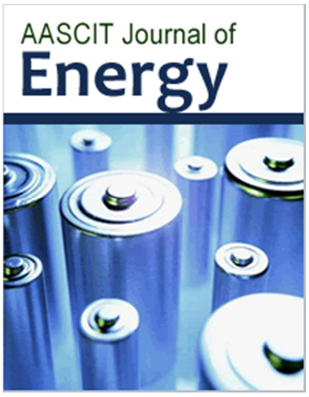
AASCIT Journal of Energy
2015; 2(4): 44-56
Published online July 10, 2015 (http://www.aascit.org/journal/energy)

American Association for Science and Technology



Application of Artificial Intelligence to Minimize Operating Costs of Smart Grid Energy Sources

Miloš Křivan

External Lecturer of Faculty of Informatics, Statistics of University of Economics, Prague, Czech Republic

Email address
krivanm@vse.cz

Citation
Miloš Křivan. Application of Artificial Intelligence to Minimize Operating Costs of Smart Grid Energy Sources. *AASCIT Journal of Energy.* Vol. 2, No. 4, 2015, pp. 44-56.

Keywords
Artificial Neural Network,
Self-Organizing Map,
Simulated Annealing,
Cluster Analysis,
Type Daily Diagrams,
Unit Commitment,
Renew-able Energy,
Smart Grid

Received: April 12, 2015
Revised: May 21, 2015
Accepted: May 22, 2015

Abstract
This paper formulates a unit commitment optimization problem for renewable and combined energy sources distributed in a smart grid. Also we present two experiments. The first experiment consists of cluster analysis of the daily diagrams of electric energy-consumption of smart grid (namely for a work day and a day off on the basis of its annual history) by self-organizing map neural network. The resulting type daily diagrams are used as a basis for the second experiment. The second experiment consists of a solution of the mentioned optimization problem by simulated annealing.

1. Introduction

With the development of computing technology and the growth of its computational power, there has been an increasing focus on artificial intelligence methods. These methods include terms such as artificial neural networks or evolutionary algorithms. However, their massive utilization in practical applications across all human activities only occurred in the eighties of the previous century, due to the development of personal computers.

Artificial neural networks are used to process and evaluate incomplete, indeterminate or inconsistent information, especially for tasks involving recognition, diagnostics, classification of objects with respect to provided categories or prediction of the time development of the given variable, compression and coding information, noise filtering, extrapolation or interpolation of the trends of a given variable and last but not least the cluster analysis of multidimensional data, as described in this article.

Evolutionary algorithms are used to find a solution with sufficient quality for large-scale general optimization tasks in a sufficiently short time. Evolutionary algorithms inspired by nature include a whole spectrum of optimization heuristic techniques, e.g. Particle Swarm resp. Ant Colony Optimization, Genetic Algorithms or Simulated Annealing. Heuristics may be described as a procedure for searching the solution space via shortcuts, which are not guaranteed to find the correct solution but do not suffer from a range of problems of conventional optimization methods such as e.g. the requirement of connectivity or differentiability of the criterion or link function, the problem of respecting constraints, being stuck in a shallow local minimum or divergence. However, their application requires the configuration of certain free parameters, which need to be setup based on the specific optimization task – these may e.g. include the starting or final temperature and the number of iterations of the simulated annealing algorithm described below and based on the evolution of thermodynamic systems. In physics, annealing is a process where an object, heated up to a certain high temperature, is being gradually cooled down to remove internal defects in the object. The high temperature causes the particles in the object to rearrange randomly, which destroys defects in the crystal lattice, and the gradual cooling then allows the particles to stabilize in equilibrium points with a lower probability of the creation of new defects.

In the context of sustainable development of human society, which depends on the planet's energy sources, covering the needs of the society requires a focus on renewable energy sources such as geothermal energy, atmospheric currents, hydro-geological cycles, solar radiation or biomass, due to their relative inexhaustibility and the minimization of the impacts of human activities on the environment related to their conversion to energy.

The implementation rate of the above proposition depends on the cost of the chain of production, transmission and consumption of energy. In relation to electricity, we are therefore minimizing production costs through the optimal operation of the electrical transmission network, i.e. a suitable choice for the connection and size of the injection active or reactive power in the network nodes, and finally by minimizing the consumption side.

Minimizing the above costs, due to the complexity of the problem as a whole, is usually realized more or less separately on three separate planes (generation, transmission and consumption), i.e. instead of one optimum we only obtain three sub-optima. In connection with the second or third plane, we speak about smart grids; however the subject of this paper is the solution of this problem on the first level, i.e. unit commitment optimization problem.

2. Unit Commitment

The task of unit commitment [11-22] is an optimization problem with a goal of minimizing the total costs of producing the volume of energy given by the prediction of its consumption for the considered period, sampled e.g. by hours. In other words, this constitutes a plan for the sorting of sources and their generated outputs covering the predicted consumption in each hour of the given period.

The optimization problem may in general formally be expressed as follows:

$$f: \mathbb{R}^n \to \mathbb{R} \quad f(\vec{x}_0) = \min_{\vec{x} \in \Omega} f(\vec{x}) \quad \Omega \subset \mathbb{R}^n \tag{1}$$

where $\vec{x}_0$ is the optimum, whereas Ω specifies the area of admissible solutions containing the optimum as given by operating-technical parameters of sources, and whereas f represents the cost function given by a sum of operating and start-up costs (Fig. 1) for sources integrated in the given period:

$$f\left(\vec{P}(t), \vec{x}(t)\right) = \sum_t \sum_i (A_i + B_i P_i(t) + C_i P_i^2(t) + D_i(1 - e^{-\frac{\Delta T_i(t)}{\tau_i}}))\, x_i(t) \tag{2}$$

where $i \in \{1, -, N\}$, $t \in \{1, -, T\}$ and $P_i(t)$ resp. $x_i(t)$ are the output resp. state of the i-th source in time t, and where A_i, B_i, C_i, D_i resp. $\Delta T_i(t)$ and τ_i are the appropriate cost coefficients resp. downtime and the time constant of the exponential growth of start-up costs for the i-th source in time t, and furthermore N resp. T is the number of sources in the network resp. the number of time snaps of the considered period.

Admissible solutions are in general specified by the following inequalities resp. equality:

$$P_i^{min} \le P_i \le P_i^{max} \tag{3}$$

$$\sum_i P_i(t) x_i(t) = C(t) \tag{4}$$

where $C(t)$ represents a prediction of the consumption in the appropriate hour of the considered period.

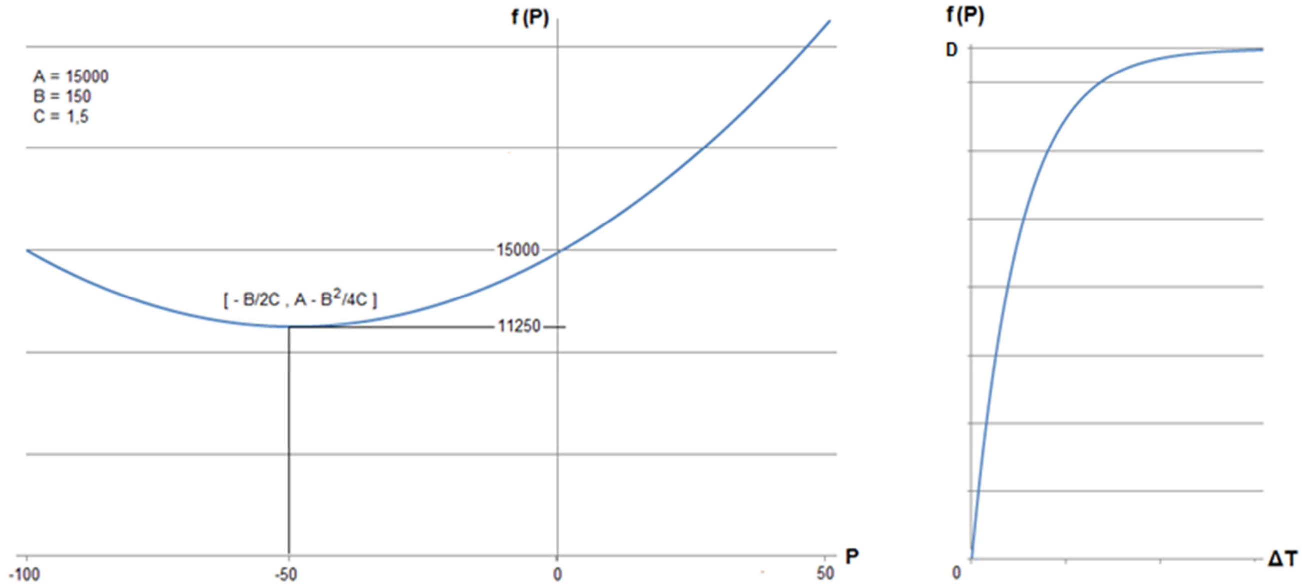


Fig 1. Dependence of Operating Costs on Power and Start-up Costs on Downtime.

