in Table VI, the TC value increased to \$549103.13 from the best TC of \$548799.32, adding \$303.81 extra cost of generation. This outcome verifies that even with a small change to the electricity supplies could lead to a price change of the test system. The best UC-ED configuration for the 10 thermal generating units with RE resources test system is available in Table VIII. The dispatched power is in MW, where the units 1 to 10 represent the thermal resources and units 11 and 12 are the solar and wind power respectively. The proposed MAI-EPL produced the TC of \$548799.32 and 0.83 seconds of execution time with 100% accuracy out of 100 trials. This result is generated using 27 population members (9 agent value and 3 duplication number).

Table 7. The performance of the mai-epl for solving the uc and the uc with re resources integration

	Agents	Duplication s	Pop. sizes	Total costs (\$)			Executed times (s)		
				Best	Mean	Worst	Best	Mean	Worst
UC	9	2	18	563937.70	563942.82	563948.63	0.49	0.84	2.13
		3	27	563937.70			0.57	1.42	2.82
		5	45				1.46	6.29	9.16
		10	90				5.15	11.52	14.73
	16	2	32	563937.70			0.71	2.17	4.84
		3	48				1.63	7.28	10.63
		5	80				4.75	10.34	14.16
		10	160				7.17	16.56	22.76
UC with RE	9	2	18	548799.32	549825.87	552307.63	0.66	1.13	2.98
		3	27	548799.32			0.83	1.73	3.64
		5	45				1.52	7.14	11.31
		10	90				6.36	12.65	16.22
	16	2	32	548799.32			0.97	2.83	5.13
		3	48				2.11	7.52	12.76
		5	80				5.32	11.76	15.24
		10	160				8.97	18.16	25.61

Table 8. The Performance Of The Mai-Epl For Solving The Uc And The Uc With Re Resources Integration

Hours	Units												Total (MW)
	1	2	3	4	5	6	7	8	9	10	11	12	
1	455	234.46	0	0	0	0	0	0	0	0	0	10.54	700
2	455	272.73	0	0	0	0	0	0	0	0	0	22.27	750
3	455	344.5	0	0	25	0	0	0	0	0	0	25.5	850
4	455	444.5	0	0	25	0	0	0	0	0	0	25.5	950
5	455	364.5	0	130	25	0	0	0	0	0	0	25.5	1000
6	455	334.5	130	130	25	0	0	0	0	0	0	25.5	1100
7	455	384.41	130	130	25	0	0	0	0	0	0.09	25.5	1150
8	455	417.04	130	130	25	0	0	0	0	0	17.46	25.5	1200
9	455	455	130	130	28.05	20	25	0	0	0	31.45	25.5	1300
10	455	455	130	130	113.49	20	25	10	0	0	36.01	25.5	1400
11	455	455	130	130	151.44	20	25	10	10	0	38.06	25.5	1450
12	455	455	130	130	162	51.57	25	10	10	10	35.93	25.5	1500
13	455	455	130	130	112.72	20	25	10	0	0	36.78	25.5	1400
14	455	455	130	130	28.59	20	25	0	0	0	31.59	24.82	1300
15	455	429.56	130	130	25	0	0	0	0	0	9.7	20.74	1200
16	455	282.46	130	130	25	0	0	0	0	0	12.92	14.62	1050
17	455	234.5	130	130	25	0	0	0	0	0	0	25.5	1000
18	455	340.96	130	130	25	0	0	0	0	0	0	19.04	1100
19	455	434.5	130	130	25	0	0	0	0	0	0	25.5	1200
20	455	455	130	130	156.98	20	25	10	0	0	0	18.02	1400
21	455	455	130	130	59.5	20	25	0	0	0	0	25.5	1300
22	455	455	0	0	123.58	20	25	0	0	0	0	21.42	1100
23	455	425	0	0	0	20	0	0	0	0	0	0	900
24	455	342.45	0	0	0	0	0	0	0	0	0	2.55	800

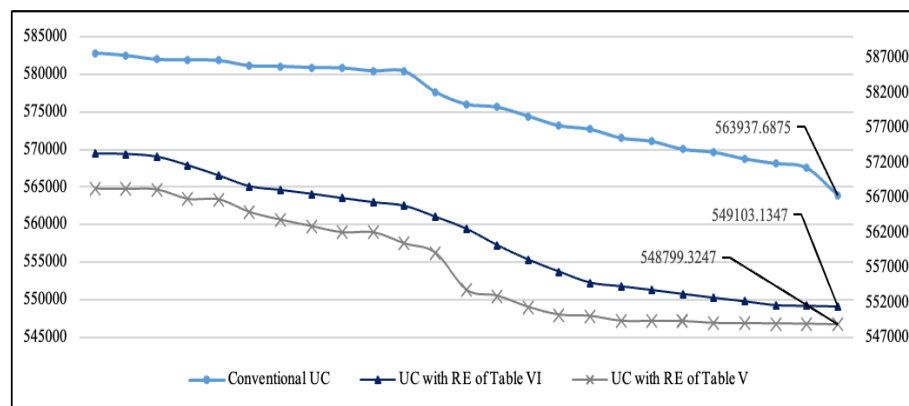


Figure 4. The plot of the MAI-EPL search process for solving the UC problem starting from the best-selected agents until convergence

