Applied Energy 88 (2011) 4687–4699

Contents lists available at ScienceDirect

Applied Energy

journal homepage: www.elsevier.com/locate/apenergy

A new memetic algorithm approach for the price based unit commitment problem

Dionisios K. Dimitroulas, Pavlos S. Georgilakis *

Electric Power Division, School of Electrical and Computer Engineering, National Technical University of Athens (NTUA), GR 15780 Athens, Greece

ARTICLE INFO

Article history:
Received 16 February 2011
Received in revised form 22 May 2011
Accepted 4 June 2011
Available online 30 June 2011

Keywords:
Price based unit commitment (PBUC)
Memetic algorithm (MA)
Genetic algorithm (GA)
Ramp rate constraints
Electric energy markets
Generation scheduling

ABSTRACT

Unit commitment (UC) is a very important optimization task, which plays a major role in the daily operation planning of electric power systems that is why UC is a core research topic attracting a lot of research efforts. An innovative method based on an advanced memetic algorithm (MA) for the solution of price based unit commitment (PBUC) problem is proposed. The main contributions of this paper are: (i) an innovative two-level tournament selection, (ii) a new multiple window crossover, (iii) a novel window in window mutation operator, (iv) an innovative local search scheme called elite mutation, (v) new population initialization algorithm that is specific to PBUC problem, and (vi) new PBUC test systems including ramp up and ramp down constraints so as to provide new PBUC benchmarks for future research. The innovative two-level tournament selection mechanism contributes to the reduction of the required CPU time. The method has been applied to systems of up to 110 units and the results show that the proposed memetic algorithm is superior to other methods since it finds the optimal solution with a high success rate and within a reasonable execution time.

1. Introduction

Unit commitment (UC) is a very important optimization task, which plays a major role in the daily operation planning of electric power systems that is why UC is a core research topic attracting a lot of research efforts [1–7].

In the regulated electricity markets, UC refers to optimizing generation resources over a daily to weekly time horizon to satisfy load demand at minimum operational cost while satisfying prevailing constraints, such as minimum up/down time, ramping up/down, and minimum/maximum generating capacity. Since the related objective would be to minimize the operational cost, UC is commonly referred to as cost-based unit commitment (CBUC). The optimal solution to the CBUC problem can be obtained by complete enumeration, which is prohibitive in practice owing to its excessive computational resource requirements [8]. The need for practical, cost-effective UC solutions, led to the development of various UC algorithms that produce suboptimal, but efficient scheduling for real sized power systems comprising hundreds of generators [9]. CBUC methods include priority list methods [8], dynamic programming [10], Lagrangian relaxation (LR) [11], branch-and-bound [12], Benders decomposition [13], and mixed-integer programming [14]. Moreover, simulated annealing [15], expert systems [16], artificial neural networks [17], genetic algorithms [18], and hybrid techniques [19] have also been used for the solution of the CBUC problem.

In the deregulated electricity markets, the unit commitment used by generating companies (GENCOs) is referred to as price based unit commitment (PBUC) in which optimization of generation resources takes place in order to maximize the GENCOs total profit [20,21]. The PBUC is a large-scale, nonconvex, nonlinear, mixed-integer optimization problem, belonging to the NP-hard class [20,21]. Because electricity markets are changing rapidly, there is strong interest on how new PBUC models are solved. That is why various methods have been proposed for the solution of PBUC problem including Lagrangian relaxation [22], mixed-integer programming [23–29], genetic algorithms [30,31], selective enumeration [32], and hybrid methods [33–35].

In [36], the memetic algorithms were introduced for the first time. Memetic algorithms can be viewed as the hybridization of evolutionary algorithms (exploration component) and local search (exploitation component). Their main advantages are their generic nature, which enables them to successfully tackle almost any optimization problem and their ease to embed problem specific knowledge and constraints [37]. Memetic algorithms have been successfully applied for the solution of difficult power system problems [38–41], but have not been applied to PBUC so far.

A new approach based on an advanced memetic algorithm (MA) is proposed in this paper for the solution of PBUC problem. The main contributions of this paper are: (i) an innovative two-level tournament selection, (ii) a new multiple window crossover, (iii) a novel window in window mutation operator, (iv) an innovative local search scheme called elite mutation, (v) new population

* Corresponding author. Tel.: +30 210 7724378.
E-mail addresses: ddimitroulas@gmail.com (D.K. Dimitroulas), pgeorg@power.ece.ntua.gr (P.S. Georgilakis).

0306-2619/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.apenergy.2011.06.009

Nomenclature

$A(i)$	constant cost coefficient of unit i (\$/h)
$B(i)$	linear cost coefficient of unit i (\$/MW h)
$C(i)$	quadratic cost coefficient of unit i (\$/MW^2 h)
$Cost(i,t)$	production cost of unit i at hour t (\$/h)
$CT(i)$	cooling constant of unit i (h)
$D(i)$	cold start staff and maintenance cost of unit i (\$/h)
$d(t)$	forecasted demand at hour t (MW)
$E(i)$	cold start cost of unit i (\$/h)
$F(i, t)$	profit of unit i at hour t (\$/h)
$FC(i, t)$	fuel cost of unit i at hour t (\$/h)
i	unit index
$I(i, t)$	operating status of unit i at hour t (1 = On, 0 = Off)
N	number of generating units
$P(i, t)$	power generation of unit i at hour t (MW)
$p_{gm}(t)$	forecasted market price for energy at hour t (\$/MW h)
$P_{max}(i)$	maximum power output of unit i (MW)
$P_{min}(i)$	minimum power output of unit i (MW)
$R_{down}(i)$	ramp down rate of unit i (MW/h)
$R_{up}(i)$	ramp up rate of unit i (MW/h)
$Rvn(i, t)$	revenue of unit i at hour t (\$/h)
$SD(i)$	shut down cost of unit i (\$)
$SU(i, t)$	start up cost of unit i at hour t (\$/h)
t	hour index
T	number of time periods of the unit commitment (UC) planning horizon (h). The UC time step is 1 h.
$T_{down}(i)$	minimum down time of unit i (h)
$T_{up}(i)$	minimum up time of unit i (h)
$X(i, t)$	if $X(i, t) > 0$, then the cumulative up time of unit i at hour t is $X(i, t)$ hours. On the other hand, if $X(i, t) < 0$, then the cumulative down time of unit i at hour t is $-X(i, t)$ hours
$X_{off}(i, t)$	cumulative down time of unit i at hour t (h)

initialization algorithm that is specific to PBUC problem, and (vi) new PBUC test systems including ramp up and ramp down constraints so as to provide new PBUC benchmarks for future research. The main advantages of the proposed memetic algorithm are: (a) flexibility in modelling problem constraints because the PBUC problem is not decomposed either by time or by unit, (b) easier convergence to the optimum solution thanks to the proposed operators and local search schemes, (c) easy implementation to work on parallel computers, and (d) production of multiple unit commitment schedules, some of which may be well suited to situations that may arise quickly due to unexpected contingencies. The method has been applied to systems of up to 110 units and the results show that the proposed memetic algorithm is superior to other methods.

The paper is organized as follows. The PBUC problem is formulated in Section 2 and a detailed description of the proposed MA used to solve it is found in Section 3. Section 4 presents MA results for test systems up to 110 units and comparisons with results obtained by other solving methods. Section 5 concludes the paper.

2. PBUC formulation

2.1. Problem formulation

The profit of unit i at hour t is calculated by the following formula [20,21]:

$$F(i,t) = Rvn(i,t) - Cost(i,t). \tag{1}$$

It must be noted that negative profit indicates losses.

Revenue and cost of unit i at hour t are calculated by:

$$Rvn(i,t) = p_{gm}(t) \cdot P(i,t) \cdot I(i,t), \tag{2}$$

$$\begin{aligned} Cost(i,t) = {} & FC(i,t) \cdot I(i,t) + SU(i,t) \cdot I(i,t) \cdot [1 - I(i,t-1)] \\ & + SD(i) \cdot I(i,t-1) \cdot [1 - I(i,t)]. \end{aligned} \tag{3}$$

The fuel cost of unit i at hour t is a quadratic function of the unit power output [21]:

$$FC(i,t) = A(i) + B(i) \cdot P(i,t) + C(i) \cdot [P(i,t)]^2. \tag{4}$$

Start up cost of unit i at hour t depends on the total off-hours that unit i was halted and is modelled by [21]:

$$SU(i,t) = D(i) + E(i) \cdot \left[1 - \exp\left(-\frac{X_{off}(i,t)}{CT(i)}\right)\right]. \tag{5}$$

Shut down cost $SD(i)$ of unit i has a constant value per shutdown.

