Algorithms For Backtracking for Objective Functions

Borislav Dimitrov[1], Marinela Yordanova[2] and Hristo Nenov[3]
[1]Associate Professor
Technical University of Varna
bdimitrov@processmodelig.org

[2]Associate Professor
Technical University of Varna
m.yordanova@tu-varna.bg

ABSTRACT: *We have developed an optimization algorithm for backtracking. To ensure to achieve the objective function we have prepared the backtracking which help to optimize the resistance furnace. These algorithms can help to do optimization procedure ushing graph theory. The applications are carried out with many functions and finally we have implemented the computational tasks.*

Keywords: Algorithm “Backtracking”, Optimization, Electric Resistance Furnaces, Energy Efficiency

Received: 26 July 2022, Revised 11 September 2022, Accepted 27 September 2022

DOI: 10.6025/jitr/2023/14/1/11-17

1. Introduction

Electric resistance furnaces (ERF) for industrial applications are powerful consumers of electric energy and their optimization according to different objective functions is an important task. Fig.1.A shows the electric resistance furnace’s typical construction. The most often applied objective function at optimization is minimum loss of energy, but also the other as minimal mass, volume and price find application. The mathematical description of the functions has been proposed in [2, 3]. It has been applied in this investigation of algorithm “backtracking” [7, 9] for optimization of ERF.

The aim of the paper is to investigate the possibility to apply the algorithm “backtracking for electric furnaces optimization. The object of the research has been real furnace equipment, subject to reconstruction in order to reduce energy losses. The results are proposed for a particular furnace which insulation is designed with the algorithm under discussion.

2. Applying Algorithm “Backtracking’ for Optimization of ERF

The principle of the algorithm backtracking [7, 9] consists in stepping solution of the problem. At each step the current solution expands the possible extensions. In case that the obtained results do not meet said requirements, the algorithm returns to the previous step and continues on an adjacent branch of the tree.

The search continues to find a solution, or to establish that the problem is unsolvable for the input conditions. The last statement can be reached by examining all possibilities. A recursive algorithm [1, 6, 7] is used - listing 1.

Listing 1

```
void try (step i){

if (i>n) { // check received decision }

else { // implement a solution in all possible ways }

for (k=1; k<=n; k++)

if (condition k is acceptable){ candidate registration;

try (i+1) // recursive call to the function remove registration;
```

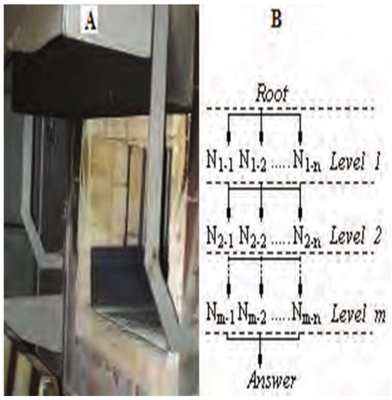

Figure 1. A– Typical construction of high temperature ERF; B– graphic interpretation of the algorithm “backtracking”

The application of the backtracking algorithm for optimal design of the ERF is illustrated by the tree structure shown in fig.1.B. The root of the tree contains the initial conditions for solving the problem. The number of levels (m) and the nodes (n) are defined by the objective function, respectively, the needed equations and database from the heat-insulating and fireproof materials. The following sequence is adopted:

- **Root (Figure 1.B) –** The input data: characteristics of the heated detail, temperature and heating time, the objective function of the optimization and other. Since the optimization procedure is directed to the heat- insulation of the furnace the database of fireproof and heat-insulating materials is necessary. Their main heat characteristics are: λ - thermal conductivity coefficient through the insulation; c - specific heat capacity; j - density; tmax - maximum temperature.

- **Level (1, 2 … m).** Each level contains an equation or computational procedure of pre-accepted methodology [4], which is calculated consistently. The number of levels is determined by the number of equations to be calculated. Moreover computing the relevant level has to determine the number of possible nodes that meet the requirements. They are marked as perspective (1, true, etc.) and passing to the next level going through. The others have no future (0). For example: the first level has a database of fireproof materials used in the first layer of insulation. Each node (N1-1, N1-2 .. etc.) contains one material. Those materials responding to specific requirement are marked.

