Indonesian Journal of Electrical Engineering and Computer Science
Vol. 12, No. 1, October 2018, pp. 17~29
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v12.i1.pp17-29

Eagle Strategy Based Crow Search Algorithm for Solving Unit Commitment Problem in Smart Grid System

Rachid Habachi, Achraf Touil, Abdelkabir Charkaoui, Abdelwahed Echchatbi
Laboratory of Mechanical, of Engineering, of Industrial Management and Innovation
The Faculty of Sciences and Technology, Hassan 1st University, PO Box 577, Settat, Morocco

Article Info

Article history:

Received Apr 21, 2018
Revised Jun 14, 2018
Accepted Jun 25, 2018

Keywords:

Smart Grid
Unit Commitment
Eagle Strategy (ES)
Crow Search Algorithm (CSA)
Lambda-Iteration Method

ABSTRACT

Eagle strategy is a two-stage optimization strategy, which is inspired by the observation of the hunting behavior of eagles in nature. In this two-stage strategy, the first stage explores the search space globally by using a Levy flight; if it finds a promising solution, then an intensive local search is employed using a more efficient local optimizer, such as hillclimbing and the downhill simplex method. Then, the two-stage process starts again with new global exploration, followed by a local search in a new region. One of the remarkable advantages of such a combina-tion is to use a balanced trade off between global search (which is generally slow) and a rapid local search. The crow search algorithm (CSA) is a recently developed metaheuristic search algorithm inspired by the intelligent behavior of crows. This research article integrates the crow search algorithm as a local optimizer of Eagle strategy to solve unit commitment (UC) problem in smart grid system. The Unit commitment problem (UCP) is mainly finding the minimum cost schedule to a set of generators by turning each one either on or off over a given time horizon to meet the demand load and satisfy different operational constraints. There are many constraints in unit commitment problem such as spinning reserve, minimum up/down, crew, must run and fuel constraints. The proposed strategy ES-CSA is tested on 10 to 100 unit systems with a 24-h scheduling horizon. The effectiveness of the proposed strategy is compared with other well-known evolutionary, heuristics and meta-heuristics search algorithms, and by reported numerical results, it has been found that proposed strategy yields global results for the solution of the unit commitment problem.

Corresponding Author:

Rachid Habachi,
Laboratory of Mechanical, of Engineering, of Industrial Management and Innovation,
The Faculty of Sciences and Technology,
Hassan 1st University, PO Box 577, Settat, Morocco.
Email: habachirachid@gmail.com

1. INTRODUCTION

Smart grids are a set of technologies, concepts and approaches, allowing the integration the generation, transmission, distribution and use into one internet by full use of advanced sensor measurement technology, communications technology, information technology, computer technology, control technology, new energy technologies [1]. However, Smart Grid uses digital technology to control grid and choosing the best mode of power distribution to reduce energy consumption, reduce costs, increase reliability and also increase transparency in the network. Therefore, the system intelligent will have will have a significant impact in the fields of finance and economics of the power industry [2]. Although, The traditional network is a one-way network in which the electrical energy produced in power plants is channeled to consumers without information to create an automated and distributed network of advanced power supplies.

The unit commitment problem plays a significant role in optimizing the cost of generating electrical power by planning production units based on the allocation of the production cost of each unit and the actual output power [3]. They involves scheduling the on/off states of generating units to minimize the operating cost for a given time horizon. The committed units must meet the systems fore-casted demand and spinning reserve requirement at minimum operating cost, subject to a large set of operating constraints. The UC problem, one of the most important tasks in short-term operation planning of modern power systems, has a significant influence on the secure and economic operation of power systems [4]. Optimal commitment scheduling cannot only save millions of dollars for power companies, it also ensures system reliability by maintaining the proper spinning reserve. The UC problem is mathematically formulated as a nonlinear, largescale and mixed integer combinatorial opti-mization problem [5], [6]. The number of combinations of 0-1 variables grows exponentially for a large-scale UC problem.

Therefore, the UC is one of the most difficult problems in the area of power system optimization. The UCP is a NP- Hard problem [7] which cannot be solved exactly in reasonable computing time for large scale problems. Research efforts, therefore, have concentrated on efficient and near-optimal UC algorithms which can be applied to realistic power systems and have reasonable storage and computation time requirements. The optimization methods for UC problems can be divided into two classes through a survey of literature as follows: The first are numerical optimization techniques such as priority list methods [4], dynamic programming [8], [9], Lagrangian relaxation methods [10], branch-and-bound methods [11], and mixed integer programming [12]. The other are stochastic search methods such as genetic algorithms (GA) [13], evolutionary programming (EP) [14], simulated annealing (SA) [15], and particle swarm optimization (PSO) [16].

To solve unit commitment problem (UCP), it has become evident that the researchers concentrated on using single metaheuristics. However, there are some limitations to this. To overcome this problem, a wide variety of hybrid approaches are proposed in the literature. The core idea of a hybrid with two or more metaheuristics was inspired by the possibility that the new hybridized algorithm combines the strengths of each of these algorithms to provide the following advantages: (i) To produce better solutions, (ii) to provide solutions in less time. Among the existing meta-heuristic algorithms, eagle strategy (ES) is a two-stage method recently proposed by [17] to solve optimization problems. In this two-stage strategy, the first stage explores the search space globally by using the so-called levy flight; if it finds a promising solution, then an intensive local search is employed using a more efficient local optimizer, such as hill-climbing and the downhill simplex method. Then, the two-stage process starts again with new global exploration, followed by a local search in a new region. One of the remarkable advantages of such a combination is to use a balanced tradeoff between global search (which is generally slow) and a rapid local search. Another advantage is that this is a methodology or strategy, not an algorithm. In fact, we can use different algorithms at different stages and at different times during iterations. Up to now, most published works on ES combined with efficient local search, such as the follower pollination algorithm (PFA) [18], and differential, 1) evolution (DE) [19] mainly focused on solving the continuous optimization problem; there is no previous work that attempts to use ES in conjunction with a local optimizer, such as crow search algorithm [20] to solve the unit commitment problem. 2) The rest of this paper is organized as follows. Section 2 contains the mathematical formulation of the UCP. Section 3 introduces the repairing mechanisms applied to the UCP. Section 4 briefly presents the basics of ES and CSA. Section 5 proposes the binary eagle strategy based crow search algorithm to solve UCP. Section 6 provides the computational results.Finally, Section 7 outlines the conclusions.

