

**Antoni Drapella**  
**Akademia Marynarki Wojennej**

---

**Małgorzata Górską**  
**Akademia Pomorska w Słupsku**

## **SOLVER II ALTERNATYWĄ NARZĘDZIA SOLVER W EXCELU**

### **STRESZCZENIE**

Artykuł dotyczy zadania znalezienia maksimum globalnego otoczonego licznymi maksimumami lokalnymi na drodze numerycznej. Przedstawiono wyniki nieudanych prób wykonania tego zadania za pomocą, skądinąd bardzo skutecznego, narzędzia, jakim jest wbudowany w Excela *Solver*. Przedstawiono algorytm narzędzia o nazwie *Solver II* oraz jego implementację w środowisku programistycznym VBA. Pokazano, że *Solver II* znakomicie radzi sobie z realizacją postawionego zadania.

#### Słowa kluczowe:

metody optymalizacyjne, znajdowanie maksimum globalnego, Excel, narzędzie *Solver*.

### **WSTĘP**

W dodatkach do środowiska obliczeniowego Excel znajdujemy narzędzie o nazwie *Solver* służące do rozwiązywania zagadnień optymalizacyjnych. Praktyka pokazuje, że jest to narzędzie skuteczne i wydajne, jeśli tylko funkcja celu ma jedno maksimum. Jeśli jednak funkcja celu ma maksimum globalne otoczone maksimumami lokalnymi, to *Solver* znajdzie najbliższe podanemu punktowi startowemu maksimum lokalne. W niniejszej pracy proponowane jest narzędzie *Solver II*, które takiej wady nie ma. Użytkownik narzędzia *Solver* ma do wyboru dwa algorytmy: Newtona i gradientowy. Narzędzie *Solver II* jest implementacją algorytmu, który „uczy się” kształtu powierzchni określonej przez funkcję celu. Nauczysz się, zacieśnia obszar przeszukiwania wokół domniemanego ekstremum globalnego. Zbiorem uczącym są losowo wybrane z dziedziny funkcji punkty oraz wartości funkcji celu w tych punktach. Algorytm *Solvera II* jest prosty i przy swej prostocie zadziwiająco efektywny. Jego ideę zaczerpnięto z trudno teraz dostępnej literatury rosyjskojęzycznej [1, 2]. Algorytm opisany w przywołanej literaturze i zastosowany w *Solverze II* różnią się,

w sposób zasadniczy, sposobem zacieśniania obszaru przeszukiwania wokół domniemanego ekstremum globalnego. Jego implementacje w VBA przedstawiono w sekcji 3.

### FUNKCJA TESTOWA

Na potrzeby niniejszej pracy skonstruowano następującą funkcję testową:

$$F_1(x_1, x_2) = \sum_{i=1}^5 C_i \cdot \exp \left[ - \left( \frac{x_1 - a_{i1}}{b_{i1}} \right)^2 - \left( \frac{x_2 - a_{i2}}{b_{i2}} \right)^2 \right]. \quad (1)$$

Funkcja testowa ma, jak wynika z (1), pięć składowych. Każda ze składowych ma pięć parametrów. Omówienie parametrów zawarto w tabeli 1. Indeks  $i$ ,  $i = 1, 2, \dots, 5$ , jest zarówno indeksem składowej funkcji celu, jak i indeksem ekstremum. Wartości parametrów podano w tabeli 2.

Funkcja ma maksimum globalne i cztery maksima lokalne. Parametry funkcji dobrano tak, by maksima leżały blisko siebie oraz by ich wartości były zbliżone, a przez to lokalizacja ekstremum globalnego była trudna. Maksima lokalne mają indeksy od 1 do 4 zgodnie z numeracją ćwiartek układu współrzędnych, w których zostały umieszczone. Maksimum globalne umieszczone w środku układu współrzędnych ma numer 5.

Funkcję testową zaprezentowano na rysunku 1. (warianty A i B — różnią się sposobem jej prezentacji).