The PBUC problem is mathematically formulated as follows [20–35]:

$$\max \sum_{t=1}^{T} \sum_{i=1}^{N} F(i,t), \tag{6}$$

subject to the following constraints:

a. Unit power generation limits [20–35]:

$$P_{\min}(i) \cdot I(i,t) \leqslant P(i,t) \leqslant P_{\max}(i) \cdot I(i,t), \quad \forall i \tag{7}$$

b. Minimum up time and down time constraints [20,22,31–35]:

$$[X(i,t-1) - T_{up}(i)] \cdot [I(i,t-1) - I(i,t)] \geqslant 0, \quad \forall i, \forall t, \tag{8}$$

$$[-X(i,t-1) - T_{down}(i)] \cdot [I(i,t) - I(i,t-1)] \geqslant 0, \quad \forall i, \forall t, \tag{9}$$

c. Ramp rate limits [20,22,31,33,35]:

$$P(i,t) - P(i,t-1) \leqslant R_{up}(i), \quad \text{as unit } i \text{ ramps up}, \tag{10}$$

$$P(i,t-1) - P(i,t) \leqslant R_{down}(i), \quad \text{as unit } i \text{ ramps down}. \tag{11}$$

d. Demand constraint [20,22,25,30–35]:

$$\sum_{i=1}^{N} P(i,t) \cdot I(i,t) \leqslant d(t), \quad \forall t. \tag{12}$$

It should be noted that (7), (10), and (11) can be combined as follows [42]:

$$P'_{\max}(i,t) = \min[P_{\max}(i), P(i,t-1) + R_{up}(i)], \quad \text{as unit } i \text{ ramps up}, \tag{13}$$

$$P'_{\min}(i,t) = \max[P_{\min}(i), P(i,t-1) - R_{down}(i)], \quad \text{as unit } i \text{ ramps down}. \tag{14}$$

Eq. (13) computes $P'_{\max}(i,t)$, which is the updated $P_{\max}(i)$ when the ramp up constraint of unit i is included. It is obvious that $P'_{\max}(i,t) \leqslant P_{\max}(i)$ meaning that the inclusion of the ramp up constraint produces an upper bound for $P(i, t)$ with lower value than when excluded.

Eq. (14) computes $P'_{\min}(i,t)$, which is the updated $P_{\min}(i)$ when the ramp down constraint of unit i is included. It is obvious that $P'_{\min}(i,t) \geqslant P_{\min}(i)$ meaning that the inclusion of the ramp down constraint produces a lower bound for $P(i, t)$ with higher value than when excluded.

The coupling demand constraint (12) complicates the solution of the PBUC problem, since the PBUC cannot be decomposed by unit. However, the demand constraint (12) appears in many PBUC formulations of the literature, e.g., [20,22,25,30–35], because each hour the GENCO has to produce no more power than the demand of that hour [20,22,25,30–35].

Based on the above mathematical formulation, the PBUC problem can be stated as follows: for a GENCO with N generating units, and given a certain market price profile of energy as well as a certain demand profile, it is required to determine the start-up/shut-down times and the power output of all the generating units at each hour t over a specified planning horizon T, so that the generator's total profit (6) is maximized, subject to the unit and demand constraints (7)–(12).

3. Proposed methodology

3.1. Memetic algorithms

Every living being obtains knowledge from two sources. The first is the gene source directly inherited from the being's parents and the second source are personal and cultural experiences.

Based upon the antecedent observation, a memetic algorithm (MA) is a hybrid computational model of those two sources. The first source is modelled by a genetic algorithm (GA) that mimics biological or Darwinian evolution and the second source is modelled by a local search algorithm that mimics cultural evolution or the evolution of ideas.

The unit of information in a GA is termed as a gene whereas in a MA is termed as a meme. Genes are improved by crossover and mutation operators that are part of a GA and memes are improved by a local search operator.

3.2. Chromosome encoding

A convenient binary mapping to a chromosome representation is selected in which zero represents the off state and one represents the on state of a unit. A candidate solution (chromosome) is a two dimensional matrix whose number of rows is N and number of columns is T.

Fig. 1 shows an example of chromosome representation of unit i for a planning horizon of $T = 8$ h. It is also given that the initial state of unit i is $X(i, 0) = -2$, i.e., unit i was 2 h continuously off at the start of the planning horizon. Since unit i is still off for the first 3 h (Fig. 1), it is concluded that $X_{off}(i, 3) = 5$. Similarly, it follows that $X_{off}(i, 6) = 1$. It is concluded that for a given chromosome, $X_{off}(i, t)$ takes a fixed value, so the start up cost takes also a fixed value, as (5) shows, thus $F(i, t)$ is a quadratic function of the unit power output, i.e., the power dispatch problem is in fact a quadratic programming problem. Fig. 1 shows that unit i is off for the first three hours, while it turns on at hour 4, so $X(i, 3) = -3$. Similarly, it follows that $X(i, 6) = -1$. Unit i is on for the hours 4 and 5, while it turns off at hour 6, so $X(i, 5) = 2$.

The information available in the chromosome together with the initial state (continuous up or down time, $X(i, 0)$, $\forall i$) of the units is all one needs to accurately model all time dependent constraints of the PBUC problem. This great modelling flexibility is one of the advantages of the proposed memetic algorithm solution, because the PBUC problem is not decomposed either by time or by unit.

Unit	Hours (1-8)							
i	0	0	0	1	1	0	1	1

Fig. 1. Example of chromosome representation. The initial state of unit i is $X(i, 0) = -2$, i.e., unit i was 2 h continuously off at the start of the planning horizon.

3.3. Population of chromosomes

A population of chromosomes is stored in a three dimensional matrix. Commitment scenarios take up two dimensions, one for the units and one for the hours of the planning horizon. The third dimension can be viewed as a stack of commitment scenarios. The top of the stack ($k = 1$) stores the elite chromosome, which is the best scenario found so far by the MA.

3.4. Overview of the proposed MA methodology

Fig. 2 illustrates the flow chart of the proposed MA for the solution of the PBUC problem.

Seven parameters namely population size, penalty, number of generations, NM, ECMP, ECMS and NEM are set by the user, as Fig. 2 shows. The acronym NM stands for number of mutations, ECMP stands for elite chromosome mutation probability, ECMS stands for elite chromosome mutated scenarios and NEM stands for number of elite mutations. Details for NM are provided in

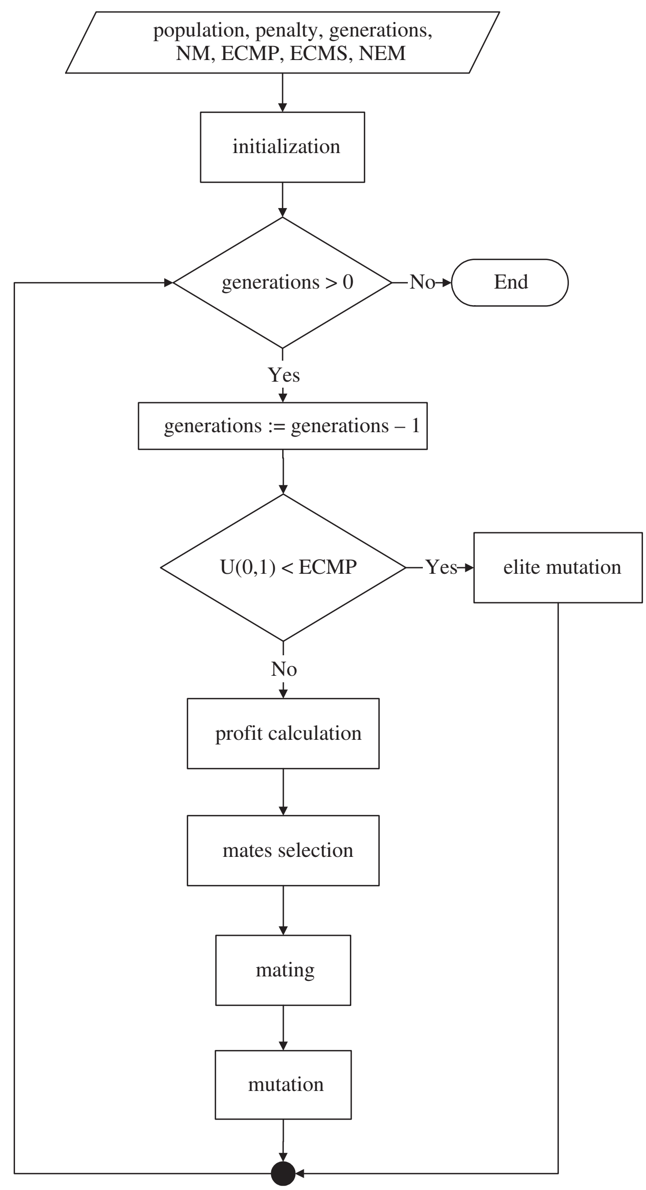

Fig. 2. Memetic algorithm block diagram. $U(0, 1)$ denotes a random number uniformly distributed in the interval (0, 1).

Section 3.9; for ECMP, ECMS and NEM in Section 3.10; and for penalty parameter in Section 3.6.