3. Competitive Model of Neural Network

We define an artificial neural network as the oriented graph with vertices and edges dynamically evaluated, i.e. as the ordered quintuplet$[V, E, \varepsilon, y, w]$:
V set of vertices (neurons)
E set of edges (synapses)
ε mapping edges with incidence vertices $(\varepsilon: E \rightarrow V \times V)$
y dynamic evaluation of vertices $(y: V \times t \rightarrow \mathbb{R})$
w dynamic evaluation of edges $(w: \varepsilon(E) \times T \rightarrow \mathbb{R})$.

The vector $\vec{y}(t) = [y_i(t)|i \in V]$ is called the network state in time t and the vector $\vec{w}(T) = [w_{ij}(T)|[i,j] \in V \times V]$ is called the network configuration in time T ($\forall[i,j] \notin \varepsilon(E) \Rightarrow w_{ij}(T) = 0$). The state resp. configuration of the network as a vector function of time t resp. T we term as the active dynamics resp. adaptive dynamics of the neural network.

Using time separation of the active and adaptive dynamics we expressed the fact that a neural network works in two time-independent modes, in an active and adaptive mode.

The active resp. adaptive dynamics of a neural network in continuous time can be defined as a solutions vector of the following systems of differential equations [1-2]:

$$\frac{d}{dt}x_j(t) + x_j(t) = \sum_i f_i(x_i(t - \Delta t))w_{ij} - \vartheta_j \quad (5)$$

resp.

$$\frac{d}{dT}w_{ij}(T) + \beta g_j(x_j(T))w_{ij}(T) = \alpha f_i(x_i(T))g_j(x_j(T)) \quad (6)$$

$i, j \in V, \alpha, \beta \in <0,1>$, and then analogously to biological neural network we call:
x_i potential of the i-th neuron
f_i activation function of the i-th neuron $(f_i(x_i) = y_i)$
g_j adaptation function of the j-th neuron
ϑ_j threshold of the j-th neuron
w_{ij} synaptic weight links of the i-th neuron to the j-th neuron
α measure of plasticity of synapses
β measure of elasticity of synapses
Δt signal delay time
and where the activation function of the neuron we approximate by sigmoid function: $f(x) = 1/(1 + e^{-px})$. The parameter $p > 0$ expresses the slope of the sigmoid. For a slope approaching zero resp. infinity we get the activation function in the shape of linearity resp. non-linearity:

$$\lim_{p\to 0} f(x) = \frac{1}{2} \quad \lim_{p\to\infty} f(x) = 0 \quad x < 0 \quad \lim_{p\to\infty} f(x) = 1 \quad x > 0$$

If we replace in (5) and (6) the derivations by analogous expressions for discrete time:

$$\frac{d}{dt}x_j(t) \equiv \frac{x_j(t+1) - x_j(t)}{t+1-t} \qquad \frac{d}{dt}w_{ij}(T) \equiv \frac{w_{ij}(T+1) - w_{ij}(T)}{T+1-T}$$

then we obtain for $\Delta t = 0$ the following systems of difference equations and the vectors of its solutions define the active and adaptive dynamics of a neural network in discrete time:

$$x_j(t+1) = \sum_i f_i(x_i(t))w_{ij} - \vartheta_j \quad \text{resp.} \quad y_j(t+1) = f_j(\textstyle\sum_i y_i(t)w_{ij} - \vartheta_j) \quad (7)$$

$$w_{ij}(T+1) = (1 - \beta g_j(x_j(T)))w_{ij}(T) + \alpha f_i(x_i(T))g_j(x_j(T)) \quad (8)$$

$i, j \in V$.

Let us divide the population of the neurons in V to two disjoint populations V_1 and V_2 ($V_1 \cup V_2 = V$, $V_1 \cap V_2 = \emptyset$, $|V_1| = n$, $|V_2| = m$), and let us connect them by edges so that there is an edge from each neuron in V_1 to each neuron in V_2 ($\varepsilon(E_1) = V_1 \times V_2$), i.e. the network is oriented from V_1 to V_2 and V_1 resp. V_2 is then understood as the input resp. output population. Let us furthermore connect neurons in V_2 by edges so that there is an edge from each neuron in V_2 to every other neuron in V_2 ($\varepsilon(E_2) = V_2 \times V_2 - \{[j,j]|j \in V_2\}$). The vector function which sets the network state for an input stimulus $x_i(0)$ we define as the network function:

$$\vec{F}(\vec{x}(0)) = \vec{y}(\infty) \qquad \vec{x}(0) = [x_i(0)|i \in V_1] \qquad \vec{y}(\infty) = [y_j(\infty)|j \in V_2]$$

Let us choose the activation function of neurons of population V1as an identity, i.e. modified linearity and the activation function of neurons of population V_2 as non-linearity. Then during the active dynamics for constantly applied stimulus $\vec{x}(0)$ attached to population V_1 we can express the active dynamics (7) for $y_k(0) = 0$ as follows:

$$y_j(t+1) = f_j(\textstyle\sum_k y_k(t)\, w_{kj} - \vartheta_j), \; -\vartheta_j = \sum_i x_i(0)w_{ij} \quad (9)$$

$i \in V_1$, $j, k \in V_2$ and let us call the parameter $-\vartheta_j$ the potential gain of the j-th neuron.

Let us choose the following initial conditions for the network configuration $w_{kj}(0) = -2$, $w_{ij}(0) = r_{ij}$ and let us add templates from the training set specified in the form $\{\vec{a}(T)|T \in \Delta T\}$ for the population V_1, where$\Delta T$is the network adaptation period. If we only let the mutual links between neurons in V_1 and V_2 adapt, and if we select the adaptation function for the neurons in V_2 to match the activation functions, then, assuming elasticity is equal to plasticity ($\alpha = \beta$), we can express the adaptive dynamics (8) as follows:

$$w_{kj}(T) = w_{kj}(T-1)$$

$$w_{ij}(T) = w_{ij}(T-1) + \alpha y_j(T)(a_i(T) - w_{ij}(T-1)) \quad (10)$$

$i \in V_1$, $j, k \in V_2$, $T \in \Delta T = \{1, -, N\}$ where N resp. r_{ij} is the number of patterns of training set resp. the value specified of the random number generator.

In each step of the adaptive dynamics (10) it is required to designate the states of neurons in V_2, i.e. the steps of the adaptive dynamics are conditioned by the active dynamics, which, from the perspective of adaptive dynamics, runs infinitely fast. Thus the state of V_2 is determined synchronously with the state of V_1.

Let us assign to each neuron in V_2 a weight vector $\vec{w}_j = [w_{ij} | i \in V_1]$. Then the neurons in V_2 together with the edges E_2 and the active dynamics (9) form a Hopfield optimization network [3-4] with the following energy function:

$$E(\vec{y}) = \sum_j \sum_k y_k y_j + \sum_j y_j \vartheta_j \,, -\vartheta_j = \sum_i a_i(T) w_{ij} = \vec{a}(T) \cdot \vec{w}_j \quad (11)$$

$i \in V_1,\ j \in V_2,\ k \in V_2 - \{j\}$.

If the vectors of the training set resp. the weight vectors are normal, then the received potential gain of each neuron will comply with $-\vartheta_j = \cos\varphi\,(\vec{a}(T), \vec{w}_j)$ and the distance between the specified vectors can be defined as the angle $\varphi \in <0, \pi>$ between them. The energy function specified above will then reach its minimum if and only if only one neuron in V_2 is excited, specifically the neuron with the maximum potential gain (11) – the so-called gain neuron.

The process of energy minimization of the state of V_2 realized by the active dynamics (9), when the excited neuron with the maximum potential gain inhibits (by negative links) other neurons, is called lateral inhibition. Lateral inhibition, which designates a corresponding state of the population of V_2 based on the presented training template, replaces the missing template association in the training set – in other words, it replaces the statement of a teacher, and we thus speak of teacher-less learning.