Tabela 1. Parametry funkcji celu

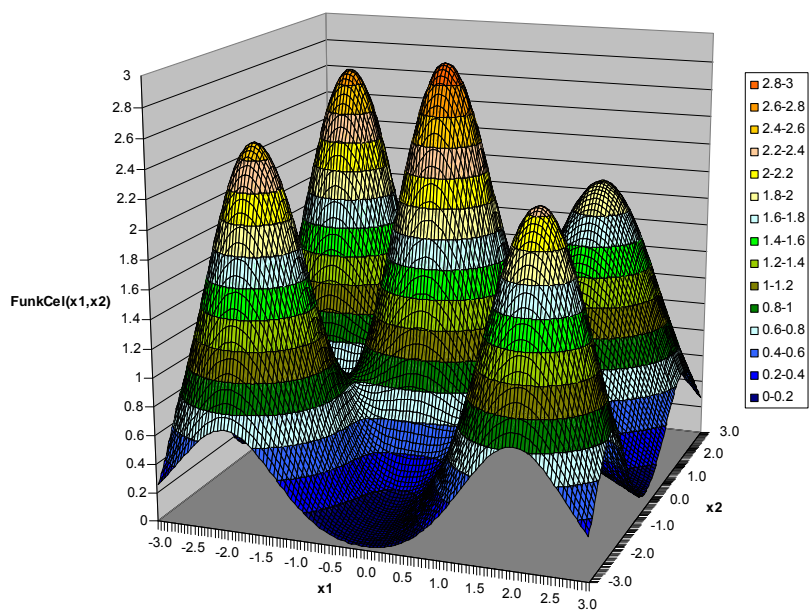
Symbol	Opis
$C_i$	Amplituda
$a_{i1}$	Parametr położenia na osi $x_1$
$a_{i2}$	Parametr położenia na osi $x_2$
$b_{i1}$	Parametr skali wzdłuż osi $x_1$
$b_{i2}$	Parametr skali wzdłuż osi $x_2$

Źródło: opracowanie własne.

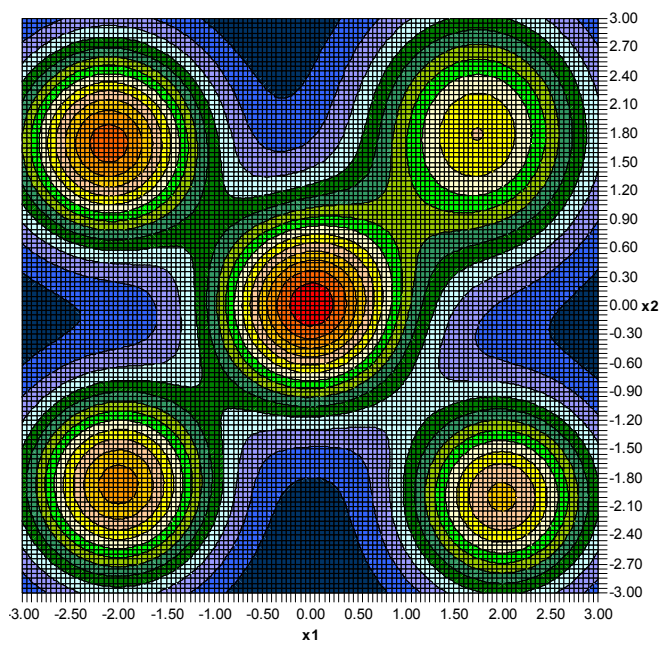
Tabela 2. Wartości liczbowe parametrów funkcji celu

Indeks $i$	$C_i$	$a_{i1}$	$b_{i1}$	$a_{i2}$	$b_{i2}$
1	2,0	1,8	1,8	1,3	1,2
2	2,3	-2,0	2,0	1,0	1,0
3	2,4	-1,9	-2,0	1,0	1,0
4	2,0	1,7	-2,1	1,0	1,0
5	2,9	0,0	0,0	1,0	1,0

Źródło: opracowanie własne.



Wariant A



Wariant B

Rys. 1. Funkcja testowa

Źródło: opracowanie własne.

### EKSPERYMENT NUMERYCZNY Z *SOLVEREM*

Aby poznać działanie narzędzia *Solver*, gdy funkcja celu ma ekstrema lokalne, przeprowadzono następujący eksperyment numeryczny. Posługując się generatorem liczb losowych o rozkładzie równomiernym, wylosowano dziesięć punktów w kwadracie o bokach długich na sześć jednostek i mającym środek umieszczony w środku układu współrzędnych. Rysunek 1. uzasadnia wybór obszaru poszukiwania ekstremum. Współrzędne wylosowanych punktów podano w tabeli 3. Były to punkty startowe dla *Solvera*. W skrajnej prawej kolumnie podano numer ekstremum znalezione przez *Solver*.

Tabela 3. Wyniki eksperymentu numerycznego z *Solverem*

Indeks $j$	Punkt startowy		Numer wskazanego maksimum
	Współrzędne		
	$x_{1j}$	$x_{2j}$	
1	-1.45964	1.61076	2
2	1.591202	-1.54034	4
3	-0.44727	0.665199	5
4	0.18789	-1.46466	5
5	-1.16524	-0.55079	5
6	-2.05665	-1.04636	4
7	2.95178	-1.29976	4
8	-1.6282	0.686285	2
9	2.730256	1.422714	1
10	-1.16169	-1.01373	3

Źródło: opracowanie własne.

Działanie narzędzia *Solver* było zatem niezadowolające. To, czy przygotowując *Solvera* do pracy, wybrano gradient sprzężony, czy metodę Newtona, nie miało wpływu na uzyskany wynik.