2. FORMULATION OF UCP

2.1. Objective function

The purpose of UCP is principally finding the minimum cost to a group of generators by turning everyone either on or off over a specific time to satisfy the loads and meet a different operational constraints. The total cost F over the whole scheduling periods is the sum of the operating cost and start-up cost for all of the unit. Therefore, the objective function of the UC problem is:

$$\min\ F = \sum_{h=1}^{H}\sum_{i=1}^{NG}[f_i(P_i^h)\ ST_i^h(1 - u_i^{h-1})]\ u_i^h \qquad (1)$$

Where H is total scheduling period; N G is number of gener-ators; Pih is generation of unit i at time h; uhi is on/off status of unit i at time h (on = 1 and off = 0); STih is start-up cost of unit i at time h. Generally, fi(Pih) denotes the fuel cost per unit which is mathematically a quadratic function:

$$f_i(P_i^h) = a_i + b_i P_i^h + c_i (P_i^h)^2 \quad (2)$$

Where ai, bi and ci represent the unit cost coefficients.
The startup cost of the ith unit is given as:

$$ST_i^h = \begin{cases} HSC_i \text{ if } T_{offi}^h \le T_{down_i} + T_{cold_i} \\ CSC_i \text{ if } T_{offi}^h > T_{down_i} + T_{cold_i} \end{cases} \quad (3)$$

Where HSC_i and CSC_i are the hot start up cost and cold start up cost of the ith unit; T_{offi}^h is the continuous off time duration of the ith unit; T_{down_i} is the minimum down time of the ith unit; T_{cold_i} is the cold start hours of the ith unit.

2.2. Constraints

2.2.1.System power balance

$$\sum_{i=1}^{NG} P_i^h \, u_i^h = P_D^h \quad (4)$$

Where P_D^h is system load demand at time t.

2.2.2. System spinning reserve requirement

$$\sum_{i=1}^{NG} P_{i,max} \, u_i^h \ge P_D^h + P_R^h \quad (5)$$

Where P_R^h is spinning reserve at time t.

2.2.3. Generation power limits

$$P_{i,min} \le P_i^h \le P_{i,max} \quad (6)$$

Where $P_{i,min}$ and $P_{i,max}$ are minimum and maximum generation limit of unit i, respectively

2.2.4.Unit minimum up time

Once a unit is started up, it should not be shut-down before a minimum up-time period is met and it math-ematically expressed for ith generating unit as follows: A unit must be on for a certain number of hours before it can be shut down.

$$T_{i,on}^h \ge T_{i,up} \quad (7)$$

Where $T_{i,on}^h$ is continuously on time of unit i up to time h, $T_{i,up}$ is unit i minimum up time.

2.2.5.Unit minimum down time

Once a unit is started down, it should not be shut-up before a minimum down-time period is met and it mathematically expressed for ith generating unit as follows: A unit must be off for a certain number of hours before it can be brought online

$$T_{i,off}^h \ge T_{i,down} \quad (8)$$

Where $T_{i,off}^h$ is continuously off time of unit i up to time h, $T_{i,down}$ is unit i minimum down time.

2.2.6.Unit initial status

The initial status at the start of the scheduling period must be taken into account.

3. THE REPAIRING MECHANISMS FOR THE UCP

Since the size of the discrete search space of the UCP increases exponentially with the increasing number of units to be scheduled, it becomes a mathematically complex com-binatorial optimization problem. It is a tough job for any algorithm to regain the feasibility of infeasible solutions of such problems, which suggests the use of some mechanisms for forcibly satisfying the constraints of a problem. In such an attempt, the following four repairing mechanisms, the first three of which are used by many researchers for the UCP [21–26], are incorporated in the proposed binary-ES-CSA addressed in Section 5:

3.1. Priority list for unit-scheduling

Priority list is made according to every unit and its parameter, and as we observe the difference in the output powers, we can see that cost per produced of a unit at its maximum output power is less than that other output power. In this research, priority list is based on fuel cost obtained gained from the average fuel cost of each unit operating at its maximum output power. The average full-load cost a of a unit is defined as the cost per unit of power ($/MW) when the unit is at its full capacity. When the fuel cost of unit is given by Equation (2), can be expressed as:

$$\alpha_i = \frac{f_i(P_{i,max})}{P_{i,max}} = \frac{a_i}{P_{i,max}} + b_i + c_i . P_{i,max} \quad (9)$$

The units are ranked by their in ascending order. Thus, the priority list of units will be formulated based on the order of i, in which a unit with the lowest i will have the highest priority to be dispatched.

