Expert Systems with Applications 40 (2013) 6080–6086

Contents lists available at SciVerse ScienceDirect

Expert Systems with Applications

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

Genetic algorithm with binary representation of generating unit start-up and shut-down times for the unit commitment problem

Grzegorz Dudek *

Department of Electrical Engineering, Czestochowa University of Technology, Al. Armii Krajowej 17, 42-200 Czestochowa, Poland

ARTICLE INFO

Keywords:
Unit commitment
Power generation dispatch
Genetic algorithms
Evolutionary computation
Combinatorial optimization

ABSTRACT

An approach for solving the unit commitment problem based on genetic algorithm with binary representation of the unit start-up and shut-down times is presented. The proposed definition of the decision variables and their binary representation reduce the solution space and computational time in comparison to the classical genetic algorithm approach to unit commitment. The method incorporates time-dependent start-up costs, demand and reserve constraints, minimum up and down time constraints and units power generation limits. Penalty functions are applied to the infeasible solutions. Test results showed an improvement in effectiveness and computational time compared to results obtained from genetic algorithm with standard binary representation of the unit states and other methods.

1. Introduction

The unit commitment (UC) problem involves scheduling the on/off status of generating units as well as their real power outputs within a power system over the future short-term period (from a day to a week). The resulting schedule should minimize the production costs and satisfy the required demand and a set of operating constraints at any time during the period. The optimal schedule may save the electric utilities millions of dollars per year in production costs and that is why the UC problem is so important in the daily operational planning of each power system.

The UC problem is a complex optimization problem with both discrete (unit commitment) and continuous (generation levels) variables. Because the complete enumeration method for UC is useless for practical systems (computer execution time for this method is usually too immense), research efforts have been focused on efficient, suboptimal UC algorithms which can be applied to realistic power systems. The method of UC can be divided as follows (Sen & Kothari, 1998; Sheblé & Fahd, 1994):

- Heuristic methods such as priority list.
- Classical optimization methods such as: dynamic programming, Lagrangian relaxation, branch-and-bound, linear programming, integer programming.
- Computational intelligence methods such as: expert systems, neural networks, simulated annealing, genetic algorithms.

The simplest UC method and also easiest to implement is the priority list method. The priority list specifies the order in which units start-up and shut-down. Unfortunately the quality of the solution is usually far from optimal due to the incomplete search of the solution space.

Many methods belonging to the second group such as branch-and-bound, dynamic and integer programming suffer from the "curse of dimensionality". This is manifested by the rapid increasing of the problem size and in consequence the computation time with the number of generating units to be committed. Several approaches have been adopted in order to reduce the search space. Most of them are based on the priority list technique: dynamic programming-sequential combination, dynamic programming-truncated combination (Pang & Chen, 1976; Pang, Sheblé, & Albuyeh, 1981), thus the solution obtained is suboptimal.

Lagrangian relaxation is a relaxation method which approximates a difficult problem of constrained optimization by a simpler one. In this approach the UC problem is decomposed into a master problem and more manageable subproblems. Each subproblem is solved independently and determines the commitment of a single unit. The subproblems are linked by the Lagrange multipliers, which are estimated at each iteration. This method has higher computational efficiency and is more flexible for handling different types of constraints compared to other approaches. However, because of the dual nature of the algorithm, its primary difficulty is associated with obtaining solution feasibility. Furthermore, the optimal value of the dual problem is not generally equal to that of the primal (original) problem.

An expert system is developed by combining the knowledge of experienced power system operators and unit commitment experts to assist operators in scheduling generating units. The

* Tel.: +48 343250896.
E-mail address: dudek@el.pcz.czest.pl

0957-4174/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.eswa.2013.05.010

knowledge is stored in an expert system rule base. However, a great deal of operator interaction is required in this approach, making it inconvenient and time-consuming.

Neural networks (most often multilayer perceptrons) are trained to recognize the most economical UC schedule associated with the pattern of the current load curve (Nayak & Sharma, 2000; Wong, Chung, & Wong, 2000). The training set contains typical load curves and corresponding UC schedules. If the neural network solution is not feasible for the entire UC period, it will be used as an initial starting point for a near-optimal solution.

Fuzzy logic provides an effective conceptual framework for dealing with the problem of knowledge representation in an environment of uncertainty and imprecision. Fuzzy approach allows taking into account many uncertainties involved in the planning and operation of power systems. The key factors such as load demand and reserve margin are treated as fuzzy variables (Padhy, 2000; Padhy, Paranjothi, & Ramachandran, 1997). A fuzzy decision system has been developed to select the units to be on or off based on these fuzzy variables expressing the forecast error and the soft limits of the spinning reserve requirements.

