On Solving Unconditional Optimization Problems in the Wolfram Mathematica System*

Vladimir R. Kristalinskii [1][0000-0003-0644-1731]

[1] Smolensk State University, Smolensk, Russia
kristvr@rambler.ru

Abstract. The article is devoted to the method of solving problems of unconditional optimization in the Wolfram Mathematica system, which can be used when conducting classes in theoretical and applied disciplines in technical universities. Algorithms for solving mathematical optimization problems are very complex. The use of electronic computing technology makes it possible to implement these algorithms but often imposes very high requirements on the competence of the person who wants to implement these algorithms. The systems of computer mathematics that have appeared in recent decades allow us to solve this problem. They have an interface that allows a specialist in any application field to solve these problems without resorting to programming in the classical sense of the word. One such system is the Wolfram Mathematica system.

Keywords: Wolfram Mathematica System, Problems of Unconditional Optimization.

1 Introduction.

The simplest example of an optimization problem is the unconditional optimization problem. In this case, we look for the largest or smallest value of the function and the point at which this value is reached. The need to solve such problems often arises when solving a variety of technical tasks [2,3]. Interest in algorithms for solving optimization problems continues [4-10]. Computer mathematics systems in this case, as in other cases, allow you to focus on the mathematical model, and its implementation is carried out through the system.

The purpose of this article is to study the capabilities of the Wolfram Mathematica system for solving unconditional optimization problems.

The solution of such tasks can be carried out by the methods of differential calculus. The Wolfram Mathematica system allows you to solve such problems, including when the parameters are set as letter coefficients.

1 Solving Inventory Management Problems in the Mathematica System.

Consider the problem of managing a single-product stock [2]. Even if there is no shortage, the stock is replenished at a time. Total expenses per unit of time consist of specific operating and non-production expenses. For simplicity, assume that operating costs are by definition proportional to the level of average inventory

$$z_{\text{exp1}} = \frac{hx}{2},$$

where h is the operating costs per unit of time associated with storing a unit of inventory, and non – production costs are inversely proportional to the length of the order cycle

$$z_{\text{n}} = \frac{kd}{x},$$

where k is the total non – production costs. Then the specific total expenses are equal

$$z_A = \frac{hx}{2} + \frac{kd}{x}$$

where d is the intensity of demand.

The optimal stock level is found from the condition

$$z = f\left(x\right) \to \min$$

Finding the derivative of the function $z_A(x)$ using the Mathematica system

*u[x]=(h*x)/2+(k*d)/x*

u1[x]=D[u[x],x]

Receive result

h/2–(d k)/x^2

Equate the derivative to zero and solve the resulting equation with respect to x

Solve[u1[x]==0,x]

$$\{\{x\to-((\sqrt{2}\ \sqrt{d}\ \ \sqrt{k}\ )/\sqrt{h}\ )\},\{\ x\to((\sqrt{2}\ \sqrt{d}\ \ \sqrt{k}\ )/\sqrt{h}\ )\}\}$$

Choosing a positive one from the two obtained roots, we get

$$x = \sqrt{\frac{2kd}{h}}$$

Now let's assume that the stock is replenished by output with an intensity of production per unit of time r that exceeds the intensity of demand d(r>d). then the specific total costs are expressed by the formula.

$$z_B = \frac{hx\left(1-\frac{d}{r}\right)}{2} + \frac{kd}{x}.$$

Find the derivative of the function $z_B(x)$ using the Mathematica system.

*u2[x]=(h*x*(1–d/r))/2+(k*d)/x*

u3[x]=D[u2[x],x]

Receive result

1/2 h (1–d/r) – (d k)/x^2

Equate the derivative to zero and solve the resulting equation with respect to x

Solve[u3[x]==0,x]

$$\{\{x \to -((\sqrt{2}\ \sqrt{d}\ \ \sqrt{k}\ \sqrt{r}\ )/\sqrt{-dh+hr}\ ])\},\{\ x \to ((\sqrt{2}\ \sqrt{d}\ \ \sqrt{k}\ \sqrt{r}\ )/\sqrt{-dh+hr}\ ])\}$$

Choosing a positive one from the two obtained roots, we get

$$x = \frac{\sqrt{2kdr}}{h(r-d)}.$$

Let's assume that the required inventory level is no longer available and the inventory is replenished at the same time. In this case, the total costs additionally include the costs z_d due to the lack of the required inventory level, and the unit total costs are determined by the expression

$$z_B = \frac{h(x-v)^2 + bv^2}{2x} + \frac{kd}{x}$$

where v is the value of the deficit, and b is the specific losses from the deficit per unit of time.

