Genetic Algorithms as a Tool of Production Process Control

Petr Klímek, Martin Kovářík
Faculty of Management and Economics, Tomas Bata University in Zlín, Czech Republic
pklimek@seznam.cz, kovarik.fame@seznam

DOI: 10.20470/jsi.v5i3.188

Abstract: The following article deals with up-to-date field of optimization and genetic algorithms use in the production process control. Theory of optimization is mentioned in the first part of this paper. The second part of this article is dedicated to genetic algorithms. Genetic algorithms are characterized by their robustness – the ability to solve difficult optimization or the tasks in which we need to decide on something. The third part of the paper focuses on practical demonstrations of two examples. The first example is on determining the optimal production. The second example is on application of genetic algorithms in process planning. Matlab and Evolver software tools were used for the data computation. Data preparation for Evolver was done in MS Excel. But the model building is only one step in knowledge discovery. The best model is often found after building models of several different types, or by trying different technologies or algorithms.

Key words: optimization, genetic algorithms, selection, mutation, crossing, Matlab, Evolver.

Introduction

The following article deals with up-to-date field of optimization and genetic algorithms use in the production process control. Theory of optimization together with one practical example is mentioned in the first part of this paper. The second part describes theory of genetic algorithms. The last part of this article is dedicated to the demonstration of two production planning examples. Data were computed in Matlab and Evolver software tools. Data were prepared for Evolver in MS Excel.

1. Classical Optimization

Optimization is a mathematical discipline in which we are looking for a minimum (or maximum) of the function \( f(x) \) on the set \( M \). This function is called the target or purpose. Other advanced techniques were also discussed in (Bezděk, 2011).

Set (called the set of admissible solutions) is typically described with some limitations, most often a set of equations or inequalities, etc. The minimum is a mathematical function whose functional value is the lowest value of all input parameters. The function of the parameters and comparing the result is the value of the parameter when compared to all other appears to be low. Maximum is a mathematical function whose functional value is the highest value of all input parameters. The function compares all parameters and the result is the value of the greatest parameter.

1.1 Methods of Classical Optimization

Classical optimization method means those which are based on optimality conditions. We set stationary points for differentiable function \( f \) i.e., all solutions of the equation \( \text{grad} \ f(x) = 0 \).

Methods for solving optimization can be divided into three categories according to the requirements for smoothness of the function \( f \) (Price, Storn, and Lampinen, 2005).

The direct selection methods – do not require calculation of derivatives.

Special attraction methods – require calculation of the gradients (e.g. steepest descent method, quasinewton method, etc.).

Newton’s method – requires the calculation of second derivatives.

Because cases need to solve global optimization especially in practice, it is very important to deal with the theoretical aspect, the general role of global optimization. Often the very simple formulation of
the optimization problem can be confusing and give the impression that the determination of general deterministic algorithm solving the global optimization task is simple. In fact, analysis shows that such a deterministic algorithm exists (Kovářík, 2008).

1.2 The Production Program Building

The engineering plant manufactures four products V1, V2, V3, and V4. In compiling the production program to be reckoned with, limited capacity production facility that is 1 200 hours and a limited amount of raw materials, which is 1 400 tons.

Products V1 and V2 are needed to produce semi-finished products, V2, V3 and V4, and can also be sold. Sales prices of these products are 300, 600, 1 000 and 3 000 per tonne. The task is to determine the production programme as a maximum value of production. Required data are listed in the following table (Tab. 1).

<table>
<thead>
<tr>
<th></th>
<th>V1</th>
<th>V2</th>
<th>V3</th>
<th>V4</th>
<th>Capacity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Z</td>
<td>1.5</td>
<td>0</td>
<td>2</td>
<td>2.5</td>
<td>1 200</td>
</tr>
<tr>
<td>S</td>
<td>2</td>
<td>1.5</td>
<td>2</td>
<td>0</td>
<td>1 400</td>
</tr>
<tr>
<td>V1</td>
<td>0</td>
<td>0.5</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>V2</td>
<td>0</td>
<td>0</td>
<td>0.5</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>Price</td>
<td>300</td>
<td>600</td>
<td>1 000</td>
<td>3 000</td>
<td></td>
</tr>
</tbody>
</table>

For the above example we can write the system of inequalities in the form.

\[
egin{align*}
1.5 V1 &+ 2.0 V3 &+ 2.5 V4 &\leq 1200 \\
2.0 V1 &+ 1.5 V2 &+ 2.0 V3 &\leq 1400 \\
- V1 &+ 0.5 V2 & &+ V4 &\leq 0 \\
- V2 &+ 0.5 V3 &+ 2.0 V4 &\leq 0 \\
300 V1 &+ 450 V2 &+ 700 V3 &+ 1500 V4 & = z \\
\end{align*}
\]