3.2. Spinning reserve constraints repairing

The spinning reserve may not meet the standards of the obtained primary unit scheduling using Binary ES-CSA . Hence, the heuristic search strategy repairs the spinning reserve constraints (5). To explain that, in order to keep the randomness nature of ES-CSA which is known as a stochastic searching algorithm, a constant pr is defined as a utilization ratio of the priority list. The procedures for repairing the spinning reserve violations in primary unit scheduling are shown as follows [26] :

Step 1: Set h = 1

Step 2: For all uncommitted units at hour t, calculate the average full-load cost $\alpha_i$using formula (9). Sort them in ascending order of α_i to obtain a list SS (α_i)

Step 3: The amount of excessive spinning reserve at each hour is calculated by:

$$R_h = \sum_{h=1}^{H} u_i^h P_{i,max} - P_D^h - P_R^h \quad (10)$$

Step 4: If Rh0, go to Step 6;

Step 5: Generate a random number $\theta \in [0; 1]$. If $\theta <$ pr, commit an uncommitted unit in SS (α_i) with the lowest α_i and return to step 3; Otherwise, randomly commit an uncommitted unit in SS(α_i) and return to step 3.

Step 6: If h < H, h = h + 1 and return to Step2. Otherwise, stop.

3.4. Minimum up and down time constraints repairing

Since the obtained unit schedule may not meet the demands of the minimum up and down time constraints/limitations, because they are failed in the previous process. Thus, they ought to be examined and fixed if the violations exist. The minimum down time is usually violated at high load levels at which the peak load hours are shorter than the minimum up time, for this cause the hills exist. In almost the same way, the minimum down time of the units, thus valleys exist. Heuristic search algorithm will be used to bank hills and fill valleys. To check for violations, on and off times of units are determined in advance. The continuously on/off times of the unit i up to hour t is calculated as follows:

$$T_{i,on}^h = \begin{cases} T_{i,on}^{h-1} + 1 & \text{if } u_i^h = 1 \\ 0 & \text{if } u_i^h = 0 \end{cases} \quad (11)$$

$$T_{i,off}^{h} = \begin{cases} T_{i,off}^{h-1} + 1 & \text{if } u_i^h = 1 \\ 0 & \text{if } u_i^h = 0 \end{cases} \quad (12)$$

The details procedures to repair violations of the minimum up and down times constraints are as follows [26]:

Step 1: Calculate the duration on and off times of all units for the whole schedule time horizon using formulas (10) and (11)

Step 2: Set h=1

Step 3: Set i=1

Step 4 : if $u_i^h = 0$ and $u_i^{h-1} = 1$ and $T_{i,on}^{h-1} < T_{i,up}$ then set $u_i^h = 1$.

Step 5 : if $u_i^h = 0$ and $u_i^{h-1} = 1$ and $h + T_{i,down} - 1 \leq H$ and $T_{i,off}^{h+Ti,down-1} < T_{i,down}$ then set $u_i^h = 1$.

Step 6 : if $u_i^h = 0$ and $u_i^{h-1} = 1$ and $h + T_{i,down} - 1 > H$ and $\sum_{h=1}^{H} u_i^h > 0$ then set $u_i^h = 1$.

Step 7: Update the duration on/off times for the unit i using using formulas (10) and (11).

Step8 : if $u < NG$, $i = i + 1$ and return to step 4.

Step9 : if $h < H$, $h = h + 1$ and return to step 3.otherwise, stop.

3.5. Decommitment of excess units

Repairing the minimum up/down time constraints of a unit may lead to extreme spinning reserves at some time instants that are not desirable from the point of operating cost. Hence a heuristic algorithm is used to decommit some units one by one, in descending order of their average full load costs as given by Equation (3.1), until the spinning reserve constraint given by Eq. (2.5) is just satisfied at any time instant. However, such decommitment is made subject to the satisfaction of the up/down time constraints of a unit, i.e., a unit will be decommitted only if no up/down time constraint of the unit is violated from such decommitment.

4. OVERVIEW OF EAGLE STRATEGY AND CROW SEARCH ALGORITHM

4.1. Egale Strategy

Eagle strategy is a two-stage optimization strategy was presented by [17]. This algorithm mimics behavior of eagles in nature. In fact, eagles use two different components to search for their prey. The first one is a random search performed by flying freely and the second one is an intensive search to catch prey when they see them. In this two-stage strategy, the first stage explores the search space globally by using a Levy flight: if it finds a promising solution, then an intensive local search is employed using more efficient local optimizer, such as hill-climbing and the down-hill simplex method. Then, the two-stage process commences another time with new global exploration, followed by local search in a new area. One of the remarkable advantages of such a combination is to use a parallel balance between global search (which is generally slow) and a rapid local search. There is another advantage that is called a methodology or strategy, not an algorithm. In fact, there are different algorithms that can be used at different times and stages during iterations. The main steps of the ES are outlined in Algorithm 1.

Algorithm 1 Eagle startegy

```
Objective function f(x)
Initialization and random initial guess x^{t=0}
while (stop criterion) do
    Global exploration by randomization (e.g. Levy flights)
    Evaluate the objectives and find a promising solution
    Intensive local search via an efficient local optimizer
    if (a better solution is found) then
        Update the current best
    end if
    Update t = t + 1
end while
```

4.2. Crow search algorithm

The crow search algorithm (CSA) is a new population-based stochastic search algorithm recently proposed by [20]. The CSA is a newly developed optimization technique to solve complex engineering optimization problems [27], [28]. It is inspired by the intelligent behavior of crows. The principles ofCSA are listed as follows [20]:

a. Crows live in the form of the flock.
b. Crows memorize the position of their hiding places.
c. Crows follow each other to commit thievery.
d. Crows protect their caches from being pilfered through probability.