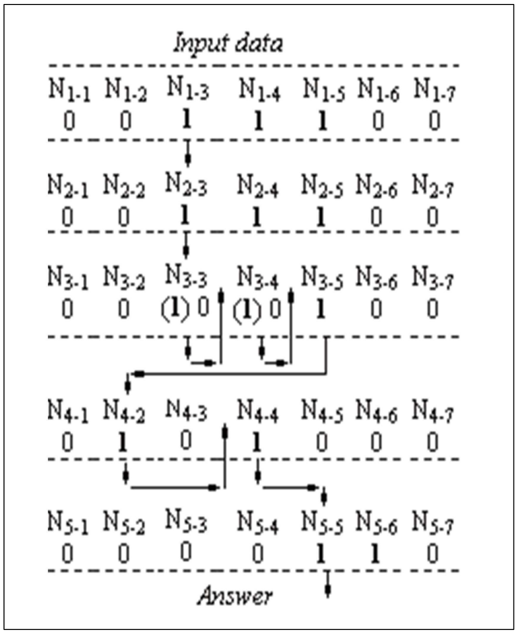

Figure 2. Graphical representation of the algorithm "backtracking" in solving a specific optimization problem

- **Conditions:** Each of the levels contains limit conditions in accordance with the equation that determine the perceptiveness of the nodes. The conditions may be limits for overheating at rated current, contact resistance, contact force, etc.

- It passes to the second level through the first perspective node of the first level (eg, $N_{1\text{-}2}$). By sequentially scanning nodes $N_{2\text{-}1}$, $N_{2\text{-}n}$ possible decisions, meeting the required conditions for this level, are searched for. If there are any, they are marked as perspective and through them it continues to the next level. If there is not perspective decision of the second level according the data in the node of the first level (N1-2), the algorithm returns to the first level. In this case $N_{1\text{-}2}$ is marked as a non-perspective and the calculation starts from the next perspective node. An example is shown in Figure 2: transition from level 3 to 4 via nodes $N_{3\text{-}3}$ and $N_{3\text{-}4}$ is impossible and they are marked as no perspective, i.e. the heat insulation materials are rejected as inappropriate to the restrictive conditions. Finally, transition is from node $N_{3\text{-}5}$.

- The same sequence repeats up to final decision. The marked nodes contain the values of each of the levels, i.e. those are the decisions of each of the equations in the realization of the task. Computing can continue in the presence of marked but not explored nodes in different levels. This allows finding more than one decision.

The basic equations for design of the thermal insulation are composed on the base of the replacing scheme of Figure 3.

The accumulated energy in each layer (i-th) of the furnace walls:

$$Q_{erf} = c_i . m_i . (\tau_i - \tau_o) \tag{1}$$

The accumulated energy in the heated detail:

$$Q_d = c_{id} . m_{id} . (\tau_{id} - \tau_o) \tag{2}$$

Where the consecutive coefficients in layers are labeled with index *i*: *ci*(*id*)– specific heat of the furnace (the load); $m_{i\,(id)}$ – mass of the furnace (the load); $\tau_{i\ (id)}$ – temperature in the furnace (the load); τ_0 – temperature of the ambient (initial temperature).