Simulated annealing is a powerful, general-purpose stochastic method for solving combinatorial optimization problems such as UC (Dudek, 2010; Mantawy, Abdel-Magid, & Selim, 1998). It has the ability of escaping local minima and converges in the limit to a globally optimal solution with probability 1. The main advantages of this method are that a complicated mathematical model of the problem under study is not needed, the starting point can be any given solution and it will attempt to improve it, the final solution does not strongly depend on the initial solution and it does not need large computer memory. One main drawback and limiting factor of this method is that it takes a great deal of CPU time to find the near-optimal solution.

Genetic algorithm (GA) is an adaptive and parallel search techniques based on the mechanism of natural selection, reproduction and mutation. GA works with a population of candidate solutions (chromosomes or individuals) which encode the variables or parameters. GA can be used with both discrete and continuous variables. It uses probabilistic transition rules during searching the solution space. A simple GA implementation using the standard crossover and mutation operators can locate near-optimal solutions. However, by adding problem-specific operators and by the proper choice of variables and their representation, better solutions to the UC problem can be obtained (Dasgupta & McGregor, 1994; Dudek, 2004; Dudek, 2007; Kazarlis, Bakirtzis, & Petridis, 1996; Mantawy, Abdel-Magid, & Selim, 1999).

This paper presents a GA with binary representation of the unit start-up and shut-down times to solve the UC problem. This definition of the decision variables reduces the solution space and computational time. Now there is no need to use the complicated algorithm such as GA with specialized mutation operators described in (Dudek, 2004) to solve this simplified problem. The same definition of the decision variables was investigated in (Dudek, 2007) but the integer representation was used.

In the proposed approach there are three fitness function definitions: one for feasible solutions and two for unfeasible ones, which are dependent on the degrees of constraint violations. The combinatorial optimization problem is solved using the GA while the unit power generation levels are determined via the conventional method of Lagrange multipliers.

2. The mathematical model of unit commitment

(The symbols that appear in the following description are listed in Table 5 in Appendix A.)

The UC problem can be mathematically formulated as follows: Objective function:

$$F = \sum_{t=1}^{T}\sum_{i=1}^{N}(\alpha_i(t)C_i(P_i(t)) + \alpha_i(t)[1-\alpha_i(t-1)]SC_i(t_{offi})) \quad (1)$$

Constraints:

(a) Load balance

$$\forall t: \quad \phi = D(t) - \sum_{i=1}^{N}\alpha_i(t)P_i(t) = 0 \quad (2)$$

(b) Unit power generation limits

$$\forall i,t: \quad \alpha_i(t)P_{\min i} \leqslant P_i(t) \leqslant \alpha_i(t)P_{\max i} \quad (3)$$

(c) Set of unit power generation limits

$$\forall t: \quad \sum_{i=1}^{N}\alpha_i(t)P_{\min i} \leqslant D(t) \quad (4)$$

$$\forall t: \quad \sum_{i=1}^{N}\alpha_i(t)P_{\max i} \geqslant D(t) + R(t) \quad (5)$$

(d) Minimum up/down time

$$\forall i: \quad t_{offi} \geqslant t_{downi} \quad (6)$$

$$\forall i: \quad t_{oni} \geqslant t_{upi} \quad (7)$$

The variable production cost of unit i at time t: $C_i(P_i(t))$ is conventionally approximated by the quadratic function:

$$C_i(P_i) = a_iP_i^2 + b_iP_i + c_i \quad (8)$$

and the start-up cost of unit i: $SC_i(t_{offi})$ is expressed as a function of the number of hours the unit has been down:

$$SC_i(T_{offi}) = e_i \exp(-g_it_{offi}) + f_i \exp(-h_it_{offi}) \quad (9)$$

To take into account the costs connected with unit shut-down at time t, in the event that it remains in an off state to the end of time period T, it is assumed that:

- unit start-up costs are evenly distributed over the number of hours of unit down time,
- unit start-up occurs at time τ after the end of the optimization period T (τ {1,2,3,...}).

Taking these assumptions into account, unit (staying in down time until the end of time period T) start-up costs in time period T are calculated using the formula:

$$SC_i(T-t) = \frac{SC_i(T-t+\tau)}{T-t+\tau}(T-t) \quad (10)$$

3. The proposed genetic algorithm approach

The GA implementation consists of random initialization, determination of unit power generation levels, cost calculations, reproduction, crossover, mutation, transposition, and elitism. The determination of the optimal outputs of generating units is called an economic dispatch problem. This problem is treated here as a subproblem of UC and is solved for each time t using the Lagrange multiplier method.

The binary tournament is used as the selection method in GA. An elitism strategy is also used which copies the best individual into the next population. GA is terminated when there is no significant improvement in the solution after a pre-specified number of generations or when the maximum number of generations is reached.