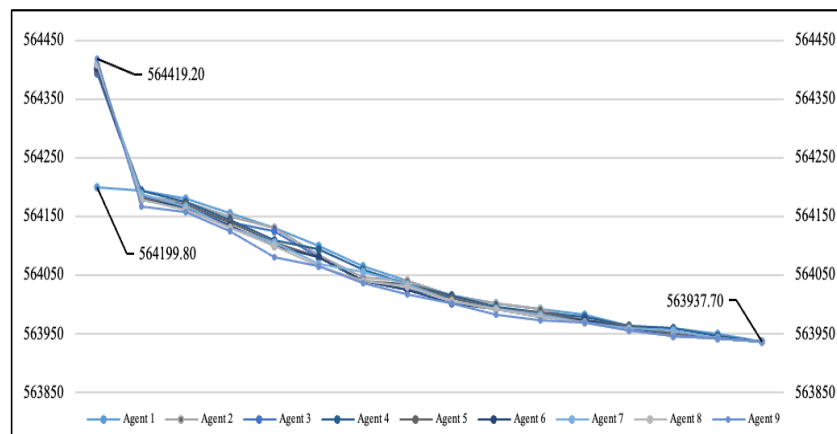


Figure 5. The plot of the search process of the duplicated agent until convergence

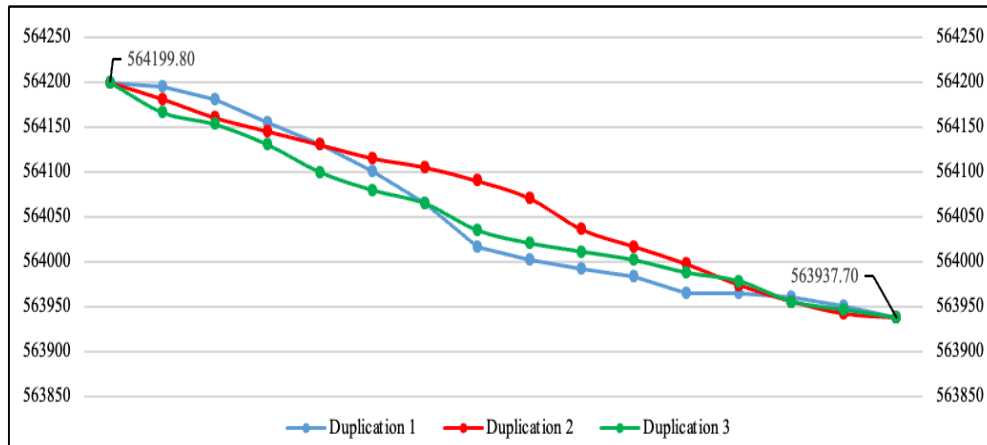


Figure 6. Comparison of the Total Cost (TC) values for the UC problem and the UC with RE resources using the data of both Table V and Table VI.

6. Conclusion

The integration of the RE resources into the conventional thermal system has increased the system complexity especially in the field of the Unit Commitment (UC) problem, however, allowing a better resolution in term of generation cost. This paper introduced a hybrid technique designated as the Multi-agent Immune incorporating Evolutionary Priority List (MAI-EPL) for solving the UC problem with the addition of RE resources as an effort to reduce the operational cost of the test system. Without jeopardizing the test system reliability, the proposed technique yielded favorable outcomes to the difficulties with the application of its primary element, the learning mechanism designated as the simultaneous cooperation and competition (SCC).

The results show that the MAI-EPL technique is capable of minimizing the test system operational cost, while at the same time maintaining the system reliability, fast computational time, and producing 100% consistency and out of 100 trials. The integration of RE resources to the existing thermal power system possesses promising potential with the use of an appropriate approach, especially the MAI-EPL technique.