### PSEUDOKOD *SOLVERA II*

Pseudokod jest słownym zapisem algorytmu ułatwiającym czytanie kodu właściwego. Pseudokody powszechnie wyparły używane dawniej sieci działań (*flow-charty*).

Niech  $k$  jest liczbą argumentów funkcji celu (f.c.),  $Nn$  liczbą losowych próbek wartości funkcji celu, a  $Mm$  liczbą próbek uwzględnianych, by wyznaczyć nowy przedział próbkowania. W toku kolejnych powtórzeń powyższego algorytmu obszar przeszukiwania zaciska się wokół maksimum globalnego.

**Procedura Solver\_II()**

Wczytaj  $k$ ,  $Nn$ ,  $Mm$

Dla  $i = 1$  do  $k$

Wczytaj dolną i górną granicę przedziału przeszukiwania wzdłuż osi każdego z  $k$  argumentów f.c.

Wylicz szerokości przedziału przeszukiwania wzdłuż  $i$ -tej osi.

Następnie  $i$

Dla  $j = 1$  do  $Nn$

Dla  $i = 1$  to  $k$

Wygeneruj realizację zmiennej losowej o rozkładzie równomiernym w przedziale  $\langle 0,1 \rangle$ .

Utwórz z niej liczbę losową o rozkładzie równomiernym w przedziale przeszukiwania wzdłuż  $i$ -tej osi.

Uzyskane w ten sposób współrzędne punktu, w którym próbkowana będzie funkcja celu, zapisz do  $i$ -tego wiersza macierzy wyników  $MW0$ .

Next  $i$

Wylicz wartość f.c. w punkcie próbkowania na podstawie uzyskanych wcześniej próbkowanych wartości jej argumentów.

Zapisz wartość f.c. do ostatniej kolumny macierzy wyników  $MW0$ .

Next  $j$

Uporządkuj  $MW0$  wierszami w kolejności wzrastania wartości jej ostatniej kolumny.

Do  $MW1$  przepisuj  $Mm$  ostatnich wierszy macierzy  $MW0$ .

W kolumnach macierzy  $MW1$  znajdź najmniejszą i największą wartość próbkową każdego argumentu funkcji celu.

Te wartości determinują następny przedział próbkowania.