Lateral inhibition in each adaptation step will ensure the adaptation of only the weight vector corresponding to the k-th gain neuron, i.e. of the weight vector as per the above-specified non-Euclidian metric of the closest presented training set template, to which it will advance on the surface of an n-dimensional ball of unit radius by an adaptation step proportional to the plasticity of the synapse:

$$\vec{w}_k(T) = \vec{w}_k(T-1) + \alpha(\vec{a}(T) - \vec{w}_k(T-1)) \quad (12)$$

and the gain neuron thus won the *competition* for the presented template of the training set. The normality of the adapted weight vector will be ensured by its subsequent normalization.

The objective function (13) will reach its minimum if and only if the weight vector is on the position with a minimal sum of distances from all vectors of the training set which excite the appropriate neuron, i.e. in the center of the cluster of the specified vectors:

$$G(\vec{w}_j) = ½ \sum_T y_j(T) \sum_i (a_i(T) - w_{ij})^2 \qquad -\frac{\partial G(\vec{w}_j)}{\partial w_{ij}} = \sum_T y_j(T)(a_i(T) - w_{ij}) \quad (13)$$

$i \in V_1,\ j \in V_2,\ T \in \Delta T$.

Adaptive dynamics (9) is a gradient descent on a lower-bounded objective function, and so, assuming that the vectors of the training set form clusters in the n-dimensional space whose size corresponds to the cardinality of V_2, the (initially randomly located) weight vectors will converge towards the centers of these clusters during adaptive dynamics.

Let us define the following categories of normal vectors:

$$C_k = \{\vec{x} \in \Omega | \varphi(\vec{x}, \vec{w}_k) < \varphi(\vec{x}, \vec{w}_j)\} \quad \Omega = \{\vec{x} \in \mathbb{R}^n | |\vec{x}| = 1\} \quad (14)$$

$k \in V_2,\ j \in V_2 - \{k\}$ and φ is a non-Euclidean metric, i.e. the angle between the vectors.

The network function will thus assign, during lateral inhibition, a vector of the canonical basis of an m-dimensional space with a one on the k-th position to an arbitrary normal network input, if and only if the network input lies in the k-th category (14). The function of the network of the competitive model can thus be understood as a classification with respect to the categories specified above.

If we set $|V_2| = m^2$, then we can interpret the neurons in V_2as elements of a square $m \times m$ grid. Let us define the square neighborhood of the r-th order of the k-th element of the grid as the set containing all grid elements which lie at a distance of less than or equal to order r, i.e. $\sigma(k, r) = \{j \in V_2 | \rho(k, j) \le r\}$, where ρ is the metric defined on the grid as the neighborhood of elements of the appropriate order, and let us adjust the adaptive dynamics (12) for the k-th gain neuron:

$$\vec{w}_j(T) = \vec{w}_j(T-1) + \alpha_j(T)(\vec{a}(T) - \vec{w}_j(T-1)) \quad (15)$$

$j \in \sigma(k, r)$ and the plasticity drops globally with the time of the adaptive dynamics and locally with the order of the distance of the appropriate neuron from the gain neuron in the population grid of V_2.

The adjustment of the adaptive dynamics specified above generalize lateral inhibition by the extension of the excitation of the gain neuron to its neighborhood, which links the above-specified metric φ with the above-specified metric ρ. If the vectors of the training set are randomly distributed in the n-dimensional space in accordance with some distribution function, then after the adaptation of the network the weight vectors will be randomly distributed in the same area in accordance with the same distribution function.

If we present a training set on an adapted network in active mode, then the map of the frequency of excitations of neurons in V_2, the so-called Kohonen's map [5-6] will provide a mapping of the clusters of vectors of the training set in an n-dimensional space. Such a generalized competitive model, under assumption of a sufficiently large cardinality of V_2, performs the cluster analysis of the training set, i.e. determines the number of clusters and their distribution in the n-dimensional space.

Let us adjust the topology of the already adapted competitive model by adding a population set V_3, connected by edges to the population V_2 so that there is an edge from each neuron in V_2 to each neuron in V_3 ($\varepsilon(E_3) = V_2 \times V_3$). Let the new output population V_3 have the same cardinality as the input population V_1, and thus the population V_2 becomes a hidden population.

Let us set the weights of edges E_3 as follows: $w_{jq(i)} = w_{ij}$, $i \in V_1$, $j \in V_2$, $q(i) \in V_3$,where $q(i)$ is the image of thei-th neuron of population V_1 in population V_3. The output populationV_3 together with the weighted edges E_3 thus forms an image of the output population V_1 together with the weighted edges E_1 mirrored over the hidden population V_2, a phenomenon which we call counter propagation [7-8] of the synaptic weights of edges E_1 to edges E_3 in the direction of the orientation of edges. Let us select the activation functions of neurons in V_3 identically to the activation functions of neurons in V_1. Then, during active dynamics after the stabilization of the state of the population of V_2, the excitation of the k-th gain neuron will lead to the following values of potentials of neurons in V_3:

$$x_{q(i)} = \sum_j y_j w_{jq(i)} = w_{kq(i)} = w_{ik}$$

$i \in V_1$, $j \in V_2$, then stimulus $\vec{x} \in C_k$ implies the following network function: $\vec{F}(\vec{x}) = \vec{w}_k$.

The function of the network in the competitive model with forward propagation of weights will thus assign a *prototype* (the closest weight vector) to each normal network input. Prototypes lie in the centers of the appropriate clusters and thus represent these clusters – they are their typical representatives.

4. Simulated Annealing

Let's the cost function argument (2) expresses the macroscopic state of a thermodynamic system with energy equal to the function value. Then we can express its thermodynamic probability as the number of micro-states corresponding to it:

$$P(E_i) = \left|\{\vec{x} \in \mathbb{R}^n \,|\, f(\vec{x}) = E_i\}\right| \tag{16}$$

If we immerse this system with various macro-states with energies E_i in a thermal reservoir, then the Boltzmann equation for the unit size of the Boltzmann constant together with the Taylor expansion of a differentiable function, allows us to express the entropy of the reservoir after the temperatures equilibration for $E = E_0 + E_i = const$ and $E \gg E_i$ as follows:

$$S(E_i) = S(E) - \frac{dS(E)}{dE_i} E_i = \ln P(E - E_i) \tag{17}$$

and then, by using the definition of temperature $dS(E)\,/dE = 1/T$ we can express the thermodynamic probability of a macro-state of the thermal reservoir as a function of the energy of the macro-state of the inserted system, i.e. by the following Boltzmann factor $(T > 0)$:

$$P(E - E_i) = ce^{-\frac{E_i}{T}} \tag{18}$$

The simulated annealing algorithm is based on the perturbation of an optimum candidate and a following decision on its replacement by a perturbation in each iteration of the algorithm based on the Metropolis criterion [9]:

$$p(\vec{x}_i \to \vec{x}_j) = \frac{P(E_j)}{P(E_i)} = e^{-\frac{\Delta E}{T}} \qquad \Delta E > 0$$

$$p(\vec{x}_i \to \vec{x}_j) = 1 \qquad \Delta E \le 0$$

which expresses the probability of the system transferring from one macro-state to another, where $\Delta E = E_j - E_i$ and $\Delta E/T$ expresses the increase of entropy, i.e. in accordance with the second law of thermodynamics an impossible event is artificially redefined as a certain event in the specified criterion.

The sequence of accepted perturbations, i.e. acceptable solutions to the optimization task, forms a Markov chain with memory of order one, i.e. the occurrence of the given solution is only conditioned by the occurrence of the previous solution. The perturbations which lie outside of the area of admissible solutions are automatically rejected.

