The 1st International Conference on Local Resource Exploitation
www.lorexp.org / info@lorexp.org
REF: LOREXP_2021_A1150 Pages: 914–929

A new parallel ant colony algorithm to solve the unit commitment problem

Un nouvel algorithme de colonie de fourmis parallèle pour résoudre le problème d'engagement des unités

Henri-Joël Akoue [1, 2,*], Pascal Ntsama Eloundou [1], Salomé Ndjakomo Essiane [3,4], Pierre Ele [5], Yves Effa [1], Benjamin Salomon Diboma [2], Mounchili Ndam Njoya Idriss[2]

[1] Department of physics, Faculty of science, The University of Ngaoundéré, Ngaoundéré, Cameroon
[2] Electrical Engineering Department, Advanced Teacher's Training College for Technical Education of Douala, University of Douala, Douala, Cameroon
[3] Laboratory of Technologies and Applied Sciences, University of Douala, Cameroon
[4] Electrical Engineering Department, Advanced Teacher's Training College for Technical Education of Ebolowa, University of Yaoundé, Yaoundé, Cameroon
[5] Electrical Engineering and Telecommunications Department, National Advanced School of Engineering, University of Yaoundé 1, Yaoundé, Cameroon
* Corresponding Author: akoueh@gmail.com

ABSTRACT:

The Ant Colony Optimization Algorithm is a metaheuristic that from its first version, produced good results in solving the traveling salesman problem. Since, this algorithm has been successfully applied to several other problems. However, its major drawbacks such as stagnation in a local optimum and its slowness for large data sizes, have led to the development of new versions such as modified versions, hybrid versions as well as parallel versions. In this paper, we propose a new parallel ant colony algorithm named MAC-UCP/MMAS (Multy Ant Colony for Unit commitment Problem based on MAX-MIN Ant System) in order to solve the unit commitment problem in a thermal power generation system. For this we develop a task parallelism using several ant colonies. The implementation of the algorithm on the multiprocessor system was done in a MIMD architecture through the SPMD computational model using explicit message passing for communication between processors. The algorithm is implemented in MATLAB software environment for two thermal unit systems, 4 and 10 generating units taken respectively over 8 and 24 hours. Thus, an increase of numbers of colonies from 1 to 6 is done to observe the behaviour of our MAC-UCP/MMAS algorithm. Results obtained shows improved solution compared to sequential MMAS, Modified Ant Colony Optimization (MACO), particle Swarm Optimization combined with Lagrange Relaxation (PSO-LR), Swarm and Evolutionary Computation (SEC), Particle Swarm Optimization combined with Genetic Algorithm (PSO-GA), Binary Grey Wolf Optimizer (BGWO). Compared to some other methods, the results show for the case of 6 colonies, a maximum coefficient of variation (change) of the total production cost of 0.168 % for the system of 10 units, and 2.37561 % for the system of 4 units. Likewise, for 6 colonies, a maximum acceleration of 2.4154 is obtained for the system of 10 units, and 2.6489 for the system of 4 units.

Keywords: Parallel ant colony algorithm, Multy Ant Colony, Unit commitment, Task parallelism, MIMD, SPMD

RÉSUMÉ :

L'Optimisation par l'algorithme de la colonie des fourmis est une métaheuristique qui dès sa première version a fait ses preuves en produisant des bons résultats dans la résolution du problème du voyageur de commerce. Depuis lors, cet algorithme a été appliqué avec succès à plusieurs autres problèmes. Cependant ses défauts majeurs tels que la stagnation dans un optimum local et sa lenteur pour des grande tailles de données, ont conduit au développement de nouvelles versions telles que les versions modifiées, les versions hybrides ainsi que les versions parallèles. Dans cet article, nous proposons un nouvel algorithme parallèle de la colonie des fourmis nommé MAC-UCP/MMAS (Multy Ant Colony for Unit commitment Problem based on MAX-MIN Ant System) pour la résolution du problème d'engagement des unités dans un système d'unités thermiques. Pour cela nous développons un parallélisme de tâches utilisant quelques colonies de fourmis. L'implémentation de l'algorithme sur un système multiprocesseur s'est faite dans une architecture MIMD à travers le modèle de calcul SPMD utilisant entre processeurs une communication explicite par passage de message. L'algorithme est implémenté dans l'environnement logiciel MATLAB pour deux systèmes d'unités thermiques, 4 et 10 unités de production prises respectivement sur 8 et 24 heures. Ainsi, le comportement de l'algorithme proposé est observé en faisant varier le nombre de colonies de 1 à 6. Les résultats obtenus montrent une amélioration de la solution lorsque l'algorithme est comparé à quelques algorithmes existant tels que MMAS séquentielle, MACO, PSO-LR, SEC, PSO-GA, BGWO. Comparé aux autres méthodes, les résultats ressortent pour le cas de 6 colonies, un coefficient de variation maximal du coût total de production de 0,168 % pour le système de 10 unités, et 2.37561 % pour le système de 4 unités. De même, pour 6 colonies, une accélération maximale de 2.4154 est obtenue pour le système de 10 unités, et 2.6489 pour le système de 4 unités.

Mots clés : Algorithme parallèle de la colonie des fourmis, Multy Ant Colony, Unit commitment, Parallélisme de tâches, MIMD, SPMD

1. INTRODUCTION

The unit commitment problem is a combinatorial optimization problem that consists of planning the switching schedule of a set of production units. Moreover, it allows to determine over a precise planning period the power that these units should produce in order to meet energy demand while respecting the economic or environmental constraints imposed. It is therefore a problem which involves both binary variables and continuous variables and for which several solutions are proposed in the literature. Unit commitment problem was considered for the first time by Lowery in 1966 through dynamic programming in order to overcome the major difficulty of enumerative methods. Indeed, the principle behind these methods is to test all possible combinations of supply with the units considered, which may require significant resources for a large number of units.

Nevertheless, many authors had proposed various methods to solve unit commitment, neural network (Jahromi et al., 2013), fuzzy logic (Zhang et al., 2015), improved simulated annealing particle swarm optimization (Zhai et al., 2020), particle swarm optimization (Khatibi et Bigdeli, 2014), artificial bee colony algorithm (Sharma et al., 2015), Binary whale optimization algorithm (Kumar et Kumar, 2020),binary-coded genetic algorithm with a particle swarm optimization (Postolov et Iliev, 2020), and ant colony algorithm (Zand et al., 2016).

Ant colony optimization algorithm it is a metaheuristic method which was introduced in 1991. It is inspired by the behaviour of ant when searching their food. This algorithm is focused on artificial ants, building their solutions in a given optimization problem and exchanging the quality of their solutions by a mechanism inspired from the behavior of real ants (Dorigo et al., 1991). Ant Colony algorithms is widely used for their great flexibility due to their distributed and adaptive nature which gives them average performance in the static case, but seem more suited to dynamic problems (Bonabeau et al., 1999). Their major drawback of its slowness of convergence, in particular when solving large scale problems. This can lead either to the best solution being obtained beyond the desired timeframe, or to an unsatisfactory solution being obtained within the desired timeframe. Depending on the applications and therefore the requirements, the authors' interest may relate either to the quality of the solution to the detriment of time (Bonabeau et al., 1999) or to the gain in time to the detriment of the quality of the solution (Twoney et al., 2010), or even both solution and time.