Following the above assumptions, the core mechanism of the CSA consists of three basic phases, namely initialization, generate a new position, and updating the memory of crows. At first, the initial population of crows represented by n dimension is randomly generated. At iteration t, the position of crow is specified by $x^{i,t} = [x_1^{i,t}, x_2^{i,t}, \ldots\ldots, x_n^{i,t}]$and it is assumed that this crow has memorized its best experience thus far in its memory $m^{i,t} = [m_{m1}^{i,t}, m_2^{i,t}, \ldots\ldots, m_n^{i,t}]$To generate a new position, crow i select randomly a crow j, for example, from the population and attempts to follow it to find the position of its hiding place (mj). In this case, according to a parameter named awareness probability (AP), two states may happen:

a. State 1: Crow j does not know that crow i is following it. As a result, the crow i will determine the hiding place of crow j.
b. State 2: Crow j knows that crow j is following it. As a result, to protect its cache from being pilfered, the crow j will fool crow i by going to another position whitin the search space.

According to States 1 and 2, the position of the crows is updated as follows:

$$x^{i,iter+1} = \begin{cases} x^{i,iter+1} + r_i \times fl^{i,iter} \times (m^{i,iter} - x^{i,iter}), \\ r_j \geq AP^{j,iter} \text{A random positonofsearch spaceotherwise} \end{cases} \quad (13)$$

Where rj is a uniformly distributed fuzzy number from [0; 1] and $AP^{j,iter}$ denotes the awareness probability of crow j at iteration iter. Finally, the crows update their memory as follows:

$$m^{i,iter+1} = \begin{cases} x^{i,iter+1}, & \text{if } f(x^{i,iter}) \text{is better than } f(m^{i,iter}) \\ m^{i,iter}, & \text{otherwise} \end{cases} \quad (14)$$

Where f(-) denotes the objective function value. It is seen that if the fitness function value of the new position of a crow is better than the fitness function value of the memorized position, the crow updates its memory by the new position. The above process is repeated until a given termination criterion (itermax) is met. Finally, the best solution of the memories is returned as the optimal solution found by the CSA. The main steps of the CSA are outlined in Algorithm 2

Algorithm 2 Crow Search Algorithm

```
Randomly initialize the position of a flock of (N_p) crows in the search space
Evaluate the position of the crows
Initialize the memory of each crow
while (iter ≤ iter_max) do
  for i = 1 to N_p do
    Randomly choose one of the crows to follow (for example j)
    Define an awareness probability
    if (r_j ≥ AP^{j,iter}) then
      x^{i,iter+1} = x^{i,iter} + r_i × fl^{i,iter} × (m^{j,iter} − x^{i,iter})
    else
      x^{i,iter+1} = a random position of search space
    end if
  end for
  Check the feasibility of new positions
  Evaluate the new position of the crows
  Update the memory of crows
end while
```

5. BINARY EAGLE STRATEGY BASED CROW SEARCH ALGORITHM FOR UCP

The binary ES-CSA is used to optimise the unit-scheduling problem in the first step, and the Lambda-iteration method [4] is used to solve the economic load dispatch problem in the second step. These two steps run iteratively until the algorithm meets the stopping criterion. Optimising the first sub problem of unit-scheduling is more difficult than the other sub-problem of ELD. So this paper mainly discusses how to model BESCSA for the first sub-problem, and the second sub-problem is solved by the traditional Lambda-iteration method. These two sub-problems are optimised iteratively until the algorithm meets the stopping criterion. The Equations (9) and (14) are transfer from continues to binary space using the following equations :

$$x^{i,iter} = \begin{cases} 1 & \text{if } s(x^{i,iter}) \geq \text{rand}(), \\ 0 & \text{if otherwise} \end{cases} \quad (15)$$

Where $s(x^{i,iter}) = \frac{1}{y}$, $y = 1 + e^{-x^{i,iter}}$ and rand() is a random number from uniform distribution [0; 1] and $x^{i,iter}$ is the updated binary position at iter iteration.

5.1. Solution representation and Initialization

Before using the proposed binary ES-CSA to solve UCP, the representation of a crow must be defined. A crow is also called an individual. Hence, we defined each unit on/off (or 1/0) status as a gene, all available unit status at each hour make up a sub-chromosome, and there are H sub-chromosomes over the time horizon H comprising an individual. An individual would display the unit commitment schedule over the time horizon H. The on/off schedule of the units is stored as an integer-matrix U with dimension N G H. A matrix representation of an individual in the population is shown as follows:

$$U = \begin{bmatrix} u_1^1 & u_1^2 & u_1^3 & \dots & u_1^H \\ u_2^1 & u_2^2 & u_2^3 & \dots & u_2^H \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ u_N^1 & u_N^2 & u_N^3 & \dots & u_N^H \end{bmatrix}$$

Where uhi is unit on/off status of unit i at time h (uhi = 1=0 for on/off).

In the initialization process, a set of individuals is created at random. For the complete N P population, the candidate solution of each individual Uj; (j = 1; 2; :::; N P) is randomly initialized. The position uhi of each crow Uj is generated using a uniform distributed random function, which generates either 0 or 1 and they are equally likely.