Way to start a system of linear inequalities is the subject of operations research and it will not be discussed here. It is essential for this case is that we want to maximize profit, which is given by utility function \( z \). Optimization can be done by the simplex method. It can be called by the command \texttt{linprog}. Because we are looking for maximum and \texttt{linprog} searches only minimum we need to transfer that role to deal with the minimum, objective function will have only negative signs: 
\[
f(x) = -300x_1 - 450x_2 - 700x_3 - 1500x_4.
\]

Equation restrictions remain unchanged. We create a new file M-file for the task solution. See the following list of Fig. 1. (The Mathworks MATLAB, 2008)

```matlab
f = [-300; -450; -700; -1500];
A = [1.5 0 2.5; 2 1.5 2 0; -1 0.5 0 1; 0 -1 0.5 2];
b = [1200; 1400; 0; 0];
lb = zeros(4,1);
options = optimset('LargeScale','off','Simplex','on');
[x,fval,exitflag,output,lambda] = linprog(f,A,b,[],[],lb,[],[],options);
x fval
```

**Fig. 1 Simplex Method Optimization** (Own processing)

Description of the commands is as follows. The first line defines a constant vector of objective function \( f \). The second line defines constants matrix inequalities and restrictive conditions. The
third line defines the right hand side vector inequalities of restrictive conditions. The next line defines a vector $lb$, left-side limitation i.e., the solution must not be less than zero. The command `optimset` sets variable options, where the parameter `LargeScale` must be turned off (off) for the simplex method. And the simplex method must be turned on (Simplex on). The optimization is running by the command `linprog` with the input and output parameters. Other lines invoke an extract of vector $x$ with the result values, value of the objective function `fval`. Nonzero values of variables `lambda.ineqlin` and `lambda.lower` inform us whether the restriction had an impact on the optimal process or not. After running the file `optimalizace.m` in Matlab we receive the final result, see the following listing in Fig. 2.

```
x =
  400
  400
   0
  200
fval =
-600000
ans =
   0
 428.5714
 557.1429
 428.5714
```

Fig. 2 Optimization Results (Own processing)

The maximum profit of CZK 600 000 appears in the event that we produce 400 pieces of product V1, 400 pieces of product V2, and 200 pieces of product V4 and product V3 will not be produced. The limiting values are listed which should have an impact on the calculation of the optimization.

2. Genetic Algorithms

Genetic algorithms are often used where the exact solution of practical problems through systematic exploration would last almost indefinitely. Genetic algorithms are based on genetic processes occurring in the nature where the evolutionary development or plant and animals breeding prefer certain desirable characteristics that are determined by combining parental chromosomes. Genetic algorithms began to emerge only in relatively recent times in the area of management or organization management (Dostál, 2002).

At the birth of genetic algorithms was the idea that in seeking better solutions to complex decision problems could be in a similar way to combine the existing solutions. Just as in genetics the term chromosome is used in the terminology of genetic algorithms. The chromosome is composed of sequentially arranged genes. Each gene controls the inheritance of one or more characters (Bäck, Hoffmeister and Schwefel, 1993). Most implementations of genetic algorithms work with binary representation of the chromosome, i.e., the original representation of the chromosome using zeros and ones so the chromosomes are binary strings that represent the majority of coded decimal numbers in applications optimized parameters of the function. For the handling of chromosomes has been proposed several genetic operators. Namely, the three most widely are used: selection, crossover, and mutation (Bäck and Schwefel, 1993).

Selection means the selection of chromosomes that become parents. When selecting at least one parent it is an important aspect so-called fitness. Basically, the principle of selection lies in the fact that stronger parental chromosome passes to the next generation. Crossing is an exchange of parts of two or more parent chromosomes, which causes modification of chromosomes, which occurs when one or more offspring is born (Goldberg, 1989).

Mutation then represents a modification of the chromosome involving a random change. However, this eventuality occurs in nature only rarely. Genetic algorithms then operate in a way that it first creates an initial population of m chromosomes and subsequently this population using genetic
operators change until the process is terminated. The reproduction process, which consists of three steps above, ie selection, crossover and mutation, is known as the epoch of the evolution of the population. The initial generation is usually obtained by random generation (Dostál, Rais and Sojka, 2005). With regard to population size, then it is logical that if the population is too small it can cause poor coverage area of the solution, while too large population increases calculation complexity. Experimental work suggests that in many cases is sufficient population size of about one hundred, respectively among n and 2n, where n is the length of a binary string. One possible change of an m-member population is to generate a new generation of m-offspring by cross-breeding. We can replace suddenly all the parental generation (Michalewicz and Fogel, 2000), (Storn and Price, 1997).

