EUROPEAN TRANSACTIONS ON ELECTRICAL POWER
Euro. Trans. Electr. Power 2007; **17**:500–511
Published online 18 December 2006 in Wiley InterScience
(www.interscience.wiley.com) DOI: 10.1002/etep.142

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

Grzegorz Dudek*,†

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

SUMMARY

This paper presents an approach for solving the unit commitment problem based on genetic algorithm with integer representation of the unit start-up and shut-down times. The new definition of the decision variables in the unit commitment problem reduces the solution space and computational time of the genetic algorithm. The method incorporates time-dependent start-up costs, demand and reserve constraints, minimum up and down time constraints, ramp rate limit 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 techniques. Copyright © 2006 John Wiley & Sons, Ltd.

KEY WORDS: unit commitment; power generation dispatch; genetic algorithms; evolutionary computation; combinatorial optimization

1. INTRODUCTION

The objective of unit commitment (UC) is to find the optimal set of generating units and their generation levels within a power system to satisfy the required demand and a large set of operating constraints at any time. The scheduling period is from a day to a week. This is a complex optimization problem with both discrete (unit commitment) and continuous (generation levels) variables. The optimization of this important problem in the daily operation and planning of the power system may save the electric utilities millions of dollars per year in production costs.

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 solution methods being used to solve the UC problem can be grouped as follows [1,2]:

- Heuristic methods such as priority list.

*Correspondence to: Grzegorz Dudek, Institute of Electrical Power Engineering, Czestochowa University of Technology, 42-200 Czestochowa, Al. Armii Krajowej 17, Poland.
†E-mail: dudek@el.pcz.czest.pl

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

Among these methods, the priority list is easy to implement and the simplest of the UC methods. This method specifies the order in which units start-up or shut-down. The quality of the solution is usually far from optimal due to the incomplete search of the solution space.

Many classical methods such as branch-and-bound, dynamic and integer programming suffer from the 'curse of dimensionality' because the problem size and the solution time increase rapidly with the number of generating units to be committed. To reduce the search space, several approaches have been adopted. Most approaches are based on the priority list technique (dynamic programming—sequential combination, dynamic programming—truncated combination [3,4]), thus the solution obtained is suboptimal.

Lagrangian relaxation approach decomposes the UC problem 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. Compared with other approaches, this method has higher computational efficiency, and is more flexible for handling different types of constraint. 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.

In the expert system approach, the knowledge of experienced power system operators and UC experts is combined to create 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) based on a database holding typical load curves and corresponding UC schedules, are trained to recognize the most economical UC schedule associated with the pattern of the current load curve [5,6]. 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 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 [7,8]. A fuzzy decision system has been developed to select the units to be on or off based on these fuzzy variables.

Simulated annealing is a general-purpose stochastic optimization method, especially for combinatorial optimization problems such as UC [9,10], which has been theoretically proved to converge with the optimum 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 algorithms (GA) represent a class of stochastic, adaptive, and parallel search techniques based on the mechanism of natural selection and genetics. They work with a coding of the parameter set, with both discrete and continuous functions. GA search from a population of points and they use probabilistic transition rules. 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 [11–14].

DOI: 10.1002/etep

This paper presents a GA with integer 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 of the GA. The fitness function is constructed as the summation of the objective function and penalty terms for some constraint violations. The combinatorial optimization problem is solved using the GA while the economic dispatch problem is solved via the conventional lambda-iteration method.