5.2. Generate new solutions

As mentioned above, the ES is a two-stage strategy, and we can use different algorithms at different stages. In the first stage, ES uses the so-called Levy flights, which represent a kind of non-Gaussian stochastic process whose step sizes are distributed based on a Levy stable distribution to generate new solutions. When a new solution is produced, the following Levy flight is applied:

$$x^{i,iter+1} = x^{i,iter} + \alpha \oplus Levy(\lambda) \quad (16)$$

Here, is the step size that is relevant to the scales of the problem. The product means entry-wise multiplications. Levy flights essentially provide a random walk while their random steps are drawn from a Levy distribution for large steps:

$$\text{Levy}(\lambda) = u = t^{-\lambda}, \quad 1 \leq \lambda \leq 3 \quad (17)$$

In this paper, we will use the Mantegna algorithm [28], which is one of the most efficient algorithms used to implement Levy flights. We assume that Levy (λ) = s, so the formula can also be described as follows:

By using Mantegna's algorithm [26], the step length s can be calculated as follows:

$$x^{i,iter+1} = x^{i,iter} + \alpha \oplus s \quad (18)$$

By using Mantegna's algorithm [26], the step length s can be calculated as follows:

$$s = \frac{\mu}{|v|}{}^{1}\!/_{\beta} \quad (19)$$

Where μ and ν draw from the normal ditrutions respectively .that is : $\mu \sim N(0, \sigma_u^2), \nu \sim N(0, \sigma_v^2)$, and σ_v, σ_u, are calculated as follows $\left(\frac{\Gamma(1+\beta).\sin(\frac{\pi\beta}{2})}{\Gamma(\frac{\beta+1}{2}).\beta.2^{(\frac{1-\beta}{2})}}\right)^{\frac{1}{\beta}}$,$\sigma_v = 1$ Here $0 \leq \beta \leq 2$ and (:) is the Gamma function. For the second stage, we can use the crow search algorithm (CSA) for the intensive local search. We know the CSA is a global search algorithm, but it can easily be tuned to do an efficient local search by limiting new solutions locally around the most promising region. As mentioned above, in the CSA, there are two specific parameters: awareness probability (AP) and flight length (f l). Small values of AP intensify the local search, while large values result in a global search. Hence, the CSA can easily be used as a local optimizer by setting the awareness probability to very small values, and for good performance, we choose the flight length f l = 2. Such a combination may produce better results than those using pure CSA.

In UCP, binary numbers 0 and 1 are used to indicate the unit status (i.e., OFF or ON). The proposed startegy is essentially a real-coded algorithm, and therefore some modifications are needed to enable it to deal with the binary variable (i.e., 0 and 1) optimization problem.

5.3. Lambda-iteration method for ELD problem

With the feasible UC schedule, classical equal lambda-iteration method [4] is used to solve the ELD problem in this paper. The ELD procedure is stopped when the tolerance, which indicates that the sum of all online units output minus the load demand, is less than the value given before hand. Once the optimal values of Pit are found, the total generation production cost is computed by adding the operating cost of all units over the time horizon H. The total start-up cost is calculated by adding the startup costs of those units that change their states from 0 to 1. Lambda-iteration method for solving ELD pseudo-codes is listed in algorithm 3.

5.4. Solution methodology

In this section, the novel binary eagle startegy based crow search algorithm is proposed to solve the UCP and then the lambda-iteration method is used to solve the sub-problem EDP. The main steps are listed as follow:

Step 1: Randomly initialize the parameters (R; AP; f l; Np), feasible vectors and initialize the memory of each crow at t = 0 as in Section 4.4.1
Step 2: Calculate priority list of units according to each unit parameters as in Section 3.3.1
Step 3: Modify unit's status of individuals in the crows satisfying spinning reserve constraints as in Section 3.3.2
Step 4: Repair each crow in the swarm for minimum up/down time violations as in Section 3.3.3.
Step 5: Decommit units of each crow in the swarm to reduce excessive spinning reserve due to minimum up/down times repairing as in Section 3.3.4
Step 6: Solve ELD problem using equal lambda-iteration method as in Section 5.3
Step 7: Fitness evaluation of all crows. Calculate the fitness function value of each agent using the objective function (1) and evaluate each crow in the population.
Step 8: Generate a set of crows X for globale exploration using the Levy flight according to Section 5.5.1 and run steps 3-7
Step 9: Update the memory of crows according to Equation (14)
Step 10: Generate randomly a set of crows around this promising solution
Step 11: Carry out an intensive local search via the crow search algorithm and run steps 3-7
Step 12: Update the memory of crows according to Equation (14)
Step 13: If (a better solution is found) Update the current best

Step 14: If the maximum iteration number is reached, then go to step 15. Otherwise, increase iteration number and go back to step 3.
Step 15: Stop and report the optimal solution of UCP.

Algorithm 3 Lambda-iteration method for solving ELD

Given a feasible solution $U = \left[u_i^h\right]_{NG \times H}$
Set tolerance τ
for $h = 1$ *to* H **do**
  Set initial values for $\lambda(t)$ and $\Delta(h)$
  for $i = 1$ *to* NG **do**
    while (not $|\epsilon| \leq \tau$) **do**
      Calculate P_i^* using $\frac{df_i(P_i^h)}{P_i^h} = \lambda(t)$
      Calculate P_i^h using
      $P_i^h = min\left[max\left[P_i^*, P_{i,min}\right], P_{i,max}\right]$
      Calculate $\epsilon = P_D^h - \sum_{i=1}^{NG} P_i^h u_i^h$
      if ($\epsilon < 0$) **then**
        Set $\lambda(t) = \lambda(t) - \Delta(t)$
      else
        Set $\lambda(t) = \lambda(t) + \Delta(t)$
      end if
      Set $\Delta(t) = \Delta(t)/2$
    end while
  end for
end for

6. RESULTS AND DISCUSSION

In order to verify the feasibility and effectiveness of the proposed method (Binary ES-CSA) for solving UCP, the proposed Binary ES-CSA is tested on different system sizes based on a basic system of 10 units from literature [10]. The scheduling time horizon H is chosen as one day with 24 intervals of one hour each. The spinning reserve requirement is set to be 10% of total load demand. For the systems of 20, 40, 60, 80 and 100 units, the basic 10-unit system is duplicated and total load demands are adjusted proportionally to the system size. The proposed Binary ES-CSA method is coded in MATLAB and implements on an Intel 2.26 GHz CPU with RAM 2GB personal computer. The parameters of proposed algorithm is given in Table 1, for the demand and generating unit data of the test system are given in Tables 2 and 3 .