In order to improve the performance of the ACO algorithms, several techniques are used in literature, among which the modification of the existing algorithms (Ameli et al., 2011; Zand et al., 2016), the hybridization of algorithms and the parallelization (Soh et al., 2020;Yu et al., 2010; Dorigo, 1992). Parallelization is one of the most efficient techniques and several parallel ACO algorithms have been deployed (Soh et al., 2020) since the first suggestion of parallelization made by Dorigo (Dorigo, 1992). In 2011, Pedemonte et al. suggested a taxonomy of five kinds of parallelizations of ACO, the master-slave model, the cellular model, the parallel independent runs model, the multicolony model and the hybrid model. In the master-slave parallelization, a « master » process lead the interaction of globals informations (pheromone matrix, solutions, etc...) between « slaves » and these ones execute their tasks (definition of search space) as received from the « master ». Besides, in the cellular model, only one colony is splitted into small quarters with their respective pheromone matrix (Pedemonte et al., 2011). In the case of parallel independent runs model, a given number of sequential algorithm having or not the same parameters is executed independently in many process. The multicolony model allows several colonies to explore simultaneously the search space and exchange informations in a given frequency in order to find the optimal solution. The hybrid models combined at least two out of the four previous models.

Several authors have applied these techniques for various problems. Out of 65 proposals listed by Pedemonte et al. (2011) on the parallel implementation of the ant colony algorithm between 1998 and 2010, we have noticed that none addressed has been the issue of unit commitment. And even today, this issue is not commonly addressed with parallelization in the literature. Some authors have been interested in the parallelization of algorithms to solve the unit commitment problem (UCP) (Dupin et Talbi, 2020; Kargarian et al., 2018; Cong et al., 2015), but none were interested to the case of Ant Colony Optimization which should however attract our attention for reasons of slowness mentioned above. This is the case for systems with a high number of production units, with several constraints to be observed for sometimes very short horizons imposed by strong load dynamics.

In order to improve the quality of the solution obtained with the classic ACO algorithms for the UCP not only in terms of computation time but also in terms of costs, this paper propose a new parallel Ant Colony algorithm for solving the thermal unit commitment problem. For that, the economic dispatch problem is solved by quadratic programming and the on/off schedule for the units over the planning period is solved by using multicolony approach of MMAS in unidirectional ring connection topology. The work is organized as follow: Section 2 is a short review on the unit commitment and the Max-min ant system algorithm. Section 3 present the materials and methods used, it is in this section that we present the proposed MAC-UCP/MMAS (Multy Ant Colony for Unit commitment Problem based on MAX-MIN Ant System) algorithm. Then section 4 is devoted to the presentation of the results and finally the conclusion is made in the section 5.

2. THE UNIT COMMITMENT AND MAX-MIN ANT SYSTEM ALGORITHM

2.1. Unit commitment

Unit commitment problem is an optimization problem whose aim is to minimize the production cost by committing available units within their constraints taken over a period (Kumar et Kumar, 2020; Dupin et Talbi, 2020; Habachi et al., 2019). The total production cost is the sum of the production cost, the startup cost and shut down cost of all the committed units. Thus, the formulation of UCP involves the objective function and various constraints. In this study, we consider a mono-objective formulation approach by considering the production cost as the only optimization criterion.

2.1.1. Objective function

The objective function is expressed as (Panwar et al., 2018; Zhao et al., 2018; Lai et al., 2012; Simon et al., 2006):

$$min\ (\sum_{i=1}^{N}\sum_{t=1}^{T} F_i(P_i(t))U_i(t) + ST_i(t)U_i(t) + SD_i(1-U_i(t))U_i(t-1)) \tag{1}$$

Where $F_i(P_i(t)) = a_i + b_i.P_i(t) + c_i.P_i(t)^2$ (2)

$$ST_i(t) = \begin{cases} HSC_i\ , si\ T_{min,i}^{off} \le T_i^{off} \le T_{min,i}^{off} + SC_i \\ CSC_i\ , si\ T_i^{off} > T_{min,i}^{off} + SC_i \end{cases} \tag{3}$$

i is the unit identification number ; N is the total number of units; T denotes the period of scheduling ; $F_i(P_i(t))$ is the fuel cost of the unit i at the time t when the unit generates a power $P_i(t)$; $U_i(t)$ represent the status of unit i at the time t; $ST_i(t)$ *and* $SD_i(1-U_i(t))$ are respectively the startup and shut down cost of unit i at the time t; a_i, b_i and c_i are fuel costs coefficient of unit i.

$T_{min,i}^{off}$ is the minimum down time of unit i; SC_i is the number of cold-start hours of unit i;

HSC_i and CSC_i are respectively the hot startup cost and cold startup cost of unit i.

2.1.2. Constraints

In order for the power system modeling to be much more realistic and reliable, some constraints must be taken into account in unit commitment problem (Zhai et al., 2020).We present here the four constraints which accompany the minimization of the objective function. More specifically, we present the load demand constraints, constraints related to spinning reserve, constraints relating to the production limits of each unit and constraints relating to minimun up and down time of each unit.

2.1.3. Load demand constraints and spinning reserve

In an electrical energy supply system, production must constantly balance demand. Any sudden drops in production can be seen while supplying the load. This can happen by prediction deviation in a real-time supply or even during a failure of one or more production units in operation (Li et al., 2013). Besides, a way to minimize such effects and to balance the losses quickly is to provid spinning reserve for the demand load at that time. This can be achieved by taking into account the spinning reserve R_t in the balance inequality of load demand constraints.

$$\sum_{i=1}^{N} U_i(t)p_i(t) \geq R_t + D_t \text{ , } t \in \{1, \dots \dots, T\} \quad (4)$$

D_t represents the load demand at the time t

2.1.4. Constraints relating to the production limits of each unit.

Due to the characteristics of each generating unit i the generated power is bounded by two limits, the lower limit denoted $Pmin_i$ and the upper limit $Pmax_i$. Thus we have:

$$Pmin_i \leq p_i(t) \leq Pmax_i . U_i(t), t \in \{1, \dots \dots, T\} \quad (5)$$

2.1.5. Constraints relating to minimum up and down time of each unit

The minimum start-up time is the time after which a unit can be stopped after it has been started. Likewise, the minimum shutdown time is the time after which a unit can reliably be considered shutdown and stable for a possible restart. These conditions are achieved by:

$$T_i^{on} \geq T_i^{up} \text{ et } T_i^{off} \geq T_i^{down} \quad (6)$$

where T_i^{up} and T_i^{down} are respectively the minimum up time and the minimum down time of unit i.

2.2. Max-Min Ant System algorithm (MMAS)

Created in 1991, the first and original version of ACO called Ant system (AS) was applied for the first time to traveller salesman problem (Dorigo et al., 1996; Dorigo et al., 1991). Morever the results obtained from this version was not competitive compared to other algorithms. As matter of fact, several improved versions came out. Out of them, we have ACS (Ant Colony System)version (Dorigo et al., 1997) , MMAS (MAX-MIN Ant System) version (Stützle et Hoos, 1996) and others. In fact, AS follows proportional random transition rule (Dupin et Talbi, 2020); the pheromone are deposit and evaporated in each path proportionally to the length of the path. Ant colony System (ACS) version use a pseudo-random transition rule and the pheromones are deposited and evaporated only on best solution (Pedemonte et al., 2011). MAX-MIN Ant System (MMAS) is an improved version developed by Thomas Stützle in 1997 and apply to some others few optimization problems (Stützle et Hoos, 2000, 1998). In this paper, MMAS is used as our basic sequential algorithm principally because it has the capacity of avoiding stagnation while others ACO versions fail to it. Furthermore other explanations come that the pheromone is bounded between a minimal value τ^{min} and a maximal value τ^{max}. Another motivation is that in MMAS only best ants are allowed to update the pheromone in their path, thus this yields a best exploitation of solutions found in each iteration of algorithm.