2. THE MATHEMATICAL MODEL OF UNIT COMMITMENT

The UC problem can be mathematically formulated as follows:

Objective function:

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

Constraints:

(a) Load balance

$$\forall t : \sum_{i=1}^{N} [\alpha_i(t) P_i(t)] = D(t) \tag{2}$$

(b) Unit power generation limits

$$\forall i, t : \alpha_i(t) P_{\text{down}i}(t) \leq P_i(t) \leq \alpha_i(t) P_{\text{up}i}(t) \tag{3}$$

(c) Set of unit power generation limits

$$\forall t : \sum_{i=1}^{N} [\alpha_i(t) P_{\text{down}i}] \leq D(t) \tag{4}$$

$$\forall t : \sum_{i=1}^{N} [\alpha_i(t) P_{\text{up}i}] \geq D(t) + R(t) \tag{5}$$

(d) Minimum up/down time

$$\forall i : t_{\text{off}i} \geq t_{\text{down}i} \tag{6}$$

$$\forall i : t_{\text{on}i} \geq t_{\text{up}i} \tag{7}$$

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

$$C_i(P_i) = a_i P_i^2 + b_i P_i + c_i \tag{8}$$

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

$$SC_i(t_{\text{off}i}) = e_i \exp(-g_i t_{\text{off}i}) + f_i \exp(-h_i t_{\text{off}i}) \tag{9}$$

DOI: 10.1002/etep

Ramp rate limit constraints are included in Equations (3)–(5). $P_{\mathrm{down}i}(t)$ and $P_{\mathrm{up}i}(t)$ in these inequalities are the lower and upper generation limit of unit i at time t, respectively, defined as follows:

$$P_{\mathrm{down}i}(t) = \begin{cases} \max[P_{\min i}, P_i(t-1) - r_{\mathrm{down}i}] & \text{if } \alpha_i(t-1) = 1 \\ P_{\min i} \quad \text{if } \alpha_i(t-1) = 0 \end{cases} \tag{10}$$

$$P_{\mathrm{up}i}(t) = \begin{cases} \min[P_{\max i}, P_i(t-1) + r_{\mathrm{up}i}] & \text{if } \alpha_i(t-1) = 1 \\ P_{\max i} \quad \text{if } \alpha_i(t-1) = 0 \end{cases} \tag{11}$$

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 ($\tau \in \{1, 2, 3, ...\}$).

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:

$$SC_i(T-t) = \frac{SC_i(T-t+\tau)}{T-t+\tau}(T-t) \tag{12}$$

3. THE PROPOSED GENETIC ALGORITHM APPROACH

The GA implementation consists of random initialization, economic dispatch and cost calculations, reproduction, crossover, mutation, transposition, and elitism.

The tournament selection method is used with tournament sizes of 2. Tournament selection is simple to implement and has none of the disadvantages of the roulette wheel selection method (it does not require scaling of the fitness function and the fitness function values can be negative). An elitism strategy is also used which copies the best parent 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

It is assumed that unit start-up can occur in the time interval $[\tau^j_{\mathrm{up}1}, \tau^j_{\mathrm{up}2}]$ when the load demand curve increases and unit shut-down can occur in the time interval $[\tau^k_{\mathrm{down}1}, \tau^k_{\mathrm{down}2}]$ when the load demand curve decreases (the superscripts j and k denote the start-up or shut-down interval number respectively). The increase and the decrease 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 Figure 1: $[\tau^1_{\mathrm{down}1} = 1, \tau^1_{\mathrm{down}2} = 4]$, $[\tau^1_{\mathrm{up}1} = 5, \tau^1_{\mathrm{up}1} = 13]$, $[\tau^2_{\mathrm{down}1} = 14, \tau^2_{\mathrm{down}2} = 15]$, $[\tau^2_{\mathrm{up}1} = 16, \tau^2_{\mathrm{up}2} = 18]$, $[\tau^3_{\mathrm{down}1} = 19, \tau^3_{\mathrm{down}2} = 24]$. There are two start-up intervals ($j = 1, 2$) and three shut-down intervals ($k = 1, 2, 3$), which are separated by the minimum and maximum values of the load.