3.1. Decision variables and their representation

In the presented approach the load demand is divided into time intervals in which the load curve increases or decreases. It is assumed that unit start-up can occur in the interval $[\tau^j_{up1}, \tau^j_{up2}]$ on the rising edge of the load demand curve and unit shut-down can occur in the interval $[\tau^k_{down1}, \tau^k_{down2}]$ on the falling edge of the load curve (the superscripts j and k denote the start-up or shut-down interval number respectively). These edges need not be monotonic. The intervals for the hourly load demand curve that is assumed in the application examples defined in Section 4 are shown in Fig. 1: $[\tau^1_{down1} = 1, \tau^1_{down2} = 4]$, $[\tau^1_{up1} = 5, \tau^1_{up2} = 13]$, $[\tau^2_{down1} = 14, \tau^2_{down2} = 15]$, $[\tau^2_{up1} = 16, \tau^2_{up2} = 18]$, $[\tau^3_{down1} = 19, \tau^3_{down2} = 24]$. There are two start-up intervals ($j = 1,2$) and three shut-down intervals ($k = 1,2,3$), which are separated by the local minimum and maximum values of the load.

The unit start-up and shut-down hours are the integer decision variables. For five intervals shown in Fig. 1, there are five variables for each unit $\mathbf{x} = [x_1, x_2, \ldots, x_5]$ defined as follows:

$$x_1 \in \Phi_1 = \{\tau^1_{down1}, \ldots, \tau^1_{down2} + 1\} \quad (11)$$

$$x_2 \in \Phi_2 = \{\tau^1_{up1}, \ldots, \tau^1_{up2} + 1\} \quad (12)$$

$$x_3 \in \Phi_3 = \{\tau^2_{down1}, \ldots, \tau^2_{down2} + 1\} \quad (13)$$

$$x_4 \in \Phi_4 = \{\tau^2_{up1}, \ldots, \tau^2_{up2} + 1\} \quad (14)$$

$$x_5 \in \Phi_5 = \{\tau^3_{down1}, \ldots, \tau^3_{down2} + 1\} \quad (15)$$

Some assumptions are adopted about variable values:

- when the variable representing the shut-down time is equal to $\tau^k_{down2} + 1$, it means that the unit shut-down in the kth interval does not occur: the unit is in on state during this interval,
- when the variable representing the start-up time is equal to $\tau^j_{up2} + 1$, it means that the unit start-up during the jth interval does not occur,
- when the unit is in off state before the optimization period T, the first interval is the shut-down type and $x_1 \neq 1$, it is assumed that the unit start-up is in the first hour,
- when the unit is in on state before the period T, the first interval is the start-up type and $x_1 \neq 1$, it is assumed that the unit shut-down is in the first hour.

For example, $\mathbf{x} = [5,5,16,16,25]$ means that the unit is in on state in the whole period T; $\mathbf{x} = [1,14,14,19,19]$ means that the unit is in off state in the whole period T; $\mathbf{x} = [2,7,14,18,25]$ means that the unit is in the on state in the intervals: until the 2nd hour, between the 7th and 13th hours and from the 18th hour to the end of period T, while it is in off state in the intervals: between the 2nd and 6th hours and between 14th and 17th hours.

GA searches the solution space through the evolution of a population of solutions. Each individual of the population is represented by an binary string composed of genes encoding consecutive variables x_i for each unit (the integer representation is also possible (Dudek, 2007)). The Gray code is used where two successive values differ in only one bit. The gene length depends on the range of x_i. To encode the five decision variables described above $B = 14$ bits are needed for each unit (the number of bits for the following decision variables are: 3, 4, 2, 2, 3). There is redundancy in such encoding: a gene can encode more values than the number of possible values of variable x_i. The solution is the possibility to assign two values of gene to one value of x_i.

The chromosome and its interpretation in Fig. 2 is presented. In Table 1 the decimal values of genes and corresponding to them the decision variable values are shown.

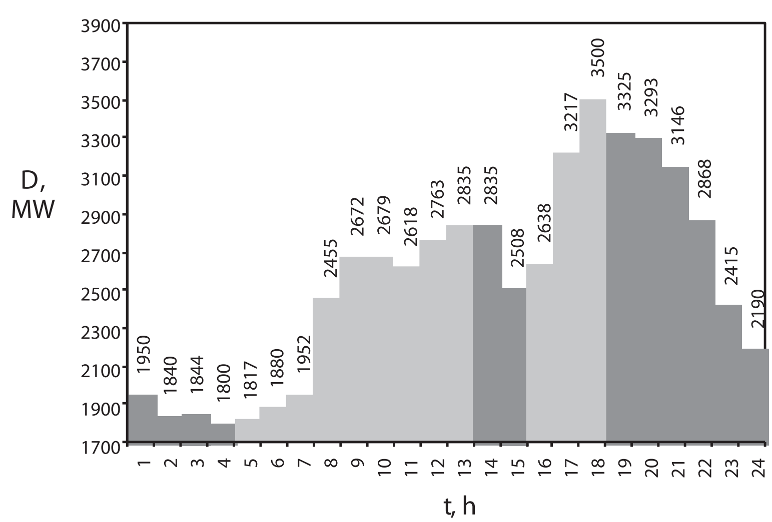

Fig. 1. The unit start-up (light bars) and shut-down (dark bars) intervals.