It can be seen from Fig. 2 that the proposed MA is composed of six main blocks: (1) initialization, (2) profit calculation, (3) mates selection, (4) mating, (5) mutation, and (6) elite mutation. These six blocks will be analytically described in Sections 3.5–3.10.

3.5. Initialization

This paper proposes a problem specific initialization according to which the population is initialized with chromosomes that always satisfy $T_{up}(i)$ and $T_{down}(i)$ constraints (8) and (9). It should be noted that the population is not randomly initialized because computational experiments showed that it was very difficult for the algorithm to produce enough feasible solutions. More specifically, a contribution of this paper is that every chromosome is initialized using randomly one of the following two proposed chromosome initialization methods:

1. The first initialization method inserts a single window, full of ones, which has random width in a manner that $T_{up}(i)$ constraint (8) is satisfied taking also into account the initial conditions $X(i,0)$. For example, with this method, the chromosome's unit 1, i.e., row 1 in Fig. 3, has been initialized. Unit 1 has $T_{up}(1) = 4$, $T_{down}(1) = 3$ and $X(1,0) = -1$. More specifically, the proposed initialization process inserted a single window of ones, from hour 4 to hour 9, as illustrated with dark shading in Fig. 3.
2. The second initialization method, which is the dual of the first method, inserts a single window, full of zeros, which has random width in a manner that $T_{down}(i)$ constraint (9) is satisfied taking also into account the initial conditions $X(i,0)$. For example, with this method, the chromosome's unit 2, i.e., row 2 in Fig. 3, has been initialized. Unit 2 has $T_{up}(2) = 2$, $T_{down}(2) = 3$ and $X(2,0) = 1$. More specifically, the proposed initialization process inserted a single window of zeros, from hour 5 to hour 9, as illustrated with light shading in Fig. 3.

3.6. Profit calculation

Profit calculation procedure carries out the following three tasks:

1. Calculates the total profit for every chromosome k in the population.
2. Calculates the average profit of the population, which is needed in mates selection (Section 3.7).
3. Following task 1, if there is a chromosome that has higher profit than the profit of the current elite, then these two chromosomes are swapped resulting in a new elite chromosome placed at location $k = 1$.

Every chromosome's k total profit, task 1, is calculated in two phases:

1. The solution of the economic load dispatch (ELD) subproblem for every hour t of the planning horizon gives the optimum power generation of every committed unit i as well as the profit for each hour. ELD is a quadratic programming (QP) problem that is solved using the active set method [43].

Unit	Hours (1-10)									
1	0	0	0	1	1	1	1	1	1	0
2	1	1	1	1	0	0	0	0	0	1

Fig. 3. Application example of the proposed chromosome initialization process.

If at some hour t, unit i happens to be decommitted and at previous hour $t-1$ has generated power greater than $R_{down}(i)$, then the chromosome must be repaired by committing unit i at hour t, otherwise ramp down constraint violation will occur.

Initially, $P'_{\min}(i,t)$ for every committed unit i is computed by (14). If $\sum_{\forall\ i\ \text{with}\ I(i,t)=1} P'_{\min}(i,t) > d(t)$ then the QP problem must exclude (12), otherwise with (12) being inevitably violated the algorithm will fail. Following the QP problem solution, the amount penalty $\cdot \left[\sum_{\forall\ i\ \text{with}\ I(i,t)=1} P(i,t) - d(t)\right]$ is subtracted from chromosome's k total profit, where penalty is the penalty parameter set by the user. On the other hand, if (12) is not violated, then the QP problem must include (12) without any amount subtracted from chromosome's k total profit.

2. The minimum up time constraint (8) and the minimum down time constraint (9) are checked for violations. Moreover, start up costs (5) and shut down costs are subtracted from chromosome's k total profit.

Whenever constraint (8) or constraint (9) is violated, then the amount $10 \cdot \text{penalty} \cdot dg$ is subtracted from chromosome's k total profit, where penalty is the penalty parameter set by the user, and dg is the degree of constraint (8) or constraint (9) violation. For example, if unit i has $X(i,t) = 3$ and $T_{up}(i) = 5$, then $dg = T_{up}(i) - X(i,t)$, i.e., $dg = 2$.

Another contribution of this paper is the use of different significance penalties. More specifically, the violation of constraint (8) and constraint (9) has significance equal to 10, whereas the violation of constraint (12) has significance equal to 1. The greater significance that (8) and (9) share in comparison with (12) results in a stronger intimidation of the algorithm to explore regions of (6) that violate (8) or (9) so as to ensure that all generating units will not exceed their technical specifications.

3.7. Mates selection

The original tournament selection [44] randomly picks a small subset of chromosomes from the population and the highest profit chromosome in this subset wins the tournament. This means that for a given chromosome, the probability to win the tournament depends on two factors: (i) the chromosome profit, and (ii) the subset in which the chromosome will be randomly put. In other words, if a chromosome becomes a member of one tournament subset it might be discarded, whereas if it becomes a member of another tournament subset it might win the tournament [44]. Consequently, the original tournament selection utilizes only one level of information that is local information: the tournament subset's profits.

This paper proposes a novel mates selection operator, which alleviates the above mentioned disadvantage of the original tournament selection by using a second and global level of information: the population's average profit. This innovative mates selection operator is named two-level tournament selection (2LTS) and selects mates based on the tournament subset's profits and the population's average profit, i.e., in a fashion currently not found in the optimization literature. In brief, the proposed mates selection operator selects the most adapted (the most fit) chromosomes that a current population has to offer. These chromosomes are named mother chromosomes and will mate with the elite chromosome giving one or in some cases two offspring. This new 2LTS operator will be explained with the help of Figs. 4 and 5.

Fig. 4 illustrates the flow chart of the proposed 2LTS operator. The 2LTS algorithm works with two variables. The first variable is k, which has a dual purpose: (1) it is a loop control variable so the algorithm can terminate, and (2) it always points to the first

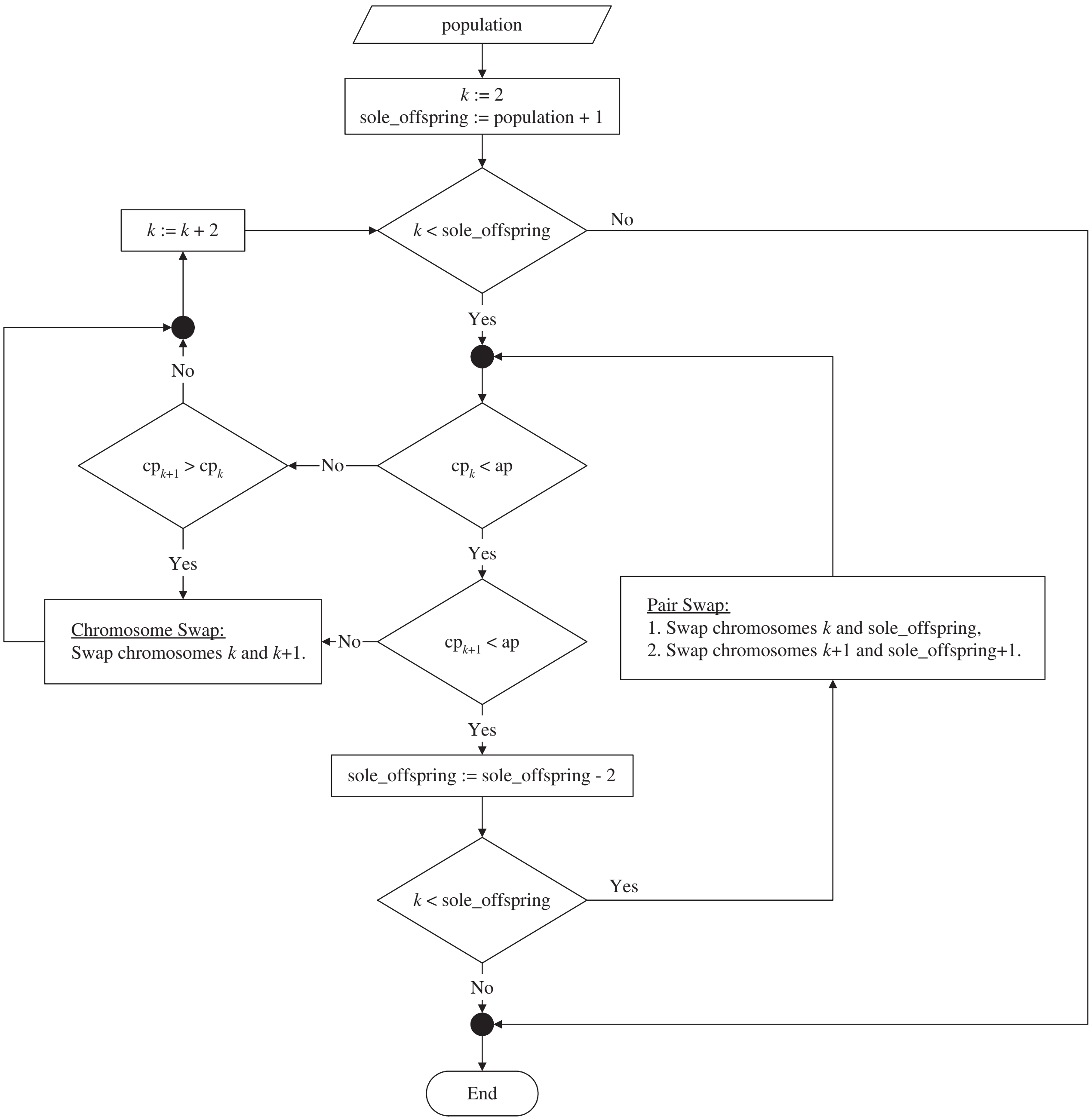

Fig. 4. Two-level tournament selection (2LTS) block diagram.