Znalezione wartości prześlij do wskazanego bloku komórek arkusza Excela

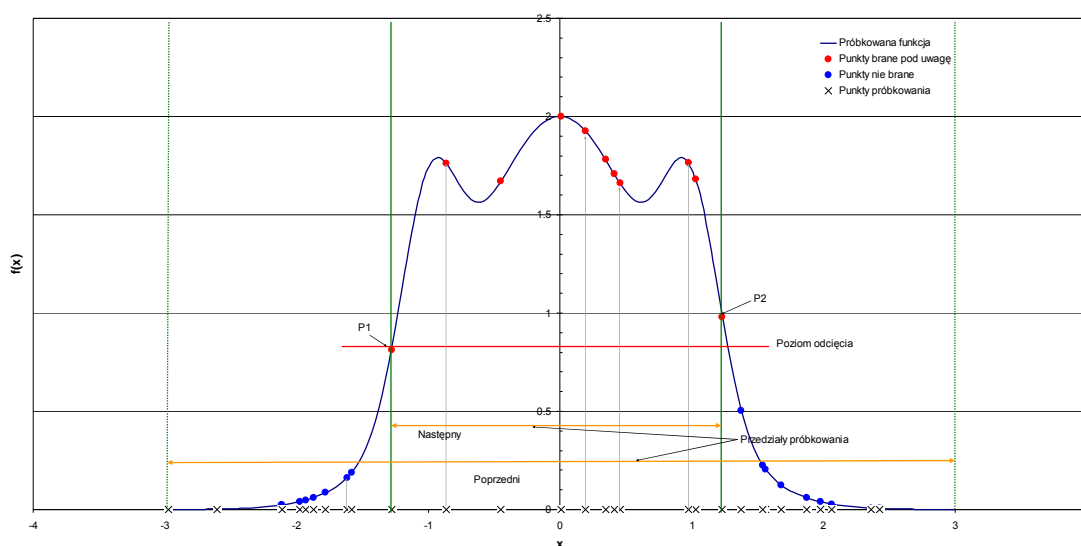
**Koniec procedury**

## ILUSTRACJA DZIAŁANIA SOLVERA II

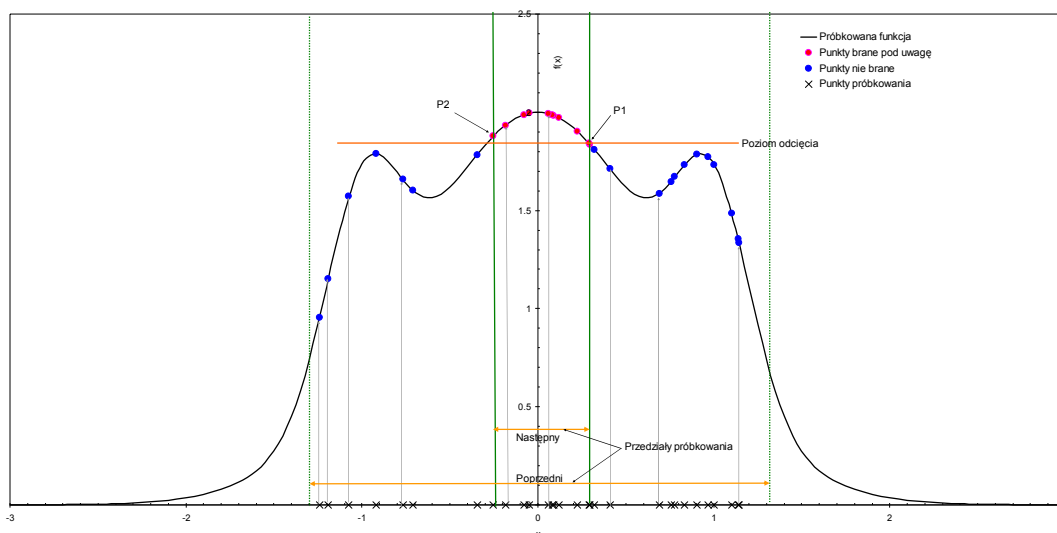
Rysunek 2. (warianty A oraz B) służy wyjaśnieniu, jak działa Solver II. W celu uproszczenia prezentacji w niniejszej sekcji artykułu poszukiwane będzie ekstremum globalne funkcji jednej zmiennej.

W wariancie A krzyżyki na osi  $x$  wskazują losowo wybrane wartości argumentu funkcji celu dalej nazywane punktami próbkowymi (PP). Jest ich trzydzieści. Wartości PP są zmiennymi losowymi podlegającymi rozkładowi równomiernemu w przedziale próbkowania. Pionowe szare strzałki naprowadzają na kropki leżące na wykresie funkcji celu. Rzędne tych punktów to wartości próbkowe (WP) funkcji celu. Dla przejrzystości rysunku strzałki naprowadzające umieszczono tylko przy niektórych PP. Spośród trzydziestu PP w dalszym toku algorytmu uwzględnia się jedenaście o największych WP. Uwzględnione PP i nieuwzględnione PP rozróżniono kolorami. Oczywiście w praktyce ogólna liczba PP idzie w tysiące, a uwzględnianych PP w setki. Dwa skrajne spośród uwzględnionych PP, oznaczone na rysunku P1 i P2, wyznaczają nowy przedział próbkowania.

Konstrukcja wariantu B jest identyczna jak A. Jednak to, co w wersji A było następnym przedziałem próbkowania, w przykładzie B jest już poprzednim przedziałem próbkowania. Następny przedział próbkowania został utworzony tak samo, jak wcześniej opisano. Niektóre z jedenastu punktów próbkowych leżą bardzo blisko siebie, stąd wydaje się, że jest ich tylko dziewięć. Warto zauważyć, jak bardzo zawężił się przedział próbkowania. Teraz leży w nim tylko maksimum globalne!



Wariant A



Wariant B

Rys. 2. Prezentacja działania Solvera II

Źródło: opracowanie własne.

## IMPLEMENTACJA SOLVERA II IMPLEMENTOWANA W VISUAL BASIC FOR APPLICATIONS (VBA)

Solver II ma postać procedury funkcyjnej, inaczej mówiąc funkcji użytkownika. Można z niej korzystać na dwa sposoby:

- wstawiać ją do arkusza wielokrotnie do uzyskania pożądanego dokładności;
- wywoływać ją z poziomu VBA w pętli z procedury zwykłej (Typu Sub) z wbudowanym stosownym warunkiem końca pętli.

Bardzo ważną cechą omawianej funkcji jest to, że po każdym jej wywołaniu generator liczb losowych równomiernych startuje z innego „ziarna”. Jest ono tworzone na podstawie czasu pobranego z zegara systemowego. Ciąg liczb losowych jest za każdym razem inny. Tak więc, przy każdym wywołaniu Solvera II funkcja celu jest próbkowana w innych punktach. W znakomity sposób podnosi to efektywność działania omawianej procedury.

Opisywana procedura ma wiele parametrów formalnych. Dzielą się one na:

- parametry próbkowania;
- parametry funkcji celu.

Parametry próbkowania to zakresy wartości argumentów funkcji celu, w których ma być ona próbkowana — liczba  $Nn$  próbek oraz liczba  $Mm$  próbek, na podstawie których zostanie wyznaczony nowy obszar próbkowania.