References

1. Trivedi, D. Srinivasan, K. Pal, and T. Reindl, "A Multi-objective evolutionary algorithm based on decomposition for unit commitment problem with significant wind penetration," *Proc. 2016 IEEE Congress on Evolutionary Computation (CEC)*, pp. 3939–3946, July 2016.
2. E. Du, N. Zhang, B.-M. Hodge, Q. Wang, Z. Lu, C. Kang, Q. Xia, "Operation of a High Renewable Penetrated Power System with CSP plants: A Look-ahead Stochastic Unit Commitment Model", *IEEE Transactions on Power Systems*, Vol. 34, pp. 140-151, Jan. 2019.
3. J. Wood, B. F. Wollenberg, *Power Generation Operation and Control*, New York: Wiley, 1996.
4. H. Ye, J. Wang, and Z. Li, "MIP Reformulation for Max-Min Problems in Two-Stage Robust SCUC", *IEEE Trans. on Power Syst.*, Vol. 32 (2), pp. 1237–1247, Mar. 2017. H. Yu, and M. Shahidehpour, 'Stochastic SCUC Solution With Variable Wind Energy Using Constrained Ordinal Optimization' *IEEE Trans. on Sustainable Energy*, Vol. 5 (2), pp. 379–388, Apr. 2014.
5. T. Halidou, H. O. R. Howlader, M. E. Lotfy, A. Yona, and Tomaru, "Unit Commitment in the Presence of Renewable Energy Sources and Energy Storage System: Case Study," *Journal of Energy and Power Eng.*, Vol. 12, pp. 322–328, 2018.
6. W. X. Wang, C. S. Li, X. Liao, and H. Qin, "Study on unit commitment problem considering pumped storage and renewable energy via a novel binary artificial sheep algorithm," *Journal of applied Energy*, Vol. 187, pp. 612–626, Feb. 2017.
7. S. Kaewpasuk, B. Intiyot, and C. Jeenanunta, "Stochastic Unit Commitment model for power system with Renewable Energy," *2017 Int. Electrical Eng. Congress (iEECON)*.
8. Y. M. Atwa, E. F. El-Saadany, M. M. A. Salama, and R. Seethapathy, "Optimal renewable resources mix for distribution system energy loss minimization," *IEEE Trans. Power Syst.*, Vol. 25 (1) pp. 360–370, Feb. 2010.
9. K. A. Juste, H. Kita, E. Tanaka, and J. Hasegawa, "An Evolutionary Programming Solution to the Unit Commitment Problem," *IEEE Trans. Power Syst.*, Vol. 14 (4), pp. 1452-1459, Nov. 1999.

10. A.Y. Saber, and G. K. Venayagamoorthy, "Efficient Utilization of Renewable Energy Sources by Grid-able Vehicles in Cyber-Physical Energy Systems," *IEEE Sys. Journal*, Vol. 4 (4), pp. 285-294, Sept. 2010.
11. S. Crotty, and R. Ahmed, "Immunological memory in humans," *Seminars in Immunology* 16, pp. 197–203, 2004.
12. L. N. De Castro, and F. J. Von Zuben, "Learning and optimization using the clonal selection principal," *IEEE Trans. on Evolutionary Computation*, Vol. 6 (3), pp. 239–251, June 2002.
13. Ferber, *Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence*, Addison-Wesley, 1999.
14. J. Fogel, A. J. Owens, and M.J. Walsh, *Artificial Intelligence through Simulated Evolution*, New York: Wiley, 1966.
15. Y. T. Fang, T. O. Ting, "Methodological Priority List for Unit Commitment Problem," *2008 International Conference in Computer Science and Software Eng.*, pp. 176–179.
16. N. C. Othman, T. K. A. Rahman, H. Mohklis, and M. M. Amin, "Solving unit commitment problem using multi-agent evolutionary programming incorporating priority list," *Arabian Journal for Science and Eng.*, Vol. 40 (11), pp. 3247–3261, Nov. 2015.
17. W. Zhong, J. Liu, and L. Jiao, "A multi-agent genetic algorithm for global numerical optimization," *IEEE Trans. Syst. Man and Cybern. Part B*, Vol. 34 (2), pp. 1128–1141, Apr. 2004.
18. D. B. Fogel, L. J. Fogel, and J.W. Atmar, "Meta evolutionary programming," *1991 Conference of Signals, Systems, and Computers, The Twenty-Fifth Asilomar*, pp. 540–545.
19. T. W. Lau, C. Y. Chung, K. P. Wong, T. S. Chung, and S. L. Ho, "Quantum inspired evolutionary algorithm approach for unit commitment," *IEEE Trans. on Power Syst.*, Vol. 24 (3), pp. 1503–1512, Aug. 2009.
20. X. Yuan, B. Ji, S. Zhang, H. Tian, and Y. Hou, "A new approach for unit commitment problem via binary gravitational search algorithm", *Applied Soft Computing*, Vol. 22, pp. 249–260, 2014.