chromosome contained in the tournament subset. The second variable is sole_offspring and when the 2LTS operator completes mates selection, as will be shown later in this section, it indicates the borderline between the collection of pairs of chromosomes that require a sole offspring and the collection of pairs of chromosomes that require two offspring so as to keep the population of chromosomes constant. The acronym cp_k stands for the profit of chromosome in position k calculated by Section 3.6 task 1, while the acronym ap stands for average profit of the population calculated by Section 3.6 task 2. It can be seen from Fig. 4 that the proposed 2LTS operator is composed of two main blocks: (1) chromosome swap, and (2) pair swap. These two blocks will be analytically described in the rest of this section.

Fig. 5 illustrates an application example of the proposed 2LTS. More specifically, Fig. 5a is the initial population before the application of the proposed 2LTS operator. A series of intermediate steps are followed in Fig. 5b–e leading finally to Fig. 5f, which is the final population after the completion of the proposed 2LTS operator.

Going into the details of chromosome swap, Fig. 5a illustrates that the elite chromosome is always stored in position $k = 1$ and the rest of the population is divided into five pairs of chromosomes, i.e., in the proposed 2LTS a tournament subset always contains two chromosomes. Chromosome χ is stored in position k of the population. Fig. 4 initializes variables k and sole_offspring to 2 and 12, respectively (the population is a set of 11 chromosomes). Pair 1 is composed of chromosomes $\chi = 2$ and $\chi = 3$. In this example ap = 80.91. The chromosome $\chi = 3$ in $k = 3$ has profit greater than ap. On the other hand, chromosome $\chi = 2$ in $k = 2$ has profit lower than ap, so chromosome swap occurs resulting in the selection of chromosome $\chi = 3$ as a mother chromosome, while chromosome $\chi = 2$ is discarded, thus resulting in Fig. 5b. As Fig. 4

a

	Elite	Pair 1		Pair 2		Pair 3		Pair 4		Pair 5	
k	1	2	3	4	5	6	7	8	9	10	11
χ	1	2	3	4	5	6	7	8	9	10	11
cp_k	150	2	121	137	38	78	46	64	16	91	147

b

	Elite	Pair 1		Pair 2		Pair 3		Pair 4		Pair 5	
k	1	2	3	4	5	6	7	8	9	10	11
χ	1	3	2	4	5	6	7	8	9	10	11
cp_k	150	121	2	137	38	78	46	64	16	91	147

c

	Elite	Pair 1		Pair 2		Pair 3		Pair 4		Pair 5	
k	1	2	3	4	5	6	7	8	9	10	11
χ	1	3	2	4	5	6	7	8	9	10	11
cp_k	150	121	2	137	38	78	46	64	16	91	147

d

	Elite	Pair 1		Pair 2		Pair 3		Pair 4		Pair 5	
k	1	2	3	4	5	6	7	8	9	10	11
χ	1	3	2	4	5	10	11	8	9	6	7
cp_k	150	121	2	137	38	91	147	64	16	78	46

e

	Elite	Pair 1		Pair 2		Pair 3		Pair 4		Pair 5	
k	1	2	3	4	5	6	7	8	9	10	11
χ	1	3	2	4	5	11	10	8	9	6	7
cp_k	150	121	2	137	38	147	91	64	16	78	46

f

	Elite	Pair 1		Pair 2		Pair 3		Pair 4		Pair 5	
k	1	2	3	4	5	6	7	8	9	10	11
χ	1	3	2	4	5	11	10	8	9	6	7
cp_k	150	121	2	137	38	147	91	64	16	78	46

Fig. 5. Application example of the proposed two-level tournament selection (2LTS) operator. (a) Initial population, before the application of the proposed 2LTS operator. Average profit of the population is ap = 80.91. (b–e) Intermediate population, during the application of the proposed 2LTS operator. (f) Final population, after the completion of the proposed 2LTS operator.

shows, following the swapping of the two chromosomes, the variable k is increased by 2, so $k = 4$, effectively advancing to pair 2.

Analyzing pair swap, in Fig. 5c pair 3 receives the attention of the 2LTS operator, so $k = 6$, which is composed of chromosomes $\chi = 6$ and $\chi = 7$. It is observed that pair 3 has both chromosomes under average profit. At first the variable sole_offspring is decreased by 2, so sole_offspring = 10, which now points to the first chromosome contained in pair 5. Then the algorithm discards chromosomes $\chi = 6$ and $\chi = 7$ by pair swap between pair's 3 and pair's 5 chromosomes resulting in Fig. 5d. Pair swap always occurs between a pair whose first chromosome is indicated by variable k and has both chromosomes under average profit, and the pair whose first chromosome is indicated by variable sole_offspring. As Fig. 4 shows, following the swapping of the two pairs, the updated pair 3, which is composed of chromosomes $\chi = 10$ and $\chi = 11$, is checked again. It is observed that the updated pair 3 of Fig. 5d has both chromosomes above average profit, so a tournament is held between these two chromosomes and finally chromosome $\chi = 11$ in $k = 7$ wins the tournament, so chromosome swap occurs resulting in the selection of chromosome $\chi = 11$ as a mother chromosome, while chromosome $\chi = 10$ is discarded, thus resulting in Fig. 5e.

Fig. 5f shows the final population after the completion of the proposed 2LTS operator. The final value of variable sole_offspring is 8, which indicates that pairs 1, 2, and 3 require a sole offspring, whereas pairs 4, and 5 require two offspring. The selected mother chromosomes are lightly shaded and located in even population locations: $k = 2$, $k = 4$, and $k = 6$. The dark shaded elite chromosome in position $k = 1$ will mate with the mother chromosome in position $k = 2$ and the sole offspring will be placed in position $k = 3$. The same mating pattern applies to pairs 2 and 3. Moreover, the elite chromosome in position $k = 1$ will remate with the mother chromosome in position $k = 2$ and the first offspring will be placed in position $k = 8$, whereas the second offspring will be placed in position $k = 9$. Similar mating pattern applies to pair 5 where the elite chromosome in position $k = 1$ will remate with the mother chromosome in position $k = 4$ and the first and second offspring will be placed in positions 10 and 11, respectively.

The previous example showed that in the proposed 2LTS the population is handled pair-wise so an even number of chromosomes are needed. The algorithm must also store the elite chromosome in position $k = 1$. Consequently, it is of the utmost importance that the population size, which is user defined, is set to an odd number greater than or equal to 3, i.e., at least three chromosomes are needed, otherwise the algorithm will fail.

From the previous example it also emerges that the proposed 2LTS has an embedded auto-reset feature that original tournament selection lacks. In original tournament selection there will always be a chromosome that wins the tournament but this is not necessarily the case with 2LTS. For instance, in 2LTS, let us suppose the case that a feasible elite chromosome exists while the rest of the population contains highly unfeasible solutions meaning that penalties will be very high and as a consequence all these chromosomes will have highly negative profits. The population's average profit will then be high enough to cause the entire population to be discarded. If this is the case, then the population (except the elite chromosome) is reinitialized, which is the embedded auto-reset feature of the proposed 2LTS.

It should be noted that the proposed mates selection mechanism contributes to the reduction of the required CPU time as will be shown in Section 4.

3.8. Mating

Mating operator blends information contained in the bit sequence of the parents. The minimal effectiveness of computational experiments that were conducted using single-point, two-point and uniform crossover lead to the conception of the proposed multiple window crossover.