Let us recall here the transition rule and the pheromone update rule of MMAS.

2.2.1. *MMAS state transition rule*

In the MMAS algorithm, ants build a solution in a probabilistic way step by step by using information related to pheromone and specifics heuristics information's of the given problem. Thus, the probability for an ant *k* to moves from discrete state *i* to *j*, is given by equation (7).

$$P_{i,j}^{m} = \frac{\tau_{i,j}^{\alpha}.\eta_{i,j}^{\beta}}{\sum_{m=1}^{c}\tau_{i,m}^{\alpha}.\eta_{i,m}^{\beta}} \tag{7}$$

With α, β represents respectively the relative importance of intensity and visibility

.$\eta_{i,j}$ is visibility of the solution;

τ_{ij} is pheromone intensity of the path;

c is the number of ants

2.2.2. *MMAS pheromone updating rule*

The pheromone update is done after each iteration. The update rule in each path is given by equation (8).

$$\begin{cases} \tau_{i,j} \leftarrow (1-\rho).\tau_{i,j} + \Delta\tau_{i,j}^{best} \text{ if } \tau^{min} < \tau_{i,j} < \tau^{max} \\ \tau_{i,j} \leftarrow \tau^{max} \text{ if } \tau_{i,j} > \tau^{max} \\ \tau_{i,j} \leftarrow \tau^{min} \text{ if } \tau_{i,j} < \tau^{min} \end{cases} \tag{8}$$

$\Delta\tau_{i,j}^{best}$ is defined by :

$$\Delta\tau_{i,j}^{best} = \begin{cases} \frac{1}{L_{best}} \text{ if the path } (i,j) \text{ is the best} \\ \quad \text{amongst the solution} \\ 0 \quad elsewhere. \end{cases} \tag{9}$$

L_{best} : best solution cost

3. MATERIALS AND METHODS

In this section, we present the materials and methods which allowed us to achieve the objectives of this work. Programming and simulation have been done in Matlab software environment version 9.2.0.538062 (R2017a) on a computer Intel Xeon 12 CPUS, 2.60 GHz, and 32 Go of RAM. The operating system installed is Windows 10 professional 64 bits (10.0, version 19041). The implementation of the algorithm on the multiprocessor system was done in a MIMD (Multiple Instruction, Multiple Data) architecture through the SPMD (Single Program-Multiple Data) computational model using explicit message passing for communication between processors. Two different systems of data were chosen to solve unit commitment problem because they appear commonly in the literature to valid the result of such this kind problem. We have test system 1 composed by 4 units on 8 hours (Khanmohammadi et al., 2010) and test system 2 composed by 10 units on 24 hours (Simon et al., 2006).

The choice made makes it possible to sufficiently test the algorithm. The parameters of the algorithm are as follow: for the 4 units system's test 1, m = 230, α = 0.8445, β = 10.5, ρ = 0.61, Q = 0.9. Then for the 10 units system test 2 simulations parameters yield: m = 230, α = 0.8445, β = 10.5, ρ = 0.61, Q = 0.9. Moreover, the total number of algorithm's iteration is 400 for the first system and 700 for the second system. The maximal number of colonies is 6 and they exchange informations every 40 iterations of the algorithm.

The proposed algorithm deals with the parallelization. Through the multicolony execution, we implement a task parallelism of which it is necessary to specify the basic essential algorithms. The presentation will be done in two phases. The first step is to present MMAS applied to thermal unit commitment problem and the second step consist to develop our parallel multicolony algorithm based MMAS.

To evaluate the parallel performance of the algorithm, the speed-up or acceleration ratio $S(p)$ and parallel efficiency $E(p)$ were respectively evaluated using formulas (10) and (11).

$$S(p) = \frac{T(1)}{T(p)} \tag{10}$$

$$E(p) = \frac{S(p)}{p} \tag{11}$$

$T(1)$ is the mean execution time of the sequential algorithm;

$T(p)$ is the mean execution time of the parallel algorithm with p colonies;

3.1 MAX-MIN Ant System for UCP

The algorithm can be divided in 5 steps: the definition of search space, economic dispatching of power by using quadratic programming algorithm, initialization, exploration of search space and pheromones update.

3.1.1. Search space definition

Firstly, all possible combinations of UCP are found in the form of binary variables by using exhaustive enumeration. Thus, for a system of 4 units we will have 16 combinations and for a system of 10 units, 1024 combinations. Furthermore, for each period all states that their power cannot satisfy load and spinning reserve are eliminated; as matter of fact, the reminding state are used to build our ant search space as shown in Figure 1.

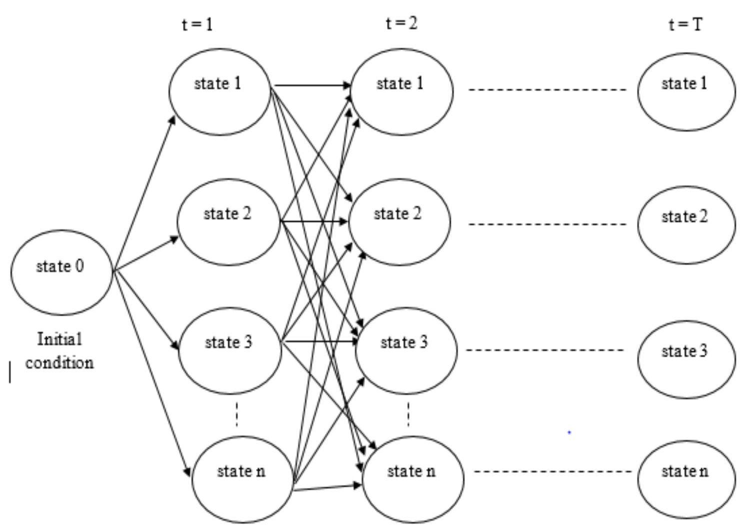


Figure 1. Ants Search space

3.1.2. Economic load dispatch

Once the search space is established, quadratic programming is used to realize Economic Dispatching (ELD) for each state in each period of scheduling. This is done by taking into account the characteristics of each unit, the load demand and the spinning reserve.

After the realization of ELD, for each state at any time on the previous search space of Figure 1, an optimal combination of actives/inactives of units is associated.

3.1.3. Initialisation

Appropriate parameters for the algorithm are well defined in initialization step. Out of them, we have: the number of ants (m), the relative importance of pheromone (α), the relative importance of visibility (β), the evaporation coefficient (ρ), as well as the initial, maximum and minimum quantities of pheromones on each trail respectively τ_0, $\tau_{\max}$ and $\tau_{\min}$. Considering that the initial quantity of pheromone is set at $\tau_{\max}$, the quantity of deposit pheromone are given by the following relationships (Lai et al., 2012; Stützle et Hoos, 2000, 1998):

$$\tau_0 = \tau_{max} = \frac{1}{\sum_{t=1}^{T} minF_t(D_t)} \tag{12}$$

$$\tau_{min} = \frac{\tau_{\max}\left(1-\sqrt[n]{P_{best}}\right)}{(\text{avg}-1)\sqrt[n]{P_{best}}} \tag{13}$$

$F_t(D_t)$ is the cost solution vector associated with a Demand power D_t at time t;

$\sum_{t=1}^{T} minF_t(D_t)$ is the sum of points with the smallest generating cost in each period;

avg is the average number of solutions from which each ant chooses;

Where P_{best} is the probability of finding the optimal solution when the MMAS algorithm converges, which is generally 0.05 (Pan et al., 2020).