To validate the results obtained with the proposed ES-CSA method, we compare the performance of the ES-CSA to those of other approaches with respect to the best total production cost and CPU execution time. The results were reported in literature when the same problem was solved using Lagrangian relaxation (LR) (S. A. Kazarlis, A. Bakirtzis, and V. Petridis 1996) [13], integer-coded GA (ICGA) (Damousis et al. 2004) [32], evolutionary programming (EP) (Juste et al. 1999) [14], and Lagrangian relaxation and genetic algorithms (LRGA) (Cheng et al. 2000) [25]. Table 4 provides comparison of the best total production cost from the ES-CSA method to those of other methods. It is clearly shown that the total production costs by the ES-CSA in all test cases are smaller than those of the above methods. From Table 4, it is obvious that the proposed ES-CSA method is superior to the mentioned methods.

The CPU execution times of the ES-CSA and other methods in literature are shown in Table 3. Although they may not be directly comparable due to different computers used, but the trend of computational time is shown that ES-CSA is able to find good optimal solutions in much smaller times than other methods. As shown in Tables 3 and 4, the total production costs of ES-CSA are shown to be less expensive than those of other methods on all generating unit systems. Obviously, ES-CSA vastly improves performance than other methods in terms of both solution quality and CPU times especially on the large-scale UCP.

In the meantime, we examine the variation in the total fuel cost of test system with evolutionary generation numbers. For differenttest systems, the convergence processes of the best solution in the 30 trials are listed in Figures 1 et 2. From Figure 1 and 2, it is easy to see the ES-CSA has satisfactory con-vergence and the algorithm escaped from the local optima at thelater iterations. It proved that the stochastic searching mechanism of ES-CSA, which is conducted by gravitational forces among agents, is efficient. And the proposed mutation strategies improved theperformance of ES-CSA.

Table 1. Parameters of different methods

Algorithms/parameters	AP	fl	β
CSA	0 .2	2	-
ESCSA	0.2	2	1 .5

Table 2. System input data for 10 units, 24 h

Unit	P_{max} (MW)	P_{min} (MW)	a	b	c	t_{up}(h)	t_{down}(h)	S_{hr}(h)	S_{cr}(h)	t_{cold}	$i_{nit.st}$
1	455	150	1000	16.19	0 .00048	8	8	4500	9000	5	8
2	455	150	970	17.26	0 .00031	8	8	5000	10,000	5	8
3	130	20	700	16.6	0.00200	5	5	550	1100	4	-5
4	130	20	680	16.5	0.00211	5	5	560	1120	4	-5
5	162	25	450	19.7	0.00398	6	6	900	1800	4	-6
6	80	20	370	22.26	0.00712	3	3	170	340	2	-3
7	85	25	480	27.74	0.00079	3	3	260	520	2	-3
8	55	10	660	25.92	0.00413	1	1	30	60	0	-1
9	55	10	665	27.27	0.00222	1	1	30	60	0	-1
10	55	10	670	27.79	0.00173	1	1	30	60	0	-1

Table 3. Load data for 10 units, 24 h

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

Table 4. Comparison of the best total production costs ($)

Methods-Number of units	10 TU's	20 TU's	40 TU's	60TU's	80 TU's	100TU's
LR [13]	565825	1130660	2258503	3394066	4526022	5657277
ELR [30]	563977	1123297	2244237	3363491	4485633	5605678
LRGA [29]	564800	1122622	2242178	3371079	4501844	5613127
DPLR [30]	564049	1128098	2256195	3384293	4512391	5640488
GA [13]	565825	1126243	2251911	3376625	4504933	5627437
GACC [31]	563977	1125516	2249715	3375065	4505614	5626514
EP [14]	564551	1125494	2249093	3371611	4498479	5623885
ICGA [32]	566404	1127244	2254123	3378108	4498943	5630838
PLEA [33]	563977	1124295	2243913	3363892	4487354	5607904
EPL [33]	563977	1124369	2246508	3366210	4489322	5608440
LMBSI [34]	563977	1123990	2243708	3362918	4483593	5602844
IPPDTM [35]	563977	-	2247162	3366874	4490208	5609782
QBPSO [36]	563977	1123297	2242957	3361980	4482085	5602486
QEA-UC [37]	563938	1123607	2245557	3366676	4488470	5609550
IQEA-UC [38]	563938	1123297	2242980	3362010	4482826	5602387
SFLA [39]	564769	1123261	2246005	3368257	4503928	5624526
ICA [40]	563938	1124274	2247078	3371722	4497919	5617913
Binary ES-CSA	563938	1123216	2242741	3360316	4480389	5600320

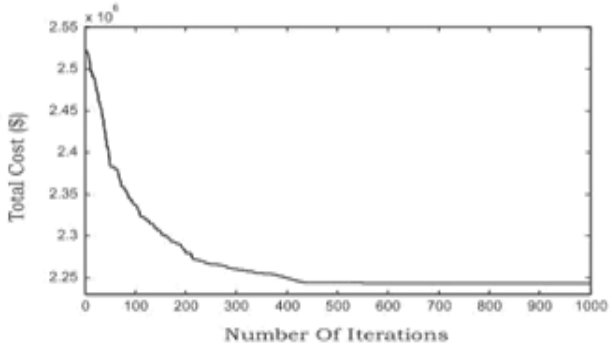

Figure 1. Convergence characteristic of fuel cost using Binary ES-CSA for 10-units based UC problem