DOI: 10.1002/etep

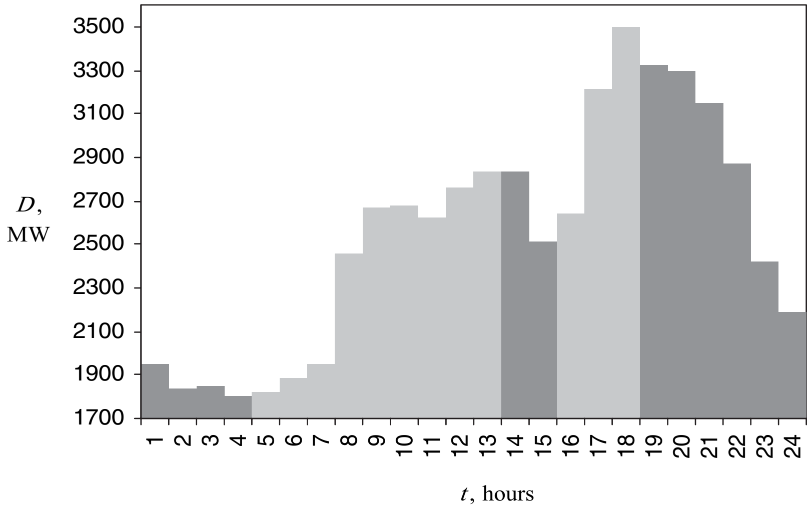

Figure 1. The unit start-up (light bars) and shut-down (dark bars) intervals.

The decision variables are the unit start-up and shut-down hours. In the case of five intervals, there are five integer decision variables for each unit $\boldsymbol{x} = [x_1, x_2, \ldots, x_5]$ defined as follows:

$$x_1 \in \Phi_1 = \{\tau^1_{\text{down1}}, \ldots, \tau^1_{\text{down2}} + 1\} \tag{13}$$

$$x_2 \in \Phi_2 = \{\tau^1_{\text{up1}}, \ldots, \tau^1_{\text{up2}} + 1\} \tag{14}$$

$$x_3 \in \Phi_3 = \{\tau^2_{\text{down1}}, \ldots, \tau^2_{\text{down2}} + 1\} \tag{15}$$

$$x_4 \in \Phi_4 = \{\tau^2_{\text{up1}}, \ldots, \tau^2_{\text{up2}} + 1\} \tag{16}$$

$$x_5 \in \Phi_5 = \{\tau^3_{\text{down1}}, \ldots, \tau^3_{\text{down2}} + 1\} \tag{17}$$

When the decision variable x_1, x_3, or x_5 assumes the value of $\tau^k_{\text{down2}} + 1$, it means that the unit shut-down in the k-th interval does not occur—the unit is in on state during this interval. Similarly when the decision variable x_2 or x_4 assumes the value of $\tau^j_{\text{up2}} + 1$, it means that the unit start-up during the j-th 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, $\boldsymbol{x} = [5, 5, 16, 16, 25]$ means that the unit is in on state in the whole period T; $\boldsymbol{x} = [1, 14, 14, 19, 19]$ means that the unit is in off state in the whole period T; $\boldsymbol{x} = [2, 7, 14, 18, 25]$ means that the unit is in the on state until the 2nd hour, between the 7th and 13th hours and from the 18th hour until the end of period T, while it is in off state between the 2nd and 6th hours and 14th and 17th hours.

GA searches the solution space through the evolution of a population of candidate solutions. Each individual of the population is represented by an integer string—the decision variables for the

DOI: 10.1002/etep

following units: $[x_1^1, x_2^1, \ldots, x_r^1, x_1^2, x_2^2, \ldots, x_r^2, \ldots, x_1^N, x_2^N, \ldots, x_r^N]$. The individual has $r{\cdot}N$ genes. The size of the solution space for this representation is $\left(\prod_{i=1}^{r} |\Phi_i|\right)^N$.