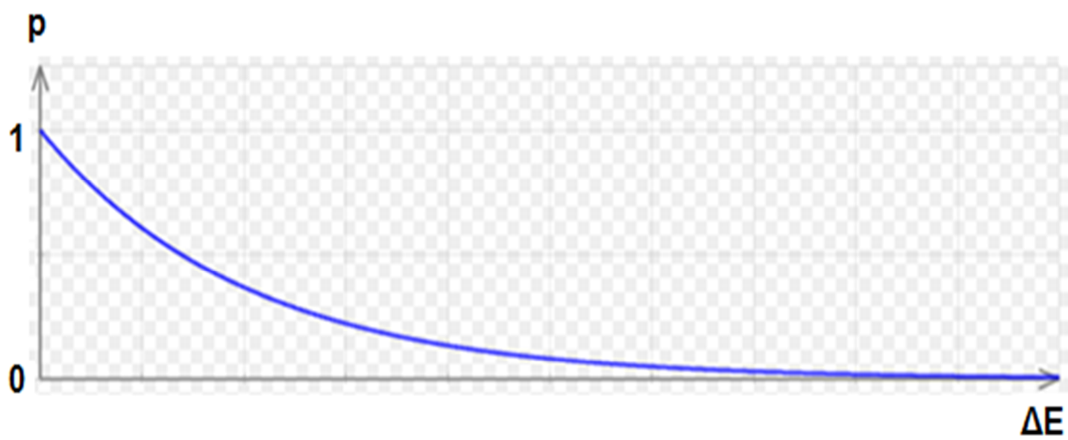


Fig 2. *Dependence of Probability on Increase of Energy.*

Fig 3. *Dependence of Probability on Temperature.*

From $p(\Delta E)$ (Fig. 2), it is clear that a significantly "worse" solution is accepted with respect to the previous solution at a much lower probability than a slightly "worse" solution. $p(T)$ (Fig. 3) may be used to control the probability of the acceptance of the solution during the iteration cycle. We initiate the iteration cycle with a sufficiently high temperature to ensure that almost every proposed solution is accepted for a certain period of time, which will allow an initial approximation of the solution to "escape" areas with shallow local minima. Later on, we reduce the temperature so that almost no "worse" solution is accepted, i.e. during the iteration cycle we cool down the system representing the optimization task from a sufficiently high temperature to a sufficiently low temperature until a solution is "frozen" in a sufficiently deep local minimum (Fig. 4). The temperature drop may be modeled e.g. as an exponentially decreasing function:

$$T = T_0 e^{-\frac{iter}{\tau}} \qquad \tau = -\frac{N}{\ln(T_\infty / T_0)} \qquad T_\infty \approx \lim_{iter \to \infty} T_0 e^{-\frac{iter}{\tau}} = 0 \tag{19}$$

where T_0 resp. T_∞ are the initial resp. final temperatures and N is the number of iterations of the algorithm.

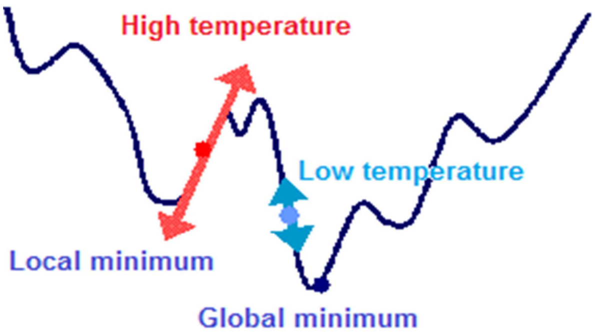


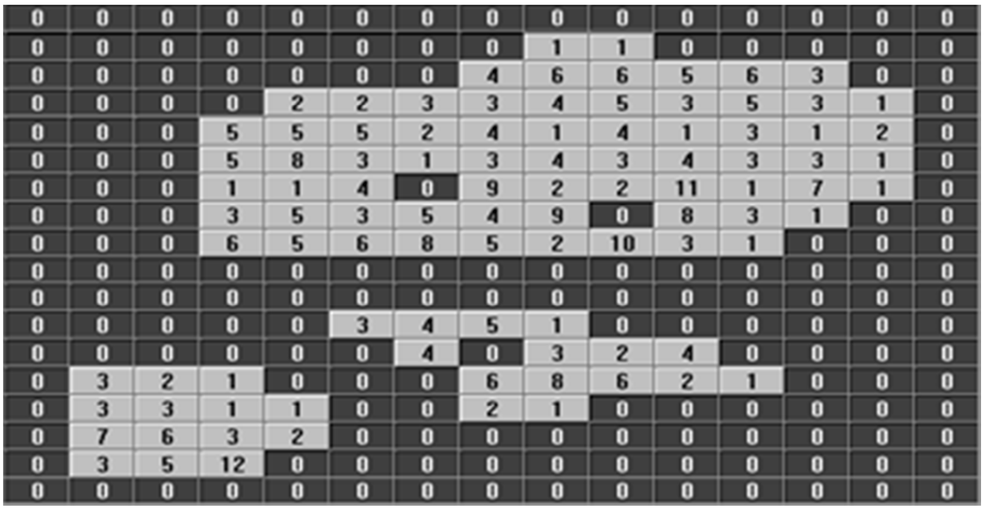
Fig 4. *Freezing of Solution.*

5. Experiment 1

The goal of this experiment is the identification of type daily diagrams of hourly smart grid consumption of electric energy on a workday, i.e. Wednesday, and on a non-work day, i.e. on Saturday, based on the recorded annual history of hour electric energy consumption of smart grid.

To allow the measuring of the efficiency of the utilized cluster analysis method, the annual history of hourly consumption of electric energy has been artificially modeled so that a typical daily diagram may be compared to a certain standard. The default standards of daily diagrams of hourly consumptions were the characteristic hourly developments of the consumption of the above-listed two days, where each hourly consumption of each of these was randomly modified by a random number generator with a normal probability distribution, as many times as was necessary to fill the annual history of hourly consumption.

Although individual daily diagrams of the annual history (Fig. 6,8) are mutually relatively different, the resulting type daily diagrams are very similar to the appropriate standards (Fig. 7,9). This documents the high efficiency of the utilized cluster analysis method. The experiment was processed in the computer program ArtInt © 2010 (Fig. 5).



0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1	1	0	0	0	0	0
0	0	0	0	0	0	0	4	6	6	5	6	3	0	0
0	0	0	0	2	2	3	3	4	5	3	5	3	1	0
0	0	0	5	5	5	2	4	1	4	1	3	1	2	0
0	0	0	5	8	3	1	3	4	3	4	3	3	1	0
0	0	0	1	1	4	0	9	2	2	11	1	7	1	0
0	0	0	3	5	3	5	4	9	0	8	3	1	0	0
0	0	0	6	5	6	8	5	2	10	3	1	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	3	4	5	1	0	0	0	0	0	0
0	0	0	0	0	0	4	0	3	2	4	0	0	0	0
0	3	2	1	0	0	0	6	8	6	2	1	0	0	0
0	3	3	1	1	0	0	2	1	0	0	0	0	0	0
0	7	6	3	2	0	0	0	0	0	0	0	0	0	0
0	3	5	12	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fig 5. *Example of Kohonen's Map from Program ArtInt.*