The multiple window crossover is implemented using the following steps:

1. Rows and columns are randomly selected, where rows correspond to units and columns correspond to hours. For example, for the 10 h planning problem of the 10 units of Fig. 6, let us suppose that the rows 1, 3, 5, 6, 9, 10 and the columns 2, 3, 6, 7, 8 have been selected.
2. The first offspring is created by copying from the elite chromosome the selected rows and columns of the first step, while any remaining blanks are filled with bits copied from the mother chromosome. For example, the elite chromosome of Fig. 6a mates with the mother chromosome of Fig. 6b and produces the offspring of Fig. 6c. More specifically, the offspring of Fig. 6c has exactly the same bits with the elite chromosome of Fig. 6a in the rows 1, 3, 5, 6, 9, 10 and in the columns 2, 3, 6, 7, 8. However, with this operation, nine windows are blank, as Fig. 6c shows. The first blank window corresponds to row 2 and column 1, i.e., position (2, 1), so in this blank window, the bit from position (2, 1) of the mother chromosome of Fig. 6b is copied, that is why 0 is put in position (2, 1) of the offsping of Fig. 6c. Similarly, the rest eight blank windows of Fig. 6c are filled.
3. If there is a need for a second offspring (e.g., for the case of pair 4 and pair 5 of Fig. 5f), then this second offspring is created by copying from the mother chromosome the selected rows and columns of the first step, while any remaining blanks are filled with bits copied from the elite chromosome. For example, the elite chromosome of Fig. 6a mates with the mother chromosome of Fig. 6b and produces the offspring of Fig. 6d. More specifically, the offspring of Fig. 6d has exacly the same bits with the mother chromosome of Fig. 6b in the rows 1, 3, 5, 6, 9, 10 and in the columns 2, 3, 6, 7, 8. However, with this operation, nine windows are blank, as Fig. 6d shows. The first blank window corresponds to row 2 and column 1, i.e., position (2, 1), so in this blank window, the bit from position (2, 1) of the elite chromosome of Fig. 6a is copied, that is why 0 is put in position (2, 1) of the offsping of Fig. 6d. Similarly, the rest eight blank windows of Fig. 6d are filled.

a

Unit	Hours (1-10)									
1	1	1	0	0	1	1	1	1	0	0
2	0	0	1	1	1	0	0	0	0	1
3	1	0	0	1	0	0	0	0	0	0
4	0	0	0	0	1	1	1	0	1	1
5	1	0	1	1	1	0	0	0	0	0
6	0	0	1	1	1	1	1	1	1	1
7	1	1	1	0	0	1	0	1	1	0
8	1	0	0	0	1	1	1	0	0	1
9	0	0	0	1	1	0	0	0	1	0
10	0	1	1	0	0	1	1	0	1	1

b

Unit	Hours (1-10)									
1	1	1	1	0	1	1	0	0	1	0
2	0	0	0	0	0	1	1	1	1	0
3	1	1	1	1	0	0	0	1	1	1
4	0	0	0	0	1	1	0	0	1	0
5	1	0	0	1	1	1	0	0	0	0
6	0	0	1	1	1	1	1	0	1	1
7	1	1	1	0	0	0	0	1	0	0
8	1	0	0	0	0	0	0	1	1	1
9	0	0	0	0	1	0	1	0	1	1
10	0	0	0	0	1	1	0	0	1	0

c

Unit	Hours (1-10)									
1	1	1	0	0	1	1	1	1	0	0
2	0	0	1	0	0	0	0	0	1	0
3	1	0	0	1	0	0	0	0	0	0
4	0	0	0	0	1	1	1	0	1	0
5	1	0	1	1	1	0	0	0	0	0
6	0	0	1	1	1	1	1	1	1	1
7	1	1	1	0	0	1	0	1	0	0
8	1	0	0	0	0	1	1	0	1	1
9	0	0	0	1	1	0	0	0	1	0
10	0	1	1	0	0	1	1	0	1	1

d

Unit	Hours (1-10)									
1	1	1	1	0	1	1	0	0	1	0
2	0	0	0	1	1	1	1	1	0	1
3	1	1	1	1	0	0	0	1	1	1
4	0	0	0	0	1	1	0	0	1	1
5	1	0	0	1	1	1	0	0	0	0
6	0	0	1	1	1	1	1	0	1	1
7	1	1	1	0	0	0	0	1	1	0
8	1	0	0	0	1	0	0	1	0	1
9	1	1	1	0	1	1	0	0	1	0
10	0	0	0	1	1	1	1	1	0	1

Fig. 6. Application example of the proposed multiple window crossover operator. (a) Elite chromosome. (b) Mother chromosome. (c) First offspring. (d) Second offspring (if required).

3.9. Mutation

Mutation operators insert unanticipated disturbances to the population in order to cause the exploration of new attractive regions of objective function (6).

This paper introduces the concept of window in window mutation operator. At first, a chromosome $k > 1$ and a unit i are randomly selected. Then, for that unit i, an initial window of ones or an initial window of zeros is randomly selected. If the selected initial window is a window of ones, then, starting at a random point inside the initial window, a full of zeros internal window (with random width greater than or equal to $T_{down}(i)$ so that the $T_{down}(i)$ constraint (9) is satisfied) overwrites previously set to one bits. On the other hand, if the selected initial window is a window of zeros,

a

Unit	Hours (1-10)									
1	0	0	0	1	1	1	1	1	1	0
2	1	1	1	1	0	0	0	0	0	1

b

Unit	Hours (1-10)									
1	0	0	0	1	0	0	0	0	1	0
2	1	1	1	1	0	1	1	1	0	1

Fig. 7. Application example of the proposed window in window mutation operator. (a) Chromosome before mutation. (b) Chromosome after mutation.

then the dual procedure is carried out and an internal window full of ones overwrites previously set to zero bits.

Fig. 7 shows the application of the proposed window in window mutation operator to the chromosome of Fig. 3, which is replicated in Fig. 7a for presentation purposes. More specifically:

1. Let us suppose that for unit 1, the initial window of six consecutive ones, shown with dark shading in Fig. 7a, has been selected. Next, an internal window width equal to 4 has been randomly selected. Next, within the initial window of ones, an internal window of four zeros starting from the randomly selected hour 5, overwrites previously set to one bits. Finally, as Fig. 7b shows, this internal window of four zeros starts at hour 5, i.e., a window full of zeros from hour 5 to hour 8 is inserted, which satisfies $T_{down}(1)$ constraint, since $T_{down}(1) = 3$.
2. Let us suppose that for unit 2, the initial window of five consecutive zeros, shown with light shading in Fig. 7a, has been selected. Next, an internal window width equal to 3 has been randomly selected. Next, within the initial window of zeros, an internal window of three ones starting from the randomly selected hour 6, overwrites previously set to zero bits. Finally, as Fig. 7b shows, this internal window of three ones starts at hour 6, i.e., a window full of ones from hour 6 to hour 8 is inserted, which satisfies $T_{up}(2)$ constraint, since $T_{up}(2) = 2$.

The window in window mutation operator is applied NM times, where NM is user defined. Larger settings for NM will result in a population with more scattered chromosomes.

3.10. Elite mutation

This paper introduces the elite mutation operator, which intensifies in the vicinity of the current best solution found by the MA. This new operator is the local search scheme of the proposed MA. The elite mutation operator is identical to the mutation operator (Section 3.9) with the only difference that it is applied on the elite chromosome of a generation using the following three user defined parameters:

1. The elite chromosome mutation probability, ECMP, which expresses the number of generations the elite mutation operator will be applied. As an example, if ECMP = 0.25 and the generations are 300, then the elite mutation will be applied to only 75 generations out of the 300 generations.
2. The elite chromosome mutated scenarios, ECMS, which expresses the total number of mutated versions of the elite chromosome that are created and evaluated in an attempt to find a higher profit chromosome than the current elite chromosome. Consequently, a steepest policy rather than a best first policy is utilized meaning that the current elite chromosome is replaced only if after the evaluation of the ECMS mutated versions, the highest profit neighbour found has higher profit than the current elite. As an example, if ECMS = 100 it means that 100 mutated versions of the elite chromosome will be evaluated.
3. The number of elite mutations, NEM, which specifies the number of times the window in window mutation operator (Section 3.9) will be applied on the current elite chromosome, thus creating a mutated version of it. As an example, if NEM = 4, it means that every mutated version of the current elite chromosome will be created by applying four times the window in window mutation operator on the current elite chromosome. NEM has to be set to a small number in order to take place the desired intensification, otherwise the elite mutation operator will act as a diversification operator and the algorithm will stagnate.

It should be noted that the GA of Section 4 does not include this elite mutation operator, meaning that the GA does not apply local search on the elite chromosome as the MA.

4. Results and discussion

4.1. Introduction

The proposed memetic algorithm is tested on six test systems, out of which two are existing test systems of the PBUC literature that do not include ramp up and ramp down constraints, while the rest four are new PBUC test systems with ramp up and ramp down constraints that are introduced in this paper so as to provide new PBUC benchmarks for future research.

All test systems are named based on the number of units N and the number of scheduling hours T, as follows: (a) $N \times T$ for the case of excluding ramp up and ramp down constraints, or (b) $N \times T$+ for the case of including ramp up and ramp down constraints. This coding shows that the postfix add sign (+) states that ramp up and ramp down constraints are included.