On contrary, there are other ways to enable overlapping generations of parents and offspring. For example, generated offspring does not substitute directly its parent but a randomly selected member of the current population. In solving optimization problems, the claim appears to be the best member of the current population appeared in the new population. It can be guaranteed for example by replacing an individual selected only from those who have below-average quality with the new chromosome. When applying genetic algorithms to problems of management and organization of production, respectively in strategic decision making, each chromosome encodes a solution to the problem. The chromosome there represents a genotype and properties (Montgomery and Keats, 1994).

Chromosomes with higher fitness or fitness value are preferred in genetic algorithms. Fitness function must be constructed so that its value is so higher that better is the value of the objective function. Practical applications of genetic algorithms usually consist of solving complex optimization problems. These are contained in the following chapter of this paper. Genetic algorithms are commonly used in practice to solve optimization problems, to search the best topology, technology and manufacturing and industrial automation and alternative learning methods such as neural networks (Wolpert and Macready, 1997). Unlike the gradient methods, which represent the search for local maxima or minima with one the solution, genetic algorithms are another approach that uses a population of interim solutions which pass through the parametric space in parallel and influence each other and modified by genetic operators. This is achieved by the fact that the population of individuals finds the right solution faster than if they searched the premises in isolation (Dostál, 2008).

3. Genetic Algorithms in Practice

3.1 Determining the Optimal Production

Genetic algorithms can be used to determine the optimal production. The demonstration shall include the following data. We should produce 12 products, 5 A-type products A, 3 B-type products, 2 C-type products, a one product of type D and of E. The production needs 6 different workplaces. Number of machines in the workplace is designed in Tab. 2 and the time for processing different types of products at different sites varies and is determined by cross-table (see Fig. 3). If we arbitrarily typed products, production would not be optimal (ie with the minimum production time). It is therefore appropriate to optimize order of entry product into production. You can select both the number of machines in each workspace and the processing time of different products at different workspaces.

<table>
<thead>
<tr>
<th>Number of working place</th>
<th>Number of Machines</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0.1</td>
<td>0.2</td>
<td>1.5</td>
<td>0.7</td>
<td>0.6</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>0.2</td>
<td>0.3</td>
<td>2.0</td>
<td>0.6</td>
<td>0.3</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>0.3</td>
<td>0.4</td>
<td>2.5</td>
<td>0.5</td>
<td>0.4</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>0.4</td>
<td>0.3</td>
<td>2.0</td>
<td>0.4</td>
<td>0.5</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>0.3</td>
<td>0.2</td>
<td>1.5</td>
<td>0.8</td>
<td>5.0</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>0.2</td>
<td>0.1</td>
<td>1.0</td>
<td>0.3</td>
<td>0.8</td>
</tr>
</tbody>
</table>
The solution itself can be viewed as calculating with the help of graphs and Best fitness and Score diversity (Fig. 4).

Fig. 3 Machines Ordering (Own processing)

Fig. 4 Graphs Best fitness and Score diversity Screenshot (Own processing)
The results of sorting device can be displayed by selecting File-Export to workspace menu and selecting other menu Export results to a MATLAB structure named (The Mathworks MATLAB, 2007).

Name optimresults can be left without change or it can be changed (Fig. 5). Then we click on OK. Result of components sort order can be received by calling garesults on the desktop and the command fval calls the value of the time production of all components after optimization. See the following commands listing Fig. 6.

```
optimresults =
x:[6 2 8 10 4 7 9 1 3 5]
fval: 12.0000
```

3.2 Application of Genetic Algorithms in Process Planning

This task was treated in the company Mitas Zlín and a part of it published as a scientific poster at the conference ROBUST2006 (Kasal, Klimek, Stříž and Říha, 2006) as a part of a larger project dealing with applications of statistical methods in the production planning process in the range of motorcycle tires of 117 sizes (Kasal, 2005).

This task was to maximize production under given conditions. It was first necessary to describe in detail the production hall. The analysis confirmed the fact that the amount of production is dependent solely on the power nonequivalence of different parts of the production line. Press machines have proved to be a weak spot. The production was dependent only on the use of 21 machines installed. Performance and capacity of other parts of the production line are not critical. They have their system optimization involving only a simple hand-highest decision-divisional staff on the basis of production and sales. The problem is therefore concerned only the maximum utilization of presses, and in one press (A), which produces nine dimensions of tires five presses (B) with 30 dimensions and 15 presses (C) with 78 dimensions. The presses are used in practice for approximately 95%. The goal was maximum utilization of presses within 14 days, after which planning is carried out. Methodology using genetic algorithms was chosen for this task due to the many independent variables. The task was solved with the help of Evolver software from Palisade who works as a supplement (Add-In) in MS Excel.

The solution will not be discussed in detail. It will be highlighted in particular the practical application of genetic algorithms. Data preparation procedure was the election of the objective function expressing the use of compression machines.