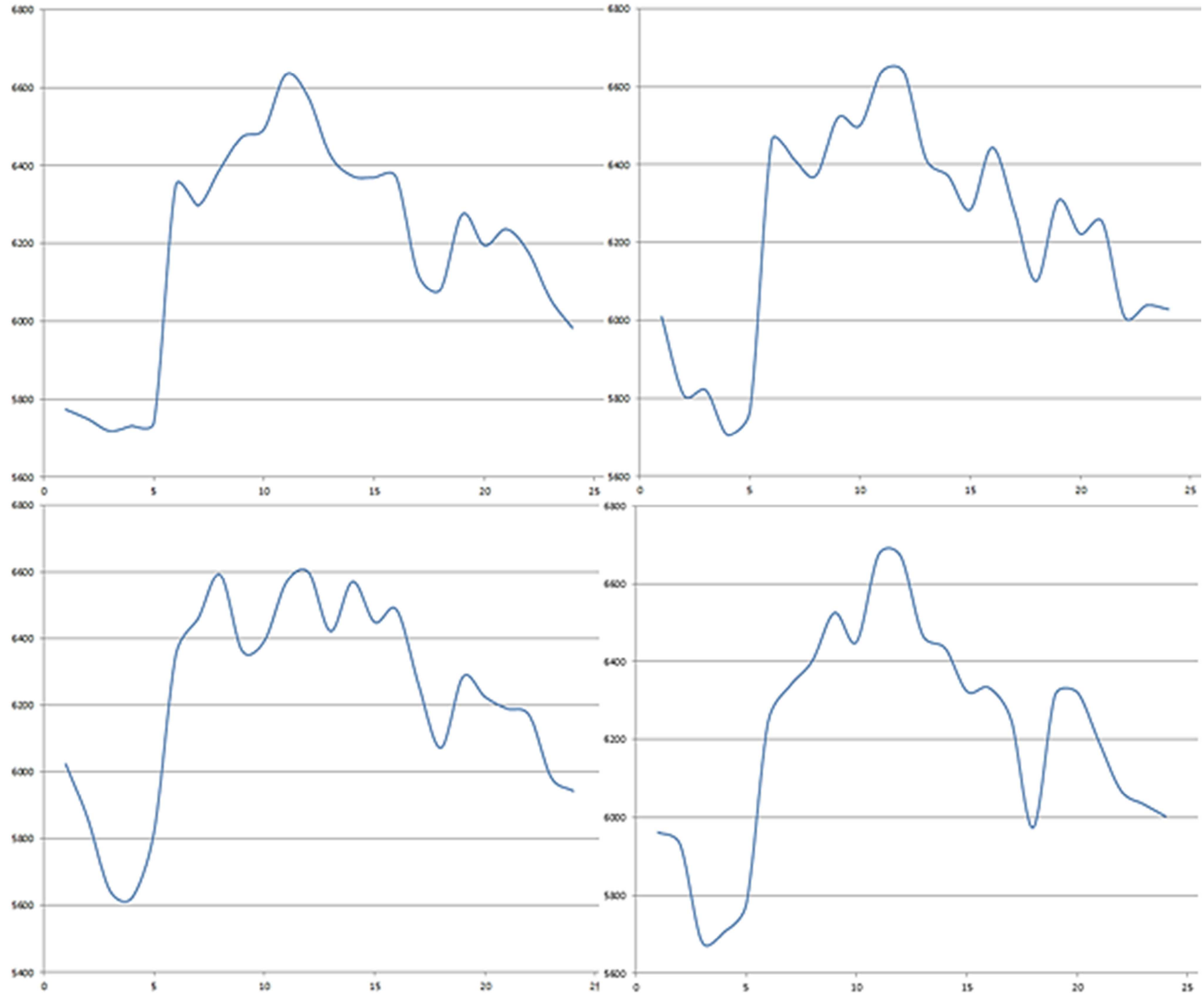


Fig 6. *Examples of Randomly Modelled Diagrams of Wednesday.*

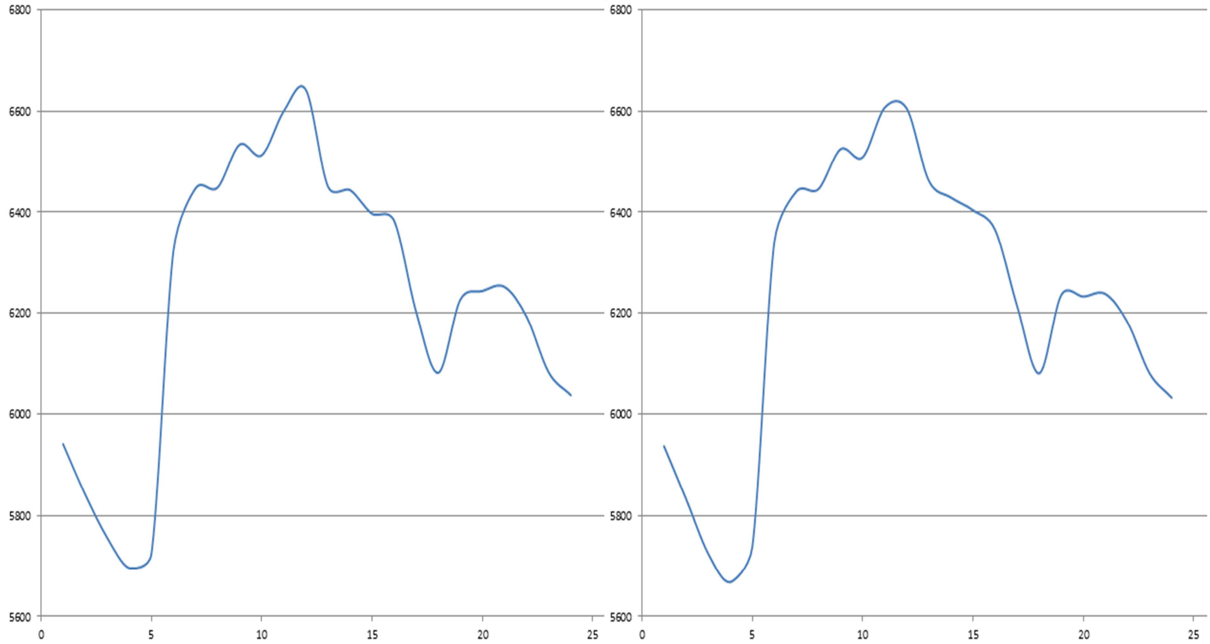


Fig 7. T*ype Daily Diagram and Standard of Wednesday.*

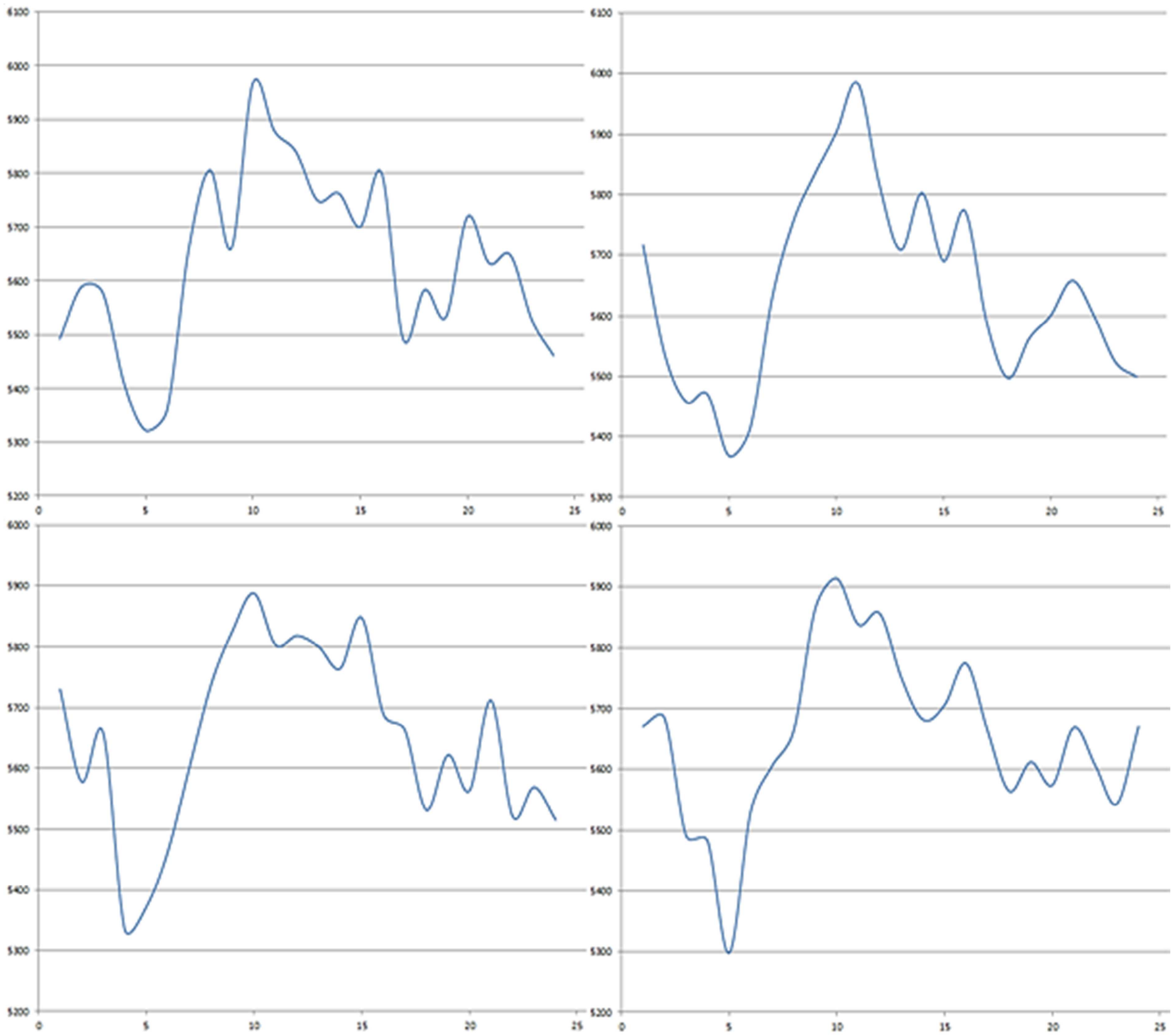


Fig 8. *Examples of Randomly Modelled Diagrams of Saturday.*

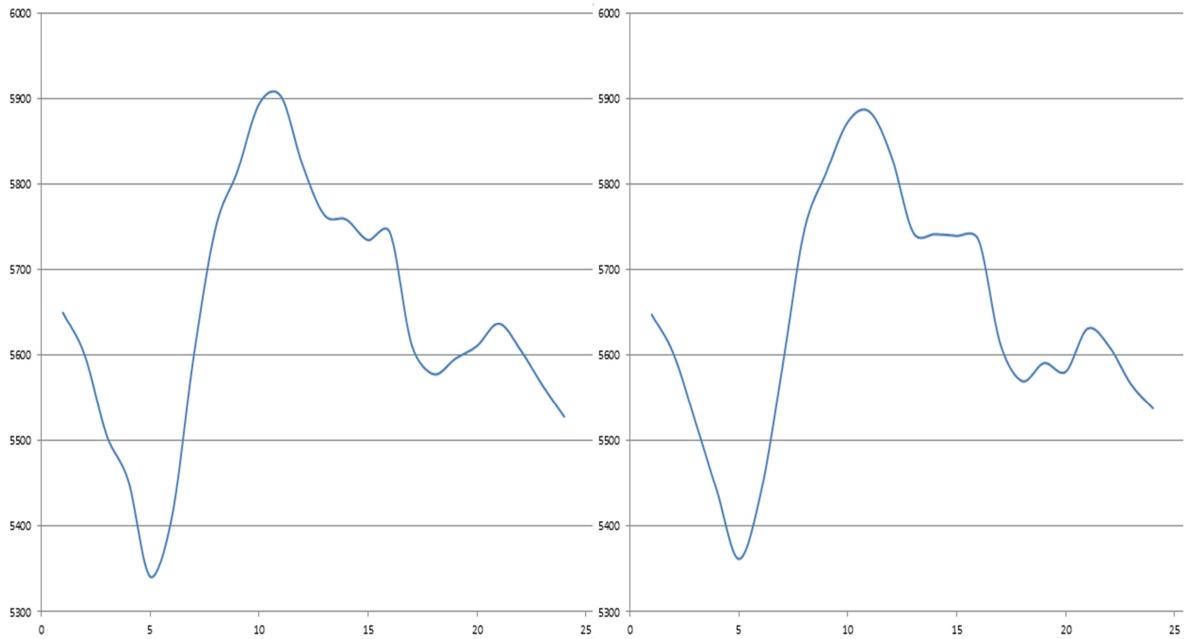


Fig 9. *Type Daily Diagram and Standard of Saturday.*

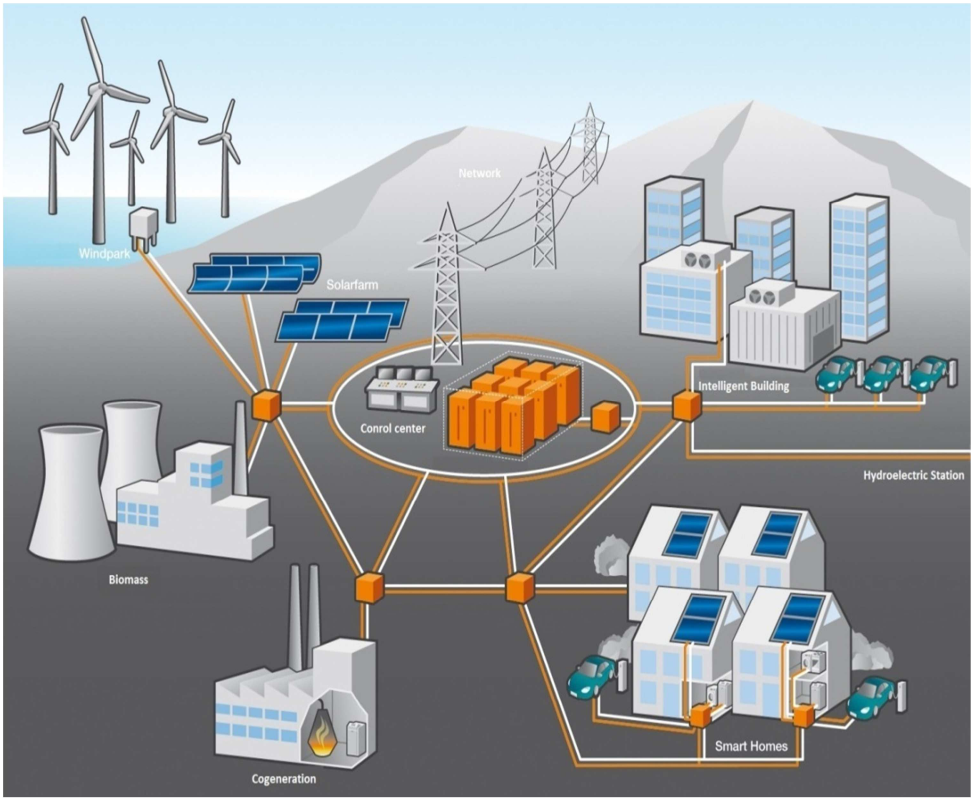


Fig 10. *Smart Grid with Distributed Generation.*

6. Experiment 2

The scenario of the following computational experiment is as follows: Let us assume a fictitious town supplied from smart grid (Fig. 10) [10]. The town, located near the foothills of a mountain range, is near a small river flowing from a lake, with a sufficient slant to build two hydro-electric plants. The vicinity of the mountains provides stable winds which are of sufficient power to build a park with wind power plants. Next to the town, there is a cogeneration plant which supplies the town with heat and power. Due to the highly developed agricultural production in the inland areas nearby, a biomass power plant has been built near the town. Due to the dominant cloudy weather in the considered period, the photovoltaic power cells located in the town do not provide sufficient output, and so these will not be included in the experiment. To retain the reliability of the delivery of power, the town is connected to two high-voltage power lines from different power suppliers.

The objective of the experiment is a proposal for an ordering of sources for a typical autumn Saturday resp. Wednesday, for which predictions of the hourly consumptions are available. In the specified experiment, the town is supplied by electrical power from thirty two rotary (synchronous generator) and eight non-rotary (transformer) machines, i.e. a total of forty sources and its cost characteristics and technical limitations are specified in Tab. 1, where HC=HC1+HC2*P is the consumption of the source itself based on the produced output. The experiment was processed in the computer program UniCom © 2010 (Fig. 11).

Tab 1. *Parameters of Sources.*

UNIT	NR	P_{min} [kW]	P_{max} [kW]	HC1 [kW]	HC2 [-]	A [CZK]	B [CZK/MW]	C [CZK/MW^2]	D [CZK]	TAU [-]