3.1.4. Exploration of search space

In this step, each ant explore the search space looking for the best solution as possible. Every ant starts with a minimal cost in the first hour till the last hour; the transition rule is given by equation (7).

In each transition, constraints related to minimum up time and minimum down time are set. If these constraints are fufilled, then startup cost are calculated, if they are not fufilled startup cost are sent to infinity.

At the last hour, the total production cost of the solution found is calculated and saved. The total production cost takes into account the fuel costs and the startup costs. We repeat the procedure to all the ants and after comparison we save the best ants solution.

3.1.5. Pheromones update rule

This operation consists of reinforcing the pheromone tracks associated with promising solutions and, on the contrary, degrading by "evaporation" that associated with bad solutions. The pheromone update rule is given by equation (8).

3.1.6. MMAS algorithm for solving UCP

The flowchart of the proposed algorithm is shown in Figure 2 and summarizes all these steps.

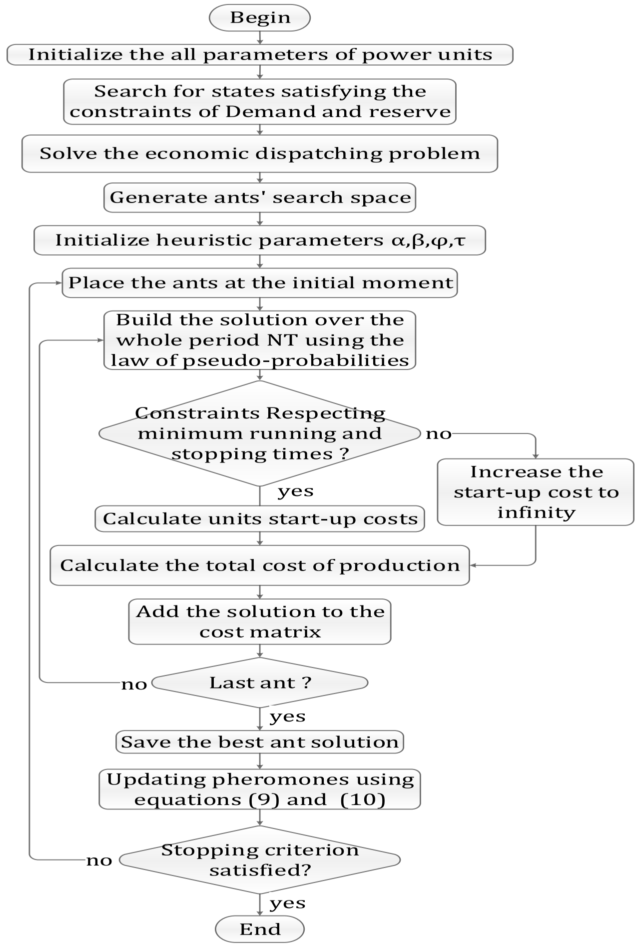


Figure. 2. MMAS algorithm applied to solve UCP

3.2 Multi Ant Colony parallel algorithm for UCP based MMAS

To parallelize the MAX-MIN Ant System for UCP we adopt the multicolony parallelization model. The main interest of this method lies in the communication between the different colonies. To this, many studies have been carried on the communication strategies between colonies (Sriyanyong et Song, 2005). Those strategies are focused on elements such as the nature of exchange information, the exchange frequency or the communication strategy. Even if this communication mechanism requires expensive buffer copying and increases synchronization costs, it is easy to understand with their send/receive primitives (Kandemir et al., 2000). The main motivation for choosing this mechanism is linked to the sharing of solutions between colonies.

To present our new algorithm, we divided it into 3 steps such as the search space generation, the exploration of search space and the exchange of information.

3.2.1. Search space generation

In this step, the search space of ants for the scheduling presented in Figure 1 is generated. It is composed of all combinations of generators able to satisfy load demand and spinning reserve by taking into account the various constraints. Then for each state of the search space, the Economic Dispatching problem is solved by using quadratic programming throughout the minimal production cost of each state is obtained. In fact, we suppose in the study that the search space generated is the same for all the colonies.

3.2.2. Exploration of search space

Once the search space is defined, ants are groups in colony and each coloniy are assigned to p available process. All the colonies have the same characteristics (α ,β, γ, ρ) and use the same basic version of MMAS algorithm. Thus, the exploration phase yields the same procedure for all the colonies.

3.2.3. Informations Exchange

In this step, each colony exchange with his neighbour the information of the best ant (the ant that have found the optimal cost). This step takes place after a predefined number of iterations of the algorithm. For our algorithm, we have adopted between the p available colonies a one-way ring communication shown in Figure. 3. Thus for all the I itérations, each colony $C_{(k)mod_p}$send to $C_{(k+1)mod_p}$the informations of the ant that has found the less production cost and receive the one of the colony $C_{(k-1)mod_p}$.If the received solution is better than the one that the colony had, then that former solution is replaced by the received ones which will be used in the update pheromone stage in the next iteration.

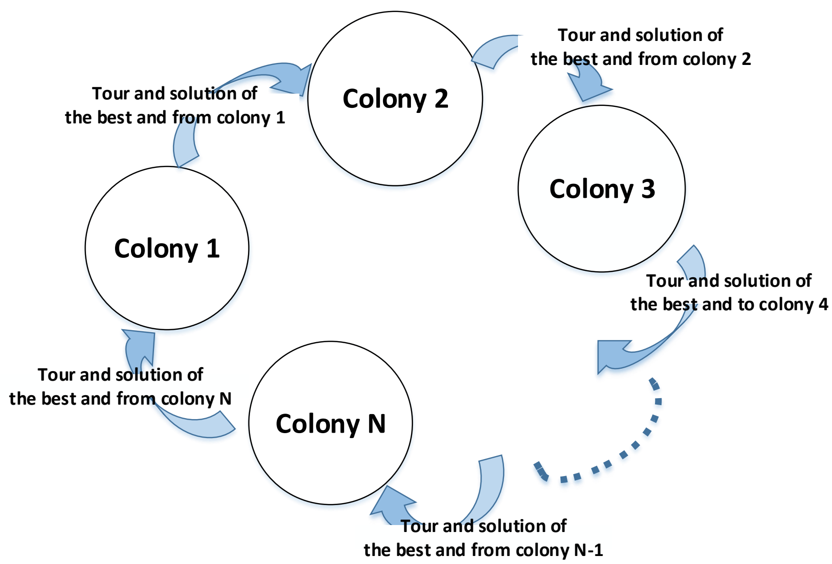


Figure 3. Inter-colony information exchange for N colonies

3.3 MAC-UCP/MMAS algorithm for UCP

The flowchart in Figure 4 summarizes our MAC-UCP/MMAS algorithm. Figure 5. is an overview of the parallel based on multiple ant-colonies in which workers are processors. It should be noted here that we can distinguish two major phases of execution of our MAC-UCP/MMAS algorithm for the unit commitment problem: the sequential execution phase which groups the steps which go from 1 to 4, and the parallel execution phase for steps 5 to 7.