The total losses Q_l are determined by the amount of the accumulated energy in the furnace walls and the losses to the environment Q_l:

$$Q_l = Q_{erf} + Q_o \qquad (3)$$

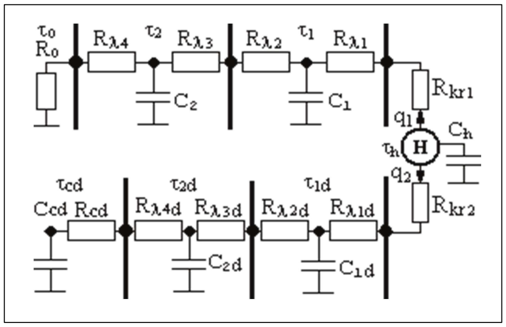

Figure 3. Replacing scheme used for the modeling of the thermal processes in the complex furnace-heated detail

System of equations describing the transient heat process is composed from the replacing scheme as follows:

$$\left|\begin{array}{l} q = \dfrac{\tau_h - \tau_1}{R_{kr1} + R_{\lambda 1}} + \dfrac{\tau_h - \tau_{1d}}{R_{kr2} + R_{\lambda 1d}} \\ C_1 \dfrac{d\tau_1}{dt} + \dfrac{\tau_1 - \tau_2}{R_{\lambda 2} + R_{\lambda 3}} = \dfrac{\tau_h - \tau_1}{R_{kr1} + R_{\lambda 1}} \\ C_{1d} \dfrac{d\tau_{1d}}{dt} + \dfrac{\tau_{1d} - \tau_{2d}}{R_{\lambda 2d} + R_{\lambda 3d}} = \dfrac{\tau_h - \tau_1}{R_{kr2} + R_{\lambda 1d}} \end{array}\right. \qquad (4)$$

Computational procedure is implemented in Matlab, based on numerical methods of Runge-Kutta [4, 5, 10]. For receiving the transient process of heating of the researched furnace ode45, ode23, ode15s are used. This procedure is carried out in one of the levels (in the example N_4).The differential equations giving the transient heat process are in the following system:

$$\left|\begin{array}{l} C_1 \dfrac{d\tau_1}{dt} = -\left(\begin{array}{l} \dfrac{1}{R_{kr1} + R_{\lambda 1} + R_{kr2} + R_{\lambda 1d}} \\ + \dfrac{1}{R_{\lambda 2} + R_{\lambda 3}} \end{array} \right) . \tau_1 + \\ + \dfrac{q_1}{1 + \left(\dfrac{R_{kr1} + R_{\lambda 1}}{R_{kr2} + R_{\lambda 1d}} \right)} \\ C_2 \dfrac{d\tau_2}{dt} = \left(\dfrac{1}{R_{\lambda 2} + R_{\lambda 3}} \right) . \tau_1 - \\ - \left(\dfrac{1}{R_{\lambda 2} + R_{\lambda 3}} + \dfrac{1}{R_{\lambda 4} + R_0} \right) . \tau_2 \\ C_{1d} \dfrac{d\tau_{1d}}{dt} = \left(\dfrac{1}{R_{kr1} + R_{\lambda 1} + R_{kr2} + R_{\lambda 1d}} \right) . \tau_{1d} \\ + \dfrac{q}{1 + \left(\dfrac{R_{kr2} + R_{\lambda 1d}}{R_{kr1} + R_{\lambda 1}} \right)} \\ C_{2d} \dfrac{d\tau_{2d}}{dt} = \left(\dfrac{1}{R_{\lambda 2d} + R_{32d}} \right) \theta_{d1} - \left(\dfrac{1}{R_{\lambda 4d} + R_{\lambda cd}} \right) . \tau_{d2} \end{array}\right.$$

The symbols in Figure 3 and those of systems of equations (4) and (5) are following:

$R_{\lambda 1 \div 4}$ - resistances of the conductive heat transfer through the walls of the furnace. They conduct the heat flow of the losses q_1 - form the chamber of the furnace to the outside environment.

R_0, R_{kr1}, R_{kr2} - resistances of convection and radiation, respectively from the housing of the ERF towards the outside environment and from the heater H towards the surface of the heat insulation layer and the heated thermal load (detail).

$R_{\lambda 1 \div 4d}$, R_{cd} - resistances of the conductive heat transfer in the heated thermal load. The useful heat flow q2 is conducted through them.

C_1, C_2, C_{1d}, C_{2d}, C_{cd}, C_h - heat capacity, respectively of the two layers of the furnace and the two layers and the center of the heated thermal load and the heater.