3.2. Economic dispatch and cost calculations. The procedure with infeasible individuals

Since the production cost (8) is a quadratic function (convex and continuous), the economic dispatch problem is solved using a lambda-iteration method [15], based on the principle of equal incremental cost. Lambda-iteration method is used for various generating unit schedules obtained by the GA. Generation levels $P_i(t)$ determined in this procedure are used to calculate unit production costs (8) and the objective function (1). This method guarantees that unit power generation limit constraints (3) are met if the set of unit power generation limit constraints (4) and (5) are met.

For solutions which violate the minimum up/down time constraints (6) or (7), but do not violate a set of unit power generation limit constraints, (4) and (5), a penalty function is created [14,16]:

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

where M is a constant calculated as follows:

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

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

$$g(i) = \sum_{k=1}^{n_{\mathrm{down}i}} \{\beta_i(k)[t_{\mathrm{down}i} - t_{\mathrm{off}i}(k)]\} \tag{20}$$

$$h(i) = \sum_{k=1}^{n_{\mathrm{up}i}} \{\gamma_i(k)[t_{\mathrm{up}i} - t_{\mathrm{on}i}(k)]\} \tag{21}$$

where:

$$\beta_i(k) = \begin{cases} 1 & \text{if } t_{\mathrm{off}i}(k) < t_{\mathrm{down}i} \\ 0 & \text{if } t_{\mathrm{off}i}(k) \geq t_{\mathrm{down}i} \vee \tau_{\mathrm{on}i}(k) > T \end{cases} \tag{22}$$

$$\gamma_i(k) = \begin{cases} 1 & \text{if } t_{\mathrm{on}i}(k) < t_{\mathrm{up}i} \\ 0 & \text{if } t_{\mathrm{on}i}(k) \geq t_{\mathrm{up}i} \vee \tau_{\mathrm{off}i}(k) > T \end{cases} \tag{23}$$

If solutions violate the set of unit power generation limit constraints (4) or (5), a penalty function is formulated as follows [16]:

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

DOI: 10.1002/etep

where W is a constant calculated as follows:

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

$f(t)$ is calculated as follows:

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

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 ensures their earlier elimination from the population. The substitute cost function (18) ensures a worse valuation of individuals violating constraints (6) or (7) from feasible individuals (M in (18) is higher than the maximum value of objective function (1)). Both substitute cost functions (18) and (24) are linearly dependent on the level of violation of constraints.

At the starting phase of the evolution process the level of violation of constraints (4) and (5) is minimized. After individuals meeting, these constraints have been found, they are evaluated by means of the function (18). At a certain point in the process, individuals that are feasible according to these constraints start to appear and become the majority in the population. Because binary tournament (not proportional selection) is used, these feasible individuals do not strongly dominate, which allows the avoidance of a premature convergence of the population into a superindividual.

3.3. Genetic operators

Three types of genetic operators are proposed: crossover, mutation, and transposition.

One-point, multi-point, uniform, arithmetical, and heuristic integer crossover [17] were tested as crossover operators.

Two types of mutation method were tested: uniform and non-uniform integer mutation. The uniform integer mutation is analogous to the binary mutation—the randomly selected genes are replaced by the random numbers in the proper ranges. The non-uniform mutation is designed for fine-tuning capabilities. The gene selected at random for mutation changes its value as follows:

$$x' = \begin{cases} x + \Delta(l, x^U - x) & \text{if } r_b = 0 \\ x - \Delta(l, x - x^L) & \text{if } r_b = 1 \end{cases} \tag{27}$$

The function $\Delta(l, y)$ returns an integer value in the range $[1, y]$ such that the value $\Delta(l, y)$ approaches 1 as the generation number l increases. This property causes the operator to search the space uniformly initially and locally at later stages. The function $\Delta(l, y)$ is given as follows:

$$\Delta(l, y) = \text{round}\{(y - 0,002)[1 - r_r^{(1-l/L)^2}] + 0,501\} \tag{28}$$

A transposition operator is introduced in binary version in [14]. 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 transposition can

DOI: 10.1002/etep

considerably help the evolution process, particularly in the last phase, penetrating the local minimums by changing the work states of pairs of units.