We find partial derivatives of function z(x,v) with respect to x and v using the Wolfram Mathematica system.

gg[x,v]=(h(x–v)^2+b*v^2)/(2*x)+(k*d)/x*

gg1[x,v]=D[gg[x,v],x]

gg2[x,v]=D[gg[x,v],v]

We equate the obtained derivatives to zero and solve the resulting system of equations for x and v

```
Solve[{gg1[x,v]==0,gg2[x,v]==0},{x,v}]
```

Let's simplify the result and get the answer.

```
FullSimplify[%]
```

```
{{x→-((Sqrt[2] Sqrt[d] Sqrt[b (b+h)] Sqrt[k])/(b Sqrt[h])),v→-((Sqrt[2] Sqrt[d]
Sqrt[h] Sqrt[k])/Sqrt[b (b+h)])},{x→(Sqrt[2] Sqrt[d] Sqrt[b (b+h)] Sqrt[k])/(b
Sqrt[h]),v→(Sqrt[2] Sqrt[d] Sqrt[h] Sqrt[k])/Sqrt[b (b+h)]}}
```

Choosing from the two obtained values x and v positive, we get

$$x=\frac{\sqrt{2dkb(b+h)}}{b\sqrt{h}},\ v=\sqrt{\frac{2kdh}{b(b+h)}}.$$

Now let's consider the case of a deficit with a uniform replenishment of the stock. In this case, the unit total expenses are defined by the expression

$$z_{\Gamma}=\frac{h\left(x\left(1-\frac{d}{r}\right)-v\right)^{2}+bv^{2}}{2x\left(1-\frac{d}{r}\right)}+\frac{kd}{x}.$$

We find partial derivatives of function z(x,v) for x and v using the Wolfram Mathematica system.

```
f[x,v]=(h*(x*(1–d/r)–v)^2+b*v^2)/(2*x*(1–d/r))+k*d/x
ff1[x,v]=D[f[x,v],x]
ff2[x,v]=D[f[x,v],v]
```

We equate the obtained derivatives to zero and solve the resulting system of equations for x and v

```
Solve[{ff1[x,v]==0,ff2[x,v]==0},{x,v}].
```

Let's simplify the result and get the answer.

```
FullSimplify[%]
```

```
{{x→-((I Sqrt[2] Sqrt[d] Sqrt[b+h] Sqrt[k] Sqrt[r])/Sqrt[b h (d–r)]),v→(I Sqrt[2]
Sqrt[d] Sqrt[k] Sqrt[b h (d–r)])/(b Sqrt[b+h] Sqrt[r])},{x→(I Sqrt[2] Sqrt[d]
Sqrt[b+h] Sqrt[k] Sqrt[r])/Sqrt[b h (d–r)],v→-((I Sqrt[2] Sqrt[d] Sqrt[k] Sqrt[b h
(d–r)])/(b Sqrt[b+h] Sqrt[r]))}}
```

The resulting result may be somewhat confusing, since the resulting expressions contain an imaginary unit (I). However, remembering that by condition r>d , hence

$$\sqrt{d-r}=i\sqrt{r-d}\ .$$

Multiplying by i, we get

$$x=\sqrt{\frac{2dkr(b+h)}{bh(r-d)}},\quad v=\sqrt{\frac{2dkh(r-d)}{r(b+h)}}$$

These examples illustrate the ability of the Wolfram Mathematica system to work with expressions in a General way. The latter case also shows that the results obtained with its help may require additional study and analysis.

2 New Tools for Solving Optimization Problems in the Latest Versions of Mathematica.

The Mathematica system allows you to solve such problems directly using built-in methods. The main methods for solving such problems using Wolfram Mathematica are widely known and discussed, for example, in [1]. However, the system is evolving and new features are emerging. For example, in version 11 of the system, the BayesianMinimization command was added. This command makes it possible to solve several new minimization problems along with well-known ones.

You can use this command to solve the usual problem of minimizing a function on an interval.

bo=BayesianMinimization[(#–2)^2+ 1&,Interval[{0,7}]]

In this example, we minimize the function

$$f(x)=(x-2)^2+1$$

on the interval [0;7]. Receive result

BayesianMinimizationObject[Minimum configuration: 2.03
Minimum value: 1.]
Let's present the results of calculations in a more convenient form
bo["MinimumConfiguration"]
2.03379
bo["MinimumValue"]
1.01162

Let's build graphs of the simulated and modeling functions (see Fig. 1).

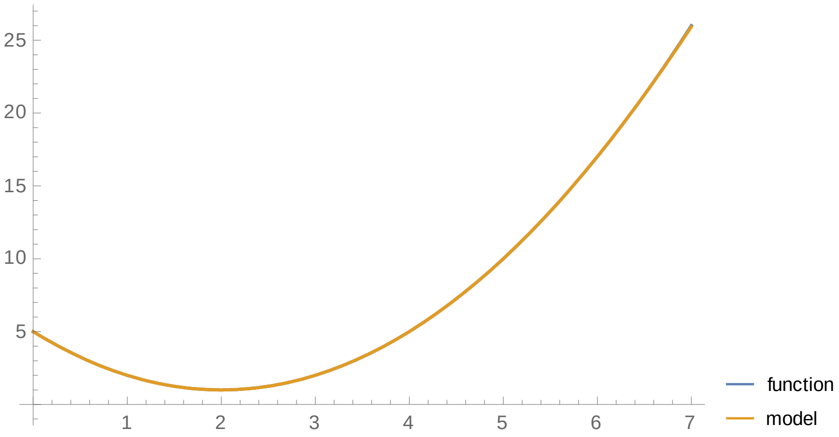

Fig. 1. The schedule is built with Mathematica.