Aby ułatwić wprowadzanie parametrów skomplikowanych funkcji celu wskazuje się cały blok komórek z wartościami parametrów. Wartości parametrów są zapisywane do dynamicznej, kolumnowej tablicy typu variant. W toku realizacji procedury wartości parametrom są z tej tablicy kolejno wydobywane i po konwersji typu zmiennej na single przypisywane zmiennym, które są zmiennymi lokalnymi w procedurze. Kod *Solvera II* ma następującą postać:

**Function Solver\_II (x1d As Single, x1g As Single, x2d As Single, x2g As Single, Nn As Integer, Mm As Integer, ParamArray Parameters())**

*'Parametry formalne procedury funkcyjnej.*

*'x1d dolna granica poprzedniego przedziału przeszukiwania zmiennej x1.*

*'x1g górna granica poprzedniego przedziału przeszukiwania zmiennej x1.*

*'x2d dolna granica poprzedniego przedziału przeszukiwania zmiennej x2.*

*'x2g górna granica poprzedniego przedziału przeszukiwania zmiennej x2.*

*'Nn liczba próbek wartości funkcji celu.*

*'Mm liczba próbek brana pod uwagę przy wyznaczaniu następnego przedziału przeszukiwania.*

*'Parameters tablica parametrów funkcji celu.*

*'Deklaracje zmiennych.*

*'Zmienne służące wydobywaniu poszczególnych parametrów funkcji celu z macierzy Parameters.*

*Dim Part, Value*

*Dim i As Integer, j As Integer, k As Integer 'Liczniki pętli.*

*Dim n As Integer 'Liczba parametrów funkcji celu.*

*Dim S As Single 'Sumator wartości składników funkcji celu.*

*Dim Z1 As Single, Z2 As Single 'Zmienne pomocnicze przy liczeniu wartości funkcji celu.*

*'Macierze kolumnowe zawierające wartości poszczególnych parametrów funkcji celu.*

*Dim a1(5) As Single, b1(5) As Single, C(5) As Single, a2(5) As Single, b2(5) As Single*

*'Macierz dynamiczna tymczasowo przechowująca parametry funkcji celu w toku wydobywania ich 'z macierzy Parameters.*



*Dim Container() As Single*

*Dim x1 As Single, x2 As Single: 'Zmienne zawierające bieżące wartości funkcji celu.*

*'Szerokości przedziałów przeszukiwania używane przy generowania losowych wartości argumentów funkcji celu w celu próbkowania funkcji celu.*

*Dim d1 As Single, d2 As Single:*

*Dim min\_x1 As Single, min\_x2 As Single, max\_x1 As Single, max\_x2 As Single*

*Dim TWS() As Single 'Tablica dynamiczna wyników próbkowania funkcji celu.*

*'Zmienne pomocnicze w procedurze porządkowania macierzy TWS według rosnących wartości trzeciej 'kolumny, czyli wartości próbkowych funkcji celu.*

*Dim Bocz\_1 As Single, Bocz\_2 As Single, Bocz\_3 As Single, Hor As Integer*

'-----  
-----

*'Rozpakowywanie tablicy Parameters w celu wydobycia z niej parametrów funkcji celu.*

*'Określanie liczby parametrów funkcji celu.*

*Let n = 0*

*For Each Part In Parameters*

*For Each Value In Part*

*Let n = n + 1*

*Next Value*

*Next Part*

*ReDim Container(n): 'Ustalenie wymiaru macierzy Container.*

*'Ustalenie wymiaru macierzy TWS mającej zawierać*

*'w pierwszej i drugiej kolumnie losowe wartości argumentów funkcji celu*

*'a w trzeciej kolumnie odpowiadające tym argumentom wartości funkcji celu w punktach próbkowania.*

*ReDim TWS(Nn, 3)*

*'Wydobywanie parametrów funkcji celu i zapis do macierzy Container.*

*Let i = 0*

*For Each Part In Parameters*

*For Each Value In Part*

*Let i = i + 1*

*Let Container(i) = Value*

*Next Value*

*Next Part*

*'Zapisywanie wartości parametrów funkcji celu do macierzy kolumnowych odpowiadających typom parametrów funkcji celu.*

*For k = 1 To 5*

*Let C(k) = Container(k): 'Zapis amplitud składowych funkcji celu.*

*Next k*

*For k = 1 To 5*

*Let a1(k) = Container(5 + k): 'Zapis wartości parametrów położenia związanych z argumentem x1.*

*Next k*

*For k = 1 To 5*

*Let a2(k) = Container(10 + k): 'Zapis wartości parametrów położenia związanych z argumentem x2.*

*Next k*

*For k = 1 To 5*

*Let b1(k) = Container(15 + k): 'Zapis wartości parametrów skali związanych z argumentem x1.*

*Next k*

*For k = 1 To 5*

*Let b2(k) = Container(20 + k): 'Zapis wartości parametrów skali związanych z argumentem x2.*