4. APPLICATION EXAMPLES

The GA described above for the UC problem was implemented in Matlab and has been applied to a practical power system with 12 units. The scheduling time horizon T is 24 hours. These experiments were done on a personal computer with a Pentium III 800 MHz processor.

The unit and load data can be found in Tables I and II respectively. The spinning reserve $R(t)$ for all t is equal to 5% of the maximum daily load demand, that is, 175 MW. It is assumed $\tau = 7$ in Equation (12). In order to compare results obtained from the proposed algorithm and from genetic algorithm with standard binary representation of the unit states [14], the ramp rate limit constraints are not taken into account ($(\forall t : P_{\mathrm{down}i}(t) = P_{\min i}, P_{\mathrm{up}}(t) = P_{\max i}$ are assumed).

On the basis of preliminary experiments, the following GA parameters were assumed:

- Population size: 100,
- Maximum no. of generations: 1000,
- Probability of individual mutations: 0.5,
- Probability of individual transpositions: 0.25,
- Probability of crossover: 0.9.

The following results were obtained:

- The minimum, maximum and average costs of the best solutions found by the algorithm in 10 runs were $64 4951, $645 065, and $645 042 respectively.
- The standard deviation of the costs of the best solutions found by the algorithm in 10 runs was $48,
- The frequency of the best solution found by GA: 0.2,
- The average number of evaluations necessary to find the best solution: 36 550,
- The average computational time necessary to find the best solution: 50 min.

One-point, multi-point, and uniform crossover gave similar results, which were better than arithmetical and heuristic crossover. The similar results were observed when uniform and non-uniform mutation was applied. As a result, one-point crossover and uniform mutation is recommended as the simplest and quickest operators. Transposition plays an important role. GA without this operator works considerably worse in unit commitment application.

The same example was solved by GA with binary representation of on/off unit status and a specialized mutation operator in which the probability of mutation is made dependent on the load demand curve, unit production, and start-up costs [14]. The best solution found by the algorithm proposed in this article is the same as that found by GA with binary representation (shown in Table III in Reference [14]), but the computational time is 2.7 times shorter without having to use such a complicated mutation method. This solution is better than those reported in Reference [14] obtained from the simple simulated annealing ($702 379), Monte Carlo method (no acceptable solution) and the heuristic method, which was used for many years in the Polish Electrical Power System ($665 634).

The second example is similar to the first one but ramp rate limit constraints are taken into account. The GA parameters were the same as in the first example. The minimum, maximum, and average costs of the best solutions found by the algorithm in 10 runs were $659 498, $664 032, and $660 094, respectively. The frequency of the best solution was 0.4 and the average number of evaluations

DOI: 10.1002/etep

Table I. Characteristics and initial state of units.