Chromosome	001	1100	11	01	111	…	011	1001	10	11	110
Interpretation	x_1	x_2	x_3	x_4	x_5	….	x_1	x_2	x_3	x_4	x_5
			for unit 1						for unit N		

Fig. 2. The chromosome and its interpretation.

Table 1
Decimal values of genes and the variable values corresponding to them.

Variable	Number of bits	Decimal value of gene															
		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
x_1	3	1	1	2	2	5	5	3	4								
x_2	4	5	5	6	6	8	8	7	7	14	14	12	13	9	9	11	10
x_3	2	14	15	16	16												
x_4	2	16	17	19	18												
x_5	3	19	20	22	21	25	25	23	24								

3.2. Economic dispatch

Since the production cost (8) is a quadratic function (convex and continuous), the economic dispatch problem for each t is solved using the method of Lagrange multipliers (Wood & Wollenberg, 1996). The Lagrange function is composed of the cost function (total variable production cost (8) of all units working at the time t) and the constraint function (2) multiplied by an undetermined multiplier λ:

$$L = \sum_{i \in \Omega(t)} C_i(P_i(t)) + \lambda \phi \tag{16}$$

The necessary conditions for a minimum of the total variable cost function result when we take the first derivative of the Lagrange function with respect to each of the independent variables (power outputs $P_i(t)$ and undetermined Lagrange multiplier λ) and set the derivatives equal to zero:

$$\frac{\partial L}{\partial P_i} = \frac{dC_i(P_i(t))}{dP_i(t)} - \lambda = 0, i \in \Omega(t). \tag{17}$$

The minimum cost operating condition is that the incremental cost rates of all units is equal to some undetermined value of λ. To this necessary condition we must add the constraint equation that the sum of the power outputs must be equal to the power demanded by the load $D(t)$. In addition inequalities (3) must be satisfied. To find the best value of λ we use the lambda-iteration method (Wood & Wollenberg, 1996). This is an iterative procedure in which we change in a systematic way the value of lambda:

1. Set an initial value for λ.
2. Find the corresponding output power of each generating unit.
3. If the total power is less than the load demand, increase λ and go to step 2.
4. If the total power is higher than the load demand, decrease λ and go to step 2.
5. The lambda-iteration procedure converges very rapidly to the global minimum for this particular type of optimization problem. Note that this method guarantees that load balance (2) is met and unit power generation limit constraints (3) are met if the set of unit power generation limit constraints (4) and (5) are met.

3.3. Cost calculations and the procedure with infeasible individuals

The solutions generated by GA can be divided into three groups: (i) satisfying all constraints (2)–(7), (ii) satisfying constraints (2)–(5) but violating minimum up/down time constraints (6) or (7), and (iii) violating set of unit power generation limit constraints (4) or (5). For solutions which satisfy all constraints generation levels $P_i(t)$ are determined using lambda-iteration method. Then we calculate unit production costs (8) and the value of objective function (1).

For solutions belonging to the group (ii), a penalty function is created (Dudek, 2004; Dudek, 2011):

$$F' = M\left(1 + \sum_{i=1}^{N} [g(i) + h(i)]\right) \tag{18}$$

where M is a constant calculated according to:

$$M = T \sum_{i=1}^{N} C_i(P_{\max i}) \tag{19}$$

$g(i)$ and $h(i)$ are discrete functions defining the level of constraint (6) and (7) violation defined as follows:

$$g(i) = \sum_{k=1}^{n_{downi}} \left(\beta_i(k)\left[t_{downi} - t_{offi}(k)\right]\right) \tag{20}$$

$$h(i) = \sum_{k=1}^{n_{upi}} \left(\gamma_i(k)\left[t_{upi} - t_{oni}(k)\right]\right) \tag{21}$$

where:

$$\beta_i(k) = \begin{cases} 1, \text{if} t_{offi}(k) < t_{downi} \\ 0, \text{if} t_{offi}(k) \geqslant t_{downi} \vee \tau_{oni}(k) > T \end{cases} \tag{22}$$

$$\gamma_i(k) = \begin{cases} 1, \text{if} t_{oni}(k) < t_{upi} \\ 0, \text{if} t_{oni}(k) \geqslant t_{upi} \vee \tau_{offi}(k) > T \end{cases} \tag{23}$$