*Next k*

*Let d1 = x1g - x1d: 'Wyliczenie szerokości przedziałów*

*Let d2 = x2g - x2d: przeszukiwania d1 i d2.*

*Randomize Timer: 'Pobranie ziarna do generatora liczb losowych z zegara systemowego.*

*For i = 1 To Nn*

*Let x1 = d1 \* Rnd(1) + x1d: x2 = d2 \* Rnd(1) + x2d: 'Wygenerowanie losowych wartości argumentów funkcji celu.*

*Let S = 0: 'Wyzerowanie sumatora składowych funkcji celu.*

*For k = 1 To 5: 'Pętla po składowych funkcji celu.*

*Z1 = (x1 - a1(k)) / b1(k): Z2 = (x2 - a2(k)) / b2(k): 'Obliczenie wielkości wyrażeń w wykładniku eksponenty funkcji celu.*

*S = S + C(k) \* Exp(-Z1 \* Z1 - Z2 \* Z2): 'Obliczenie wartości kolejnej składowej funkcji celu.*

*Next k*

*Let TWS(i, 1) = x1: Let TWS(i, 2) = x2: Let TWS(i, 3) = S: 'Zapis wyniku i-tego próbkowania funkcji celu.*

*Next i*

*'Porządkowanie macierzy wyników próbkowania funkcji celu według trzeciej kolumny, to jest wartości funkcji celu.*

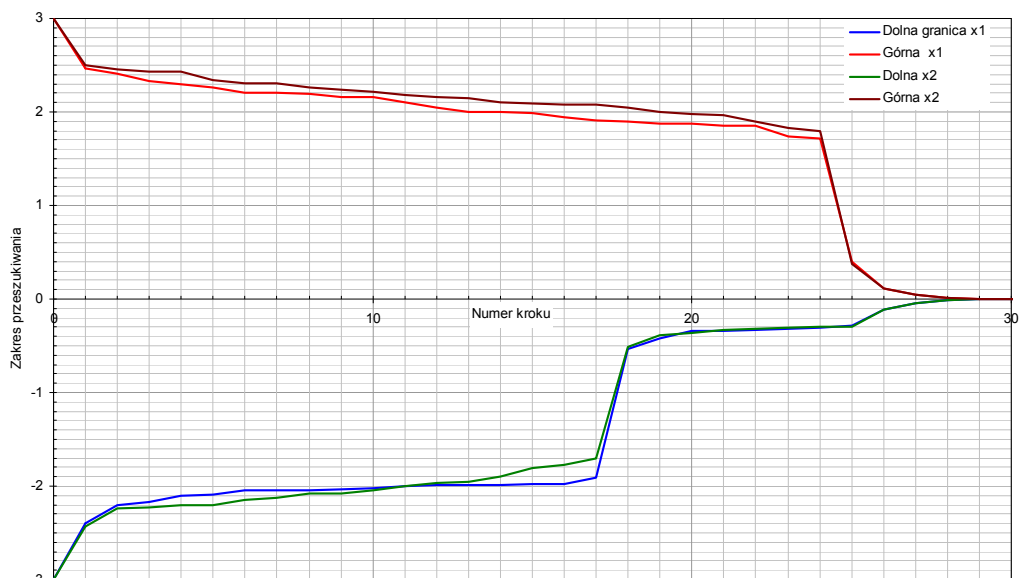
```

Powrot1:
Let Hor = 0
For i = 1 To Nn - 1
  If TWS(i, 3) <= TWS(i + 1, 3) Then GoTo Dalej2
  Let Bocz_1 = TWS(i, 1): Bocz_2 = TWS(i, 2): Bocz_3 = TWS(i, 3)
  Let TWS(i, 1) = TWS(i + 1, 1): TWS(i, 2) = TWS(i + 1, 2): TWS(i, 3) = TWS(i + 1, 3)
  Let TWS(i + 1, 1) = Bocz_1: TWS(i + 1, 2) = Bocz_2: TWS(i + 1, 3) = Bocz_3:
  Let Hor = 1
Dalej2:
Next i
If Hor = 1 Then GoTo Powrot1
'Ustalanie dolnych i górnych granic nowego zakresu nowego przeszukiwania.
Let min_x1 = 10 ^ 6: Let min_x2 = 10 ^ 6: max_x1 = 10 ^ (-6): max_x2 = 10 ^ (-6):
'Ustalenie stosownych wartości wyjściowych.
For i = Nn - 100 To Nn
  If TWS(i, 1) < min_x1 Then min_x1 = TWS(i, 1)
  If TWS(i, 2) < min_x2 Then min_x2 = TWS(i, 2)
  If TWS(i, 1) > max_x1 Then max_x1 = TWS(i, 1)
  If TWS(i, 2) > max_x2 Then max_x2 = TWS(i, 2)
Next i
'Przesłanie do arkusza wyniku działania procedury w postaci macierzy wierszowej
'zawierającej granice następnego przedziału przeszukiwania.
Let NextRange = Array(min_x1, max_x1, min_x2, max_x2)
End Function

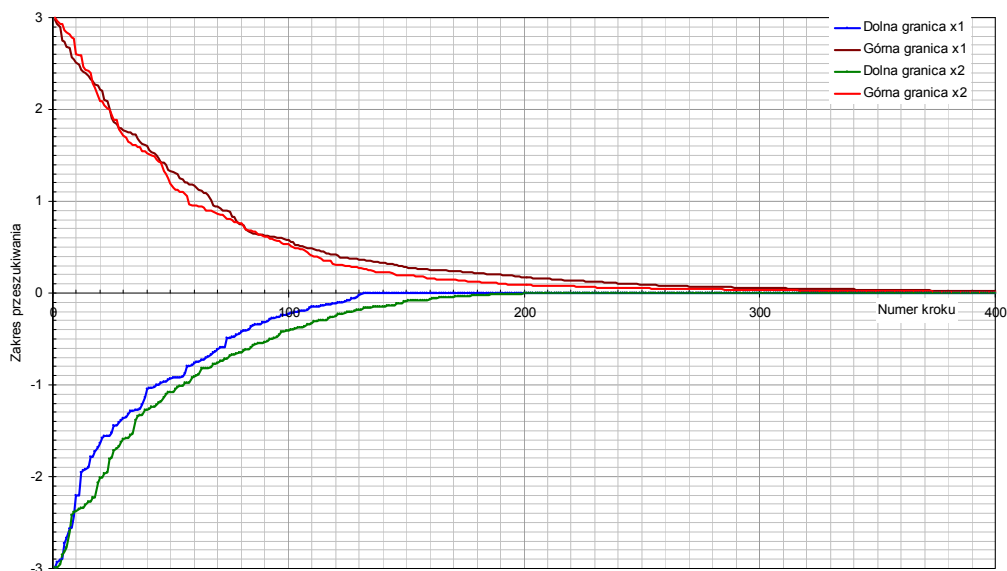
```