Here we see that it is better to use the usual Minimize function to solve this problem. However, the BayesianMinimization team has features that previously existing Teams did not have. We can look at the set of values that were used to calculate the minimum

```
bo["EvaluationHistory"]
Configuration               Value
6.30544                     19.5368
3.91381                     4.66267
3.13817                     2.29542
1.07336                     1.85866
2.16175                     1.02616
1.86308                     1.01875
1.88728                     1.01271
0.0977517    4.61855
6.82761                     24.3058
3.04529                     2.09263
2.05689                     1.00324
4.27194                     6.16173
4.80517                     8.86897
3.09127                     2.19086
0.924357    2.15701
```

We can also perform function minimization on a discrete set.

```
bo=BayesianMinimization[(#^2-5*#+2)&,{5.7,1.1,3.4,6.8,2.3,1.2}]
bo["MinimumConfiguration"]
2.3
bo["MinimumValue"]
–4.21
```

We can minimize the function on a set constructed using a random number generator

```
bo1=BayesianMinimization[ (#^2-5*#+4)&,RandomReal[{0,10}]&]
bo1["MinimumConfiguration"]
```

```
2.54493
 bo1["MinimumValue"]
–2.24796
```

All these new features allow us to expand the range of problems that can be considered with students when studying problems whose mathematical models are reduced to problems of unconditional optimization.

4 Conclusion.

Along with the examples discussed in this article, the system allows you to solve a wide range of different optimization problems: linear programming problems (in particular, the transport problem), the assignment problem, and dynamic programming problems. Methods for solving multi-criteria optimization problems using the Mathematica system are developed.

The Mathematica system also allows you to solve a large number of problems related to the most diverse areas of mathematics and applied disciplines. In the latest versions of the system, it includes some artificial intelligence algorithms (in particular, image recognition). All this makes it an indispensable tool for teaching students of mathematical and engineering specialties, as well as for conducting scientific research.

References

1. Kristalinsky, V. R., Cherny, S. N. Solving Problems of Dynamic Programming in the Wolfram Mathematica System. International journal of open information technologies. Vol. 7. - no. 2. (2019). (In Russian)
2. Kristalinsky, V. R., Kristalinsky, R. E.: Solution of the Dirichlet Problem for Multi-Connected Domains. Modern Information Technologies and IT education. pp. 220-226. (2019). (In Russian)
3. Kristalinsky, V. R., Kristalinsky, R. E.: On the Use of the Wolfram Mathematica System in Statistical Data Analysis. Distance Educational Technologies. pp. 58-61. (2019). (In Russian)
4. Rao, R. Jaya: A Simple and New Optimization Algorithm for Solving Constrained and Unconstrained Optimization Problems. International Journal of Industrial Engineering Computations. V. 7. №. 1. PP. 19-34. DOI: 10.5267/j.ijiec.2015.8.004 (2016).
5. Tejani, G. G. et al.: An Improved Heat Transfer Search Algorithm for Unconstrained Optimization Problems. Journal of Computational Design and Engineering. V. 6. №. 1. PP. 13-32. DOI:10.1016/j.jcde.2018.04.003 (2019).
6. Yuan, G., Hu, W.: A Conjugate Gradient Algorithm for Large-Scale Unconstrained Optimization Problems and Nonlinear Equations. Journal of Inequalities and Applications. V. 2018. №. 1. PP. 113. DOI:10.1186/s13660-018-1703-1. (2018).
7. Yuan, G., Wang, X., Sheng, Z.: The Projection Technique for Two Open Problems of Unconstrained Optimization Problems. Journal of Optimization Theory and Applications. V. 186. №. 2. PP. 590-619. DOI:10.1007/s10957-020-01710-0. (2020).
8. Zang W. et al. A genetic algorithm using triplet nucleotide encoding and DNA reproduction operations for unconstrained optimization problems. Algorithms. V. 10. №. 3. PP. 76. DOI:10.3390/a10030076. (2017).

9. Ghazali, K. et al.: Newton-SOR Iteration for Solving Large-Scale Unconstrained Optimization Problems with an Arrowhead Hessian Matrices. Journal of Physics: Conference Series. IOP Publishing. V. 1358. №. 1. PP. 012054. (2019)
10. Ghazali, K. et al.: Newton-SOR Iteration for Solving Large-Scale Unconstrained Optimization Problems with an Arrowhead Hessian Matrices. Journal of Physics: Conference Series. IOP Publishing, 2019. V. 1358. №. 1. PP. 012054. DOI:10.1088/1742-6596/1358/1/012054