For solutions belonging to group (iii), a penalty function is defined as follows (Dudek, 2011):

$$F'' = W\left(1 + \sum_{t=1}^{T} f(t)\right) \tag{24}$$

where W is a constant calculated according to:

$$W = M\left(1 + \frac{T}{2}\sum_{i=1}^{N} [(t_{downi} - 1) + (t_{upi} - 1)]\right) \tag{25}$$

and $f(t)$ is calculated as follows:

$$f(t) = \begin{cases} \sum_{i \in \Omega(t)} P_{\min i} - D(t), \text{if} \sum_{i \in \Omega(t)} P_{\min i} > D(t) \\ D(t) + R(t) - \sum_{i \in \Omega(t)} P_{\max i} \text{if} D(t) + R(t) > \sum_{i \in \Omega(t)} P_{\max i} \\ 0, \text{otherwise} \end{cases} \tag{26}$$

Such fitness function definitions ensure that individuals violating constraints (4) or (5) are evaluated worse than individuals violating constraints (6) or (7) (because W in (24) is higher than the maximum value of function (18)). This leads to their earlier elimination from the population. The penalty function (18) ensures a worse valuation of individuals violating constraints (6) or (7) from feasible ones (M in (18) is higher than the maximum value of objective function (1)). Both penalty functions (18) and (24) are linearly dependent on the level of violation of constraints.

During the initial phase of the evolution process, when there is no feasible solutions, the level of violation of constraints (4) and (5) is minimized. After individuals meeting these constraints have been found, they are evaluated using function (18). This solutions

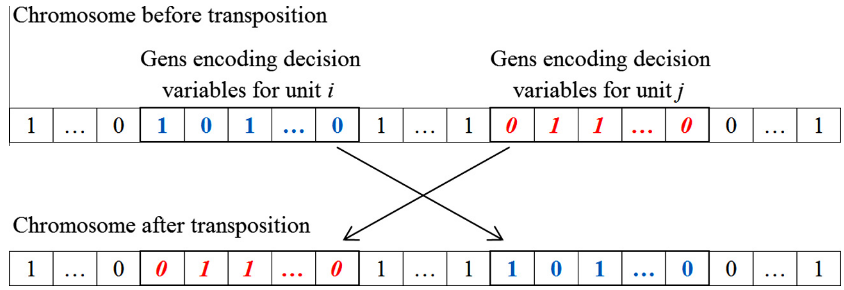

Fig. 3. An illustration of the transposition.

become the majority in the population because their fitness is better than the fitness of the solutions violating constraints (4) and (5). At a certain point in the searching process individuals that are feasible according to (6) and (7) start to appear and become the majority in the population. In this stage GA searches the feasible solutions.

Due to the using of binary tournament (not proportional selection) better fitted individuals do not strongly dominate, which allows the avoidance of a premature convergence of the population into a superindividual.

3.4. Genetic operators

Three types of genetic operators are applied: crossover (one-point, multi-point and uniform), classical binary mutation and transposition.

The crossover and mutation operators used in this application are conventional GA operators (Michalewicz, 1994). Transposition introduced in (Dudek, 2004) is untypical operator dedicated to the UC problem. It operates on one chromosome and generates offspring by exchanging fragments of the chromosome that encode all decision variables of two randomly chosen units. Transposition occurs with probability p_t. This operation can considerably help the evolution process, particularly in the last phase, penetrating the local minimums by changing the work states of pairs of units.

The transposition operator is illustrated in Fig. 3.

4. Application example

The proposed GA was verified on a practical UC problem with 12 units and 24-h scheduling time horizon. The number of all decision variables is $5 \times 12 = 72$. Calculations were performed in Matlab.

The unit and load data can be found in Table 2 and Fig. 1, respectively. The spinning reserve $R(t)$ for all t is equal to 5% of the maximum daily load demand, i.e. 175 MW. It is assumed: $\tau = 7$ in (10) and i: $P_{\min i} = 180$ MW, $P_{\max i} = 350$ MW, $t_{downi} = t_{upi} = 5$ h.

In the first stage of our tests we run the algorithm many times to find the best settings and parameter values. In this study the UC problem was reduced to three units (units 1, 2 and 3). The initial status of the units was assumed to be −24 h and load demand was reduced to one fourth of the load demand assumed for 12 units. This simplification to 15 decision variables made it possible to find an optimal solution using the enumeration method and compare with solutions found by GA.

It is worth noting that the algorithm without transposition rarely found the optimal solution, whilst using this operator the optimal solution was found in each run. One-point, multi-point and uniform crossover gave similar results.