Unit	Initial status[a] (hours)	Initial P_i (MW)	a \$/(MW^2 ·hour)	b \$/(MW ·hour)	c (\$/hour)	e (\$)	f (\$)	g (hour^{-1})	h (hour^{-1})	$P_{\min i}$ (MW)	$P_{\max i}$ (MW)	$t_{\text{down}i}$ (hour)	$t_{\text{up}i}$ (hour)	$r_{\text{down}i}$ (MW/ hour)	$r_{\text{up}i}$ (MW/ hour)
1	−24	0	0.004531	7.3968	643.24	−2889.45	5466.28	0.3680	−0.0112	180	350	5	5	75	60
2	−4	0	0.004683	7.5629	666.27	−2893.81	5474.51	0.3680	−0.0112	180	350	5	5	75	60
3	−4	0	0.004708	7.4767	672.77	−2888.84	5465.13	0.3680	−0.0112	180	350	5	5	75	60
4	On	180	0.004880	7.4742	686.58	−2882.77	5453.66	0.3680	−0.0112	180	350	5	5	75	60
5	On	199	0.004214	7.2995	601.53	−2863.94	5418.07	0.3680	−0.0112	180	350	5	5	75	60
6	On	182	0.004582	7.3102	641.99	−2843.13	5378.74	0.3680	−0.0112	180	350	5	5	75	60
7	On	180	0.004267	7.5494	609.07	−2876.16	5441.15	0.3680	−0.0112	180	350	5	5	75	60
8	On	325	0.003572	6.6577	531.63	−2903.29	5492.22	0.3680	−0.0112	180	350	5	5	75	60
9	On	180	0.004788	7.7184	678.40	−2892.73	5472.47	0.3680	−0.0112	180	350	5	5	75	60
10	On	350	0.003485	6.2115	503.60	−2928.65	5540.14	0.3680	−0.0112	180	350	5	5	75	60
11	On	332	0.003658	6.5492	528.19	−2894.88	5476.32	0.3680	−0.0112	180	350	5	5	75	60
12	On	349	0.003671	6.4137	527.81	−2915.53	5515.34	0.3680	−0.0112	180	350	5	5	75	60

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

DOI: 10.1002/etep

Table II. Load demand D (MW).

Hour	1	2	3	4	5	6	7	8	9	10	11	12
D	1950	1840	1844	1800	1817	1880	1952	2455	2672	2679	2618	2763
Hour	13	14	15	16	17	18	19	20	21	22	23	24
D	2835	2835	2508	2638	3217	3500	3325	3293	3146	2868	2415	2190

Table III. The best power sharing (MW) of example 2.

Hour	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_{10}	P_{11}	P_{12}
1	0,00	0,00	0,00	180,00	180,00	180,00	180,00	282,01	0,00	350,00	290,27	307,72
2	0,00	0,00	0,00	180,00	180,00	180,00	180,00	253,57	0,00	323,91	262,49	280,03
3	0,00	0,00	0,00	180,00	180,00	180,00	180,00	254,57	0,00	324,94	263,47	281,01
4	0,00	0,00	0,00	180,00	180,00	180,00	180,00	243,51	0,00	313,60	252,66	270,24
5	0,00	0,00	0,00	180,00	180,00	180,00	180,00	247,78	0,00	317,98	256,84	274,40
6	0,00	0,00	0,00	180,00	180,00	180,00	180,00	263,63	0,00	334,22	272,32	289,83
7	0,00	0,00	0,00	180,00	180,00	180,00	180,00	282,69	0,00	350,00	290,93	308,38
8	211,12	0,00	0,00	188,10	238,56	218,21	206,31	342,69	0,00	350,00	350,00	350,00
9	213,61	0,00	197,08	190,42	241,24	220,68	208,96	350,00	0,00	350,00	350,00	350,00
10	214,78	0,00	198,20	191,50	242,49	221,83	210,20	350,00	0,00	350,00	350,00	350,00
11	204,64	0,00	188,45	182,09	231,59	211,80	199,43	350,00	0,00	350,00	350,00	350,00
12	228,74	0,00	211,64	204,46	257,50	235,63	225,02	350,00	0,00	350,00	350,00	350,00
13	240,70	0,00	223,15	215,57	270,37	247,47	237,73	350,00	0,00	350,00	350,00	350,00
14	240,70	0,00	223,15	215,57	270,37	247,47	237,73	350,00	0,00	350,00	350,00	350,00
15	184,46	0,00	180,00	180,00	209,90	191,85	180,00	337,41	0,00	350,00	344,38	350,00
16	207,96	0,00	191,64	185,17	235,17	215,09	202,96	350,00	0,00	350,00	350,00	350,00
17	237,25	211,82	219,83	212,37	266,66	244,06	234,07	350,00	190,93	350,00	350,00	350,00
18	272,94	246,35	254,17	245,50	305,03	279,34	271,96	350,00	224,70	350,00	350,00	350,00
19	283,00	256,09	263,86	0,00	315,86	289,30	282,65	350,00	234,23	350,00	350,00	350,00
20	278,43	251,67	259,47	0,00	310,95	284,78	277,80	350,00	229,90	350,00	350,00	350,00
21	257,44	231,36	239,26	0,00	288,37	264,02	255,51	350,00	210,04	350,00	350,00	350,00
22	246,22	220,50	228,46	0,00	276,31	252,92	243,59	350,00	0,00	350,00	350,00	350,00
23	180,00	180,00	180,00	0,00	201,31	180,00	180,00	310,38	0,00	350,00	317,98	335,33
24	180,00	180,00	180,00	0,00	180,00	180,00	180,00	251,05	0,00	321,33	260,03	277,58