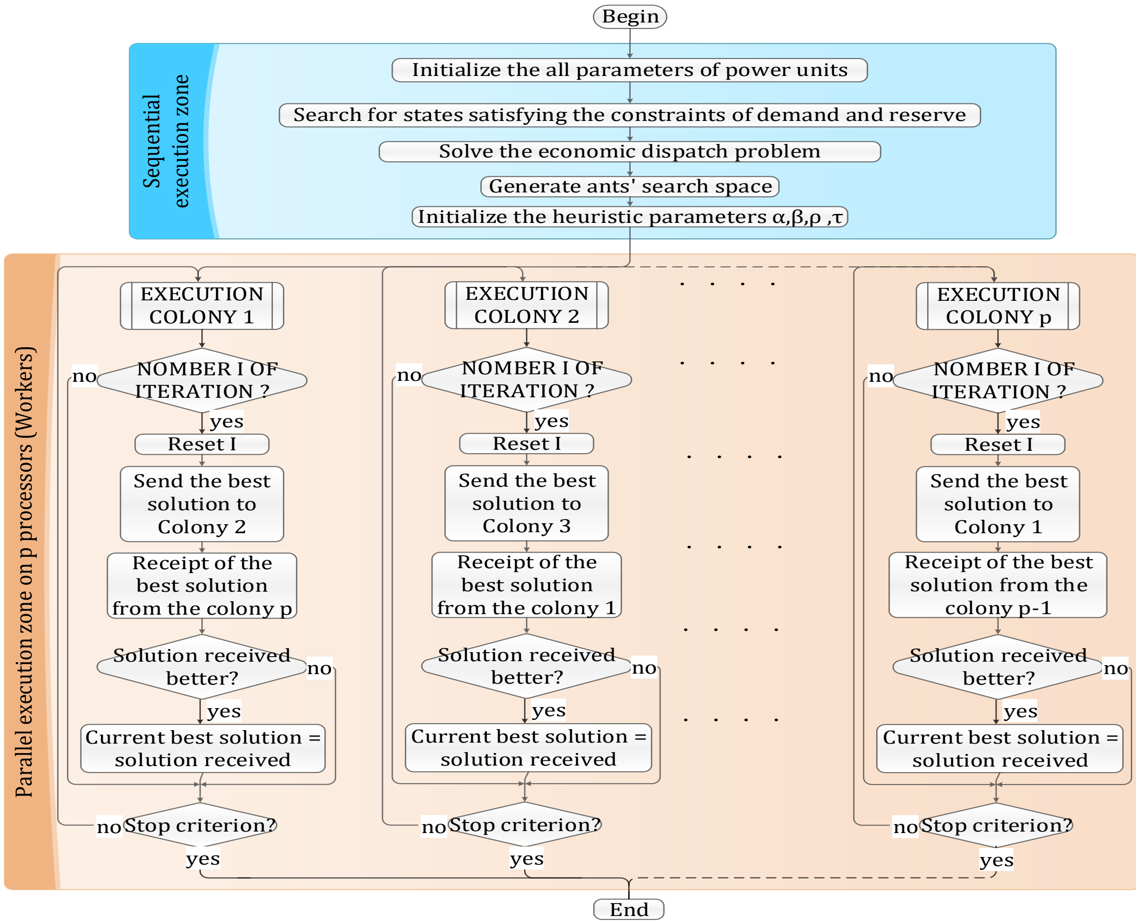


Figure 4. MAC-UCP/MMAS algorithm

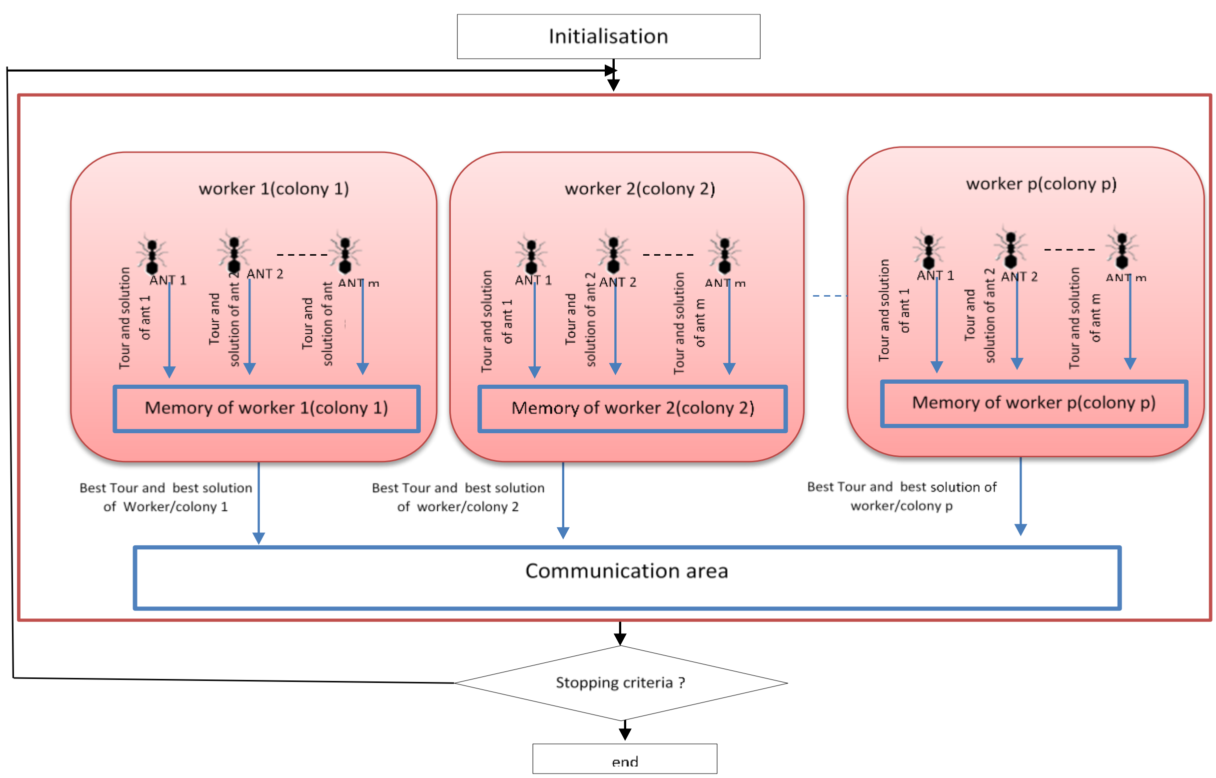


Figure 5. Schematic overview of the parallel based on multiple ant-colonies.

4. RESULTS AND DISCUSSIONS

In this section, we present the results that come out of our various simulations of proposed parallel MAC-UCP/MMAS algorithm.

Tables 1 and 2 show the results of the units switching on/off programs , the powers generated by each generator over the planning period as well as the cumulative costs, respectively for the systems of 4 units and 10 units. We can extract from these tables the best total cost obtained on test system 1 for the 8 hours of the planning horizon, namely $ 73,444.69. For test system 2 the total cost of production including the running cost of the generators and their start-up costs over the 24 hours period is $ 83351.372. In Tables 3 and 4 we present the results obtained when we vary the number of colonies respectively for the systems of 4 and 10 units. For each number of colonies, 10 consecutive runs of the algorithm were performed. It is therefore clear that increasing the number of colonies significantly reduces the total cost of production, as well as the range of variation of the result.

In order to better highlight this, we also plotted the cost difference between the MAC-MMAS version and the sequential MMAS version as the number of colonies increases. As can be seen in Figure 6 for both systems, it appears that the cost savings compared to the sequential version of the algorithm increases with the number of colonies used.