On the basis of these preliminary experiments the following GA parameters were assumed in 12-unit UC problem:

- population size: 100,
- maximum no. of generations: 1000,
- probability of chromosome mutation: 0.5,
- probability of chromosome transposition: 0.25,
- crossover operator: one-point crossover,
- probability of crossover: 0.9.

The optimal solution in Table 3 is shown. The results are presented in Table 4 (marked by GA1), where: $F_{\min}$, $F_{\max}$ and F_{av} are the minimum, maximum and average costs of the best solutions found by the algorithm in 10 runs, $_F$ is the standard deviation of the cost, f_{opt} is the frequency of finding the optimal solution, n_{opt} is the average number of evaluations necessary to find the optimal solution and t_{opt} is the average computational time necessary to find the optimal solution.

In the same table the results of other optimization methods to the same UC problem are presented:

Table 2
Characteristics and initial state of units.

Unit	Initial status[a] (h)	Initial P_i (MW)	a (\$/(MW^2 h))	b (\$/(MW h))	c (\$/h)	e (\$)	f (\$)	g (h^{-1})	h (h^{-1})
1	−24	0	0.004531	7.3968	643.24	−2889.45	5466.28	0.3680	−0.0112
2	−4	0	0.004683	7.5629	666.27	−2893.81	5474.51	0.3680	−0.0112
3	−4	0	0.004708	7.4767	672.77	−2888.84	5465.13	0.3680	−0.0112
4	on	180	0.004880	7.4742	686.58	−2882.77	5453.66	0.3680	−0.0112
5	on	199	0.004214	7.2995	601.53	−2863.94	5418.07	0.3680	−0.0112
6	on	182	0.004582	7.3102	641.99	−2843.13	5378.74	0.3680	−0.0112
7	on	180	0.004267	7.5494	609.07	−2876.16	5441.15	0.3680	−0.0112
8	on	325	0.003572	6.6577	531.63	−2903.29	5492.22	0.3680	−0.0112
9	on	180	0.004788	7.7184	678.40	−2892.73	5472.47	0.3680	−0.0112
10	on	350	0.003485	6.2115	503.60	−2928.65	5540.14	0.3680	−0.0112
11	on	332	0.003658	6.5492	528.19	−2894.88	5476.32	0.3680	−0.0112
12	on	349	0.003671	6.4137	527.81	−2915.53	5515.34	0.3680	−0.0112

[a] "on" indicates unit is in the on-state, "−x" indicates unit is in the off-state for x hours.

Table 3
The optimal solution found by GA.

Hour	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
P_1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
P_2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	281.04	284.45	260.02	255.55	235.03	195.52	180.00	180.00
P_3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	238.06	239.40	227.70	255.51	269.32	269.32	206.60	231.54	288.69	292.07	267.77	263.33	242.92	203.62	180.00	180.00
P_4	180.00	180.00	180.00	180.00	180.00	180.00	180.00	234.78	229.95	231.25	219.96	246.80	260.12	260.12	199.59	223.66	278.81	282.07	258.63	254.34	234.64	196.73	180.00	180.00
P_5	180.00	180.00	180.00	180.00	180.00	180.00	180.00	292.61	287.03	288.53	275.45	306.53	321.97	321.97	251.88	279.74	343.60	347.38	320.23	315.27	292.46	248.55	190.18	180.00
P_6	180.00	180.00	180.00	180.00	180.00	180.00	180.00	267.92	262.78	264.16	252.14	280.72	294.91	294.91	230.46	256.08	314.81	318.28	293.31	288.75	267.78	227.40	180.00	180.00
P_7	180.00	180.00	180.00	180.00	180.00	180.00	180.00	259.69	254.18	255.66	242.75	273.44	288.68	288.68	219.47	246.98	310.05	313.78	286.97	282.06	259.54	216.18	180.00	180.00
P_8	282.01	253.57	254.57	243.50	247.78	263.63	282.69	350.00	350.00	350.00	350.00	350.00	350.00	350.00	350.00	350.00	350.00	350.00	350.00	350.00	350.00	350.00	314.16	251.05
P_9	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	261.97	238.07	233.70	213.63	180.00	0.00	0.00
P_{10}	350.00	323.91	324.95	313.60	317.98	334.22	350.00	350.00	350.00	350.00	350.00	350.00	350.00	350.00	350.00	350.00	350.00	350.00	350.00	350.00	350.00	350.00	350.00	321.34
P_{11}	290.27	262.49	263.47	252.66	256.84	272.32	290.93	350.00	350.00	350.00	350.00	350.00	350.00	350.00	350.00	350.00	350.00	350.00	350.00	350.00	350.00	350.00	321.66	260.03
P_{12}	307.72	280.03	281.01	270.24	274.40	289.83	308.38	350.00	350.00	350.00	350.00	350.00	350.00	350.00	350.00	350.00	350.00	350.00	350.00	350.00	350.00	350.00	339.00	277.58

Table 4
Result comparision.