necessary to find the best solution was 83 500. The best solution found by the algorithm is shown in Table III.

5. CONCLUSIONS

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 in the first example was 0.018% ($114) and in the second example was 0.69% ($4534). In this method, we decide

DOI: 10.1002/etep

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 examples in the binary representation is $2^{N \cdot T} \approx 5 \cdot 10^{86}$, and now in the proposed integer representation is reduced to $3 \cdot 10^{43}$, so about 10^{43} times. In general, the reduction degree is exponentially dependent on the number of units: $\left(2^T \Big/ \prod_{i=1}^{r} |\Phi_i|\right)^N$.

The effectiveness of the algorithm was achieved by defining start-up and shut-down times as the decision variables and the introduction of an operator specific to the problem, that is, transposition searching through local minimums.

The calculation time can be reduced by implementing the algorithm in a programming environment that is faster than Matlab, and doing the calculations in a parallel machine environment.

6. LIST OF SYMBOLS AND ABBREVIATIONS

GA	genetic algorithm
UC	unit commitment
$\lvert\cdot\rvert$	power of a set
$\alpha_i(t)$	on/off status of the i-th unit at the t-th hour, $\alpha_i(t) \in \{0, 1\}$
$\tau_{\text{down1}}, \tau_{\text{down2}}$	initial and final hour of the shut-down interval
$\tau_{\text{off}i}(k), \tau_{\text{on}i}(k)$	shut-down/start-up hour of unit i after the k-th on/off state period
$\tau_{\text{up1}}, \tau_{\text{up2}}$	initial and final hour of the start-up interval
Φ_i	set of hours in the i-th 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
$C_i[P_i(t)]$	variable production cost of unit i at time t (\$/hour)
$D(t)$	load demand at the t-th hour (MW)
e_i, f_i, g_i, h_i	start-up cost function parameters of unit i
L	maximal generation number
l	generation number
N	total number of units
$n_{\text{down}i}, n_{\text{up}i}$	number of periods in which unit i is in continuous off/on state during the optimization period T
$P_{\text{down}i}(t), P_{\text{up}i}(t)$	rated lower/upper generation limit of unit i at time t (MW)
$P_i(t)$	power generation of unit i at time t (MW)
$P_{\text{min}i}(t), P_{\text{max}i}(t)$	lower/upper generation limit of unit i (MW)
$R(t)$	spinning reserve requirement at the t-th hour (MW)
r	number of start-up and shut-down intervals
r_b	binary random number
$r_{\text{down}i}, r_{\text{up}i}$	ramp-down/ramp-up rate limit for the i-th unit (MW/hour)
r_r	uniform real random number in [0,1]
round(.)	round to the nearest integer
$SC_i(t_{\text{off}i})$	start-up cost of unit i after $t_{\text{off}i}$ hour off state (\$)
T	number of hours in the study period
$t_{\text{off}i}, t_{\text{on}i}$	time period during which unit i is continuously off/on (hour)

DOI: 10.1002/etep

$t_{\text{off}i}(k)$, $t_{\text{on}i}(k)$	down/up time period of unit i during the k-th period of off/on state (hour)
$t_{\text{up}i}$, $t_{\text{down}i}$	minimum up/down time of unit i (hour)
x, x^L, x^U	decision variable—the unit start-up or shut-down hour, its lower and upper value respectively.