Water1	1	600	980	5	0.065	12046	142.0	0.029	48429	6.257
Water1	2	600	980	5	0.065	12046	142.0	0.029	48429	6.257
Water2	1	390	440	5	0.065	14667	161.2	0.030	53826	5.948
Water2	2	390	440	5	0.065	14667	160.7	0.030	53826	5.948
Water2	3	390	440	5	0.067	14667	161.4	0.030	53826	5.948
Water2	4	390	440	5	0.068	14667	162.4	0.030	53826	5.948
Wind1	1	120	200	10	0.078	15104	170.7	0.383	220797	6.633
Wind1	2	120	200	10	0.078	15104	170.7	0.380	220797	6.633
Wind1	3	120	200	10	0.078	15104	170.7	0.387	220797	6.633
Wind1	4	120	200	10	0.078	15104	170.7	0.390	220797	6.633
Wind1	5	120	200	10	0.078	15104	170.7	0.393	220797	6.633
Wind2	1	140	210	10	0.114	16480	188.3	0.448	352258	7.164
Wind2	2	140	210	10	0.114	16659	188.3	0.459	352258	7.164
Wind2	3	140	210	10	0.114	16480	188.3	0.451	352258	7.164
Wind2	4	140	210	10	0.114	16659	188.3	0.462	352258	7.164
Wind2	5	140	210	10	0.114	16480	188.3	0.455	352258	7.164
Cogener1	1	300	500	15	0.054	15902	198.8	0.152	278466	5.788
Cogener2	1	130	200	15	0.088	15815	176.8	0.437	230474	7.948
Cogener2	2	130	200	15	0.088	15815	176.8	0.431	230474	7.948
Cogener2	3	130	200	15	0.088	15815	176.8	0.434	230474	7.948
Cogener2	4	130	200	15	0.088	15815	176.8	0.427	230474	7.948
Biomass1	1	100	150	5	0.131	8483	199.5	0.884	167598	9.224
Biomass1	2	100	150	5	0.131	9534	200.7	0.853	155921	9.076
Biomass1	3	100	150	5	0.120	9273	224.0	0.701	153513	9.044
Biomass2	1	100	150	5	0.132	7948	204.4	0.939	72779	7.447
Biomass2	2	100	150	5	0.132	7948	204.4	0.943	72779	7.447
Biomass2	3	100	150	5	0.132	7948	204.4	0.947	72779	7.447
Biomass2	4	100	150	5	0.132	7948	204.4	0.950	72779	7.447
Biomass3	1	100	150	5	0.173	10505	282.3	1.101	128197	8.669
Biomass3	2	100	150	5	0.173	10505	282.3	1.051	128197	8.669
Biomass3	3	100	150	5	0.173	10505	282.3	1.001	128197	8.669
Biomass3	4	100	150	5	0.173	10505	282.3	0.951	128197	8.669
Network1	1	100	200	1	0.084	20903	338.8	1.554	50000	5
Network1	2	100	200	1	0.084	20903	338.8	1.550	50000	5
Network1	3	100	200	1	0.084	20903	338.8	1.545	50000	5
Network1	4	100	200	1	0.084	20903	338.8	1.541	50000	5
Network2	1	100	200	1	0.071	24409	387.2	1.574	50000	5
Network2	2	100	200	1	0.071	24409	387.2	1.569	50000	5
Network2	3	100	200	1	0.071	24409	387.2	1.563	50000	5
Network2	4	100	200	1	0.071	24409	387.2	1.580	50000	5