Algorithm	F_{min} (\$)	F_{max} (\$)	F_{av} (\$)	${}_F$ (\$)	f_{opt}	n_{opt}	t_{opt} (min)
GA1	644,951	645,065	645,013	57	0.4	33,800	47
GA2	644,951	646,229	645,264	381	0.2	70,650	135
GA3	644,951	645,065	645,042	48	0.2	36,550	50
SA	644,951	645,344	645,116	180	0.5	57,637	145
GrA	692,317	716,938	704,549	12,310	0	–	–
MC	No feasible solutions						
HM	649,589	649,589	649,589	0	0	–	–

GA2 is the GA with binary representation of on/off unit status using transposition and the specialized mutation operator in which the probability of mutation is made dependent on the load demand curve, unit production and start-up costs (Dudek, 2004).

- GA3 is the GA with integer representation of unit start-up and shut-down times using transposition, one-point crossover and uniform mutation (Dudek, 2007).
- SA is the simulated annealing method with an adaptive cooling schedule and specialized operators: mutation and transposition (Dudek, 2010).
- GrA is the greedy algorithm where the standard mutation operator is employed to generate new solutions.
- MC is the Monte Carlo method where points in the solution space are randomly chosen from the uniform distribution, remembering the best solution.
- HM is the heuristic method of limit time characteristics (Toroń, 1962), which was used for many years in the Polish Power System.

The number of evaluations of the cost function in these algorithms has been set at 100,000, similar to the proposed GA, and the calculations for every algorithm are done ten times.

The proposed GA found the optimal solution in 4 out of 10 trials. The average value of the best solution costs is the lowest for the proposed algorithm, as well as the number of evaluations necessary to find the optimal solution and the computational time, which is almost three times lower than the computational time for GA and SA with binary representation of on/off unit status. The results for GA1 are quite stable (${}_F = \$57$).

In Fig. 4 the convergence of the proposed algorithm is shown. From this figure it can be seen that the algorithm quickly finds solutions which are feasible according to constraints (4) and (5), and then, after about 40 generations, solutions feasible according to all constraints.

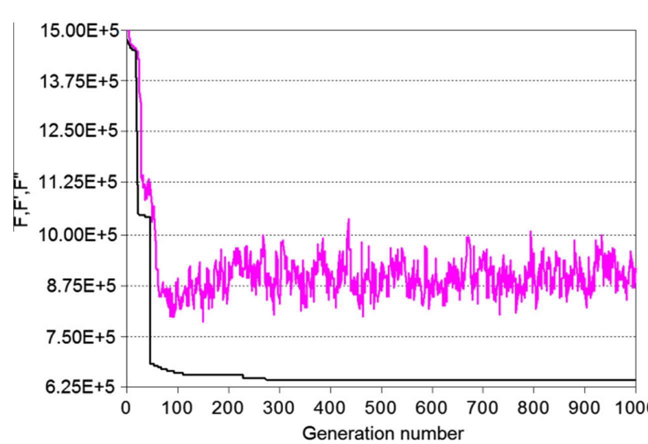

Fig. 4. Cost of the best solution and the average cost of the population in successive generations.

5. Conclusions

The UC problem is a very important one in daily operational planning of power systems. The scheduling optimization of the generating units can bring significant savings in production costs. If instead of the limit time characteristic method used for years in the Polish Power System, the proposed method was applied to the system with 100 units (this corresponds to the Polish Power System), nearly 13 million dollars in cost savings per year can be expected.

In the proposed method we decide at which intervals of the day units can start-up and shut-down, and in this way we reduce the solution space. The size of the solution space for the application example using the classical binary representation is $2^{N \cdot T} \approx 5 \times 10^{86}$, and now in the proposed representation is reduced to $2^{N \cdot B} \approx 4 \times 10^{50}$, so about 10^{36} times. In general the reduction degree is exponentially dependent on the number of units: $(2^{T-B})^N$.

The proposed GA for the UC problem gives a stable and acceptable solution that is near-optimal. The difference between the cost of the best and worst solution found in 10 runs of the algorithm was only 0.018% ($114). The new way of the decision variable definition and their binary encoding improve algorithm efficiency as well as the problem specific operator, transposition searching through local minimums.

Appendix A

Table 5
List of symbols.