The following procedure is implemented for all test systems:

1. The optimal parameter settings of the genetic algorithm (GA) are identified. These parameters are: (i) the population size, (ii) the penalty, (iii) the number of generations, and (iv) the NM. It should be noted that the GA does not include the elite mutation operator of Section 3.10. Next, 100 simulation runs of the GA are executed, using the GA optimal parameter settings, and the results are stored for comparison purposes.
2. The same optimal settings of the genetic algorithm are also used for the memetic algorithm. Moreover, the optimal settings of the additional parameters of the MA are identified. These parameters are: (i) the ECMP, (ii) the ECMS, and (iii) the NEM. Next, 100 simulation runs of the MA are executed, using the MA optimal parameter settings, and the results are stored for comparison purposes.
3. Each PBUC problem is also solved using the Lagrangian relaxation (LR) and the simulated annealing (SA) method.
4. The solutions of the above methods (GA, MA, SA, and LR) are compared and conclusions are drawn. It should be noted that these four methods have been implemented in C++ on a computer with Pentium 1.5 GHz processor.

4.2. Existing test systems

Two existing test systems are used: (a) test system 10 × 24, the data of which can be found in [32,45], where it is called A10 test system, and (b) test system 110 × 24, the data of which can be

Table 1
Price and demand forecast for 4 × 8+ test system.

Hour	1	2	3	4	5	6	7	8
Price ($/MW h)	32.1	44.56	51.47	48.34	41.18	34.1	28.4	22.1
Demand (MW)	214	227	238	235	225	214	210	205

Table 2
Unit data for 4 × 8+ test system.

Unit i	$P_{min}(i)$ (MW)	$P_{max}(i)$ (MW)	$A(i)$ (\$/h)	$B(i)$ (\$/MW h)	$C(i)$ (\$/MW^2 h)	$T_{up}(i)$ (h)	$T_{down}(i)$ (h)	$X(i, 0)$ (h)	$R_{up}(i)$ (MW/h)	$R_{down}(i)$ (MW/h)	$P(i, 0)$ (MW)	$D(i)$ (\$/h)	$E(i)$ (\$/h)	$CT(i)$ (h)	$SD(i)$ (\$)
1	27	90	47.38	21.3913	0.06512	4	3	−1	61	74	0	25	30	5	8
2	38	150	43.49	19.1342	0.06124	3	2	2	45	63	72	105	100	4	12
3	80	230	39.67	16.2916	0.05768	4	3	−1	105	82	0	220	225	5	23
4	115	350	36.93	17.7604	0.05957	3	2	2	120	116	143	255	250	4	31

Table 3
Ramp up and ramp down constraints for 10 × 24+ test system.

Unit i	1	2	3	4	5	6	7	8	9	10
$R_{up}(i)$ (MW/h)	35	66	80	55	143	128	271	55	161	143
$R_{down}(i)$ (MW/h)	55	70	93	118	139	261	276	83	150	87

found in [32,45,46], where it is called A110 test system. The aforementioned systems exclude ramp up and ramp down constraints.

4.3. New test systems

Four new test systems are introduced in this paper:

1. 4 × 8+ test system, the data of which are presented in Tables 1 and 2. This test system was on purpose selected to be relatively small, so it can be solved by complete enumeration, which guarantees the discovery of the global optimum, and this global optimum is compared with the solution obtained by the MA as an initial check of the ability of the proposed MA to find the global optimum.
2. 10 × 24+ test system. This test system shares exactly the same data as test system 10 × 24 with the addition of the ramp up and ramp down constraints shown in Table 3.
3. 60 × 24+ test system. It is based on 10 × 24+. The 60 × 24+ test system is created by copying six times the units of 10 × 24+ (it means that units 1, 11, 21, 31, 41, and 51 are the same; units 2, 12, 22, 32, 42, and 52 are the same, etc.), by multiplying by six the forecasted demand at each hour, while the price forecast is the same as in 10 × 24 and 10 × 24+.
4. 110 × 24+ test system. This test system shares exactly the same data as test system 110 × 24 with the addition of the ramp up and ramp down constraints shown in Table 4.

It should be noted that in the above mentioned systems, there is a big variability in the electricity price during the scheduling horizon. More specifically, three different electricity price profiles are considered:

1. The electricity price for the 4 × 8+ test system ranges from \$22.100/MW h to \$51.470/MW h, as can be seen from Table 1.
2. The 10 × 24, 10 × 24+, and 60 × 24+ test systems have the same electricity price profile. This electricity price is very low, since it ranges from \$1.899/MW h to \$2.498/MW h [32]. This electricity price profile can be seen in Fig. 9.
3. The 110 × 24 and 110 × 24+ test systems have the same electricity price profile. This electricity price ranges from \$12.976/MW h to \$24.050/MW h [46].

4.4. Optimal parameter settings

Tables 5 and 6 present the optimal parameter settings of the GA and MA, respectively, which were found after enough trials. As already mentioned, the same optimal parameter settings of the GA are also used for the MA. The optimal settings of the three additional parameters (i.e., ECMP, ECMS, NEM) of the MA are also shown in Table 6.

Table 4
Ramp up and ramp down constraints for 110 × 24+ test system.

Unit i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
$R_{up}(i)$ (MW/h)	5	11	10	5	3	11	6	9	8	34	18	66	16	26	83	98
$R_{down}(i)$ (MW/h)	11	9	10	5	10	12	6	5	6	27	32	38	32	90	58	53
Unit i	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
$R_{up}(i)$ (MW/h)	144	106	95	124	148	192	163	279	118	230	429	402	79	74	34	19
$R_{down}(i)$ (MW/h)	125	92	89	140	117	183	157	143	243	232	456	341	53	90	23	11
Unit i	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48
$R_{up}(i)$ (MW/h)	63	167	254	379	31	42	60	67	78	114	320	362	505	688	15	22
$R_{down}(i)$ (MW/h)	36	123	351	203	33	53	88	102	107	77	337	448	544	590	25	6
Unit i	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64
$R_{up}(i)$ (MW/h)	26	49	44	13	55	18	14	34	54	48	87	60	78	50	79	57
$R_{down}(i)$ (MW/h)	52	14	46	15	57	13	32	31	50	91	81	59	68	114	128	150
Unit i	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80
$R_{up}(i)$ (MW/h)	126	122	110	128	157	313	318	278	136	183	87	19	342	513	74	74
$R_{down}(i)$ (MW/h)	81	94	111	168	142	248	233	165	122	222	64	15	406	519	59	84
Unit i	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96
$R_{up}(i)$ (MW/h)	26	32	69	97	243	422	16	55	39	42	63	43	116	380	142	376
$R_{down}(i)$ (MW/h)	44	34	53	77	145	272	22	54	24	148	118	90	390	106	515	214
Unit i	97	98	99	100	101	102	103	104	105	106	107	108	109	110		
$R_{up}(i)$ (MW/h)	12	8	19	10	49	42	55	117	93	213	101	83	160	97		
$R_{down}(i)$ (MW/h)	11	11	9	21	45	22	51	67	76	123	498	54	217	82		

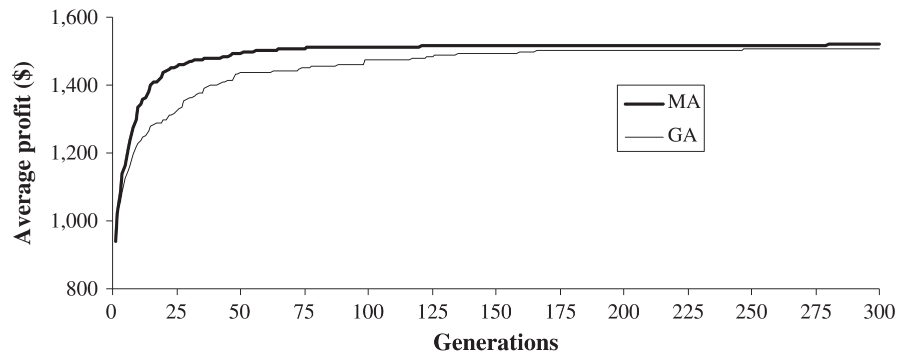

Fig. 8. Average profit versus MA and GA generations for the 10 × 24+ test system.