It was therefore necessary for each of the 117 assigned time stamping products including handling (column C in Fig. 7). Other input data were the number of individual machines and their assignment to the corresponding dimensions (column D), stocks (P) and the daily requirements (U) the amount for every dimension.
The principle of the solution was to determine the maximum number of units produced using the total time fund 22.5 hours, i.e. 3 work shifts per 7.5 hours. Since this is precisely the solution to the bottleneck (i.e. presses) which have actually been used for nearly 100% we consider other sections of the production. Further, the principle requires establish production of products that must be necessarily produced. This fulfills the column U (potřeba—need). Software Evolver offers column Q for determining the number of units produced for the selected product. We move on the application of genetic algorithms and run the program Evolver settings (Settings) (Fig. 8) and do the following steps.

We insert function we would like to minimize, maximize or close a particular value (we will maximize) in the cell “For the cell”. To do this, it is also necessary to identify the cells that will be modified by the algorithm to achieve the best combination of output and also establish conditions and limitations.

Fig. 7 Data Sorting in MS Excel (Kasal, Klimek, Stříž and Říha, 2006)

Fig. 8 Settings in Evolver (Kasal, Klimek, Stříž and Říha, 2006)
are inserted through the field "Subject to the Constraints". Column R (produce this product: yes/no) will contain only the binary values and column Q (size of production) which we denote as a column for editing we set the integers here but with the limits for example <0,500> and a method for calculation ("Solving method") in both cases "Recipe", see Fig. 9. Although the program offers other methods, it is appropriate for the task purposes.

Fig. 9 Further Options in Evolver (Kasal, Klímek, Stříž and Říha, 2006)

Mutation constant is set to a low value of 0.02, which is a very common value in practice and crossing threshold to 0.9, which determines the probability that the gene of chromosome will not change. Software is not limited to just the basic (default) algorithms for the process of selection-mutation-crossing, and provides the modified algorithms suitable for specific problems, see Fig. 10.

Fig. 10 Algorithm Options (Kasal, Klímek, Stříž and Říha, 2006)

It is therefore possible to use following procedures (algorithms): arithmetic crossover, heuristic crossover, Cauchy mutation, Boundary mutation, non-uniform mutation, linear and local search. If we choose all of these algorithms (which the program allows) the process selects the best possible solution procedure. The problem for one-day production was solved within two minutes even though the production was set only by 90 % and the algorithm was not given help in the form of determining the required production at the beginning of the solution. It was a very satisfying result.
4. Conclusion

Genetic algorithms are characterized by their robustness – the ability to solve difficult optimization or the tasks in which we need to decide on something. They can be characterized by properties such as multimodality and multicriteriality. Their deployment is effective in tasks that can be defined as follows:

- The solution space is too large and lack expertise that would enable the narrow space of promising solutions.
- It is unable to perform a mathematical analysis of the problem.
- Traditional methods fail.
- It is a task with multiple extremes, criteria and constraints.

Before using genetic algorithm, we should realize that the particular type of algorithm is suitable for solving only a certain range of problems. But genetic algorithms are not suitable for solving all problems. We should therefore not forget the variety of classical methods (method of direct selection, Newton method, etc.) which are able to solve many optimization problems with relatively low computational complexity. Summary of the disadvantages of genetic algorithms is listed below:

- The large time demand is typical for many tasks.
- You can not test whether it is the global optimum.
- Provide a solution too far from optimum for very large problems.

The genetic algorithms were chosen for this paper due to complexity of the manufacturing processes. Genetic algorithms this process significantly improved and they are perhaps the only possible tool for solving very complex problems with many variables. After calculating prediction by choosing the best mathematical or statistical method and after the analysis of the production process a model was designed in MS Excel that fits in the required direction the true nature of the production. Production efficiency can be interpreted as the number of units produced with maximum utilization of capacity, so there was chosen as the objective function equation which result determined the minimum required quantity of products. The data were transferred to a special software tool Evolver from Palisade. It is running as an add-in in MS Excel environment. The other problems included determination of suitable rank of recipes introduced into production. The task could be also solved by use of the genetic algorithm in Matlab.

Acknowledgements

This paper was supported by the IGA UTB Project No. IGA/73/FaME/10/D – Development of Use Mathematical and Statistical Methods in Quality Control.

References

Bezděk, V., 2011: Possible Use of Fuzzy Logic in Database, JSI 2(2), pp. 31–46
Dostál, P., 2002: Moderní metody ekonomických analýz. UTB, FaME
Dostál, P., 2008: Pokročilé metody analýz a modelování v podnikatelství a veřejné správě. Akademické nakladatelství CERM®, s.r.o.
Kovářík, M., 2008: Počítačové zpracování dat v programu MATLAB. Martin Stříž, Bučovice

**JEL Classification:** C61, L23