Table 1. UCP Results with MAC-UCP/MMAS for 4-units system

Hour	Demand (MW)	Status of Units 1 2 3 4	Power generated for each unit (MW) 1	2	3	4	Total Power Generated (MW)	Fuel Cost ($)	Transition Cost ($)	Total cumulative Cost ($)
1	450	1 1 0 0	300	150	0	0	450	9109.360	0	9109.360
2	530	1 1 0 0	300	230	0	0	530	10593.040	0	19702.400
3	600	1 1 0 1	300	250	0	50	600	12412.860	0.02	32115.280
4	540	1 1 0 0	300	240	0	0	540	10782.280	0	42897.560
5	400	1 1 0 0	276.1905	123.809	0	0	400	8205.360	0	51103.349
6	280	1 1 0 0	196.1905	83.8095	0	0	280	5525.780	0	57170.50
7	290	1 1 0 0	202.8571	87.149	0	0	290	5706.050	0	63414.33
8	500	1 1 0 0	300	200	0	0	500	10030.360	0	73444.69

Table 2. Results provided by MAC-UCP/MMAS for 10-Units system

HOUR	DEMAND (MW)	UNITS 1 2 3 4 5 6 7 8 9 10	CUMULATIVE TOTAL COSTS ($)
1	1170	1 1 1 1 1 1 1 1 0 1	2717.524
2	1250	1 1 1 1 1 1 1 1 0 1	5323.400
3	1380	1 1 1 1 1 1 1 1 1 1	8304.781
4	1570	1 1 1 1 1 1 1 1 1 1	11600.605
5	1690	1 1 1 1 1 1 1 1 1 1	15179.271
6	1820	1 1 1 1 1 1 1 1 1 1	19085.683
7	1910	1 1 1 1 1 1 1 1 1 1	23232.089
8	1940	1 1 1 1 1 1 1 1 1 1	27461.806
9	1990	1 1 1 1 1 1 1 1 1 1	31839.993
10	1990	1 1 1 1 1 1 1 1 1 1	36218.180
11	1970	1 1 1 1 1 1 1 1 1 1	40535.245
12	1940	1 1 1 1 1 1 1 1 1 1	44764.962
13	1910	1 1 1 1 1 1 1 1 1 1	48911.368
14	1830	1 1 1 1 1 1 1 1 1 1	52843.915
15	1870	1 1 1 1 1 1 1 1 1 1	56882.319
16	1830	1 1 1 1 1 1 1 1 1 1	60814.866
17	1690	1 1 1 1 1 1 1 1 11	64393.532
18	1510	1 1 1 1 1 1 1 1 1 1	67554.400
19	1420	1 1 1 1 1 1 1 1 1 1	70519.636
20	1310	1 1 0 1 1 1 1 1 1 1	73271.306
21	1620	1 1 0 1 1 1 1 1 1 1	75885.569
22	1210	1 1 0 1 1 1 1 1 1 1	78394.308
23	1250	1 1 0 1 1 1 1 1 1 1	80987.321
24	1140	1 1 0 1 1 1 1 1 1 1	**83351.372**

Table 3. Impact of the number of colonies on the solution (4-units system)

Number of colonies	Best cost ($)	Worst cost ($)	Mean($)	Variation (%)	Gap with one colony version (%)
1(sequential MMAS)	73513.8700	74232.3582	73839.1315	0.9730	0
2	73444.6857	74105.3582	73821.5701	0.8949	0.0238
3	73444.6857	73797.6571	73645.0543	0.4793	0.2635
4	73444.6857	73534.4575	73501.0713	0.1221	0.4600
5	73444.6857	73513.8734	73492.7821	0.0941	0.4713
6	73444.6857	73444.6857	73444.6857	0	0.5371

Table 4. Impact of the number of colonies on the solution (10-units system)

Number of colonies	Best cost ($)	Worst cost ($)	Mean ($)	Variation (%)	Gap with one colony version (%)
1	83371.2087	83495.9491	83425.4416	0.1495	0
2	83351.3723	83373.215	83368.9351	0.0262	0,0678
3	83351.3723	83369.439	83363.1257	0.0217	0.0747
4	83351.3723	83366.0031	83357.5375	0.0175	0.0815
5	83351.3723	83351.3723	83351.3723	0	0.0889
6	83351.3723	83351.3723	83351.3723	0	0.0889

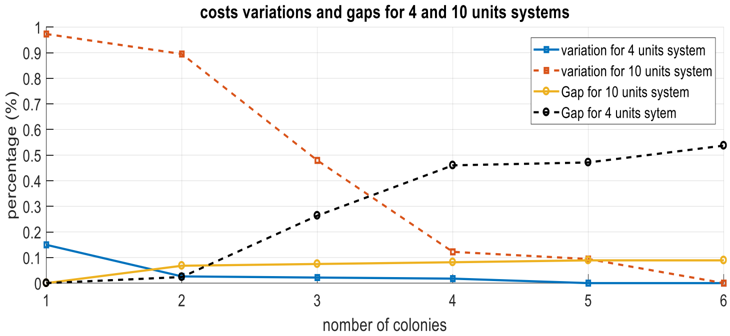
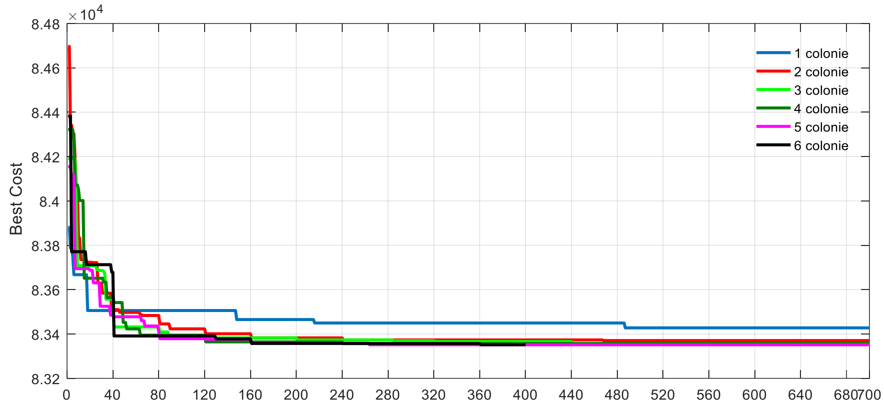


Figure 6. Cost variations and gaps for 4 and 10 units systems using MAC-UCP/MMAS and sequential MMAS

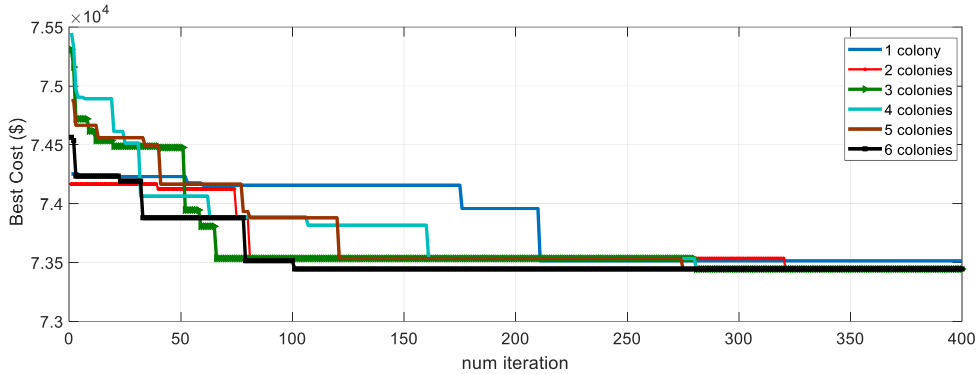
Figure 7 shows for the system of 10 units and 4 units, the different convergence curves of the algorithm as a function of the number of colonies. It is clear from this figure that the algorithm converges faster as the number of colonies increases. Figure 8 makes it possible to highlight for certain selected period the exchange processes through synchronization in the change of certain cost values between colonies. With the areas circled on the diagram, we can observe the adoption of better solutions by some colonies.