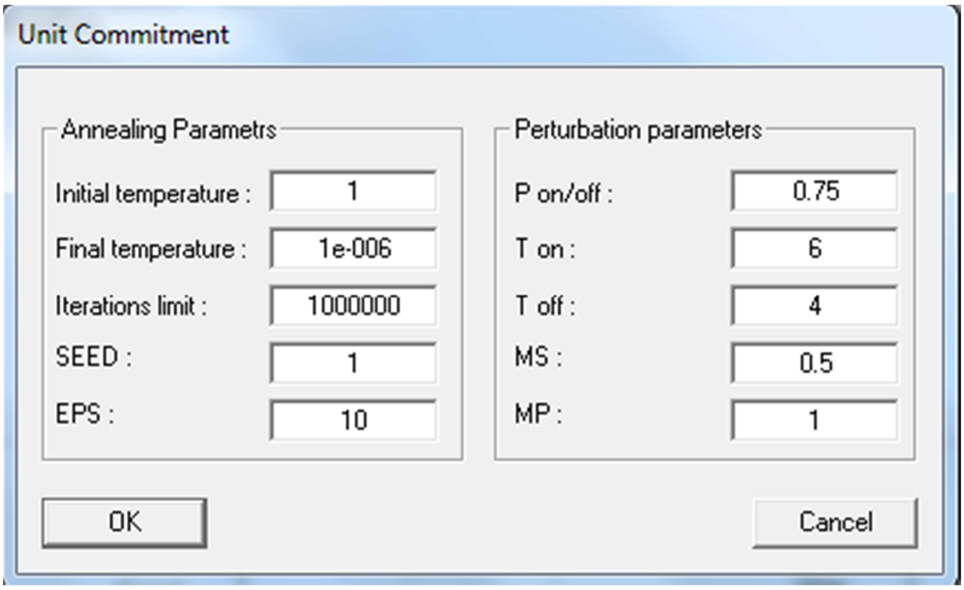


Fig 11. *Program UniCom.*

Supply	Nr.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
Water1	1	979	979	980	980	970	978	976	980	970	978	969	980	960	980	979	976	975	979	979	978	956	959	974	969
Water1	2	980	980	979	978	976	976	978	980	979	978	975	976	973	972	969	979	976	975	979	974	974	966	964	975
Water2	1	438	438	438	438	438	438	440	438	438	438	438	440	438	438	436	437	438	438	435	436	437	438	438	440
Water2	2	438	438	438	438	438	440	440	437	438	438	438	437	437	438	438	436	438	438	438	440	436	436	438	435
Water2	3	440	440	440	440	439	439	439	440	439	440	437	438	440	438	435	440	440	440	439	440	439	439	437	440
Water2	4	440	439	440	440	440	439	440	438	440	437	439	433	440	439	439	437	438	439	437	437	440	432	439	438
Wind1	1	198	195	197	196	194	198	197	197	198	200	192	195	197	196	194	196	195	198	196	198	197	195	194	198
Wind1	2	198	196	196	198	191	198	196	196	196	197	198	195	198	192	197	198	192	197	196	198	196	198	198	198
Wind1	3	196	197	196	197	196	197	197	198	198	198	195	198	198	198	194	196	195	185	194	197	198	200	198	197
Wind1	4	200	196	197	196	198	198	198	197	197	200	196	197	197	196	198	191	198	198	192	194	185	193	198	191
Wind1	5	197	195	200	198	198	198	196	198	198	198	195	197	198	198	198	193	191	193	194	195	196	196	196	198
Wind2	1	178	170	175	166	158	206	205	206	208	210	204	199	182	205	205	206	188	167	191	194	200	185	175	197
Wind2	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Wind2	3	170	153	153	151	166	210	205	203	204	205	210	205	199	206	205	205	205	161	187	184	206	179	201	183
Wind2	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Wind2	5	171	153	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Cogener1	1	488	484	487	486	487	498	499	498	496	499	498	499	488	487	498	490	497	496	497	496	492	497	488	487
Cogener2	1	197	173	175	197	195	198	198	198	198	195	198	186	197	200	186	191	174	195	176	197	196	197	187	194
Cogener2	2	198	176	187	198	194	197	197	200	198	197	198	193	196	198	194	195	187	200	195	183	190	186	185	187
Cogener2	3	196	186	187	187	192	198	198	196	197	198	198	190	198	198	176	197	198	178	198	196	194	194	174	194
Cogener2	4	197	198	186	196	197	198	197	198	198	197	198	196	196	198	195	189	192	198	195	183	173	193	175	184
Biomass1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Biomass1	2	102	100	102	0	101	141	148	133	143	137	141	136	102	141	146	133	108	101	115	118	114	106	100	102
Biomass1	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Biomass2	1	0	0	0	0	0	127	139	149	130	140	139	139	114	115	117	121	100	100	108	102	122	101	101	0
Biomass2	2	0	0	0	0	0	145	132	140	132	123	143	145	122	116	130	101	102	102	102	108	104	100	109	102
Biomass2	3	0	0	0	0	0	145	136	134	144	132	128	109	117	114	107	131	102	100	106	105	107	102	105	104
Biomass2	4	0	0	0	0	0	117	139	140	140	129	146	137	117	106	109	126	102	107	109	102	109	101	0	0
Biomass3	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Biomass3	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Biomass3	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Biomass3	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Network1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Network1	2	0	0	0	0	0	0	100	103	103	100	157	118	101	0	0	0	0	0	101	100	103	105	103	101
Network1	3	0	0	0	0	0	0	0	0	0	0	0	106	101	100	100	141	100	0	0	0	0	0	0	0
Network1	4	0	0	0	0	0	0	0	0	101	101	143	128	100	104	101	0	0	0	0	0	0	0	0	0
Network2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Network2	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Network2	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Network2	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Total	[kW]:	6601	6486	6353	6280	6368	7079	7190	7197	7283	7265	7373	7372	7206	7173	7146	7105	6931	6785	6959	6955	6964	6898	6777	6714
Load	[kW]:	5937	5833	5725	5668	5738	6341	6441	6446	6525	6509	6607	6605	6462	6429	6404	6365	6214	6081	6236	6233	6238	6182	6081	6033
Consum	[kW]:	674	663	638	622	640	748	759	761	768	766	776	777	754	754	752	750	727	714	733	732	736	726	706	691