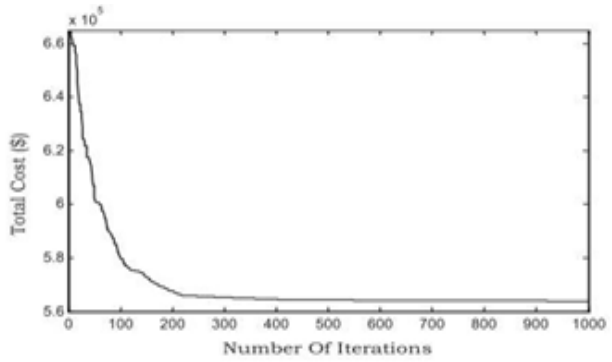

Figure 2. Convergence characteristic of fuel cost using Binary ES-CSA for 40-units based UC problem

7. CONCLUSION

In this paper, we proposed an eagle strategy based crow seach algorithm (ES-CSA) to solve unit commitment problem in smart grid system. The proposed method is a combination of binary eagle strategy based crow seach algorithm with Lambda-iteration method, which is enhanced by priority list to handle the spinning reserve constraints and a heuristic search strategy to handle minimum up/down time constraints.
The simulation results for solving several UCP instances with thenumber of units in the range of 10–100 shows that the proposed method is effective for solving UCP. The total production costs over the scheduled time horizon by ES-CSA are less expensive than any otheroptimization methods reported in the literature especially on the large-scale UCP. Moreover, the CPU times of the proposed method increase linear with the UCP system size, which is favorable for large-scale implementation.

REFERENCES

[1] C.He-Rui, P. Xu, "Study on Smart Grid System Based on System Dynamics", TELKOMNIKA Indonesian Journal of Electrical Engineering Vol. 12, No. 12, pp. 7979-7986, December 2014.
[2] Shahinzadeh H, Hasanalizadeh-Khosroshahi A. "Implementation of Smart Metering Systems: Challenges and Solutions". TELKOMNIKA Indonesian Journal of Electrical Engineering. 2014; 12(7).
[3] Ajenikoko GA, Olabode OE. Optimal Power Flow with Reactive Power Compensation for Cost and Loss Minimization on Nigerian Power Grid System. Indonesian Journal of Electrical Engineering and Informatics. 2017; 5(3): 236-247.
[4] A. J. Wood and B. F. Wollenberg, Power generation, operation, and control. John Wiley & Sons, 2012.
[5] R. Burns and C. Gibson, “Optimization of priority lists for a unit commitment program,” in IEEE Transactions on Power Apparatus and Systems, vol. 94, no. 6. IEEE-INST ELECTRICAL ELECTRONICS ENGINEERS INC 345 E 47TH ST, NEW YORK, NY 10017-2394, 1975, pp. 1917–1917.
[6] G. B. Sheble, “Solution of the unit commitment problem by the method of unit periods,” IEEE Transactions on Power Systems, vol. 5, no. 1, pp. 257–260, 1990.
[7] C. Tseng, “On power system generation unit commitment problems.” 1998.