In Figure 9, we present the different convergence characteristics obtained when we vary the period of information exchange between colonies. We apply it to the 4-unit system and the results obtained attest that increasing the exchange frequency (reducing the exchange period) accelerates convergence while maintaining the quality of the solution. Tables 5 and 6 show the mean execution times for the four-unit and ten-unit systems respectively. It also shows the speed-up and efficiency values of the parallel algorithm for each of the colony number values.

In both cases we can observe a decrease in execution time as the number of colonies increases. Thus for the test system of 4 units we go from 4.535 s for the sequential version to 1.712 s for the parallel version with 6 colonies, i.e. a speed-up of 2.65 and therefore an efficiency of 44.15 %. Likewise for the system of 10 units we go from 5.0612 s for the sequential version, to 2.0954 s for the parallel version with 6 colonies, i.e. a speed-up of 2.41 and an efficiency of 40.26 %.



(a) 4 units systems



(b) 10 units systems

Figure 7. Convergence characteristics of MAC-UCP/MMAS as a function of the number of colonies, for the system of 4 units (a) and 10 units (b).

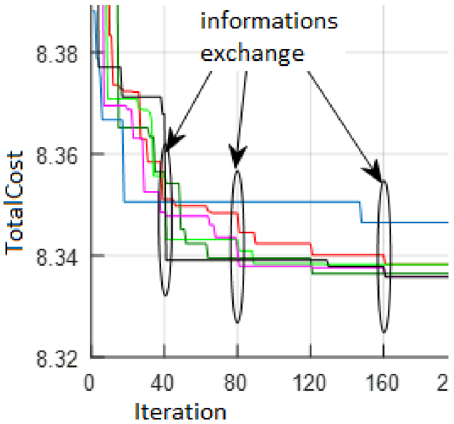


Figure 8. Highlighting exchanges on the 4-unit

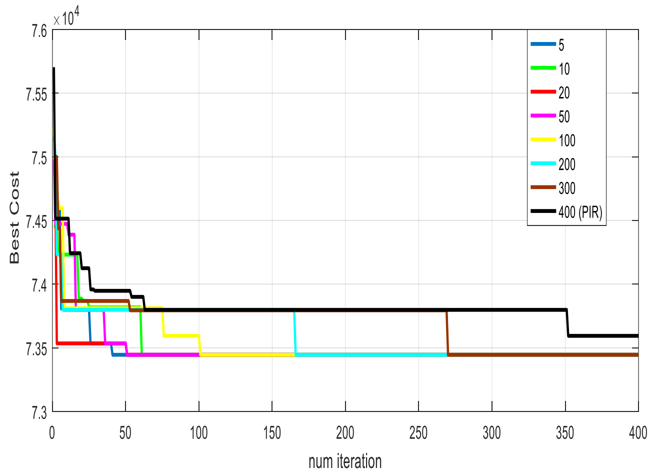


Figure 9. Convergence characteristics of MAC-UCP/MMAS system with different periods of information exchange

Table 5. Mean execution times of MAC-UCP/MMAS for 4-Units System

Nomber of colonies	1	2	3	4	5	6
Mean execution time (s)	4.53488	4.567	4.144	4.056	3.819	1.712
Speed-up (p.u.)	1	0.9930	1.0943	1.1181	1.1874	2.6489
Parallel efficiency (%)	100	49.65	36.48	27.95	23.75	44.15

Table 6. Mean execution times of MAC-UCP/MMAS for 10-Units System

Nomber of colonies	1	2	3	4	5	6
Mean execution time (s)	5.0612	2.8375	2.4145	2.3723	2.2759	2.0954
Speed-up (p.u.)	1	1.7837	2.0962	2.1335	2.2239	2.4154
Parallel efficiency (%)	100	89.19	69.87	53.34	44.48	40.26

Through Table 7 we establish a comparison between the cost obtained for our MAC-UCP/MMAS algorithm with the costs obtained in the literature by sequential versions proposed by other authors using the same system of 4 units.

The results allow us to assert that the proposed algorithm significantly improves the quality of the solution.

Table 8 presents a comparison of the results of our parallel algorithm and other sequential algorithms found in the literature for the 10-unit system.

We also note that for this dataset, the algorithm shows better results than some existing sequential versions of ACO (Simon et al., 2006).

Table 7. Comparison of total cost of production with other methods for 4-unit systems

Method	ILR (Kamboj et al., 2016)	LR-PSO (Kamboj et al., 2016)	MACO (Ameli et al., 2011)	Hybrid HS and Random Search Algorithm (Kamboj et al., 2016)	IBCS (Zhao et al., 2018)	BGWO (Panwar et al., 2018)	MMAS	Proposed MAC-MMAS
TOTAL COST ( )	75231.9	74808	74520.3	74476.0	74240.7	73933.1	73513.87	**73444.6857**
Change (%)	2.37561	1.8224	1.4434	1,3848	1.0722	0.6606	0,09411	**0**

Table 8. Comparison of total cost of production with other methods for 10-unit systems

Method	PROPOSED MAC-MMAAS	Proposed MMAS	Dynamics programming (Simon et al., 2006)	BRANCH AND BOUND (Simon et al., 2006)	Ant colony sytem (Simon et al., 2006)
TOTAL COST ($)	83351.3	83427.5	83652.4	83475.25	83491.42
Change (%)	0	0.0913	0.3599	0.148	0.168

5. CONCLUSION AND PERSPECTIVES

In this article, a new parallel algorithm of ant colony optimization for solving the thermal unit commitment problem was proposed and performed. This new approach is based on the Multicolony MMAS (MAX-MIN Ant System) algorithm with unidirectional circular exchange. The implementation on MATLAB was done through an SPMD (Single Program-Multiple Data) model with message passing communication. Several repetitive and consecutive tests of our algorithm were carried out on two datasets, namely a set of 4 thermal units and one of 10 thermal units. In both cases, the results obtained were compared with those existing in the literature. These comparisons show that the proposed algorithm significantly improves the quality of the solution in terms of cost while improving execution time. In particular, the sensitivity of this algorithm is highlighted through to the number of colonies and the frequency of exchange. With the increasing use of energy storage device, it is important to investigate on their impact on the UCP and therefore be able to develop robust UCP models.

6. CONFLICT OF INTEREST DECLARATION

We declare that there is no conflict of interest and this work respect our views on the issue of UCP.

7. REFERENCES

Ameli A., Safari A. et Shayanfar H.A., 2011. Modified Ant Colony Optimization technique for solving Unit commitment problem. *International Journal on Technical and physical problems of engineering*, **3**(4), 29–35.

Baldaniya R. B.et Mehta H. D., Solving Unit Commitment Problem using Ant Colony Optimization (ACO) Technique. *International Journal for Scientific Research & Development (IJSRD)*, **2,** 2321–0613.

Bonabeau E., Dorigo M., Marco D. de R.D.F, Theraulaz G., et Théraulaz G., 1999. *Swarm intelligence: from natural to artificial systems*. Oxford university press.