τ_h, τ_1, τ_2, τ_0, τ_{1d}, τ_{2d}, τ_{cd} - temperatures, respectively: of the heater; of the two layers of the furnace; the outside environment; the two layers and the center of the heated thermal load. The temperatures are given as initial values and they are determined each iteration of the computing procedure. Only the temperature of the outside environment t0 is excluded from this rule - it is set as constant. In this way a constraint from third order is set.

3. Processing Results With Graph Theory

The optimization procedure allows numerous results to be processed. The nodes, marked as promising, represent directed graphs, an example of which is shown in Figure 4. This allows using algorithms to work with graphs.

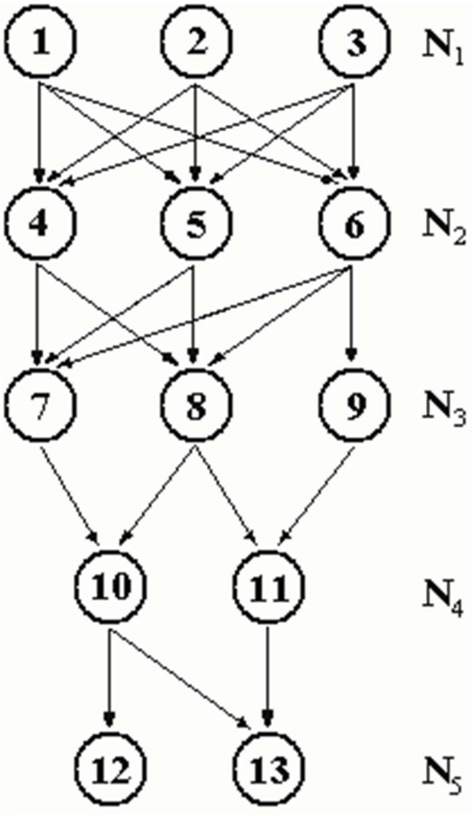

Figure 4. Oriented graph, consisting of optimization data of ERF

```
}}
```

To solve the optimization problem in question algorithm "BFS - Breath-First-Search" is used. Processing begins from the top vertex - i, i.e. in the specific task of level N_1, node 1 and all of its immediate neighbors are reviewed.

Then it proceeds to further search – search in the width of each of its neighbors. Generally, according to the graph theory BFS algorithm requires passing through all the nodes in this manner - i.e. sequential selection (random) of starting vertex until all nodes of the graph are not searched.

In this particular task it should always start from node of level N_1, because it makes no sense to start working on another part of the results. BSF function uses the adjacency matrix of A (Figure 2).

The function used with C / C++ syntax is given in listing 2 [1, 6, 7]. Further studies have been made on the application of algorithms "DFS - Depth-First-Search" and Dijkstra's algorithm for finding minimum paths in a graph.

Listing 2:

```
#define MAXN 300
unsigned n; //counting the nodes of graph
unsigned v=1; // starting node for searching
void BSF (unsigned i){ // the function is called with BSF (v-1)
unsigned k, j, p, queue[MAXN], currentVert, LevelVertex,
queueEnd;
for (k=0; k<n; k++)
queue [k] = 0;
for (k=0; k<n; k++)
used [k] = 0 ;
queue [0] = i ;
used [i] = 1 ;
currentVert = 0 ;
levelVertex = 1 ;
queueEnd = 1 ;
while ( currentVert < qeueEnd ) {
for ( p = currentVert; p < levelVertex; p++ ) {
printf (“%u”, queue[p] + 1)
currentVert ++ ;
for ( j=0; j<n; j++ )
if ( A[queue[p]][j] && !used[j] ) {
queue [queueEnd++] = j ;
used[j] = 1 ;
}}
printf (“\n”);
levelVertex = queueEnd ;
}}
```