Fig 12. *Unit Commitment of Wednesday.*

Supply	Nr.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
Water1	1	979	979	980	976	978	976	974	978	970	976	961	970	971	975	963	974	963	972	976	965	959	974	975	952
Water1	2	979	976	980	976	976	976	976	979	980	979	975	979	975	980	978	978	975	975	978	970	973	976	967	980
Water2	1	440	440	438	438	438	440	438	440	438	438	438	438	440	438	437	437	437	438	438	438	437	440	438	434
Water2	2	438	438	438	438	438	438	438	438	438	438	440	438	437	440	438	436	429	438	440	438	438	438	433	435
Water2	3	440	440	440	440	440	440	440	439	439	438	440	438	439	438	439	436	439	437	437	437	439	438	439	438
Water2	4	440	440	440	439	440	439	439	439	439	440	440	440	439	438	439	433	433	439	440	438	438	433	437	437
Wind1	1	196	196	197	197	196	181	194	198	196	180	198	198	195	185	189	189	183	191	197	183	175	184	198	185
Wind1	2	198	197	198	196	184	184	197	193	193	198	192	198	197	183	184	198	195	192	183	173	198	198	173	181
Wind1	3	195	194	195	198	194	197	180	185	198	198	185	194	185	192	196	198	196	198	198	191	183	195	198	184
Wind1	4	198	193	196	198	193	193	195	180	197	196	196	198	198	183	198	198	192	178	196	197	197	198	170	194
Wind1	5	198	194	192	198	198	180	195	197	193	198	197	197	180	196	187	198	187	173	173	182	200	192	193	175
Wind2	1	162	162	144	151	153	144	142	173	185	162	159	159	153	177	173	206	196	160	153	141	179	171	184	171
Wind2	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Wind2	3	170	152	152	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Wind2	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Wind2	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Cogener1	1	472	482	444	483	442	453	471	490	482	500	497	488	489	484	481	485	490	472	489	498	492	486	490	482
Cogener2	1	184	175	174	164	171	175	169	185	185	151	187	176	176	186	198	173	179	158	163	195	176	149	162	172
Cogener2	2	187	186	169	186	176	161	173	175	164	180	167	174	140	186	195	174	163	176	173	175	192	185	185	174
Cogener2	3	197	183	176	160	163	183	176	196	159	176	176	186	167	198	184	163	176	174	175	172	164	170	174	172
Cogener2	4	186	182	170	176	151	164	196	192	198	175	183	190	191	186	181	179	176	195	181	185	187	183	147	169
Biomass1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Biomass1	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Biomass1	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Biomass2	1	0	0	0	0	0	100	102	100	101	100	102	0	0	0	0	0	0	0	0	0	0	0	0	0
Biomass2	2	0	0	0	0	0	0	100	106	107	102	100	101	106	0	0	0	0	0	0	0	0	0	0	0
Biomass2	3	0	0	0	0	0	0	0	101	100	101	102	109	100	101	101	102	109	104	100	102	108	101	101	101
Biomass2	4	0	0	0	0	0	0	0	0	105	102	101	106	102	101	107	100	0	0	0	0	0	0	0	0
Biomass3	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Biomass3	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Biomass3	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Biomass3	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Network1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Network1	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Network1	3	0	0	0	0	0	0	0	0	0	101	108	100	100	101	101	104	100	100	101	101	103	101	101	101
Network1	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Network2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Network2	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Network2	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Network2	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Total	[kW]:	6259	6209	6123	6014	5931	6024	6195	6384	6467	6529	6544	6477	6380	6368	6369	6361	6218	6170	6191	6181	6238	6212	6165	6137
Load	[kW]:	5648	5602	5523	5441	5362	5438	5586	5746	5813	5873	5885	5833	5744	5742	5740	5735	5613	5570	5591	5581	5631	5610	5566	5538
Consum	[kW]:	621	617	610	583	579	596	619	648	664	666	669	654	646	636	639	636	615	610	610	610	617	612	609	609

Fig 13. *Unit Commitment of Saturday.*

7. Conclusion

By comparing Fig. 7 to Fig. 12, it is clear that the Wednesday consumption is covered in a similar fashion, but due to the increased volumes, i.e. a larger area below its progression, the cluster of outputs supplied by the first and second biomass power plants covering the mid-day and evening peaks is larger. Three transformers of one of the distribution companies, specifically the one with the lower price of energy, were turned on during the midday peak – one of which was turned off temporarily between the peak hours. The third biomass power plant was not used at all due to its higher start-up costs and relatively high operating costs. Referential resp. optimal costs for the coverage of the Wednesday energy consumption then amount to 69 216 resp. 45 006 CZK.

By comparing Fig. 9 to Fig. 13, it is clear that the power consumption on Saturday is primarily covered by sources with more or less lower production costs, such as both hydro-electric plants and the first wind-power park together with cogeneration units, which contribute by supplying the town with heat. The mid-day consumption peak corresponds well with the cluster of outputs supplied by the biomass power plant and the start-up of one transformer, which together with one source of the specified cluster also covers the evening consumption peak. Referential resp. optimal costs for the coverage of the Saturday energy consumption then amount to 65 254 resp. 38 446 CZK.

In practice, unit commitment optimization problem was usually solved by Lagrange multipliers method, but it does not work correct with bivalent independent variables from objective function, i.e. with state of source. Therefore, heuristic methods seem more appropriate to solve our optimization problem.

References

[1] S. Grossberg, Studies of Mind and Brain, Reidel, Dordrecht 1982

[2] S. Grossberg, The Adaptive Brain, North-Holland 1987

[3] J. J. Hopfield, Neural networks and physical systems with emergent collective computational abilities, Proceedings of the National Academy of Sciences of the USA, 79(8) 1982

[4] J. J. Hopfield, Artificial Neural Networks, IEEE Circuits & Devices 4 1988

[5] T. Kohonen, Self-Organized Formation of Topologically Correct Feature Maps, Biological Cybernetics 43(1) 1982

[6] T. Kohonen, The Self-Organizing Map, Proceedings of IEEE 78(9) 1990

[7] R. Hecht-Nielsen, Counter-propagation networks, Appl. Opt. 26(23) 1987

[8] R.Hecht-Nielsen, Applications of Counter-propagation networks, Neural Networks 1(2) 1988

[9] N. Metropolis, Equations of State Calculations by Fast Computing Machines, Journal of Chemical Physics, 21(6) 1953

[10] URL:http://powertown.no/wp-content/uploads/2011/11/SmartGrid_Ueberblick_ohneLegende.jpg,

[11] © 2010 ABB, Deutsche Telekom (translated)

[12] B. Garlík, M. Křivan,Renewable energy unit commitment, with different acceptance of balanced power, solved by simulated annealing, Energy and Buildings, Volume 67, December (2013)

[13] A.H. Mantawy, Y.L. Abdel-Magid, S.Z. Selim, A simulated annealing algorithm for unit commitment, IEEE Transactions on Power Systems 13 (1998)

[14] A.H. Mantawy, Y.L. Abdel-Magid, S.Z. Selim, Integrating genetic algorithms, Tabu search and simulated annealing for the unit commitment problem, IEEE Transactions on Power Systems 14 (1999)

[15] G.B. Sheble, T.T. Maifeld, Unit commitment by genetic algorithm and expert system, Electric Power Systems Research (1994)

[16] H. Sasaki, M. Watanabe, R. Yokoyama, A solution method of unit commitment by artificial neural networks, IEEE Transactions on Power Systems 7 (1992)

[17] J.M. Arroyo, A.J. Conejo, A parallel repair genetic algorithm to solve the unit commitment problem, IEEE Transactions on Power Systems 17 (2002)

[18] K.A. Juste, H. Kita, E. Tanaka, J. Hasegawa, An evolutionary programming solution to the unit commitment problem, IEEE Transactions on Power Systems 14 (1999)

[19] K.S. Swarup, S. Yamashiro, Unit commitment solution methodology using genetic algorithm, IEEE Transactions on Power Systems 17 (2002)

[20] M.P. Walsh, M.J.O. Malley, Augmented Hopfield network for unit commitment and economic dispatch, IEEE Transactions on Power Systems 12 (1997)

[21] N.S. Sisworahardjo, A.A. El-Kaib, Unit commitment using ant colony search algorithm, in: Proceedings of the 2002 Large Engineering Systems Conference on Power Engineering, (2002)

[22] P.C. Yang, H.T. Yang, C.L. Huang, Solving the unit commitment problem with a genetic algorithm through a constraint satisfaction technique, Electric Power Systems Research 37 (1996)

[23] U.D. Annakkage, T. Numnonda, N.C. Pahalawaththa, Unit commitment by parallel simulated annealing, in: Proceedings of IEE-Generation Transmission and Distribution, (1995)