[8] W. L. Snyder, H. D. Powell, and J. C. Rayburn, "Dynamic programming approach to unit commitment," IEEE Transactions on Power Systems, vol. 2, no. 2, pp. 339–348, 1987.
[9] Z. Ouyang and S. Shahidehpour, "An intelligent dynamic programming for unit commitment application," IEEE Transactions on Power Systems, vol. 6, no. 3, pp. 1203–1209, 1991.
[10] F. Zhuang and F. D. Galiana, "Towards a more rigorous and practical unit commitment by lagrangian relaxation," IEEE Transactions on Power Systems, vol. 3, no. 2, pp. 763–773, 1988.
[11] A. I. Cohen and M. Yoshimura, "A branch-and-bound algorithm for unit commitment," IEEE Transactions on Power Apparatus and Systems, no. 2, pp. 444–451, 1983.
[12] J. A. Muckstadt and R. C. Wilson, "An application of mixed-integer programming duality to scheduling thermal generating systems," IEEE Transactions on Power Apparatus and Systems, no. 12, 1968.
[13] S. A. Kazarlis, A. Bakirtzis, and V. Petridis, "A genetic algorithm solution to the unit commitment problem," IEEE transactions on power systems, vol. 11, no. 1, pp. 83–92, 1996
[14] K. Juste, H. Kita, E. Tanaka, and J. Hasegawa, "An evolutionary pro-gramming solution to the unit commitment problem," IEEE Transactions on Power Systems, vol. 14, no. 4, pp. 1452–1459, 1999.
[15] F. Zhuang and F. Galiana, "Unit commitment by simulated annealing," IEEE Transactions on Power Systems, vol. 5, no. 1, pp. 311–318, 1990.
[16] B. Zhao, C. Guo, B. Bai, and Y. Cao, "An improved particle swarm optimization algorithm for unit commitment," International Journal of Electrical Power & Energy Systems, vol. 28, no. 7, pp. 482–490, 2006.
[17] X.-S. Yang and S. Deb, "Eagle strategy using levy´ walk and firefly algorithms for stochastic optimization," in Nature Inspired Cooperative Strategies for Optimization (NICSO 2010). Springer, 2010, pp. 101– 111.
[18] X.-S. Yang, S. Deb, and X. He, "Eagle strategy with flower algorithm," in Advances in Computing, Communications and Informatics (ICACCI), 2013 International Conference on. IEEE, 2013, pp. 1213–1217.
[19] X.-S. Yang and S. Deb, "Two-stage eagle strategy with differential evolution," International Journal of Bio-Inspired Computation, vol. 4, no. 1, pp. 1–5, 2012.
[20] A. Askarzadeh, "A novel metaheuristic method for solving constrained engineering optimization problems: crow search algorithm," Computers & Structures, vol. 169, pp. 1–12, 2016.
[21] Y.-W. Jeong, W.-N. Lee, H.-H. Kim, J.-B. Park, and J.-R. Shin, "Thermal unit commitment using binary differential evolution," Journal of Elec-trical Engineering and Technology, vol. 4, no. 3, pp. 323–329, 2009.
[22] S. Patra, S. Goswami, and B. Goswami, "A binary differential evolution algorithm for transmission and voltage constrained unit commitment," in Power System Technology and IEEE Power India Conference, 2008. POWERCON 2008. Joint International Conference on. IEEE, 2008, pp. 1–8.
[23] S. Patra, S. Goswami, and B. Goswami, "Differential evolution algorithm for solving unit commitment with ramp constraints," Electric power components and systems, vol. 36, no. 8, pp. 771–787, 2008.
[24] A. S¸. Uyar, B. Turkay¨ and A. Keles¸, "A novel differential evolution application to short-term electrical power generation scheduling," Inter-national Journal of Electrical Power & Energy Systems, vol. 33, no. 6, pp. 1236–1242, 2011
[25] X. Yuan, A. Su, H. Nie, Y. Yuan, and L. Wang, "Application of enhanced discrete differential evolution approach to unit commitment problem," Energy Conversion and Management, vol. 50, no. 9, pp. 2449–2456, 2009.
[26] X. Yuan, A. Su, H. Nie, Y. Yuan, and L. Wang, "Unit commitment problem using enhanced particle swarm optimization algorithm," Soft Computing-A Fusion of Foundations, Methodologies and Applications, vol. 15, no. 1, pp. 139–148, 2011.
[27] A. Askarzadeh, "Electrical power generation by an optimised au-tonomous pv/wind/tidal/battery system," IET Renewable Power Gener-ation, 2016.
[28] D. Oliva, S. Hinojosa, E. Cuevas, G. Pajares, O. Avalos, and J. Galvez,´ "Cross entropy based thresholding for magnetic resonance brain images using crow search algorithm," Expert Systems with Applications, vol. 79, pp. 164–180, 2017.
[29] C.-P. Cheng, C.-W. Liu, and C.-C. Liu, "Unit commitment by lagrangian relaxation and genetic algorithms," IEEE transactions on power systems, vol. 15, no. 2, pp. 707–714, 2000.
[30] W. Ongsakul and N. Petcharaks, "Unit commitment by enhanced adaptive lagrangian relaxation," IEEE Transactions on Power Systems, vol. 19, no. 1, pp. 620–628, 2004.
[31] T. Senjyu, H. Yamashiro, K. Uezato, and T. Funabashi, "A unit commit-ment problem by using genetic algorithm based on unit characteristic classification," in Power Engineering Society Winter Meeting, 2002. IEEE, vol. 1. IEEE, 2002, pp. 58–63.
[32] I. G. Damousis, A. G. Bakirtzis, and P. S. Dokopoulos, "A solution to the unit-commitment problem using integer-coded genetic algorithm," IEEE Transactions on Power systems, vol. 19, no. 2, pp. 1165–1172, 2004.
[33] D. Srinivasan and J. Chazelas, "A priority list-based evolutionary algo-rithm to solve large scale unit commitment problem," in Power System Technology, 2004. PowerCon 2004. 2004 International Conference on, vol. 2. IEEE, 2004, pp. 1746–1751.
[34] I. C. Silva, S. Carneiro, E. J. de Oliveira, J. Pereira, P. A. Garcia, and A. L. Marcato, "A lagrangian multiplier based sensitive index to determine the unit commitment of thermal units," International Journal of Electrical Power & Energy Systems, vol. 30, no. 9, pp. 504–510, 2008.
[35] K. Chandram, N. Subrahmanyam, and M. Sydulu, "Unit commitment by improved pre-prepared power demand

table and muller method," International Journal of Electrical Power & Energy Systems, vol. 33, no. 1, pp. 106–114, 2011.
[36] Y.-W. Jeong, J.-B. Park, S.-H. Jang, and K. Y. Lee, "A new quantum-inspired binary pso: application to unit commitment problems for power systems," IEEE Transactions on Power Systems, vol. 25, no. 3, pp. 1486–1495, 2010.
[37] T. Lau, C. Chung, K. Wong, T. Chung, and S. Ho, "Quantum-inspired evolutionary algorithm approach for unit commitment," IEEE Transac-tions on Power Systems, vol. 24, no. 3, pp. 1503–1512, 2009.
[38] C. Chung, H. Yu, and K. P. Wong, "An advanced quantum-inspired evolutionary algorithm for unit commitment," IEEE Transactions on Power Systems, vol. 26, no. 2, pp. 847–854, 2011.
[39] J. Ebrahimi, S. H. Hosseinian, and G. B. Gharehpetian, "Unit com-mitment problem solution using shuffled frog leaping algorithm," IEEE Transactions on Power Systems, vol. 26, no. 2, pp. 573–581, 2011.
[40] M. M. Hadji and B. Vahidi, "A solution to the unit commitment problem using imperialistic competition algorithm," IEEE Transactions on Power Systems, vol. 27, no. 1, pp. 117–124, 2012.