Cong G., Meyers C., Rajan D. et Parriani T., 2015. Parallel Strategies for Solving Large Unit Commitment Problems in the California ISO Planning Model. *2015 IEEE International Parallel and Distributed Processing Symposium*, 710–719

Dorigo M. et Gambardella L.M., 1997. Ant colony system: a cooperative learning approach to the traveling salesman problem. *IEEE Trans. Evol. Computat,* **1**(1), 53–66.

Dorigo M., 1992. Optimization, learning and natural algorithms. PhD Thesis, Politecnico di Milano.

Dorigo M., Colorni A. et Maniezzo V., 1991. Distributed optimization by ant colonies. *Proceedings of ecal91 - european conference on artificial life*, 134–142.

Dorigo M., Maniezzo V. et Colorni A., 1996. Ant system: optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, **26**(1), 29–41.

Dupin N. et Talbi E., 2020. Parallel matheuristics for the discrete unit commitment problem with min-stop ramping constraints. *Intl. Trans. in Op. Res.* **27**(1), 219–244.

Habachi R., Touil A., Boulal A., Charkaoui A. et Echchatbi A., 2019. Solving economic dispatch and unit commitment problem in smart grid system using eagle strategy based crow search algorithm. *Indonesian Journal of Electrical Engineering and Computer Science*, **4**(3), 1087-1096.

Jahromi M. Z., Bioki M. M. H., NEJAD M. R. et Fadaeinedjad R., 2013. Solution to the unit commitment problem using an artificial neural network. *Turkish Journal of Electrical Engineering & Computer Sciences*, **21**(1), 198–212.

Kamboj V.K.,. Bath S. K et Dhillon J. S., 2016. Implementation of hybrid harmony search/random search algorithm for single area unit commitment problem. *International Journal of Electrical Power & Energy Systems*, **77,** 228–249.

Kandemir M., Choudhary A., Banerjee P., Ramanujam J., et Shenoy N., 2000. Minimizing data and synchronization costs in one-way communication. *IEEE Transactions on Parallel and Distributed Systems*, 11(12), 1232–1251.

Kargarian A., Mehrtash M. et Falahati B., 2018. Decentralized Implementation of Unit Commitment With Analytical Target Cascading: A Parallel Approach. *IEEE Trans. Power Syst.*, **33**(4), 3981–3993, doi: 10.1109/TPWRS.2017.2787645.

Khanmohammadi S., Amiri M., et. Haque M. T, 2010. A new three-stage method for solving unit commitment problem. *Energy*, **35**(7), 3072–3080.

Khatibi M. et Bigdeli M., 2014. Transient stability improvement of power systems by optimal sizing and allocation of resistive superconducting fault current limiters using particle swarm optimization. *Advanced Energy: An International Journal (AEIJ)*, **1**(3), 11–27.

Kumar V. et Kumar D., 2020. Binary whale optimization algorithm and its application to unit commitment problem. *Neural Computing and Applications*, **32**(7), 2095–2123, 2020.

Lai Y., Bai Y. et Fan Y. 2012. Optimization of Unit Commitment based on improved Max-Min Ant System algorithm. *2012 IEEE International Conference on Intelligent Control, Automatic Detection and High-End Equipment*, Beijing, China, 192–195.

Li Y.-F., Pedroni N. et Zio E., 2013. A Memetic Evolutionary Multi-Objective Optimization Method for Environmental Power Unit Commitment. *IEEE Transactions on Power Systems*, **28**(3), 2660–2669.

Middendorf M., Reischle F. et Schmeck H., 2002. Multi colony ant algorithms. *Journal of Heuristics*. **8**(3), 305–320, 2002.

Pan H., You X., et Liu S., 2020. High-Frequency Path Mining-Based Reward and Punishment Mechanism for Multi-Colony Ant Colony Optimization. *IEEE Access*. **8**, 155459–155476.

Panwar L.K., Reddy S., Verma A., Panigrahi B.K. et Kumar R., 2018. Binary grey wolf optimizer for large scale unit commitment problem. *Swarm and Evolutionary Computation*, **38**, 251–266.

Pedemonte M., Nesmachnow S., et Cancela H., 2011. A survey on parallel ant colony optimization. *Applied Soft Computing*, **11**(8), 5181–5197.

Postolov B. et Iliev A., 2020. Adaptive Genetic Algorithm for Hydro-thermal Unit Commitment Considering the Security Constraints. *International Journal of Electrical Engineering and Computing*, **4**(2), 61–69.

Randall M. et Lewis A., 2002. A parallel implementation of ant colony optimization. *Journal of Parallel and Distributed Computing*. **62**(9), 1421-1432.

Simon S. P., Padhy N.P. et Anand R.S., 2006. An ant colony system approach for unit commitment problem. *International Journal of Electrical Power & Energy Systems*, **28**(5), 315–323.

Singhal P.K., Naresh R. et Sharma V., 2015. A novel strategy-based hybrid binary artificial bee colony algorithm for unit commitment problem. *Arabian Journal for Science and Engineering*, **40**(5), 1455–1469.

SOH M., Nguimeya B. Tsofack, etTayou Djamegni C., 2020. A Multi Ant Colony Optimization Approach For The Traveling Salesman Problem. *Revue Africaine de la Recherche en Informatique et Mathématiques Appliquées*.

Sriyanyong P.et Song Y.H., 2005. Unit commitment using particle swarm optimization combined with Lagrange relaxation. *IEEE Power Engineering Society General Meeting*, 2752–2759.

Stützle T. et Hoos H.H., 1996. Improving the Ant System: A detailed report on the MAX–MIN Ant System. *FG Intellektik, FB Informatik, TU Darmstadt, Germany, Tech. Rep. AIDA–96–12*.

Stützle T.et Hoos H.H., 2000. MAX–MIN ant system. *Future generation computer systems*, **16**(8), 889–914.

Stützle T.et Hoos H., 1998. Improvements on the ant-system: Introducing the max-min ant system. *Artificial neural nets and genetic algorithms*, 245–249.

Twomey C., Stützle T., Dorigo M., Manfrin M. et Birattari M., 2010. An analysis of communication policies for homogeneous multi-colony ACO algorithms. *Information Sciences*, **180**(12), 2390–2404.

Yu D., Wang Y. et Guo R., 2010. A Hybrid Ant Colony Optimization Algorithm Based Lambda-Iteration Method for Unit Commitment Problem. *2010 Second WRI Global Congress on Intelligent Systems*, 19–22.

Zand A., Bigdeli M. et Azizian D., 2016. A Modified ANT COLONY ALGORITHM FOR SOLVING THE UNIT COMMITMENT PROBLEM. *Advanced Energy: An International Journal (AEIJ)*, **3**(2/3), 15–27.

Zhai Y., Liao X., Mu N. et Le J., 2020. A two-layer algorithm based on PSO for solving unit commitment problem. *Soft Computing*, **24**(12), 9161-9178.

Zhang N., Hu Z., Han X., Zhang J. et Zhou Y., 2015. A fuzzy chance-constrained program for unit commitment problem considering demand response, electric vehicle and wind power. *International Journal of Electrical Power & Energy Systems*, **65**, 201–209.

Zhao J., Liu S., Zhou M., Guo X., et Qi L., 2018. An Improved Binary Cuckoo Search Algorithm for Solving Unit Commitment Problems: Methodological Description. *IEEE Access*, **6**, 43535–43545.