## EKSPERYMENT NUMERYCZNY Z SOLVEREM II

W każdym kroku funkcja celu była próbkowana w  $Nn = 1000$  punktów. Do wyznaczania następnego przedziału próbkowania brano  $Mm = 100$  punktów. Na rysunku 3. w wariancie A pokazano dolne i górne granice przedziałów próbkowania wartości argumentów  $x_1$  i  $x_2$  uzyskiwane w kolejnych krokach. Ekstremum globalne zostało zlokalizowane w dwudziestym czwartym kroku, a precyzyjnie określone w dwudziestym siódmym. Solver II jest zatem godny polecenia. Na próbkowaniu nie należy oszczędzać. W wariancie B pokazano proces lokalizacji ekstremum globalnego tej samej funkcji celu, gdy  $Nn = 100$ , a  $Mm = 10$ . Proces zaciskania obszaru przeszukiwania wokół maksimum globalnego jest znacznie wolniejszy. Należy też pamiętać, że gdy liczba próbek w jednym kroku iteracji maleje, rośnie prawdopodobieństwo zgubienia maksimum globalnego.



Wariant A



Wariant B

Rys. 3. Próbkowanie funkcji celu

Źródło: opracowanie własne.

## PODSUMOWANIE

Uzyskane za pomocą *Solvera II* wyniki są zachęcające. *Solver II* dość skutecznie znajduje ekstremum globalne otoczone przez liczne i zbliżone mu wartościami ekstrema lokalne.

Ekstremum globalne funkcji testowej zostało zlokalizowane w iteracji dwudziestej czwartej, a precyzyjnie określone w dwudziestej siódmej. *Solver II* jest zatem godny polecenia.

Oczywiście można się posługiwać do tego samego celu *Solverem* i zastosować metodę wielu startów [3]. Polega ona na wielokrotnym uruchamianiu *Solvera* z losowo wybranych punktów należących do dziedziny funkcji celu, a następnie stosownej obróbce wyników. Aby takiej obróbki dokonać, trzeba jednak dysponować aż czterema procedurami:

- uruchamiania *Solvera* w pętli, w kolejnych losowych punktach startowych;
- selekcji tych punktów, w których *Solver* zakończył pracę z powodu bardzo małej wartości modułu gradientu, a w których maksimum nie ma;
- klasteryzacji, aby zlokalizować wszystkie ekstrema;
- wyboru ekstremum globalnego spośród zlokalizowanych.

*Solver II* jest lepszy od *Solvera* również w tym, że zróżnicowanie wartości bezwzględnych poszczególnych argumentów funkcji celu nie wpływa na jego działanie.