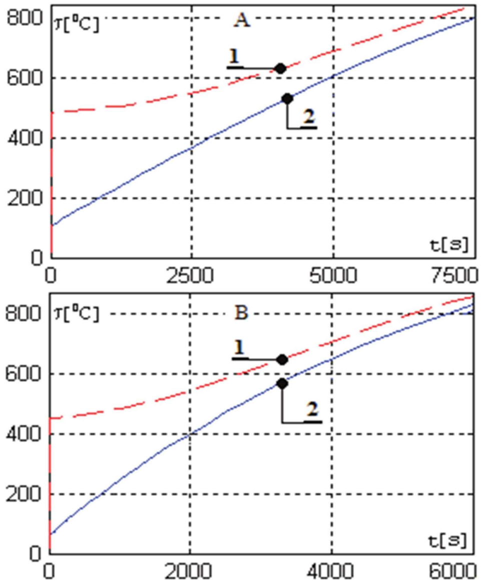

Figure 5. Transient process of heating of ERF before (A) and after reconstruction of insulation (B) dimensioned with the proposed optimization procedure. 1 - temperature of the heater, 2 - the surface temperature of the load

Experimental studies have been made for the reconstructing middle- temperature ERF with installed capacity of 120kW. The reconstruction consists of replacement of heat-insulation.

The new layer of the insulation is designed with the exposed procedure with the objective function - minimal losses. Figure 5 shows the results before reconstruction. The operating temperature of 8000C of the surface of heated load (Figure 2) is achieved in approximately 7500 sec (2h). After designing of the furnace with the proposed optimization procedure (figure 5.B) at the same installed capacity, operating temperature is reached approximately 5500 sec. due to reduced losses. The furnace consumes about 34 kW/ h less per cycle. The example shown in Figure 5 is optimized with objective functionminimum losses, but the described methodology is applicable with objective functions- minimum mass, volume and price. In those cases the database must contain and price lists of materials. For practical purposes complete software for design of electro-thermal equipment is designed.

4. Conclusion

The optimization algorithm „backtracking" and the processing results with BFS – Breath-First-Search can be applied for optimization of ERF. The correct work of algorithm requires large database for fire- proof and heatinsulating materials.

Acknowledgement

This paper is developed in the frames of project "Improving energy efficiency and optimization of electro technological processes and devices", № МУ03/163 financed by the National Science Fund.

References

[1] Ameral, L. (2001). *Algorithms and Data Structures in C++*. Sofia.

[2] Dimitrov, B., Tahrilov, Hr. & Nikolov, G. (2012) *Study and analysis of optimization approaches for insulation of an industrial grade furnace with electrical resistance heaters International scientific conference on information, communication and energy systems and technologies*, Vol. XLVII. ICEST: Veliko Tarnovo, Bulgaria.

[3] Dimitrov, B. B. (2012) Hr. Tahrilov, A. *Marinov. Inproving Energy Efficiency of Industrial Grade Furnaces with Electrical*

Resistance Heaters and Comparative Model-Experiment Analysis XLVII *International scientific conference on information, communication and energy systems and technologies*. ICEST: Veliko Tarnovo, Bulgaria.

[4] Dimitrov, B. (2006) Analysis of the calculation process, a mathematical model of the electric resistance. *Magazine Electrical and Electronics* E + E, 11–12, 78–83.

[5] Dimitrov, B. & Tahrilov, Hr. (2005) Comparison of numerical methods for solving transient resistance in the electric furnace. *Magazine Electrical and Electronics E + E*, 5–6, 26–31.

[6] Liberty, J. (1999). C++ Unleashed. Sams Publishing.

[7] Nakov, P. & Dobrikov, P. (2003) Programming C++. Algorithms, (Sofia).

[8] Tahrilov, Hr. & Dimitrov, B. (2009). *Design, Modeling and Optimization of Electric Resistance Furnaces. TU:* Varna, Bulgaria.

[9] Floudas, C. & Pardalos, P. (2009). *Encyclopedia of Optimization*. Springer: Berlin.

[10] Hahn, B. & Valentine, D. (2010). *Essential MATLAB for Engineers and Scientists.* Elsevier: Amsterdam.