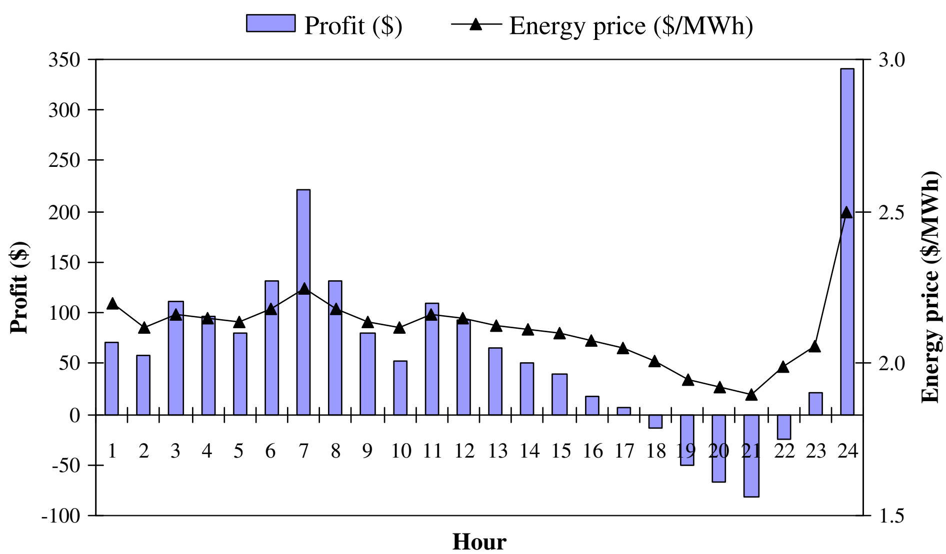

Fig. 9. Energy price per hour and total profit (computed by the MA) per hour for the 10 × 24+ test system.

4.5. Results for 4 × 8+ test system

The 4 × 8+ test system was used as an initial test of the capabilities of the proposed MA. Since 4 × 8+ is a relatively small test system, the PBUC problem for this system was solved by complete enumeration and the global optimum found had a total profit of $24647.62, as computed by (6), while the respective global optimum PBUC schedule is shown in Table 7.

It can be seen from Table 8 that the MA as well as the GA managed to find the global optimum solution, using the optimal parameter settings of Tables 6 and 5, respectively. LR and simulated annealing (SA) also found the global optimum, as Table 8 shows. Moreover, after 100 simulation runs, the average profit found by the proposed memetic algorithm is $24647.61. The

Table 5
GA optimum parameter settings.

Test system	Population	Penalty	Generations	NM
4 × 8+	25	10^7	300	200
10 × 24	25	10^7	300	25
10 × 24+	25	10^7	300	25
60 × 24+	25	10^7	1000	50
110 × 24	25	10^7	1000	6000
110 × 24+	25	10^7	1000	6000

Table 7
Global optimum PBUC schedule for 4 × 8+ test system.

Unit	Hours (1–8)							
1	0	0	1	1	1	1	1	0
2	1	1	1	1	1	1	1	1
3	0	0	0	1	1	1	1	1
4	1	1	1	0	0	0	0	0

Table 6
MA optimum parameter settings.

Test system	Population	Penalty	Generations	NM	ECMP	ECMS	NEM
4 × 8+	25	10^7	300	200	0.25	100	4
10 × 24	25	10^7	300	25	0.25	100	5
10 × 24+	25	10^7	300	25	0.25	100	5
60 × 24+	25	10^7	1000	50	0.40	150	2
110 × 24	25	10^7	1000	6000	0.40	150	2
110 × 24+	25	10^7	1000	6000	0.40	150	2

Table 8
Profit ($) comparison for six test systems and four optimization methods.

Test system	Parameter	LR	SA	GA	MA
$4 \times 8+$	Best profit ($)	24647.62[a]	24647.62[a]	24647.62[a]	24647.62[a]
$4 \times 8+$	Average profit ($)		24646.79	24647.27	24647.61
$4 \times 8+$	Worst profit ($)		24638.53	24642.12	24646.95
$4 \times 8+$	Success rate (%)		86	91	97
10×24	Best profit ($)	1897.01	1899.41[a]	1899.41[a]	1899.41[a]
10×24	Average profit ($)		1898.85	1899.21	1899.39
10×24	Worst profit ($)		1894.38	1896.15	1898.83
10×24	Success rate (%)		84	88	96
$10 \times 24+$	Best profit ($)	1533.09	1534.40[a]	1534.40[a]	1534.40[a]
$10 \times 24+$	Average profit ($)		1533.89	1534.06	1534.39
$10 \times 24+$	Worst profit ($)		1530.23	1531.47	1533.92
$10 \times 24+$	Success rate (%)		83	86	97
$60 \times 24+$	Best profit ($)	9102.09	9128.45	9131.69	9154.83[a]
$60 \times 24+$	Average profit ($)		9126.77	9129.83	9154.78
$60 \times 24+$	Worst profit ($)		9123.58	9125.45	9153.51
$60 \times 24+$	Success rate (%)		0	0	94
110×24	Best profit ($)	1369726.41	1373455.12	1374808.60	1378549.12[a]
110×24	Average profit ($)		1372783.58	1373858.28	1378415.76
110×24	Worst profit ($)		1369567.15	1370185.91	1376057.19
110×24	Success rate (%)		0	0	92
$110 \times 24+$	Best profit ($)	1347553.75	1351588.38	1352677.25	1356988.62[a]
$110 \times 24+$	Average profit ($)		1350645.23	1351293.37	1356763.04
$110 \times 24+$	Worst profit ($)		1346989.12	1347539.75	1354316.53
$110 \times 24+$	Success rate (%)		0	0	89

[a] This value is considered as the optimum solution (solution with best profit among all methods) for the respective test system and success rates are computed according to this optimal solution for the respective test system.

success rate of the proposed memetic algorithm is 97%, that is, 97 times out of 100 simulation runs the same optimal answer (i.e., $24647.62) is obtained. In addition to the best, average and worst profit, the success rate is another very good measure for error analysis and comparison of metaheuristic methods (such as SA, GA, and MA) [47], that is why the value of the success rate is given in Table 8 for all the considered test systems and metaheuristic optimization methods. Since LR is a deterministic optimization method, it always provides the same solution for the same problem, that is why only this single value (best profit) is given in Table 8 for the LR method.

4.6. PBUC results for test systems without ramp up and ramp down constraints

Table 8 shows that for the 10×24 test system, the MA, GA, and SA found the same best solution with $1899.41 profit, which is 0.13% higher than the profit found by LR.

In case of 110×24 test system, the best solution was found only by the MA, as Table 8 shows, that is why the success rate is zero for GA and SA. More specifically, the best solution of MA has 0.27%, 0.37%, and 0.64% higher profit in comparison with the best profit of GA, SA, and LR, respectively.

In conclusion, among all the considered metaheuristic optimization methods and for all the examined test systems, the proposed MA provides the highest success rate in finding the optimal solution, as Table 8 shows.

The average CPU time needed by the MA to solve the 10×24 and the 110×24 test system is 4.04 and 492.84 s, respectively.

4.7. PBUC results for test systems with ramp up and ramp down constraints

Table 8 shows that the proposed MA outperforms GA and SA for $4 \times 8+$, $10 \times 24+$, $60 \times 24+$, and $110 \times 24+$ test systems, since MA provides higher average total profit over 100 simulation runs, higher worst profit, and higher success rate in comparison with GA and SA. It can be also seen from Table 8 that for large test systems with 60 units or more, the MA constantly outperforms the LR, since the profit calculated even by the worst MA solution is always higher than the profit calculated by the LR method.

The inclusion of the ramp up and ramp down constraints obviously reduces the profit. In particular, the average profit computed by the MA is higher for the 10×24 test system in comparison with the $10 \times 24+$ system, as Table 8 shows. The same conclusion exists for the 110×24 and $110 \times 24+$ test systems.

Table 9
MA based PBUC schedule for $10 \times 24+$ test system.

Unit	Hours (1–24)																						
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0
3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
4	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
8	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
9	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
10	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Fig. 8 shows the average profit, after 100 simulation runs, during each generation of the MA and the GA for the $10 \times 24+$ test system. It can be seen that both the MA and GA have quite fast convergence rates, however the MA converges faster than the GA.

Fig. 9 shows the electricity price profile together with the profit curve that was found by the MA for the $10 \times 24+$ test system. It is concluded from Fig. 9 that the maximum profit, i.e., $340.89, is obtained during hour 24, when the energy price takes its maximum value ($2.498/MW h). On the other hand, there is loss during hours 18–22, when the energy price takes very low values.

Table 9 shows the MA based PBUC schedule for the $10 \times 24+$ test system that corresponds to the maximum total profit of $ 1534.40 found by the MA.

The average CPU time needed by the MA to solve the $4 \times 8+$, $10 \times 24+$, $60 \times 24+$, and $110 \times 24+$ test systems is 0.28, 5.11, 170.85, and 657.12 s, respectively.

5. Conclusion

An advanced memetic algorithm (MA) solution to the price based unit commitment problem has been presented. The MA is a valuable tool in searching large discrete solution spaces, and in PBUC the solution space is quite large, making the MA ideal for the PBUC problem. The main contributions of this paper are: (i) an innovative two-level tournament selection, (ii) a new multiple window crossover, (iii) a novel window in window mutation operator, (iv) an innovative local search scheme called elite mutation, (v) new population initialization algorithm that is specific to PBUC problem, and (vi) new PBUC test systems including ramp up and ramp down constraints so as to provide new PBUC benchmarks for future research.