## SUPLEMENT A

Pseudokod algorytmu opisanego w przywołanej literaturze [1, 2], nazwanego *Ilin*, jest taki, jak podano poniżej. Lista parametrów formalnych funkcji *Ilin* zawiera te same parametry co lista *Solvera II* oraz parametry *PoprzMaks* niosące największą wartość funkcji celu znaną w poprzednim wywołaniu funkcji *Ilin* oraz *ObnizMaks*, którego rola jest wyjaśniona w tekście pseudokodu.

### ***Function Ilin (Lista parametrów formalnych)***

*Wczytaj parametry funkcji celu.*

*Dla  $i = 1$  to  $k$*

*Wczytaj dolną i górną granicę przedziału przeszukiwania wzdłuż osi  
każdego z  $k$  argumentów  $f.c.$*

*Wylicz szerokości przedziału przeszukiwania wzdłuż  $i$ -tej osi.*

*Następne  $i$*

*Dla  $j = 1$  to  $Nn$*

*Dla  $i = 1$  to  $k$*

*Wygeneruj realizację zmiennej losowej o rozkładzie równomiernym w przedziale  $<0,1>$ .*

*Utwórz z niej liczbę losową o rozkładzie równomiernym w przedziale przeszukiwania wzdłuż  $i$ -tej osi.*

*Następne  $i$*

*Wylicz wartość f.c. w punkcie próbkowania na podstawie uzyskanych wcześniej próbkowanych wartości jej argumentów.*

*Jeżeli wyliczona f.c. jest większa od  $PoprzMaks$ , wówczas zwiększ licznik  $r$  takich przypadków o 1*

*Uzyskane w ten sposób współrzędne punktu, w którym próbkowana była funkcja celu, zapisz do  $r$ -tego wiersza w kolumnach od 1 do  $k$  macierzy wyników  $MW0$*

*Zapisz wartość f.c. do  $k+1$  kolumny  $r$ -tego wiersza macierzy wyników  $MW0$ .*

*Następne  $j$*

*Jeżeli  $r < 2$  wówczas przyjmij  $PoprzMaks = PoprzMaks - ObnizMaks$  i realizuj algorytm od początku.*

*Jeżeli  $r \geq 2$  wówczas w kolumnach macierzy  $MW1$  znajdź najmniejszą i największą wartość próbkową każdego argumentu funkcji celu. Te wartości determinują następny przedział próbkowania.*

*Znalezione wartości prześlij do wskazanego bloku komórek arkusza Excela*

### **End Function**

Eksperymenty numeryczne z funkcją *Ilin* pokazały, że *PoprzMaks* albo było wielokrotnie korygowane o *ObnizMaks* albo funkcja *Ilin* gubiła maksimum globalne. Działo się tak, bowiem liczba uwzględnianych PP była mała.

Algorytm *Solvera II* zapewnia stałą dużą liczbę PP uwzględnianych przy wyznaczaniu nowego obszaru przeszukiwania.

## **SUPLEMENT B**

Zasadnicze zmiany wyglądu OFFICE 2007 i 2010 nie ominęły również okien dialogowych *Solvera*. Metoda gradientowa, jako podstawowa metoda znajdowania ekstremów funkcji gładkich, w nowym Solverze oczywiście pozostała. Zatem wyniki niniejszej pracy dotyczą również nowego *Solvera*.

*Solver II* znacznie łatwiej niż *Solver* można przystosować do znajdowania maksimum, gdy oprócz ograniczeń na wartości argumentów funkcji celu nałożone są dodatkowe ograniczenia mające postać równości lub nierówności, na przykład

$x_1 \leq 0, x_2 \geq 0, x_1 + x_2 = 15$ . Generuje się kolejne PP z obszaru zmienności argumentów, ale próbuje jedynie w tych, których współrzędne spełniają nałożone ograniczenia.

## BIBLIOGRAFIA

- [1] Растрингин Л. А., *Статистические методы поиска*, Наука, Москва 1968.
- [2] Илин В. Н., *Машинное проектирование электронных устройств*, Энергия, Москва 1972.
- [3] Zieliński R., Neuman P., *Stochastyczne metody poszukiwania minimum funkcji*, WNT, Warszawa 1986.

## **SOLVER II AN ALTERNATIVE TO SOLVER IN EXCEL**

### ABSTRACT

This paper deals with the problem of finding a global maximum of the task function when this maximum is surrounded by several local maximums. A tool named *Solver*, offered in Excel spreadsheet, was employed to find the global maximum. Unfortunately, instead of finding the global maximum, *Solver* finds the local one that is nearest to the point from which it starts. The paper puts forward an alternative procedure that reliably finds the surrounded global maximum. This procedure randomly samples an area where all the maximums are located, investigates the shape of the task function and in consecutive iterations tightens the area around the global maximum. When this area becomes sufficiently small, coordinates of the center of the area are taken as approximate coordinates of unknown actual coordinates of the global maximum.

Keywords:

optimization methods, finding global maximum, Excel, *Solver* tool.