Symbol	Description
$\alpha_i(t)$	on/off status of the ith unit at the tth hour, $\alpha_i(t)$ {0,1}
τ_{down1}, τ_{down2}	initial and final hour of the shut-down interval
$\tau_{offi}(k)$, $\tau_{oni}(k)$	shut-down/start-up hour of unit i after the kth on/off state period
τ_{up1}, τ_{up2}	initial and final hour of the start-up interval
Φ_i	set of hours in the ith shut-down or start-up interval
$\Omega(t)$	set of units in on state at time t
a_i, b_i, c_i	production cost function parameters of unit i
B	number of bits encoding the decision variables for one unit
$C_i(P_i(t))$	variable production cost of unit i at time t ($/h)
$D(t)$	load demand at the tth hour (MW)
e_i, f_i, g_i, h_i	start-up cost function parameters of unit i
F, F', F''	cost functions for the feasible solutions, the solutions that violate constraints (6) or (7) but do not violate constraints (4) and (5) and the solutions that violate constraints (4) or (5), respectively
N	total number of units
n_{downi}, n_{upi}	number of periods in which unit i is in continuous off/on state during the optimization period T
$P_i(t)$	power generation of unit i at time t (MW)
$P_{\min i}(t)$, $P_{\max i}(t)$	lower and upper generation limit of unit i, respectively (MW)
$R(t)$	spinning reserve requirement at the tth hour (MW)
$SC_i(t_{offi})$	start-up cost of unit i after t_{offi} hour off state ($)
T	number of hours in the optimization period
$t_{offi}(k)$, $t_{oni}(k)$	down/up time period of unit i during the kth period of off/on state (h)
t_{upi}, t_{downi}	minimum up/down time of unit i (h)
x_l	decision variables

References

Dasgupta, D., & McGregor, D. R. (1994). Thermal unit commitment using genetic algorithms. *IEE Proceedings-Generation Transmission and Distribution, 141*(5), 459–465.

Dudek, G. (2004). Unit commitment by genetic algorithm with specialized search operators. *Electric Power System Research, 72*, 299–308.

Dudek, G. (2007). Genetic algorithm with integer representation of unit start-up and shut-down times for the unit commitment problem. *European Transactions on Electrical Power, 17*, 500–511.

Dudek, G. (2010). Adaptive simulated annealing schedule to the unit commitment problem. *Electric Power Systems Research, 80*(4), 465–472.

Dudek, G. (2011). *New optimization methods for the electric power unit commitment problem*. Czestochowa University of Technology Publishing House (in Polish).

Kazarlis, S. A., Bakirtzis, A. G., & Petridis, V. (1996). A genetic algorithm solution to the unit commitment problem. *IEEE Transaction on Power Systems, 11*(1), 83–92.

Mantawy, A., Abdel-Magid, Y., & Selim, S. (1998). A simulated annealing algorithm for unit commitment. *IEEE Transaction on Power Systems, 13*(1), 197–204.

Mantawy, A., Abdel-Magid, Y., & Selim, S. (1999). Integrating genetic algorithms, tabu search, and simulated annealing for the unit commitment problem. *IEEE Transaction on Power Systems, 14*(3), 829–836.

Michalewicz, Z. (1994). *Genetic algorithms + data structures = evolution programs*. New York: Springer-Verlag.

Nayak, R., & Sharma, J. (2000). A hybrid neural network and simulated annealing approach to the unit commitment problem. *Computers & Electrical Engineering, 26*(6), 461–477.

Padhy, N. (2000). Unit commitment using hybrid models: A comparative study for dynamic programming, expert system, fuzzy system and genetic algorithms. *Electrical Power and Energy Systems, 23*, 827–836.

Padhy, N., Paranjothi, S., & Ramachandran, V. (1997). A hybrid fuzzy neural network – Expert system for a short term unit commitment problem. *Microelectronics and Reliability, 37*(5), 733–737.

Pang, C. K., & Chen, H. C. (1976). Optimal short-term thermal unit commitment. *IEEE Transaction on Power Apparatus and Systems, 95*(4), 1336–1342.

Pang, C. K., Sheblé, G. B., & Albuyeh, F. (1981). Evaluation of dynamic programming based methods and multiple area representation for thermal unit commitments. *IEEE Transaction on Power Apparatus and Systems, 100*(3), 1212–1218.

Sen, S., & Kothari, D. (1998). Optimal thermal generating unit commitment: A review. *Electrical Power and Energy Systems, 20*(7), 443–451.

Sheblé, G. B., & Fahd, G. N. (1994). Unit commitment literature synopsis. *IEEE Transaction on Power Systems, 9*(1), 128–136.

Toroń, M. (1962). *A method of selecting the optimal set of generating units in the power systems*. Ph.D. dissertation. Dept. Elect. Eng., Technical University of Wroclaw, Poland (in Polish).

Wong, M., Chung, T., & Wong, Y. (2000). An evolving neural network approach in unit commitment solution. *Microprocessors and Microsystems, 24*, 251–262.

Wood, A., & Wollenberg, B. (1996). *Power generation, operation, and control*. New York: John Wiley & Sons.