ACKNOWLEDGEMENTS

This study was supported by the Polish State Committee for Scientific Research under Grant 8T10B03921.

REFERENCES

1. Sheblé GB, Fahd GN. Unit commitment literature synopsis. *IEEE Transaction on Power Systems* 1994; **9**(1):128–136.
2. Sen S, Kothari D. Optimal thermal generating unit commitment: a review. *Electrical Power and Energy Systems* 1998; **20**(7):443–451.
3. Pang CK, Chen HC. Optimal short-term thermal unit commitment. *IEEE Transaction on Power Apparatus and Systems* 1976; **95**(4):1336–1342.
4. Pang CK, Sheblé GB, Albuyeh F. Evaluation of dynamic programming based methods and multiple area representation for thermal unit commitments. *IEEE Transaction on Power Apparatus and Systems* 1981; **100**(3):1212–1218.
5. Nayak R, Sharma J. A hybrid neural network and simulated annealing approach to the unit commitment problem. *Computers and Electrical Engineering* 2000; **26**(6):461–477.
6. Wong M, Chung T, Wong Y. An evolving neural network approach in unit commitment solution. *Microprocessors and Microsystems* 2000; **24**:251–262.
7. Padhy N, Paranjothi S, Ramachandran V. A hybrid fuzzy neural network—expert system for a short term unit commitment problem. *Microelectronics Reliability* 1997; **37**(5):733–737.
8. Padhy N. Unit commitment using hybrid models: a comparative study for dynamic programming, expert system, fuzzy system and genetic algorithms. *Electrical Power and Energy Systems* 2000; **23**:827–836.
9. Mantawy A, Abdel-Magid Y, Selim S. A simulated annealing algorithm for unit commitment. *IEEE Transaction on Power Systems* 1998; **13**(1):197–204.
10. Wong S. An enhanced simulated annealing approach to unit commitment. *Electrical Power and Energy Systems* 1998; **20**(5):359–368.
11. Dasgupta D, McGregor DR. Thermal unit commitment using genetic algorithms. *IEEE Proceedings—Generalized Transmission and Distribution* 1994; **141**(5):459–465.
12. Kazarlis SA, Bakirtzis AG, Petridis V. A genetic algorithm solution to the unit commitment problem. *IEEE Transaction on Power Systems* 1996; **11**(1):83–92.
13. Mantawy A, Abdel-Magid Y, Selim S. Integrating genetic algorithms, tabu search, and simulated annealing for the unit commitment problem. *IEEE Transaction on Power Systems* 1999; **14**(3):829–836.
14. Dudek G. Unit commitment by genetic algorithm with specialized search operators. *Electric Power System Research* 2004; **72**:299–308.
15. Wood A, Wollenberg B. *Power Generation, Operation, and Control*. John Wiley & Sons: New York, 1996.
16. Dudek G. Economic dispatch and unit commitment using evolutionary algorithms. Ph.D. Dissertation (in Polish), Dept. Elect. Eng., Czestochowa University of Technology, Poland, 2003.
17. Michalewicz Z. *Genetic Algorithms + Data Structures = Evolution Programs*. Springer-Verlag: New York, 1994.

AUTHOR'S BIOGRAPHY

Grzegorz Dudek (born in 1969) received his M.Sc. degree in 1994 and Ph.D. degree in 2003 in Electrical Engineering from the Czestochowa University of Technology, Poland. He is currently a Lecturer at the Department of Electrical Engineering, Czestochowa University of Technology. His research interests are in using artificial intelligence and pattern recognition in the areas of generation planning and scheduling, unit commitment, and load forecasting.