The method has been challenged on test systems of up to 110 units and the results show that in every case examined the proposed MA converged to higher profit PBUC schedules than the genetic algorithm, the simulated annealing, and the Lagrangian relaxation method. Moreover, among all the considered metaheuristic optimization methods and for all the examined test systems, the proposed MA provides the highest success rate in finding the optimal solution. Furthermore, for large test systems with 60 units or more, the proposed MA constantly outperforms the LR, since the profit calculated even by the worst MA solution is always higher than the profit calculated by the LR method. Additionally, the proposed MA is feasible from a computational viewpoint. Future work includes the investigation of ecological constraints and the simultaneous optimization of energy and ancillary services markets.

References

[1] Georgopoulou CA, Giannakoglou KC. Metamodel-assisted evolutionary algorithms for the unit commitment problem with probabilistic outages. Appl Energy 2010;87:1782–92.
[2] Niknam T, Khodaei A, Fallahi F. A new decomposition approach for the thermal unit commitment problem. Appl Energy 2009;86:1667–74.
[3] Hawkes AD, Leach MA. Modelling high level system design and unit commitment for a microgrid. Appl Energy 2009;86:1253–65.
[4] Georgopoulou CA, Giannakoglou KC. Two-level, two-objective evolutionary algorithms for solving unit commitment problems. Appl Energy 2009;86:1229–39.
[5] Delarue E, D'haeseleer W. Adaptive mixed-integer programming unit commitment strategy for determining the value of forecasting. Appl Energy 2008;85:171–81.
[6] Thorin E, Brand H, Weber C. Long-term optimization of cogeneration systems in a competitive market environment. Appl Energy 2005;81:152–69.
[7] Voorspools KR, D'haeseleer WD. Long-term unit commitment optimisation for large power systems: unit decommitment versus advanced priority listing. Appl Energy 2003;76:157–67.
[8] Wood AJ, Wollenberg BF. Power generation, operation and control. 2nd ed. New York: Wiley; 1996.
[9] Sheblé GB, Fahd GN. Unit commitment literature synopsis. IEEE Trans Power Syst 1994;9:128–35.
[10] Snyder WL, Powell HD, Rayburn JC. Dynamic programming approach to unit commitment. IEEE Trans Power Syst 1987;2:339–48.
[11] Zhuang F, Galiana FD. Towards a more rigorous and practical unit commitment by Lagrangian relaxation. IEEE Trans Power Syst 1988;3:763–73.
[12] Cohen AI, Yoshimura M. A branch-and-bound algorithm for unit commitment. IEEE Trans Power Apparatus Syst 1983;102:444–51.
[13] Baptistella LFB, Geromel JC. Decomposition approach to problem of unit commitment schedule for hydrothermal systems. IEE Proc Control Theory Appl 1980;127:250–8.
[14] Zhai Q, Guan X, Yang J. Fast unit commitment based on optimal linear approximation to nonlinear fuel cost: error analysis and applications. Elect Power Syst Res 2009;79:1604–13.
[15] Zhuang F, Galiana FD. Unit commitment by simulated annealing. IEEE Trans Power Syst 1990;5:311–8.
[16] Wang C, Shahidehpour SM. A decomposition approach to nonlinear multi-area generation scheduling with tie-line constraints using expert systems. IEEE Trans Power Syst 1992;7:1409–18.
[17] Sasaki H, Watanabe M, Kubokawa J, Yorino N, Yokoyama R. A solution method of unit commitment by artificial neural networks. IEEE Trans Power Syst 1992;7:974–81.
[18] Kazarlis SA, Bakirtzis AG, Petridis V. A genetic algorithm solution to the unit commitment problem. IEEE Trans Power Syst 1996;11:83–92.
[19] Wong KP, Wong YW. Thermal generator scheduling using hybrid genetic/simulated annealing approach. IEE Proc Gener Transm Distrib 1995;142:372–80.
[20] Yamin HY. Review on methods of generation scheduling in electric power systems. Elect Power Syst Res 2004;69:227–48.
[21] Padhy NP. Unit commitment – a bibliographical survey. IEEE Trans Power Syst 2004;19:1196–205.
[22] Shahidehpour M, Yamin H, Li Z. Market operations in electric power systems: forecasting, scheduling, and risk management. New York: Wiley; 2002.
[23] Delarue E, Van Den Bosch P, D'Haeseleer W. Effect of the accuracy of price forecasting on profit in a price based unit commitment. Elect Power Syst Res 2010;80:1306–13.
[24] Bisanovic S, Hajro M, Dlakic M. Hydrothermal self-scheduling problem in a day-ahead electricity market. Elect Power Syst Res 2008;78:1579–96.
[25] Simoglou CK, Biskas PN, Bakirtzis AG. Optimal self-scheduling of a thermal producer in short-term electricity markets by MILP. IEEE Trans Power Syst 2010;25:1965–77.
[26] Li T, Shahidehpour M. Price-based unit commitment: a case of Lagrangian relaxation versus mixed integer programming. IEEE Trans Power Syst 2005;20:2015–25.
[27] De La Torre S, Arroyo JM, Conejo AJ, Contreras J. Price maker self-scheduling in a pool-based electricity market: a mixed-integer LP approach. IEEE Trans Power Syst 2002;17:1037–42.
[28] Arroyo JM, Conejo AJ. Optimal response of a power generator to energy, AGC, and reserve pool-based markets. IEEE Trans Power Syst 2002;17:404–10.
[29] Arroyo JM, Conejo AJ. Optimal response of a thermal unit to an electricity spot market. IEEE Trans Power Syst 2000;15:1098–104.
[30] Richter CW, Sheblé GB. A profit-based unit commitment GA for the competitive environment. IEEE Trans Power Syst 2000;15:715–21.
[31] Georgilakis PS. Genetic algorithm model for profit maximization of generating companies in deregulated electricity markets. Appl Artif Intell 2009;23:538–52.
[32] Pokharel BK, Shrestha GB, Lie TT, Fleten S-E. Profit based unit commitment in competitive markets. In: Proc int conf power system technology. p. 1728–33; 2004.
[33] Yamin HY, Shahidehpour SM. Unit commitment using a hybrid model between Lagrangian relaxation and genetic algorithm in competitive electricity markets. Elect Power Syst Res 2003;68:83–92.
[34] Attaviriyanupap P, Kita H, Tanaka E, Hasegawa J. A hybrid LR-EP for solving new profit-based UC problem under competitive environment. IEEE Trans Power Syst 2003;18:229–37.
[35] Yamin HY, El-Dwairi Q, Shahidehpour SM. A new approach for GenCos profit based unit commitment in day-ahead competitive electricity markets considering reserve uncertainty. Int J Electr Power Energy Syst 2007;29:609–16.
[36] Moscato P. On evolution, search, optimization, genetic algorithms and martial arts: towards memetic algorithms. Technical report caltech concurrent computation program, rep. 826. Pasadena (CA, USA): California Institute of Technology; 1989.
[37] Krasnogor N, Smith J. A tutorial for competent memetic algorithms: model, taxonomy, and design issues. IEEE Trans Evol Comput 2005;9:474–88.
[38] Augugliaro A, Dusonchet L, Riva-Sanseverino E. Service restoration in compensated distribution networks using a hybrid genetic algorithm. Elect Power Syst Res 1998;46:59–66.
[39] Mendes A, França PM, Lyra C, Pissara C, Cavellucci C. Capacitor placement in large-sized radial distribution networks. IEE Proc Gener Transm Distrib 2005;152:496–502.
[40] Burke EK, Smith AJ. Hybrid evolutionary techniques for the maintenance scheduling problem. IEEE Trans Power Syst 2000;15:122–8.
[41] Urdaneta AJ, Gómez JF, Sorrentino E, Flores L, Díaz R. A hybrid genetic algorithm for optimal reactive power planning based upon successive linear programming. IEEE Trans Power Syst 1999;14:1292–8.

[42] Damousis IG, Bakirtzis AG, Dokopoulos PS. A solution to the unit-commitment problem using integer-coded genetic algorithm. IEEE Trans Power Syst 2004;19:1165–72.
[43] Nocedal J, Wright SJ. Numerical optimization. 2nd ed. New York: Springer; 2006.
[44] Haupt RL, Haupt SE. Practical genetic algorithms. 2nd ed. New Jersey: John Wiley & Sons; 2004.
[45] Orero SO, Irving MR. Large scale unit commitment using a hybrid genetic algorithm. Int J Elect Power Energy Syst 1997;19:45–55.
[46] Huse ES, Wangensteen I, Faanes HH. Thermal power generation scheduling by simulated competition. IEEE Trans Power Syst 1999;14:472–7.
[47] Georgilakis PS. Differential evolution solution to transformer no-load loss reduction problem. IET Gener Transm Distrib 2009;3:960–9.